INDUCTIVE CLUSTERING: AUTOMATING LOW-LEVEL SEGMENTATION IN HIGH RESOLUTION IMAGES*

Annie Chen[a], Gary Donovan[a], Arcot Sowmya[a], John Trinder[b]

[a] School of Computer Science and Engineering – (anniec, garyd, sowmya)@cse.unsw.edu.au
[b] School of Surveying and Spatial Information Systems - j.trinder@unsw.edu.au
University of New South Wales, Sydney 2052, Australia

KEY WORDS: Classification, Edge, GIS, High Resolution, Identification, Land Use, Segmentation, Spatial Infrastructures.

ABSTRACT:

In this paper we present a new classification technique for segmenting remotely sensed images, based on cluster analysis and machine learning. Traditional segmentation techniques which use clustering require human interaction to fine-tune the clustering algorithm parameters and select good clusters. Our technique applies inductive learning techniques using C4.5 to learn the parameters and pick good clusters automatically. The techniques are demonstrated on level 1 of RAIL, a hierarchical road recognition system we have developed.

1. INTRODUCTION

Road detection and recognition from remotely sensed imagery is an important process in the acquisition and update of Geographical Information Systems (GIS). Recent research has tended towards automatic approaches such as snakes, and has combined additional image information like multiple scales (Baumgartner et al, 1999; Baumgartner et al, 2000; Laptev et al, 2000) and image bands (Desachy et al, 1999). See (Laptev et al, 2000) for an extensive summary of current road extraction work.

In our previous papers (Teoh et al, 2000a; Teoh et al 2000b; Trinder et al, 1999) we described the RAIL/Recoil/KBRoad system, which is a semi-automatic, multi-level, adaptive and trainable edge-based road recognition system. RAIL is intended to demonstrate the use of various Artificial Intelligence approaches in road extraction, particularly for choosing parameters. These methods are relevant for most applications, even though we only demonstrate them on the RAIL system.

Recently, we have developed a new learning technique designed to automatically combine different algorithms in the best way for a given image - inductive clustering. This uses inductive learning to improve the results obtained from clustering, by learning the optimal clustering parameters for any situation.

The details of this inductive clustering method are presented in this paper. Section 2 gives a brief overview of RAIL, inductive learning and clustering. In Section 3, we introduce the framework that combines inductive learning with clustering, and discuss the experimental components of that framework. Finally, in Section 4 we present some results from evaluating the framework.

2. BACKGROUND INFORMATION

Our research builds upon the existing RAIL system, which includes components for inductive learning and clustering algorithms, amongst other features. RAIL has been used for many experiments in the past, including preliminary trials of Artificial Intelligence techniques in road-detection; a new junction recognition algorithm (Teoh et al, 2000a); and as part of an expert system written in PROLOG (Trinder et al, 1998).

RAIL works by classifying images using different techniques, such as inductive learning or various clustering methods (described below). Classification is the process of sorting image-level features (e.g. edges) into different classes based on their attributes. Our classification systems are designed to arrange all the road edges into one class.

2.1 RAIL

RAIL is a multi-level edge-based road-extraction program, where straight edges are detected from a single-spectrum image using VISTA's implementation of the Canny operator (Pope et al, 1994). Level 1 joins pairs of opposite edges together, whilst Level 2 links the edge pairs together to make road sections. Levels 3 and 4 relate to intersection detection and integration, respectively. At each level, a different classification technique can be applied.

At level 1, the objective is to join matching edges into edge pairs, or road segments. A road segment can be thought of as a pair of edges which are part of a road, and oppose each other. The attributes, or properties, of such a road segment at RAIL Level 1 are listed in Table 1.

* A preliminary version of this paper (Chen, 2002) was presented at ICML2002, Machine Learning in Computer Vision Workshop.

Attributes / Relations	Road Property Addressed
Average Enclosed Intensity inside segment	Roads generally have high gray-scale intensities
Width – Average distance between edges	Road widths usually fall within a certain range
Difference in spatial direction between edges	Roads generally appear as pairs of spatially parallel boundaries
Difference in gradient direction between edges	Road boundaries have opposite gradient directions
Intensity difference (between inside and outside the segment)	Road appears brighter than its surroundings

Table 1. Description of Road Segment Attributes

Initial tests on our sample data set using cluster analysis revealed that the last three attributes do not usefully distinguish between different road segments. Since these attributes were ineffective, they were not used in clustering.

2.2 Inductive Learning

Inductive learning is the process of generating a decision tree by having the computer "learn" rules, based on pre-classified examples provided to it. The resulting decision tree can then be used to classify new examples. One example application of an inductive learner to road recognition is to calculate thresholds, used for selecting edges that match road-sides. Traditionally these thresholds would be determined by human experts, but inductive learning can provide a more customized and locally applicable result.

2.3 Clustering

Clustering is the process of automatically grouping a given set of data into separate clusters such that data points with similar characteristics will belong to the same cluster. While there are many different algorithms for clustering, in this paper we focus on the KMeans and kNN algorithms. Here we describe these algorithms briefly (see Weiss et al, 1991 for further details).

The clustering data set in RAIL is made up of points described by level 1 attributes, where each point represents an edge pair. This data set is to be segmented into n clusters. Each cluster centre is initialized with a random point from the data set, and each remaining point is then grouped into the closest cluster, one at a time.

KMeans measures the distance between the point and the centre of every cluster, eventually placing the point in the closest cluster. After all the data points are clustered, the centre is re-evaluated for each cluster, then the data points are clustered again iteratively. The kNN algorithm differs in that, when grouping the data points, it looks at the k nearest neighbours (i.e. the closest points from existing clusters), and the data point is placed in the cluster containing the most neighbours. We also used the modification suggested by (Barandela et al, 2001) to improve kNN's performance.

Once the clusters have been formed, clusters of interest are identified by visual inspection. In road extraction, clustering can be used, for example, to create a group of edges (or other image-level objects) that have similar shape, intensity, and so on, and hence form part of a road.

A large number of experiments need to be run with different parameters in order to find the setting that produces the best result for a given problem. This whole clustering process requires a lot of hand tuning, to find a suitable algorithm, select the associated parameters, and finally pick out the useful clusters. In this paper we will suggest ways to automate this laborious process by applying inductive learning techniques to each of these stages

3. INDUCTIVE CLUSTERING

Our inductive clustering framework has been designed to learn from cluster descriptions what constitutes a good road cluster, and apply the learned knowledge to perform clustering automatically. The ultimate goal is to allow the system to take a new image and deduce, from the characteristics of the image, the optimal algorithm and parameters to use. It will then automatically identify the road cluster for the user.

This framework uses a multi-level learning strategy to tackle the process systematically at the following three stages (Figure 2):

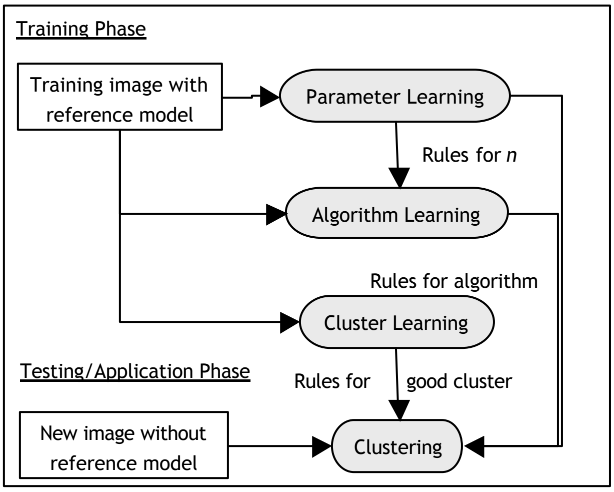

Figure 2. Inductive Clustering Framework Overview

Parameter Learning: Learn the parameters that will give the best result for a given algorithm and image type. Parameters include *n* (the number of clusters) and *k*, in kNN clustering.

Algorithm Learning: Learn which algorithm is most suitable for a given image type. The previous stage determines the parameters to use for each algorithm.

Cluster Learning: We then learn to identify the road clusters by comparing their characteristics to known road and non-road clusters.

Inductive learning methods are used to derive rules at each level. These rules are combined at the end to allow a single step clustering process for extracting road clusters. We currently use the attribute-value learning program C4.5 (Quinlan, 1996) for the inductive learning process.

To train our system we have to run a large number of experiments for each level. In order to automate each of these experiments we used a reference model for each data set, so that the evaluation of the results could be done automatically for each experiment.

Since we used edge-based recognition algorithms, we were able to store the goal state as a simple set of edges. This reference model (or set) of edges was chosen by an experienced human operator. The quality of a cluster could then be measured by comparing the number of reference edges that have been chosen correctly. When compared with our old method of visually inspecting the image, this provides an objective (i.e. observer-independent) and quantifiable measure of the result.

However there are still problems with this reference model system. The model deals with different objects than those created by the levels of RAIL, which leads to false positives. Moreover, this reference format models what the *computer detects*, not what is actually there in the *real world*. In consequence, our stated accuracies are mildly optimistic, at the least. A more suitable reference model would involve storing the actual shape of the road (e.g. its centerline and width). However, we have not yet implemented such a model.

The measures we use to quantify our results are taken from (Harvey, 1999). They are percentage values, given by:

$$complete = \frac{TP}{size_{reference}} \qquad (1)$$
$$correct = \frac{TP}{size_{derived}}$$

where *TP* = Number of True Positive's
size = Number of edges in that image.

High completeness means that the cluster has covered the road edges well, whereas high correctness implies that the cluster does not contain many (incorrect) non-road edges. There is usually a trade off between the two measures, since a complete cluster is more likely to contain spurious non-road edges, and hence be less correct.

It is computationally easier if there is only one criterion to distinguish between clusters. At Level 1 of RAIL, completeness is more important than correctness since we do not want to remove any information at this lower level. Hence our weighted filtering criterion is:

$$cxc = complete^3 \times correct \qquad (2)$$

Clearly, this measure is biased towards completeness. We also used a threshold to ensure that the cluster reached a minimum stage of completeness. These two tests can be expressed together as:

$$complete \geq 80\% \text{ and } cxc \geq 95 \times 10^6 \qquad (3)$$

The thresholds in Equation (3) are based upon empirical observations.

3.1 Parameter Learning

In the parameter learning stage we want to deduce rules for the value of *n* (the number of clusters) to use on a given algorithm and image.

Name	Description	Value
Size	Number of edge pairs.	Continuous
Algorithm	Clustering algorithm used.	KMeans, kNN
n	Number of clusters.	[2, 30)
Classes	Whether attributes produce a good road cluster.	Good, Bad

Table 3. Parameter Learning Attributes

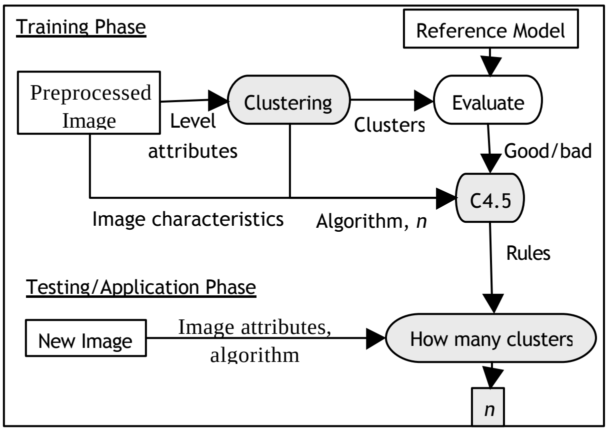

Figure 4. Parameter Learning Experimental Design

The attributes that we use to learn clustering parameters are described in Table 3. This includes image characteristics, along with the clustering parameters we need to determine. Each set of attributes is classified as either "Good" or "Bad". We then classify each setting as capable of generating good or bad road clusters by evaluating the best cluster produced against Eq (3).

There are two phases in the parameter learning process, as shown in Figure 4. In the training phase, Level 1 RAIL attributes of the given image are calculated and used to cluster with different parameters. The clusters generated are evaluated against the reference model to determine which ones are "Good". The inductive learner is then used to generate rules for choosing *n* in other, unseen, images.

The purpose of the Testing/Application phase is obvious: we simply apply the generated rules on a new image to obtain the values of *n* to use for a given algorithm on that image.

3.2 Algorithm Learning

The purpose of algorithm learning is to learn which algorithm to use for a given image. The learning attributes here are image characteristics, and also the algorithm used (see Table 5). For each algorithm, the optimal *n* deduced from parameter learning was used. The algorithm that gives the best result on an image is classified as “Good”, with the others being labeled “Bad”.

Name	**Description**	**Value**
Size	Number of edge pairs.	Continuous
Algorithm	Clustering algorithm used.	KMeans, kNN
Classes	Whether the algorithm produced a good road cluster.	Good, Bad

Table 5. Algorithm Learning Attributes

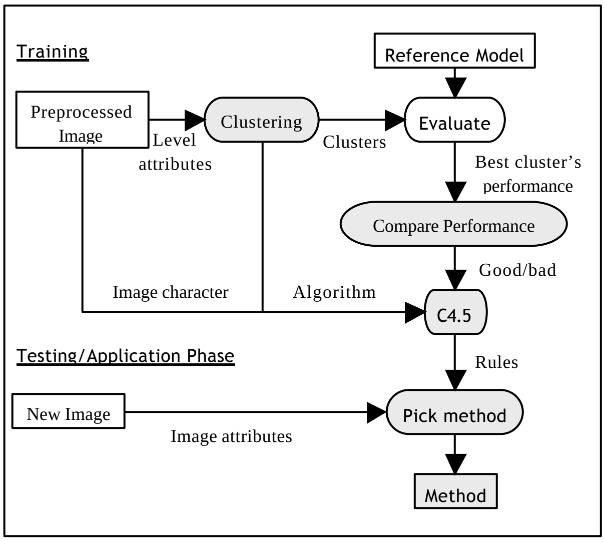

Figure 6. Algorithm Learning Experimental Design

Algorithm learning has two phases of inductive learning, as shown in Figure 6. The details of algorithm learning are similar to those of parameter learning. First, level 1 RAIL attributes of the given image were calculated and used to cluster with different algorithms. The clusters generated were evaluated against the reference model, and the algorithm producing the best road cluster was classified as “Good”, with the other algorithm being labelled as “Bad”. The learning attributes (see Table 5) together with the classification of each run were used to generate a decision tree for application on new images.

3.3 Cluster Learning

In cluster learning we want to deduce rules for identifying the road cluster of each clustering experiment. The learning attributes we have identified in Table 7 are cluster characteristics and image characteristics. We can classify each cluster as “Good” or “Bad” by evaluating it against the reference model.

Name	**Description**	**Value**
Size	Number of edge pairs	Continuous
Aspect ratio	$Width_{max} / Height_{max}$	Continuous
Area	$Width_{max} \times Height_{max}$	Continuous
Centroid	centre of the cluster.	Continuous
Classes	Whether the cluster contains road edges.	Good, Bad

Table 7. Cluster Learning Attributes

Clustering experiments with different algorithms and parameters were run. All the clusters generated were evaluated against the reference model and classified based on the evaluation. The learning attributes from Table 7 together with the classification of each run were used in the inductive learner. This process is shown in Figure 8.

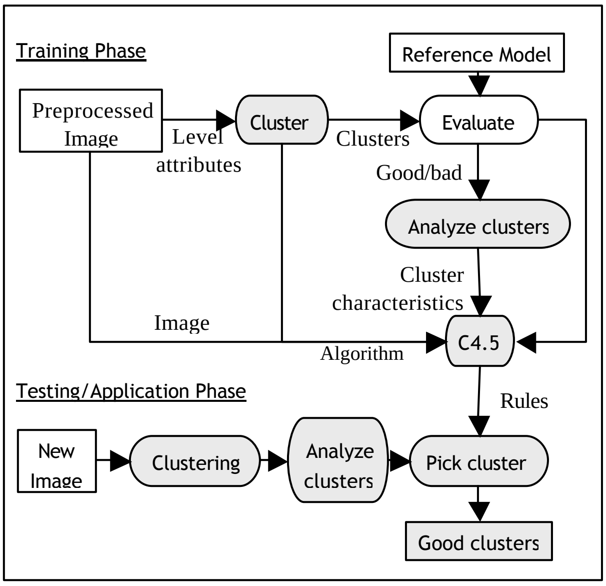

Figure 8. Cluster Learning Experimental Design

4. RESULTS

The inductive clustering framework was initially tested on two digital aerial images of a suburban area of France. These images have a ground resolution of 045m/pixel. One image contains 1956 edges (Figure 9), and the other contains 6481 edges (Figure 10).

Since two images are not enough to learn from and test on, we divided these images into 5 sub-images, giving us 10 sets of edge pairs to experiment on. This subdivision was implemented by forming all possible edge pairs in an image, and calculating the RAIL level 1 attributes for those edge pairs. The resultant attributes were randomly split into 5 subsets, which were treated as independent sets for the purpose of testing.

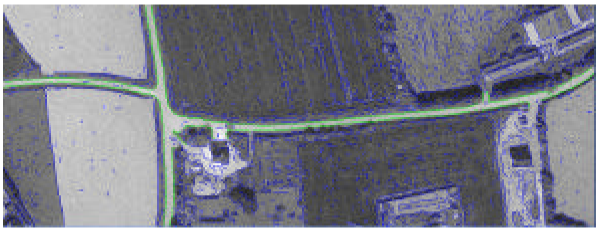

Figure 9: Image A

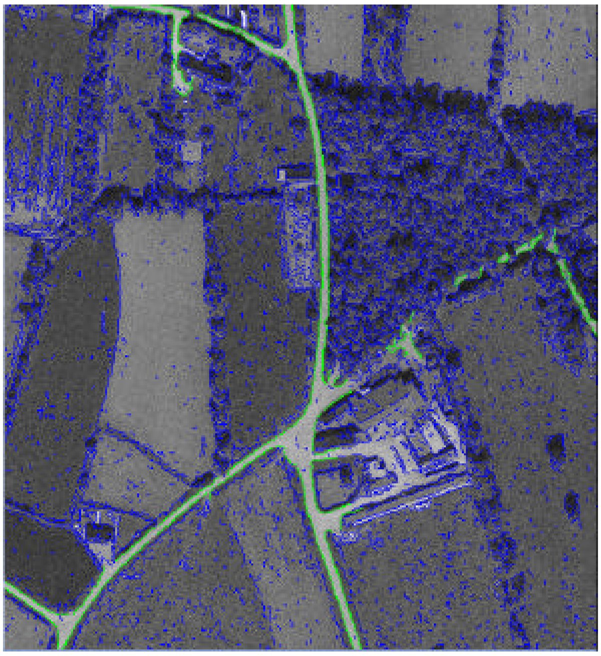

Figure 10: Image B

Clustering experiments were then run on each subset. Unbiased error rates were calculated using 5 fold cross validation (Weiss et al, 1991) for each stage of the framework.

4.1 Rules

The results of inductive clustering are presented in the form of rules. These rules can be applied to new data in order to determine how to classify it. In this section we present a summary of the rules generated by inductive clustering on our two images. Note that the inductive clustering framework can be adapted to different applications and all sorts of images. However, the rules shown below should only be applied to images with similar characteristics (e.g. resolution, complexity, etc.) to the ones we have used.

Each rule identifies a partition of data via its learning attributes and gives a classification for that partition. The percentage after the classification indicates the accuracy of this rule when it is applied to the training data (note that this accuracy measure is optimistically biased). A proper evaluation of the accuracy of these rules is given in Section 4.2.

4.1.1 Parameter Learning[*]

```
Rule 16:
           size > 7023
           size <= 7057
           algorithm = kmeans
           n > 10
           ->  class good  [70.7%]
Rule 6:
           size > 1804
           size <= 1806
           n > 6
           ->  class good  [70.2%]
... etc. ...
Rule 2:
           n <= 5
           ->  class bad  [95.1%]


Default class: bad
```

Summary: For small images use between 5 and 7 clusters. Use more than 8 clusters and KMeans for larger images.

4.1.2 Algorithm Learning

```
Rule 1:
           algorithm = knn
           ->  class good  [68.7%]
Rule 2:
           algorithm = kmeans
           ->  class bad  [68.7%]

Default class: good
```

Summary: kNN generally produces better results than KMeans.

4.1.3 Cluster Learning

```
Rule 4:
           size > 8919
           enclosed_intensity > 142.342
           area > 1961.34
           ->  class good  [80.9%]
Rule 5:
           aspect_ratio > 1.43097
           ->  class bad  [99.8%]
Rule 1:
           area <= 1961.34
           ->  class bad  [99.7%]
Rule 3:
           enclosed_intensity <= 142.342
           ->  class bad  [99.0%]

Default class: bad
```

[*] To make the results clearer, some rules in the parameter learning section have been omitted.

Summary: Road clusters have an enclosed intensity (measured in the range [0, 255]) greater than 142 and area (see Table 7) greater than 1961.

4.2 Evaluation

Table 11 shows the evaluation of the rules presented in the last section. We performed 5-fold cross validation on our data, in order to determine unbiased error rates. This process involves randomly dividing the data into 5 partitions of approximately equal sizes. Leaving out one partition at a time, the remaining data is used to generate a decision tree, which is tested on the 5th partition. Given some assumptions, the average of the 5 error rates is then the error rate for the decision tree. This produces an unbiased measure.

Learning	Param.	Algo.	Cluster
Data Size	521	52	4945
Fold 1	96.8 %	70 %	99.5 %
Fold 2	91.2 %	88.9 %	98.9 %
Fold 3	94.1 %	75 %	99.6 %
Fold 4	95.1 %	90 %	98.9 %
Fold 5	93.6 %	63.3 %	98.9 %
Avg Accuracy Rate	**94.1 %**	**77.4%**	**99.2 %**

Table 11. 5-Fold Cross-Validation Results

5. CONCLUSION AND FUTURE WORK

In this paper we have introduced a way of automating clustering for road classification using inductive learning techniques. We have implemented and tested this concept on our RAIL system, and preliminary results are encouraging.

In the future we hope to improve the accuracy of our testing. One avenue for doing this is to develop a better reference model, addressing the inherent shortcomings of our current edge based one. We plan to train our system using a larger set of images in order to generate better rules. The evaluation measures used (*cxc* and *complete*) are handpicked and their thresholds set empirically. Automation of this process is a future goal. We also plan to extend this clustering framework to other levels of RAIL.

6. REFERENCES

Barandela, R. and Juarez, M., 2001. Ongoing Learning for Supervised Pattern Recognition. In: *XIV Brazilian Symposium on Computer Graphics and Image Processing,* Florianopolis, Brazil.

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W. and Ebner, H., 1999. Automatic Road Extraction Based on Multi-Scale, Grouping and Context. In: *Photogrammetric Engineering and Remote Sensing*, pp. 777-785.

Baumgartner, A. and Hinz, S., 2000. Multi-Scale Road Extraction Using Local and Global Grouping Criteria. In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Amsterdam, Netherlands, Vol XXXIII, part B3/1, pp. 58-65.

Chen, A., Donovan, G. and Sowmya, A., 2002. Recent Progress on RAIL: Automating Clustering and Comparison of Different Road Classification Techniques on High Resolution Remotely Sensed Imagery. *ICML 2002 Workshop on Machine Learning in Computer Vision*, Sydney, Australia, pp. 1-8..

Desachy, J., Bessettes, V. and Dherete, P., 1999. Remote Sensed Images Processing and Geographic Features. In: *Information Processing for Remote Sensing*, C.H. Chen, Ed.: World Scientific, pp. 433-457.

Harvey, W. A., 1999. Performance Evaluation for Road Extraction. In: *The Bulletin de la Société Française de Photogrammétrie et Télédétection,* n. 153(1999-1), pp.79-87

Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C. and Baumgartner, A., 2000. Automatic Extraction of Roads From Aerial Images Based on Scale-Space and Snakes. In: *Machine Vision and Applications: MVA 12,* pp. 22-31.

Pope, A. R., Lowe D. G., 1994. Vista: A Software Environment for Computer Vision Research. In *IEEE Computer Vision and Pattern Recognition 1994*, pp. 768-772.

Quinlan, J. R., 1996. *C4.5: Programs For Machine Learning*. Morgan Kaufmann, San Mateo, California.

Teoh, C.Y., Sowmya, A., 2000a. Junction Extraction from High Resolution Images by Composite Learning. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, Amsterdam, Netherlands, Vol. XXXIII, Part B3, pp882-888.

Teoh, C.Y., Sowmya, A., Bandyopadhyay S., 2000b. Road Extraction from high resolution images by composite learning, In: *Proceedings of International Conf. Advances in Intelligent Systems: Theory and Applications*, Amsterdam, Netherlands, pp308-313.

Trinder, J. C., Wang, Y., 1998. Automatic 4D Feature Extraction from Aerial Imagery. In: *Digital Signal Processing.* 8, pp. 215-224

Trinder, J. C., Nachimuthu, A., Wang, Y., Sowmya, A., Singh, A., 1999. Artificial Intelligence Techniques for Road Extraction from Aerial Images. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* Vol. XXXII, Part 3-2W5, pp. 113-118

Weiss, S., Kulikowski, C., 1991. *Computer Systems That Learn*. Morgan Kaufmann, San Francisco, California.

7. ACKNOWLEDGMENTS

This research is partially supported by an Australian Research Council grant.