Knowledge Transfer in Semi-automatic Image Interpretation

Jun Zhou[1], Li Cheng[2], and Terry Caelli[23] and Walter F. Bischof[1]

[1] Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{jzhou, wfb}@cs.ualberta.ca
[2] Canberra Laboratory, National ICT Australia,
Locked Bag 8001, Canberra ACT 2601, Australia
{li.cheng, terry.caelli}@nicta.com.au
[3] Research School of Information Science and Engineering,
Australian National University,
Bldg.115, Canberra ACT 0200, Australia

Abstract. Semi-automatic image interpretation systems utilize interactions between users and computers to adapt and update interpretation algorithms. We have studied the influence of human inputs on the image interpretation by examining several knowledge transfer models. Experimental results show that the quality of system performance depended not only on the knowledge transfer patterns but also on the user input, indicating how important it is to develop user-adapted image interpretation systems.

1 Introduction

It is widely accepted that semi-automatic methods are necessary for robust image interpretation [1]. For this reason, we are interested in modelling the influence of human input on the quality of image interpretation. Such modelling is important because users have different working patterns that may affect the behavior of computational algorithms[2]. This involves three components: first, how to represent human inputs in a way that computers can understand; second, how to process the inputs in computational algorithms; and third, how to evaluate the quality of human inputs. In this paper, we describe how we deal with these three aspects in a real world application for updating road maps using aerial images.

2 Road annotation in aerial images

Updating of road data is important in map revision and for ensuring that spatial data in GIS databases remain up to date. This requires normally an interpretation of maps where aerial images are used as the source of update. In real-world map revision environments, for example the software environment used at the United State Geological Survey, manual road annotation is mouse- or command-driven. A simple road drawing operation can be implemented by either clicking a tool icon on the tool bar followed by clicking on maps using a mouse, or by entering a key-in command. The tool icons correspond to road classes and view-change operations, and the mouse clicks correspond to the road axis points, view change locations, or a reset that ends a road annotation. These inputs represent two stages of human image interpretation, the detection of linear features and the digitizing of these features.

We have developed an interface to track such user inputs. A parser is used to segment the human inputs into action sequences and to extract the time and locations of road axis points **inputs**. These time-stamped points are used as input to a semi-automatic system for road tracking. During tracking, the computer interacts with the user, keeping the human at the center of control. A summary of the system is described in the next section.

3 Semi-automatic Road Tracking System

The purpose of semi-automatic road tracking is to relieve the user from some of the image interpretation tasks. The computer is trained to perform road feature tracking as consistent with experts as possible. Road tracking starts from an initially provided road segment indicating the road axis position. The computer learns relevant road information, such as range of location, direction, road profiles, and step size for the segment. On request, the computer continues with tracking using a **road axis predictor**, such as a particle filter or a novelty detector [3, 5]. Observations are extracted at each tracked location and are compared with the knowledge learned from the human operator. During tracking, the computer continuously updates road knowledge from observing human tracking while, at the same time, evaluating the tracking results. When it detects a possible problem or a tracking failure, it gives control back to human, who then enters another segment to guide the road tracker.

Human input affects the tracker in three ways. First, the input affects the parameters of the road tracker. When the tracker is implemented as a **road axis predictor**, the parameters defines the initial state of the system that corresponds to the location of road axis, the direction of road and the curvature change. Second, the input represents the user's interpretation of a road situation, including dynamic properties of the road such as radiometric changes caused by different road materials, and changes in road appearance caused by background objects such as cars, shadows, and trees. The accumulation of these interpretations in a database constitutes a human-to-computer knowledge transfer. Third, human input keeps the human at the center of the control. When the computer fails tracking, new input can be used to set the correct the tracking direction. The new input also permits prompt and reliable correction of the tracker's state model.

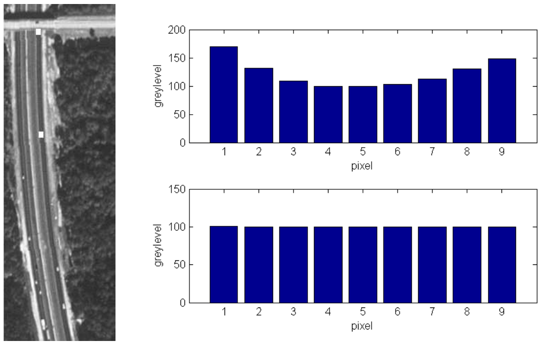

Fig. 1. Profiles of an road segment. In the left image, two white dots indicates the starting and ending points of road segment input by human. The right graphs shows the road profiles perpendicular to (upper) and along (lower) the road direction.

4 Human Input Processing

The representation and processing of human input determines how the input is used and how it affects the behavior of image interpreter.

4.1 Knowledge Representation

Typically, a road is long, smooth, homogenous, and it has parallel edges. However, the situation is far more complex and ambiguous in real images, and this is why computer vision systems often fail. In contrast, humans have a superb ability to interpret these complexities and ambiguities. Human input to the system embeds such interpretation and knowledge on road dynamics.

The road profile is one way to quantize such interpretation in the feature extraction step [4]. The profile is normally defined as a vector that characterize the image greylevel in certain directions. For road tracking applications, the road profile perpendicular to the road direction is important: Image greylevel values change dramatically at the road edges and the distance between these edges is normally constant. Thus, the road axis can be calculated as the mid-points between the road edges. The profile along the road is also useful because the greylevel value varies very little along the road direction, whereas this is not the case in off-road areas.

Whenever we obtain a road segment entered by the user, the road profile is extracted at each road axis point. The profile is extracted in both directions and combined into a vector (shown in figure 1). **Both the individual vector at each road axis point and an average vector for the whole input road segment are calculated and stored in a knowledge base. They characterize a road situation that human has recognized.** These vectors form the template profiles that the computer uses when observation profile is extracted during road tracking.

4.2 Knowledge Transfer

Depending on whether machine learning is involved in creating a road axis point predictor, there are two methods to implement the human-to-computer knowledge transfer using the created knowledge base. The first method is to select a set of road profiles from the knowledge base so that a road tracker can compare to during the automatic tracking. An example is the Bayesian filtering model for road tracking [5]. At each predicted axis point, the tracker extracts an observation vector that contains two directional profiles. This observation is compared to template profiles in knowledge base for a matching. Successful matching means that the prediction is correct, and tracking continues. Otherwise, the user gets involved and provides new input. The second method is to learn a road profile predictor from stored road profiles in the knowledge base, for example, to construct profile predictors as one-class support vector machines [6]. Each predictor is represented as a weighted combination of training profiles obtained from human inputs in the Reproducing Kernel Hilbert space space, where past training samples in the learning session are associated with different weights with a proper time decay.

Both knowledge transfer models are highly dependent on the knowledge obtained from the human. Direct utilizing of human inputs is risky because low quality inputs lower the performance of the system. This is especially the case when profile selection model without machine learning is used. We propose that human inputs can be processed in two ways. First, similar template profiles may be obtained from different human inputs. The knowledge base then expands quickly with redundant information, making profile matching inefficient. Thus, new inputs should be evaluated before being added into the knowledge base, and only profiles that are quite different should be accepted. Second, the human input may contain points of occlusions, for example when a car is in a scene. This generates noisy template profile. On the one hand, such profiles deviate from the dominant road situation. Other the other hand, they expand the knowledge based with barely useful profiles. To solve this problem, we remove those points whose profile has a low correlation with the average profile of the road segment.

5 Human Input Analysis

5.1 Data Collection

Eight participants were required to annotate roads by mouse in an software environment that displays the aerial photos on the screen. None of the users was experienced in using the software and the road annotation task. The annotation was performed by selecting road drawing tools, followed by mouse clicks on the

Table 1. Statistics on users and input

	user1	user2	user3	user4	user5	user6	user7	user8
gender	F	F	M	M	F	M	M	M
total number of inputs	510	415	419	849	419	583	492	484
total time cost (in seconds)	2765	2784	1050	2481	1558	1966	1576	1552
average time per input (in seconds)	5.4	6.6	2.5	2.9	3.7	3.4	3.2	3.2

perceived road axis points in the image. Before performing the data collection, each user was given 20 to 30 minutes to become familiar with the software environment and to learn the operations for file input/output, road annotation, viewing change, and error correction. They did so by working on an aerial image for the Lake Jackson area in Florida. When they felt confident in using the tools, they were assigned 28 tasks to annotate roads 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. Thus, the user had to decide how close the image should be zoomed in to identify the true road axis. Furthermore, the road had to be smooth, i.e. abrupt changes in directions should be avoided and no zigzags should occur.

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

5.2 Data Analysis

We obtained eight data sets, each containing 28 sequences of road axis coordinates tracked by users. Such data was used to initialize the particle filters, to regain control when road tracker had failed, and to correct tracking errors. It was also used to compare performance between the road tracker and manual annotation.

Table 1 shows some statistics on users and data. The statistics include the total number of inputs, the total time for road annotation, and average time per input. The number of inputs reflects how close the user zoomed in the image. When the image is zoomed in, mouse clicks traverse the same distance on the screen but correspond to shorter distances in the image. Thus, the user needed to input more road segments. The average time per input reflects the time that users required to detect one road axis and annotate it.

From the statistics, it is obvious that the users had performed the tasks in different patterns, which influenced the quality of the input. For example, more inputs were recorded for user 4. This was because user 4 zoomed the image into more detail than the other users. This made it possible to detect road axis

Table 2. Performance of semi-automatic road tracker. The meaning of n_h, t_t, and t_c is described in the text.

	user1	user2	user3	user4	user5	user6	user7	user8
n_h	125	142	156	135	108	145	145	135
t_t (in seconds)	154.2	199.2	212.2	184.3	131.5	196.2	199.7	168.3
t_c (in seconds)	833.5	1131.3	627.2	578.3	531.9	686.2	663.8	599.6
time saving (%)	69.9	59.4	40.3	76.7	65.9	65.1	57.8	61.4

locations more accurately in the detailed image. Another example is that of user 3, who spent much less time per input than the others. This was either because he was faster at detection than the others, or because he performed the annotation with less care.

6 Experiments and Evaluations

We implemented the semi-automatic road tracker using profile selection and particle filtering. The road tracker interacted with the recorded human data and used the human data as a virtual user. We counted the number of times that the tracker referred to the human data for help, which is considered as the number of human inputs to the semi-automatic system. In evaluating the efficiency of the system, we computed the savings in human inputs and savings in annotation time. The number of human inputs and plotting time are related and so reducing the number of human inputs also decreases plotting time. Given an average time for a human input, we obtained an empirical function for calculating the time cost of the road tracker:

$$t_c = t_t + \lambda n_h, \tag{1}$$

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

$$\lambda_i = \frac{\text{total time for user i}}{\text{total number of inputs for user i}} \quad 1 \leq i \leq 8. \tag{2}$$

The performance of semi-automatic system is shown in Table 2. We observe a large improvement in efficiency compared to a human doing the tasks manually. Further analysis showed that the majority of the total time cost came from the time used to simulate the human inputs. This suggests that reducing the number of human input can further improve the efficiency of the system. This can be achieved by improving the robustness of the road tracker.

The performance of the system also reflects the quality of human input. Input quality determines how well the template road profiles can be extracted. When an input road axis deviates from the true road axis, the corresponding template profile may include off-road content perpendicular to the road direction. Moreover, the profile along the road direction may no more be constant. Thus, the road tracker may not find a match between observations and template profiles, which in turn requires more human inputs, reducing the system efficiency.

Figure 2 shows a comparison of system with and without processing of human input during road template profile extraction. When human input processing is skipped, noisy template profiles enter the knowledge base. This increases the time for profile matching during the observation step of the Bayesian filter, which, in turn, causes the system efficiency to drop dramatically.

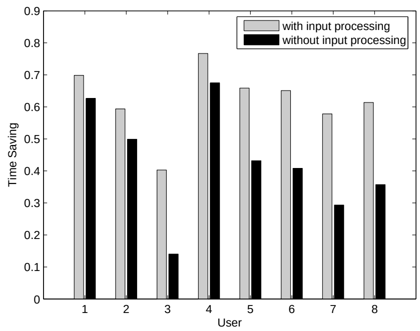

Fig. 2. Efficiency comparison of semi-automatic road tracking.

7 Conclusion

Studying the influence of human input to the semi-automatic image interpretation system is important, not only because human input affects the performance of the system, but also because it is a necessary step to develop user-adapted systems. We have introduced a way to model these influences in an image annotation application. The user inputs were transferred into knowledge that computer vision algorithm can process and accumulate. Then they were processed to optimize the road tracker in profile matching. We analyzed the human input patterns and pointed out how the quality of the human input affected the efficiency of the system.

References

1. Myers, B., Hudson, S., Pausch, R.: Past, present, and future of user interface software tools. ACM Transactions on Computer-Human Interaction **7** (2000) 3–28
2. Chin, D.: Empirical evaluation of user models and user-adapted systems. User Modeling and User-Adapted Interaction **11** (2001) 181–194
3. Isard, M., Blake, A.: CONDENSATION-conditional density propagation for visual tracking. International Journal of Computer Vision **29** (1998) 5–28

4. Baumgartner, A., Hinz, S., Wiedemann, C.: Efficient methods and interfaces for road tracking. International Archives of Photogrammetry and Remote Sensing **34** (2002) 28–31
5. Zhou, J., Bischof, W., Caelli, T.: Road tracking in aerial image based on human-computer interaction and bayesian filtering. ISPRS Journal of Photogrammetry and Remote Sensing **61** (2006) 108–124
6. Zhou, J., Cheng, L., Bischof, W.: A novel learning approach for semi-automatic road tracking. In: Proceedings of the 4th International Workshop on Pattern Recognition in Remote Sensing, Hongkong, China (2006) 61–64