Human-Computer Interaction in Map Revision Systems

Jun Zhou and Walter F. Bischof

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{jzhou, wfb}@cs.ualberta.ca

Terry Caelli

National ICT Australia
Canberra Laboratory
Locked Bag 8001, Canberra ACT 2601
Australia
Terry.Caelli@rsise.anu.edu.au

Abstract

Cartographic map revision cannot be automated completely because, for legal reasons, a human operator is ultimately responsible for all revisions. We present a human-centered map revision system in which the human operator retains complete control over the operations with the computer acting as an apprentice and, later, as an assistant. The apprentice learns simple tasks from the human operator by tracking, parsing and modeling all input operations in tasks such as road tracking. The computer also learns by itself during the tracking process. Eventually, the apprentice can take over these tasks from the human and execute them, returning control to the human operator whenever problems arise. Initial results prove the efficiency and reliability of this approach compared map revision performed by the human operator, per se.

1 Introduction

Computer vision research and development is often aimed at replacing humans performing perceptual tasks. However, in applications such as map revision, humans cannot be replaced completely, not only for legal reasons, but also because computer vision algorithms are not sufficiently robust. Classic semi-automatic systems allow a human to initiate the update process, but then proceed with little or no human-computer interaction, leaving final editing to the human (Mckeown & Denlinger, 1988; Baumgartner, Hinz & Wiedemann, 2002). Although the resulting systems can improve the speed of the map revision process, their robustness, efficiency and accuracy are far behind the real-world requirements.

To progress beyond this approach we need to study how humans and computers can understand each other and how they can work cooperatively. This includes developing models that can integrate the human-computer interface, user behaviour models, computer vision algorithms, knowledge transfer schemes and performance evaluation criteria. The human-computer interface provides useful information by tracking the human. The user modeling studies human action patterns from recorded human information while effective computer vision algorithms can replace human for performing tedious and trivial task components. Performance estimation and tracking (machine learning) models then allow the computer vision algorithms to be trained using parameters extracted from human actions. Finally, performance evaluation enables the system to eliminate noise from human input, decide whether to accept or reject the input, and let the human-in-the-loop to gain control over the whole process (see Figure 1). This prototype can be applied to systems that require mutual understanding and different level of interactions between human and computer. For example, similar, but simpler systems have been studied in The Lumiere project (Horvitz et al., 1998) and user supporting systems (Encarnacao & Stoev, 1999).

In this paper, we apply this approach to computer-aided map revision. We introduce a human-centered approach in which the human works as a tutor and decision maker with the computer acting as an apprentice and, after training, as an assistant. The computer tracks, parses and models all user actions and, on request, takes over simple tasks to provably reduce human effort.

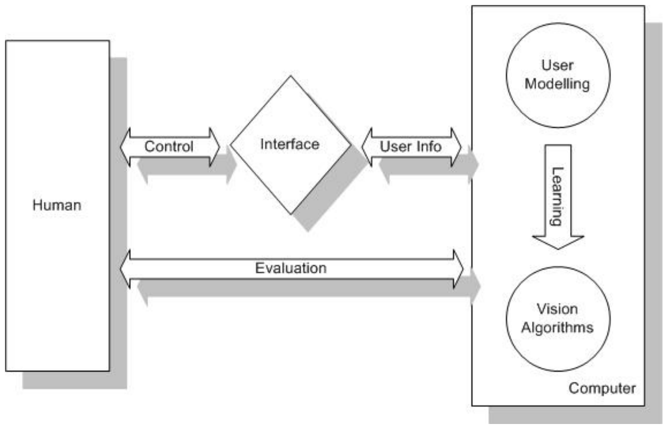

Figure 1: Human-computer interaction prototype

2 Human-Computer Interface and User Modeling in Map Revision

The standard work environment for human-based map revision involves simultaneously displaying an old map with the latest aerial photos. The human operator compares them visually and modifies the map whenever a discrepancy is found between the two. Figure 2 displays such an environment, which is the platform of Raster Graph Revision (RGR) system used in United States Geological Survey (USGS).

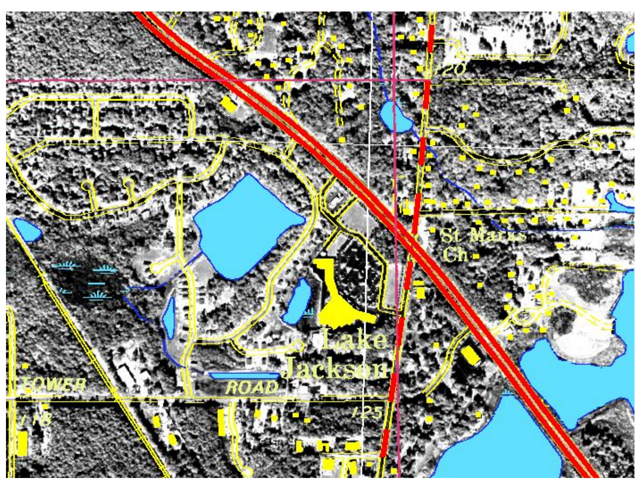

Figure 2: Map revision environment. Old map layers are displayed simultaneously with the latest aerial photos.

In RGR systems, a simple drawing operation can be implemented by either clicking a tool icon on the tool bar followed by clicking on maps using 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 as 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, which includes:

- Task ID
- Key-in command
- Event time
- Event type
- X and Y coordinates of the mouse clicking

By doing so, we have been able to fully capture and record the time-stamped system-level event sequence. This sequence 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.

Altogether there are 278 tools in RGR software, each corresponding to a user action. Among them, 144 actions are related to the drawing actions, each of which may be composed of a tool selection, a sequence of coordinate clicks, viewing changes, as well as reset operations indicating the end of the action. We focus our study on these actions. If we consider a complete action as a sentence, its components can be viewed as words. Syntactic and semantic information lies in both sentence and word level.

We built a semantic lexicon to store the action information. The lexicon has two parts, the first part containing the spelling information of each action. The second part is a semantic marker which shows the usage of the action, such as how many coordinate clicks and resets should an action have, and the meaning of the position of the coordinates in an action given the resets.

A parser was designed according to the syntactic and semantic information of each action. First, a scan generator is applied to the sequence of system-level events. Then the events are grouped into words according to the spelling information of each action. Finally, the sequence of words is segmented into sequence of sentences, or complete actions.

The sequence of actions is arranged into a tree structure. The root of the tree is a project, which is defined as the revision of a map. Each branch contains tasks defined as the revision of one ground object, such as a road, a block of buildings, a lake, and others. The user actions are stored into an XML format database. These data have been used successfully to model view change patterns (Zhou, Bischof & Caelli, 2004).

3 Road Tracking and Human-Computer Interactions

The availability of human data makes it possible for the computer to perform simple tasks in collaboration with the human in map revision. We have developed a road tracking system based on human-computer interactions.

3.1 System Overview

In the road tracking system the human and computer both contribute to the tracking tasks. On the one hand, the computer does not have any initial knowledge of the specific roads to be tracked except some general description of what roads might look like. All other knowledge is learned from human actions. After training, it takes over most of the tracking jobs. On the other hand, the human operator works as a tutor and decision maker, passing on knowledge on roads that is of interest to the computer. The human operator also makes decisions what to do when the computer fails.

The road tracking starts from initial human inputs of road segments. The computer learns relevant road information, such as range of location, direction, road profiles, and step size from this initial human input. At the same time, the computer preprocesses the image to facilitate extraction of road features from images. The extracted road features are compared with the knowledge learned from the human operator. On request, the computer continues with tracking using a Kalman filter (Kalman, 1960). During tracking, the computer continuously updates road knowledge from observing the human tracking. It also evaluates the tracking results. When it detects a possible problem or failure, it gives control back to human who then enters another segment to guide the road tracker. Figure 3 shows the architecture of the system.

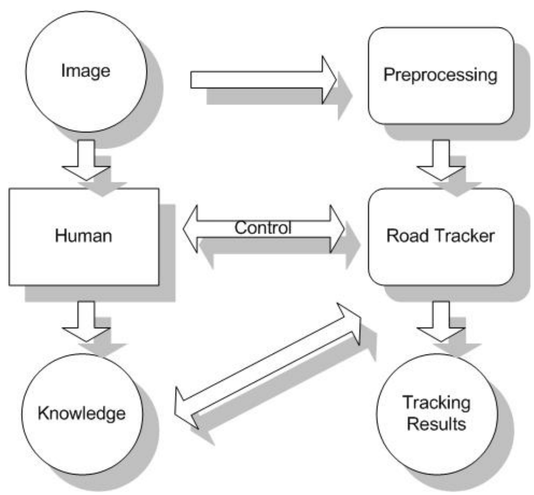

Figure 3: Block diagram for road tracking system.

3.2 Human-Computer Interactions

The human-computer interactions encompass several aspects.

- We have to define what knowledge is to be passed to the computer. This defines the relevant road properties.
- Knowledge has to be transferred to the computer in a way that the computer can understand. This defines the relevant knowledge representation.

- The computer should be able to make effective use of the knowledge, which includes using the acquired knowledge to guide the tracking, and to correct the past tracking errors. This defines the roles of human and computer in the system.

3.2.1 Knowledge about Roads and Its Representation

Two kinds of properties of road can be defined, stationary and dynamic. The stationary properties are those that apply to most of the roads and describe the physical characteristics of the roads. The dynamic properties are those change with different roads, or even change within a road.

Some useful stationary properties are (Bajcsy & Tavakoli, 1976; Vosselman & Knecht, 1995):

- Roads are elongated. The road surface often has a good contrast with the adjacent areas.
- The surface of a road is smooth and homogenous.
- The curvature of roads has an upper bound.
- The width of roads is bounded. The upper and lower bounds of the width depend on the importance of the road.
- Roads are networked.

The stationary properties determine what road tracking method can be used. It is the conceptual-level knowledge of human that guides the designing of computer vision algorithms. For example, the curvature property suggests that road position extrapolation methods can be used to predict the road position (Mckeown & Denlinger, 1988). The contrast and homogenous properties make it possible to use edge detection methods to detect the parallel road edges (Laptev et al., 2000; Baumgartner et al., 1997).

The dynamic properties characterize the changes of road features, such as:

- radiometric changes caused by different road materials,
- special properties of crossings, bridges, and ramps,
- road appearance changes caused by background objects such as cars, shadows, trees, and others.

The dynamic properties cannot be predicted completely. Although a lot of research in computer vision and machine learning has been devoted to these problems, uncertainty is the major reason that causes the failure of automated systems. In contrast, humans can easily detect these properties. Thus, human-computer interactions can help the computer in acquiring such human knowledge, so that to the computer can adapt to changes in road features. One of the effective representations of the dynamic properties is to use road profiles. Road profiles can be extracted as image grey-level pixels related to the road direction.

For this system, we developed algorithms based on edge detection, profile matching and Kalman filtering. Edge detection is built upon the stationary properties, profile matching keeps track of the dynamic properties. The Kalman filter predicts road location as linear systems with random noise and thus approximates both properties.

3.2.2 Roles of the Human and Computer in Road Tracking

The human operator is at the center of the system. The operator affects the tracker in two ways. First, the operator tells the computer the starting location and direction of the road by initializing road segments. These inputs are used by the computer to extract reference profiles, to detect the road edges, and to estimate the road width. They are also used to set the state model of the Kalman filter in the tracking. When the computer fails, the operator observes the road changes, diagnoses the failure reason, and indicates the right direction of the tracking. The new input enables prompt and reliable correction of the state model of the tracker. Second, reference profiles extracted from human inputs are stored, and the road tracker gradually accumulates knowledge on the reference profiles. These profiles represent different road situations that the tracker has not yet seen. This knowledge passing process makes the tracker more and more robust.

The computer also accumulates knowledge by itself. During the tracking, it continues updating the matched reference profiles with the latest tracking results. This enables the tracker to adapt to smooth road changes so that human inputs can be reduced.

The tracker performance is always evaluated in so far as when there is lack of confidence over several consecutive positions control is returned to the human and it simply waits for the next input. This evaluation is performed via cross-correlation where 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 to reduce the tracking performance.

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

4 Road Tracking Algorithm

The road tracking algorithm is an extension of the one used by Vosselman and colleagues (Vosselman & Knecht 1995). Aside from the interaction with humans, it consists of three components: preprocessing, tracking and self-evaluation.

4.1.1 Preprocessing

In the preprocessing step, the image is first convolved with a Gaussian filter to reduce high frequency noise. Then the road width is estimated using edge detection. A road segment is entered by the human operator with two consecutive mouse clicks. A line joining the points defines the road axis. We assume that the road edges are straight and parallel lines on both sides of the road axis. Road width can be estimated by calculating the distance between the roadsides. In this step, knowledge about road characteristics also helps determining road edges because road width varies as a function of road class.

The edge detector is based on gradient profiles. To reduce the uncertainty in the road width estimation, the edge detector first estimates the true upper and lower bound of the road width, with the road width definitions from USGS serving as a reference (USGS, 1996). At each axis point, a profile is extracted perpendicular to the road axis. The gradient of the profile is calculated, and one point is selected on both sides of the axis where the maximum gradient is found. The distance between the two points is considered an estimate of the road width at this axis point. Then a new bound is calculated as a function of the accumulated distance. Using the new bounds, the edge detector determines the new road width at each axis point, extracts new edge points and computes the average distances between edge points as the final estimate of the road width.

Initial reference profiles are extracted from the road segment entered by the human operator according to the road width. Later, new profiles are extracted and added into profile base for further use each time a human input occurs. To avoid noise in the profile, cross correlation is used to remove the points in input road segments.

4.1.2 Kalman Filtering

The Kalman filter is a recursive procedure to estimate the parameters of linear filtering problems. In this system, the Kalman filter is used to predict road center points with the state update equation and to correct the predictions with the measurement update equation. The state model is defined as

$$\widehat{x} = \begin{pmatrix} x \\ y \\ \theta \\ \theta' \end{pmatrix},$$

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 measurement model is calculated as a function of the profile matching result. The recursive update equations can be found in Welch and Bishop's tutorial (Welch & Bishop, 2005) and are not described here.

During the measurement update process, both parallel and perpendicular road features are used to extract the road profiles. Thus the risk of off-road tracking is reduced, and, in turn, tracking errors are reduced.

The tracker evaluates the tracking result using normalized cross-correlation between the reference profile and the road profiles at the current position. When multiple reference profiles are obtained from human inputs, 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 where the cross-correlation is above a preset threshold. These points are skipped. Control is returned to the human operator if too many points are skipped.

We also developed an error correction mechanism using the latest human input. The Kalman filter is used not only to perform prediction but also to backtrack to the previous results, given that all the previous states are recorded. Among all the previous results, the points closest to the failure positions have the highest error probability, and this failure is often caused by changes in road texture. Once a failure is detected, the Kalman filter stops. The human operator enters a new road segment, from which a new reference profile is extracted. This profile contains information that is closely related to the road texture near the failure position. The system uses the new reference profile and tracks from the starting point of the new road segment, but on the opposite direction indicated by the human, and modify errors.

5 Experimental Results

Experiments were done on 26 images cropped from the digital orthophoto quadrangles (DOQs) supplied by the United States Geological Survey. DOQs are orthogonally rectified images produced from aerial photos taken at height of 20,000 feet, with an approximate scale of 1:40,000, and having a ground resolution of 1 meter. These images contained roads in both urban and rural scenes with different complexity.

A human operator was required to draw roads by hand in the RGR software environment as used at USGS. Human data was collected on the images with the number of human inputs and time cost recorded. A total of 472 human inputs were recorded with a total time of 1882 seconds. Hence each image took on average 72.4 seconds and 18.2 inputs. Then the system was run on these images interacting with the recorded data.

The tracking performance was evaluated in three respects, correctness, savings in human input, and savings in plotting time. Using the proposed road tracking system lead to a substantial cost saving, with the number of human inputs and the time cost reduced by 50.4% and 40.9%, respectively. Tracking errors happened in only two images. Thus, the correctness was maintained.

Most previous semi-automatic approaches did not demonstrate how much they really help the human operator. The system described by Baumgartner and colleagues reduced the plotting time by 50% to 70% (Baumgartner et al., 1997) on simple rural scenes. In our work, most of the images were extracted from suburban scenes, which include complex road textures. Some tracking results are shown in Figures 4-5.

6 Conclusion

This paper introduces a human-computer interaction system for map revision where the machine tracks all actions of the human operator and learns a task by comparing the human actions with the image input. On request, the computer takes over simple tasks, returning control to human as soon as confidence rating gets too low. Performance of the system was evaluated using saving in time and input operations.

The proposed interaction prototype is an attempt to fill the gap between human and computer in automatic or semi-automatic computer vision systems. It can also be used in other applications, such as medical image processing and object tracking. It is supposed to reduce the human effort in these tasks while guaranteeing accurate results, because the human is never removed from the process.

In the future, the interaction theory will be further studied, especially the knowledge representation and learning scheme. We consider the knowledge transfer as an online learning process, thus techniques like reinforcement learning should fit in well to such problems.

7 References

Bajcsy, R., & Tavakoli M. (1976). Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(9), 623-637.

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W. (1997). Multi-Resolution, Semantic Objects, and Context for Road Extraction. In W. Forstner & L. Plumer (Ed.), *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*. Birkhäuser Verlag, 140-156

Baumgartner, A., Hinz, S., & Wiedemann, C. (2002). Efficient methods and interfaces for road tracking. *Proceedings of the ISPRS-Commision III Symposium "Photogrammetric Computer Vision" (PCV'02)*, Graz; *International Archives of Photogrammetry and Remote Sensing*, Vol. 34, Part 3-B, pp. 28-31.

Encarnacao, M., & Stoev, S. (1999). An application independent intelligent user support system exploiting action-sequence based user modeling. *Proceedings of the Seventh International Conference on User Modeling*, Banff, Canada, 245-254.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison,* WI, 256-265.

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

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

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 662-672.

USGS (1996). *Standards for 1:24000-Scale Digital Line Graphs and Quadrangle Map.* United States Geological Survey, U.S. Department of Interior.

Welch, G., & Bishop, G. (2001) An Introduction to the Kalman Filter. Retrieved February 14, 2005 from http://www.cs.unc.edu/~tracker/ref/s2001/kalman/

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, 265-274.

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

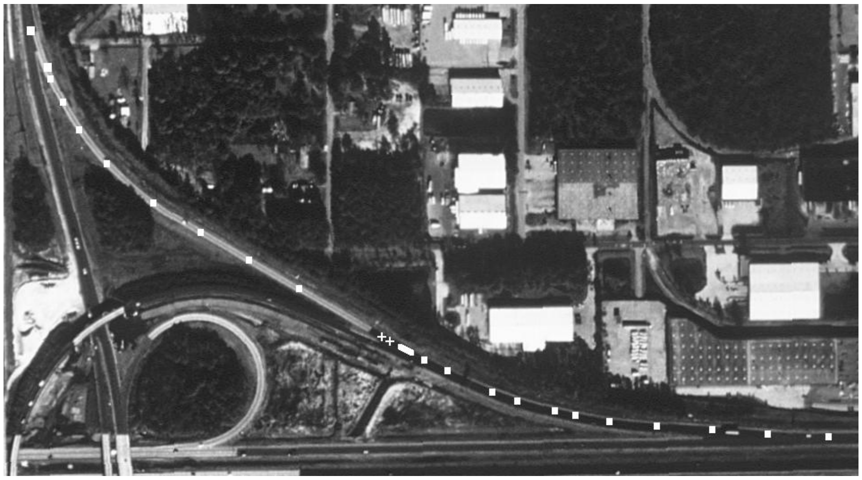

Figure 4: Tracking result. The white dots are the road axis points detected by the tracker. The leftmost two dots are the initial inputs from human. The white line segment in the middle shows the location of the interaction where computer lost control.

Figure 5: Tracking result. The white crosses are the backtracking results by the computer from the point where it failed and get a new human input.