A Machine Learning Approach for Automatic Road Extraction

Xiongcai Cai¹, Arcot Sowmya¹, and John Trinder²

¹ School of Computer Science and Engineering, University of New South Wales and National ICT Australia
² School of Surveying and Spatial Information Systems, University of New South Wales

Sydney, NSW 2052, Australia, {xcai, sowmya}@cse.unsw.edu.au,
j.trinder@unsw.edu.au
**Introduction**

- **Automatic Road Extraction**
  - concerned with development of computer vision algorithms for pattern recognition and vector delineation of roads from remotely sensed scenes
  - fundamental step in acquisition and maintenance of geographical databases

- **Machine Learning and Parameter Tuning**
  - many extraction algorithms exist, but heuristic or with manually tuned parameters
  - parameter tuning is key to success of fully automated methods

- **Novel Automatic Method**
  - based on fast marching level set method
  - machine learning for parameter tuning
  - information fusion for refinement of object delineation
System Overview

Figure 1 Overview of the proposed system

Training Images
Reference Images

Test Images

Seed Learning

Parameter Searching

Selection Rule Learning

Contour & Centreline Extraction

Information Fusion

Rules

Rues

Roads
Fast Marching Level Set Method

- Similar to, but has advantages over, classic “snakes”
  - consists of a moving contour, and
  - user defined terms for introducing speed constraints
- Smartly handles sharp corners and topological changes
Let $\gamma_0$ be a closed, nonintersecting initial curve. Assume $\phi(x,t), x \in \mathbb{R}^2$, is a scalar function such that at time $t$ the zero-level set of $\phi(x,t)$ is the curve $\gamma_1$, which consists of all pixel $x$ satisfied $\phi(x,t) = 0$.

Let $\phi(x,0) = \pm d(x)$, where $d(x)$ is the distance from $x$ to the curve $\gamma_0$.

Let such level set of $\Phi(x,t) = z$ evolve along its gradient field with speed $F$. The particle speed $\partial x / \partial t$ in the direction $n$ normal to the level set is given by the speed function $F$.

$$\frac{\partial x}{\partial t} \cdot n = F \Rightarrow \frac{\partial x}{\partial t} \cdot \frac{\nabla \phi}{|\nabla \phi|} = F \Rightarrow \frac{\partial x}{\partial t} = F \frac{\nabla \phi}{|\nabla \phi|}$$

By the chain rule, $\phi(x,t) = z \Rightarrow \phi_t + \frac{\partial x}{\partial t} \cdot \nabla \phi = 0 \Rightarrow \phi_t + F \cdot |\nabla \phi| = 0$
Assume that $F > 0$, $\Phi(x,y,t) = 0$ becomes single-valued in $t$, i.e. each pixel is visited once. This leads to the fast marching level set method:

$$F|\nabla \mu(x, y)| = 1, \text{ where}$$

$\mu$ - arrival time of the contour

$F$ - the speed function

Combined with an optimal sorting technique, this leads to a very fast solution.
Road Recognition

- Two Problems with fast marching level set method
  - Seed selection
  - Parameter tuning for speed function

- Our Primary Contribution
  - Automatic Seed selection (1)
  - Automatic parameter tuning (2)

- Further Performance Improvement
  - Information Fusion at decision level (3)
A Machine Learning Approach for Automatic Road Extraction

(1) Seed Selection

Figure 2 Seeds learning
(1) Seed Selection (ctd)

- Candidate seeds extracted from junction centre points
- Feature subset selection for the seeds
- Feature fusion for combining texture features from candidate seeds and segments
- C4.5 & SVM as learning algorithms
- Stacking for decision fusion
- Texture features (see next slide)
(1) Seed Selection (ctd)

Table 1. Feature List

<table>
<thead>
<tr>
<th>Co-occurrence Matrix Based</th>
<th>Histogram Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>energy</td>
<td>mean</td>
</tr>
<tr>
<td>correlation</td>
<td>variance</td>
</tr>
<tr>
<td>contrast</td>
<td>skewness</td>
</tr>
<tr>
<td>dissimilarity</td>
<td>kurtosis</td>
</tr>
<tr>
<td>homogeneity</td>
<td>energy</td>
</tr>
<tr>
<td>entropy</td>
<td>entropy</td>
</tr>
<tr>
<td>maximum</td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td></td>
</tr>
</tbody>
</table>
(2) Parameter Tuning

Speed function formulation due to Keaton and Brokish (2003):

First attempt:

\[
F(x, y) = \left\{ \left( e^{-\frac{1}{2}(\hat{c}(x,y) - \mu_0)} + e^{-|T(x,y) - T_0|} \right) \times \frac{1}{1 + |\nabla I(x,y)|} \right\} - 10^{-t}. \tag{1}
\]

\[
\mu_0 = \frac{1}{n} \sum_{i=1}^{n} I_i, \quad T_0 = \frac{1}{n} \sum_{i=1}^{n} T_i
\]

- Add a term \(10^t\) to enable learning the speed according to characteristics of the image
- Texture feature vector \(T\) is based on the ‘best’ feature subset of a larger feature set
(2) Parameter Tuning (ctd)

Second attempt:

\[ F(x, y) = \left( e^{-\frac{1}{2}(\hat{c}(x,y) - \mu_0)\Sigma^{-1}(\hat{c}(x,y) - \mu_0)^T} + e^{-|T(x,y) - T_0|} \right) \times \frac{1}{1 + |\nabla I(x, y)|^p}. \]

- Term \( \Sigma \) and \( P \) are parameters that will be automatically tuned.
(2) Parameter Tuning (ctd)

Figure 3 Parameter learning
(2) Parameter Tuning (ctd)

**Sequential Search** - for Speed Function (1)

- Sequential Search
  Sequentially test a range of parameter values

- Evaluate and Compare
  The extractor itself is used as part of the evaluation function
  Compare the performance for each parameter value attempted

- Select the best parameter value that produces the best performance of the extractor

- SVM regression to learn the relationship between image characteristics and best parameter value
(2) Parameter Tuning (ctd)

**Genetic search** - for Speed Function (2)

- Genetic Algorithm
  - Random search method rather than analytical methods or exhaustive search
  - Avoid constructing a complicated model using *a priori* knowledge
  - Reduce the computation burden

- Evaluate and Compare – same as sequential search

- Select the best parameter value

- SVM regression for learning
(2) Parameter Tuning (ctd)

**Figure 4 Genetic Algorithm**

```plaintext
initialize θ, P_{co}, P_{mut}, L n-bit chromosomes
do determine fitness of each chromosome
  rank the chromosomes
  do random select two chromosomes
    if Rand[0, 1] < P_{co} then
      crossover the pair at a randomly chosen bit
    else
      change each bit with probability P_{mut}
  remove the parent chromosomes
  until L offspring have been created
  until reach the maximum iteration limitation θ
return highest fitness chromosome
```
(2) Parameter Tuning (ctd)

Figure 5 Parameter Tuning Using Genetic Algorithm

Training:
initialize training images and references
for each training image
find optimal parameters by GA
create features
build parameter selection rules by learning
return parameter selection rules

Testing:
initialize parameter selection rules, new images
for each new image
calculate features
find parameters using selection rules
return parameters
(3) Decision Fusion

Figure 6 Decision fusion

Majority Vote Rule:

Assign ROAD to class $\omega_j$ if $\nabla R_1 \cap \nabla R_2$, otherwise NONROAD to class $\omega_j$. $\nabla R_1$ and $\nabla R_2$ are the decisions of the individual classifiers.
Experimental Setup

Dataset

• Dataset consists of 11 grey-scale high resolution remotely sensed images from a rural area.

• Size of each image is 1024*1024 pixels cropped from a larger image of ground resolution 1.3 meters per pixel.

• For experiments using genetic search, each image is further split into 9 patches to construct a 99 image patch training set.

• Leave-one-out cross validation is used in order to learn from the largest available dataset and obtain effective test sets.
Experimental Setup (ctd)

Evaluation Metrics

• We use the centerline vector reference model (Wiedemann et al. 1998).

• Manually delineated references are provided as line vectors.

• Evaluation is performed by comparing the recognized road centerline vectors against manual reference.

• The evaluation measures are given by:

$$ \text{completeness} = \frac{\text{length}_{TP}}{\text{length}_{reference}}, \quad \text{correctness} = \frac{\text{length}_{TP}}{\text{length}_{classified}}, $$

$$ \text{length}_{TP} = \text{length}_{(reference \cap classified)}.$$  

• The two measures above are combined into a general measure of quality:

$$ \text{CXC} = \text{completeness} \times \text{correctness}^2 $$

CXC is also used as the fitness for the genetic algorithm.
Experimental Results - Seed selection

- Achieved 89% correctness by leave-one-out cross validation
- Only 4% false positives, causes incorrect object contours and centrelines
- Final centreline was improved by decision fusion
Table 2. Results of parameter learning by algorithm one

<table>
<thead>
<tr>
<th>Image</th>
<th>Learned t</th>
<th>Optimal t</th>
<th>Based on learned t</th>
<th>Based on optimal t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Complete</td>
<td>Correct</td>
<td>CXC</td>
<td>Complete</td>
</tr>
<tr>
<td>1</td>
<td>293</td>
<td>320</td>
<td>42% 100% 42%</td>
<td>42%</td>
</tr>
<tr>
<td>2</td>
<td>270</td>
<td>270</td>
<td>61% 93% 52%</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>270</td>
<td>270</td>
<td>49% 82% 33%</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>287</td>
<td>220</td>
<td>18% 100% 18%</td>
<td>9%</td>
</tr>
<tr>
<td>5</td>
<td>320</td>
<td>320</td>
<td>52% 100% 52%</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>270</td>
<td>270</td>
<td>23% 85% 17%</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>290</td>
<td>300</td>
<td>2% 78% 1%</td>
<td>2%</td>
</tr>
<tr>
<td>8</td>
<td>290</td>
<td>290</td>
<td>49% 68% 23%</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>236</td>
<td>220</td>
<td>85% 92% 71%</td>
<td>81%</td>
</tr>
<tr>
<td>10</td>
<td>279</td>
<td>270</td>
<td>11% 75% 6%</td>
<td>11%</td>
</tr>
<tr>
<td>11</td>
<td>303</td>
<td>320</td>
<td>13% 100% 13%</td>
<td>16%</td>
</tr>
</tbody>
</table>
Experimental Results - Sequential Search (ctd)

Figure 7 Some experimental results using our method on remotely sensed images by algorithm one
Experimental Results – Genetic search

Table 3. Comparative results for shown images (CXC values) — High CXC is better.

<table>
<thead>
<tr>
<th></th>
<th>All (average)</th>
<th>Image A (average)</th>
<th>Patch a</th>
<th>Patch b</th>
<th>Patch c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning approach</td>
<td>34%</td>
<td>64%</td>
<td>90%</td>
<td>98%</td>
<td>87%</td>
</tr>
<tr>
<td>Standard approach</td>
<td>20%</td>
<td>55%</td>
<td>67%</td>
<td>31%</td>
<td>53%</td>
</tr>
<tr>
<td>Improvement</td>
<td>14%</td>
<td>9%</td>
<td>23%</td>
<td>67%</td>
<td>34%</td>
</tr>
</tbody>
</table>
Experimental Results - Genetic search (ctd)

Figure 8

Left: Image A (1024*1024 pixels) by learning approach (CXC 0.64 average).

Right: Patches results: a1, b1 and c1 by learning approach; a2, b2 and c2 by standard approach.
Experimental Results - Genetic search (ctd)

Figure 9 Comparative Results

CXC

Image Patch

Standard Approach

Our Approach
Experiment Results (ctd)

Figure 10-1 Experimental Results of One image Over 4 Steps in the Learning Approach
Experiment Results (ctd)

Edges & Seeds

Figure 10-2 Experimental Results of One image Over 4 Steps in the Learning Approach
Experiment Results (ctd)

Figure 10-3 Experimental Results of One image Over 4 Steps in the Learning Approach
Experiment Results (ctd)

Figure 10-4 Experimental Results of One Image Over 4 Steps in the Learning Approach
Experiment Results (ctd)

Mapping Back

Figure 10-5 Experimental Results of One image Over 4 Steps in the Learning Approach
Conclusion

- Region growing approach based on fast marching level set method for road recognition
- Automatic seed selection and parameter tuning using machine learning
  - relationships: seeds class, seed characteristics and image characteristics
  - relationships: parameters and image characteristics
- Information fusion to refine road centreline