A Novel Learning Approach for Semi-Automatic Road Tracking

Jun Zhou and Walter F. Bischof
Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
{jzhou,wfb}@cs.ualberta.ca

Li Cheng
Canberra Laboratory
National ICT Australia
Locked Bag 8001, Canberra ACT 2601, Australia.
li.cheng@nicta.com.au

Abstract

We tackle the problem of semi-automatic road tracking in aerial photos. Our solution to this human-computer interaction problem is to provide a learning approach that integrates naturally input from human experts with automatic tracking of roads. More specifically, our system learns an ensemble of road predictors from human inputs and uses them to further track a road, alerting the human expert when novel situations are encountered. The proposed approach is shown to outperform existing manual and semi-automatic methods.

1 Introduction

Road tracking is an important task in remote sensing and map revision. It's normally assumed that roads are elongated objects with homogenous surfaces and adequate contrast to adjacent areas. In real scenes, road surfaces vary considerably (see Figure 1 for an example). This is the main source of problems with fully automatic systems. One solution 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 the task [4]. In this approach, knowledge can be transferred dynamically to the computer, not only when necessary, but also to guide the computer. In the past, several semi-automatic road tracking systems have been proposed, which allow humans to initiate the tracking process [3, 7, 1]. Road tracking is performed by road-trajectory prediction using different models and using cross-correlation between templates and observation road profiles.

In this paper, we propose a learning approach that naturally integrates inputs from human experts and automatic tracking of roads. Human inputs provide the online learner with training examples to generate road profile predictors. Thus an ensemble of road predictors is learned incrementally from human inputs, one by one. The predictors are then used to automatically track the road, and when novel situations are encountered, control is returned back to the human expert. In this way, we avoid the problem of having to explicitly define the off-road class, and we enable explicit learning from human inputs. The proposed approach is computationally efficient, and it can rapidly adapt to dynamic situations where the distributions of road feature changes. Experimental results confirm the effectiveness of our approach, and it is shown to be superior to existing methods.

Figure 1. An image sample of size 663 by 423 pixels extracted from an aerial photo.

2 System Framework

A road annotation task starts with an aerial photo with the target of tracking, and the system proceeds with "human input → sampling → prediction" iterations. The first session is aimed at learning from human input whereas the last sessions are aimed at automatic road tracking.

A human input is initiated by two mouse clicks on an image. The line joining the click positions defines the road axis and indicates the road direction. Along the road direction, a set of road profiles are extracted at consecutive axis points normal to and along the road axis. The length of each profile is determined by the road width estimated from the distance between the road edges, which in turn are obtained with a gradient-based edge detection method [8]. We denote a road profile as $x \in \mathcal{X} \subseteq \mathbb{R}^d$ where $\mathcal{X}$ is the profile space with dimension d, and denote one human input as one

learning session $s \in \mathcal{I}_S$ where $\mathcal{I}_S$ indexes the set of learning sessions that occurred for this road tracking task. A road profile x is associated with a label $y \in \mathcal{Y}$ and a state $\sigma \in \Sigma$, where $\mathcal{Y}$ is the set of feasible labels (e.g. $y = 1$: on the road, $y = 0$: off-road), σ encodes its location as

$$\sigma = \left[\begin{array}{ccc} u & v & \theta \end{array} \right]', \tag{1}$$

where u and v are the coordinates of road axis point, and θ is the direction of the road. Then let the triplet $z = (x, \sigma, y)$ represent an *example*. We further restrict that, at an appropriate road axis location, there has to be exactly one profile with $y = 1$, hence $y = 1$ for all training examples. To keep the notation simple, we assume that there are T road profiles along the entire road, and define a set of time steps $\mathcal{T} = \{1, \cdots, T\}$ with one corresponding road profile for each time step. As will become clear later, a predictor (i.e. road profile predictor) $f_s \in \mathcal{F}$ is learned from a learning session s, where $\mathcal{F}$ denotes the set of predictors learned in sessions indexed by $\mathcal{I}_S$.

We assume that, at time t, the system has experienced a sequence of learning sessions $(1, \cdots, s)$, and obtained an ensemble of predictors $\mathcal{F}_1^s \triangleq (f_1, \cdots, f_s)$ by $\mathcal{F}_1^s = \mathcal{F}_1^{s-1} \cup f_s$. It then proceeds with the sampling session, where the system searches the neighborhood (bounded by a fixed range of angles and depths) along the current road axis for candidate road profiles $\tilde{\mathcal{X}} = \{x_1, \cdots, x_m\}$, where m denotes the number of candidates being sampled. The state at time t is sampled using the following non-linear function

$$\sigma_t = \left[\begin{array}{c} u_{t-1} + \varrho \cos(\theta_{t-1}) \\ v_{t-1} + \varrho \sin(\theta_{t-1}) \\ \theta_{t-1} \end{array} \right] \tag{2}$$

where ϱ is the step size of the sampling determined by the road width and tracking status.

After the sampling session, the prediction session starts with a set of candidate profiles $\tilde{\mathcal{X}}$ and their associated states, with the goal of picking a predicted profile $\hat{x}_t$ and then predicting its label $\hat{y}_t$. If $\hat{y}_t = 0$ (not on the road), control is handed back to the human expert for an input. In this manner, switching between human inputs and automatic tracking continues until the tracking task is completed.

Ideally, for predicting an example $\hat{z}_t = (\hat{x}_t, \hat{\sigma}_t, \hat{y}_t)$, we hope to be close to the true one $z_t = (x_t, \sigma_t, y_t)$, that is, $\hat{x}_t \to x_t$, $\hat{\sigma}_t \to \sigma_t$ and $\hat{y}_t = y_t = 1$. Unfortunately, this is not always the case in real scenarios. Therefore, a heuristic strategy was developed to overcome some often-encountered situations. We employ a jump-over strategy, where ϱ is increased to jump over the current state when the system fails to find a road profile with $\hat{y} = 1$. Then a new sampling session occurs from the previous state. The strategy is particularly useful in dealing with small occlusions on the road, for example, when cars and long trucks are presented such that no candidate road profile is predicted on the road. When failures continue, even with the jump-over strategy, the system recognizes a tracking failure and returns control to the human expert, who then inputs another road segment from which new road examples with $y = 1$ can be extracted.

3 Proposed Learning Approach

The interactions between human and computer lead to a situation, where learning sessions are mixed with automatic tracking runs. The first learning session is initialized by the first human input. Each successive learning session s starts when an outlier is detected ($\hat{y}_t = 0$) for the current predicted profile $\hat{x}_t$ at time t, and control is handed back to a human expert. For the sake of simplicity, assume each learning session contains exactly S examples. Therefore, the learning session finishes when the expert finishes teaching with an input that consists successive examples $(z_{t+1}, \cdots, z_{t+S})$, where $z_i = (x_i, \sigma_i, y_i), \forall i \in \{t+1, \cdots, t+S\}$.

3.1 The Learning Algorithm

One learning session corresponds to one human input, with the goal of obtaining a reasonable predictor f_s. Assume the current learning session s starts at time t and contains exactly S examples $(z_{t+1}, \cdots, z_{t+S})$. Further, define a kernel mapping $k(\cdot, \cdot)$ from profile space to a Hilbert feature space, $\mathcal{X} \to \mathcal{H}$ as $x \mapsto k(x, \cdot) \in \mathcal{H}$. Here $\mathcal{H}$ denotes the Reproducing Kernel Hilbert space (RKHS) with induced kernel $k(\cdot, \cdot)$ such that $f(x) = \langle k(x, \cdot), f(\cdot) \rangle$, and $\langle \cdot, \cdot \rangle$ gives the inner product. The norm in this case is naturally defined as $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$.

In the online learning algorithm in [2], the RKHS predictor $f \in \mathcal{H}$ is represented as *a weighted combination of training profiles in the RKHS space*, where past examples in the learning session $\{z_t\}_{t=1}^S$ are associated with different weights $\{\alpha_t\}_{t=1}^S$ (with a proper time decay) that are derived formally from the large margin principle [6]. We extend this algorithm to incorporate learning from human inputs and to deal with the novelty detection scenario, so that the learning problem is naturally formulated as a novelty detection by solving online 1-SVMs **(one class Support Vector Machines)**.

Given a profile x_t at time t, the novelty detection formulation (1-SVM in [5]) is

$$\begin{array}{ll} \min\limits_{\|f\|=1, \xi} & C\xi \\ s.t. & \langle f, k(x_t, \cdot) \rangle \geq 1 - \xi, \\ & \xi \geq 0 \end{array} \tag{3}$$

where $C > 0$ is a constant, and ξ is the positive slack variable. The loss function is then defined as

$$l_t \triangleq l(f_t; x_t) = (\gamma - (1 - \tau)\langle f_t, k(x_t, \cdot) \rangle)_+ \tag{4}$$

where $(\cdot)_+ \triangleq \max\{\cdot, 0\}$. From [2], we can minimize the following regularized risk function. Denote the Bregman divergence as $R_{\rm div}(f) = \|f - f_t\|^2/2$, which measures the distance of the predicted f from the previous prediction f_t. Given this constraint (e.g. $R_{\rm div}$), consider minimizing the regularized risk

$$R_{\rm reg}(f) = \underbrace{\frac{\lambda}{2}\|f\|^2}_{\lambda R_{\rm cap}(f)} + \underbrace{C\xi + \varsigma(\gamma - \langle f, k(x_t, \cdot)\rangle - \xi) - \zeta\xi}_{R_{\rm inst}(f)}, \tag{5}$$

where ς and ζ are Lagrangian multipliers, and $\lambda \geq 0$ is a regularization parameter. The risk function consists of two terms. The capacity risk, $R_{\rm cap}(f)$, controls the complexity of the prediction f, and the instantaneous risk, $R_{\rm inst}(f)$, is the Lagrangian function of the linear programming optimization problem. The separating function is given in [2] as $f(\cdot) = \sum_{i=1}^{t} \alpha_i k(x_i, \cdot)$, where the weights are

$$\begin{cases} \alpha_i & \Leftarrow (1-\tau)\alpha_i \quad \forall i < t \\ \alpha_t & \Leftarrow \min\left\{\frac{l_t}{k(x_t,x_t)},\ (1-\tau)C\right\} \end{cases} \tag{6}$$

As stated in [2], the resultant weight updating formula has several advantages. First, we adopt a robust hinge loss in the weight updating so that α_t is always upper bounded by $(1-\tau)C$. This ensures limited influence from outliers. Second, the decay rate $\tau \in (0, 1)$ is able to balance between adapting to the current example and keeping memory of past examples.

The Learning Algorithm

Input: The cut-off value C, decay rate τ, current learning session s.

Initialize: $f_{t+1} \Leftarrow \underline{0}$.

for $i = t+1$ to $t+S$ **do**

Observe profile x_i

Compute $l_i^{(s)}$ according to Eq. (4)

Compute $(\alpha_j^{(s)})_{j=t+1}^{i}$ according to Eq. (6)

end for

Output: The sequences $(\alpha_j^{(s)})_{j=t+1}^{t+S}$, s, f_s.

Based on the above algorithm, the predictor f_s is obtained given the weight sequence $(\alpha_i^{(s)})_{i=t+1}^{t+S}$ and corresponding training examples. For a sequence of length S, the space complexity of the proposed learning algorithm is $(d+1)S$, and the time complexity is $O(S^2)$.

3.2 The Tracking Algorithm

Assume at time t, after the *sth* learning session, the newly learned predictor f_s is incorporated into the set of predictors as $\mathcal{F}_1^s = \mathcal{F}_1^{s-1} \cup f_s$. The automatic tracking starts by searching the neighborhood along the current road axis for candidate profiles. The losses of the candidates are calculated using Eq. (4) and the one with the minimum loss, $\hat{x}_t$, is picked as the input to the predictors. If it is considered to be on the road ($\hat{y}_t = 1$ with the predictor $f \in \mathcal{F}$, which produces the least loss), the state $\hat{\sigma}_t$ of the profile is used to set the current road axis point and the origin of the next prediction. **Otherwise, when $\hat{y}_t = 0$, a novelty is detected and human is involved to start a new learning session.**

The Tracking Algorithm

Input: Decay rate τ, threshold ϵ, the set of learned predictors so far $\mathcal{F}_1^s$.

Obtain a set of m candidate profiles $\tilde{\mathcal{X}}$ from the sampling session.

for $i = 1$ to m, $j = 1$ to s **do**

Compute $l_{i,j}$ according to Eq. (4)

end for

$(l^*, \hat{x}_t) \leftarrow \min_{i,j}\{l_{i,j}\}$

Predict label as

$$\hat{y}_t = \begin{cases} 0, & l^* \geq \epsilon \\ 1, & otherwise \end{cases} \tag{7}$$

Output: $\hat{z}_t = (\hat{x}_t, \hat{\sigma}_t, \hat{y}_t)$

---

4 Experimental Results

We conducted experiments with humans annotating roads in aerial photos with a ground resolution of 1 meter. Gaussian kernels were adopted for the proposed algorithm in all experiments, and the internal parameters of all algorithms were tuned for good performance. Although the decay factor is a decreasing function of t in the theoretical analysis, we fixed it to appropriate constant values in the following experiments.

Eight users were required to plot roads by hand in an image annotation environment. They were assigned 28 tasks to annotate roads on the aerial photo of the Marietta area in Florida. The 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 and various road conditions. We obtained 8 data sets each containing 28 sequences of road axis coordinates marked by users. These data were used to initialize the online learning, to regain control when road tracker had failed, and to correct tracking errors.

Figure 2 illustrates two examples of road tracking. In both cases, the road tracking starts from the left of the image, with the white/black line segment showing the location of the human input, and the white/black dots showing road

Table 1. Comparison of road tracking results.

	input saving (%)	time saving (%)	distance saving (%)	RMSE (in pixels)
CCKF	71.9	63.9	85.3	1.86
CCPF	72.3	62.0	85.6	1.90
OLT1	73.7	67.9	87.7	1.44
OLT2	77.0	71.9	88.5	1.58

axis points detected by the computer.

Figure 2. Examples of road tracking.

To evaluate the performance of the proposed algorithm, we used the efficiency and accuracy criteria reported in [8]. To evaluate efficiency, savings in number of human inputs, in plotting time, and in tracking distance were considered. Tracking accuracy was evaluated as the root mean square error between the road tracker and ground truth (which was obtained with complete manual input).

We compared the results of the proposed Online Learning and Tracking (OLT) algorithm with two algorithms reported in [8]. These two algorithms implement profile matching by cross-correlation with state prediction based on Kalman filtering (CCKF) and particle filtering (CCPF). Table 1 shows the performance comparison. Due to the factored sampling involved in the particle filtering, the tracker may perform differently for each Monte Carlo trial. For this reason we evaluated the CCPF algorithm over 10 Monte Carlo trials and report the average performance. The experimental results show substantial improvement of the tracking performance using the proposed OLT algorithm compared to the CCKF and CCPF algorithms.

We tested the OLT algorithm with and without time decay in the learning. The decay factors were set to 0 and 0.05 for OLT1 and OLT2, respectively. The result shows that a small decay factor can improve the efficiency of the tracking system, due to the fact that the latest samples are assigned higher weights in the learning sessions, permitting to better characterize the gradual changes of road features.

5 Conclusion

We have presented a learning approach for road tracking in aerial images within a human-computer interaction framework that enables natural switching between human inputs and automatic tracking. Besides conceptual advantages, the experiments on real world tasks validated the superior performance of the approach. This approach is very generic and could be applied to similar applications that requires intensive human-computer interactions.

References

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

[2] L. Cheng, D. Schuurmans, S. Wang, T. Caelli, and S. V. N. Vishwanathan. Online learning with sparse kernels. Technical report, University of Alberta, 2006.

[3] D. McKeown and J. Denlinger. Cooperative methods for road tracking in aerial imagery. In *Proceedings of the IEEE Conference in Computer Vision and Pattern Recognition*, pages 662–672, Ann Arbor, MI, June 1988.

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

[5] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[6] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

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

[8] J. Zhou, W. Bischof, and T. Caelli. Robust and efficient road tracking in aerial images. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (CMRT05)*, volume XXXVI, pages 35–40, Vienna, Austria, August 2005.