Machine Learning Based Urban Change Detection by Fusing High Resolution Aerial Images and Lidar Data

Kaibin Zong[1], Arcot Sowmya[1], and John Trinder[2]

[1] School of Computer Science and Engineering, The University of New South Wales, Sydney 2052, Australia
{kaibinz,sowmya}@cse.unsw.edu.au
[2] School of Civil and Environmental Engineering, The University of New South Wales, Sydney 2052, Australia
j.trinder@unsw.edu.au

Abstract. Map databases usually suffer from obsolete scene details due to frequently occurring changes, therefore automatic change detection has become vital. Generally, change detection is done by spectral analysis of multi temporal images without including elevation information. In this paper, we describe a method for urban change detection by fusing high resolution aerial images with airborne lidar data which provides elevation information. For dealing with radiometric differences, three supervised learning algorithms are introduced which reduce the need for radiometric corrections. Three experiments are conducted on different training sets for each algorithm, to evaluate their performance on change detection and their sensitivity to unbalanced and noisy datasets. All algorithms are also compared with the standard PCA method. Experimental results demonstrate the capabilities of these methods and a detailed theoretical analysis of the achieved results is also presented.

Keywords: Remote sensing, change detection, supervised learning, Lidar.

1 Introduction

City mapping and GIS are widely used in many fields and have also become the focus of recent research [3]. However, the databases usually suffer from obsolete scene details due to frequently occurring changes and require timely updating. Therefore, the development of change detection technology is critical. Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times [1]. Traditionally, changes are detected from multi-temporal images using spectral information alone. Many algorithms have been developed for dealing with this problem, such as image difference, PCA and post-classification comparison [2]. However, with developments in sensor technology, high resolution images bring more challenges for those traditional algorithms and make their performance unstable. By using images alone, it is difficult to distinguish between classes with similar spectral features, or to separate objects made of the same material but with different semantic meaning [3].

F. Bian et al. (Eds.): GRMSE 2013, Part I, CCIS 398, pp. 522–532, 2013.

Airborne lidar (Light Detection and Ranging) is an active acquisition system that scans the terrain normal to the flight direction by emitting infrared laser pulses at high frequency [4]. By exploiting the elevation information provided by lidar, researchers have explored change detection by image and lidar data fusion [5-6]. Trinder and Salah [4] evaluated the contribution of aerial images and lidar data to pixel level change detection using four methods, namely image difference, principal component analysis (PCA), minimum noise fraction transformation (MNF) and post-classification (P-C) comparison. For each method, their results show an improvement in terms of best detection accuracy and omission and commission errors when the two data sources are fused. By fusing image and airborne lidar data, low contrast and shadow effects in images can be compensated for by the more accurately defined planes in lidar, and the poorly defined edges in lidar data can be compensated for by the accurately defined edges in aerial images [4]. However, the performance of P-C comparison is highly reliant on the classification accuracy in each individual image. For the other three methods, a thresholding step must to be implemented, which usually lacks justification. Besides, the choosing of the most suitable components is still problematic when using PCA and MNF methods.

In this paper, we consider fusion of high resolution aerial images and airborne lidar data for urban change detection due to their complementary advantages. In order to overcome the drawbacks of traditional methods and make full use of lidar elevation data, three supervised learning algorithms are utilized for pixel level change detection, namely artificial neural network (ANN) [7], support vector machine (SVM) [8] and Logitboost [9]. ANN and SVM are well known off-the-shelf algorithms that can handle the nonlinear relationships and noise in the training set. Logitboost has been rarely used so far for change detection. It combines the ideas of additive models and logistic regression [9] and the performance of the base learner can usually be improved by the boosting process. By using supervised learning techniques, changes can be predicted directly without using thresholds and the radiometric differences between two images can be handled robustly as well.

In order to demonstrate the capability of these algorithms, we conducted three experiments using different training sets. All algorithms were first tested on a dataset that contains 8 manually selected sub-images. Then, to address the imbalance in the data, we decreased the number of unchanged training instances using random sampling. Finally, some patches were hand-selected from those sub-images to create the third training set. All the three experiments were repeated 10 times with 10-fold cross validation (CV). After that, we further tested the performance of the three learning algorithms on all 8 sub-images based on the best model built in each experiment.

The remainder of this paper is organized as follows. A brief overview of the three learning algorithms is presented in Section 2. The experiments and results of applying those algorithms to change detection are reported in Section 3 followed by detailed results discussion. Finally, conclusions are drawn and some perspectives are given in Section 4.

2 Learning Algorithms

2.1 Artificial Neural Network

Artificial neural network (ANN) is an information processing algorithm inspired by biological nervous systems [7]. It is composed of a large number of interconnected neurons with different weights on the edges. Such a network is able to learn highly complex and nonlinear relationships among data. This is achieved by a nonlinear transformation of the weighted sum of inputs at each individual unit and then combining them together. Among all structures of ANN, the Multi-Layer Perceptron (MLP) with back propagation learning algorithm is the best known [7]. The weights in the network are initialized to small values randomly and are optimized iteratively by minimizing the sum of squared errors. In order to avoid being trapped in local minima, a momentum term is added so that the neural network can step over local minima or flat regions during the gradient descent process and may achieve better performance.

2.2 Support Vector Machine

Support vector machine (SVM) is a statistical learning algorithm aimed at finding a linear separating hyperplane between two classes [8]. This hyperplane can be described using the function:

$$f(x) = sign(< w, x > + b) \tag{1}$$

SVM searches for the best hyperplane among all candidates by maximizing the geometric margins. This results in a classifier that separates training instances of different classes with the largest gap. SVM handles nonlinear relationships by adopting the kernel trick. It maps the feature vector into a higher dimensional space using a nonlinear function and then creates the hyperplane for separation, which corresponds to a nonlinear decision surface in the original feature space. When considering potential outliers in the dataset, a regularization term is added into the target function and the optimal plane is the one that provides the best trade-off between training errors and complexity of the decision surface.

2.3 Logitboost

Logitboost algorithm was introduced by Friedman et al. [9] and combines the ideas of additive model and logistic regression [9]. Logitboost fits its model by maximizing the binomial log-likelihood using a generalized backfitting algorithm called "Local Scoring". Instead of using a weighted linear combination of input attributes, the posterior class probability is estimated based on a sum of smooth functions in the following form:

$$p(x) = P(y = 1|x) = \frac{e^{F(x)}}{1 + e^{F(x)}}$$
$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{2}$$

Similar to Boosting, in each loop, a separate Logistic regression model is built based on a reweighted training set and is added into the ensemble classifier for maximizing the probability in (2). Cross validation is used in this process to avoid overfitting.

3 Experiments and Results

3.1 General Introduction

In order to demonstrate the capabilities of the three supervised learning algorithms on change detection, bi-temporal high resolution aerial images of Coffs Harbour in NSW Australia and corresponding airborne lidar datasets were used (provided by NSW Department of Land and Property Information). Data information is summarized in Table 1.

Table 1. Dataset information

First set of aerial images	Resolution Time	50cm 9/2009
Second set of aerial images	Resolution Time	10cm 12/2009
Two date aerial lidar data	Density Time	1.5pts/m^2 9/2009 & 12/2009

This dataset covers an urban area that includes buildings, roads, trees, water and ground. Due to the short time interval of 3 months between the bi-temporal data acquisition, newly constructed buildings are the main changes estimated by visual interpretation. Hence, the change detection problem can be simplified to a binary classification task. Firstly, a digital surface model (DSM) was generated from the first return lidar data using nearest neighbor interpolation to avoid creating new height values and keeping the well-defined planes. This step results in a lidar image in which pixel values denote ground surface heights (see Fig. 1). After resampling the second aerial image to 50cm resolution with bilinear interpolation, the two images were registered to each other and matched against their corresponding lidar DSM by using a projection transformation.

In this paper, we used a feature vector that consists of 8 attributes for each pixel. The first 3 attributes are the original pixel values (3 colour image bands) in the first aerial image and the fourth attribute is its height information in the DSM. The remaining 4 attributes were derived from the second aerial image and DSM in the same order. As can be seen in Fig. 1, due to the different seasons and sun angle, the radiometric characteristics and shadows are different in the images. Instead of using any pre-processing technique, we aim to overcome these confusing factors by using lidar elevation information and the flexibility of the learning algorithms.

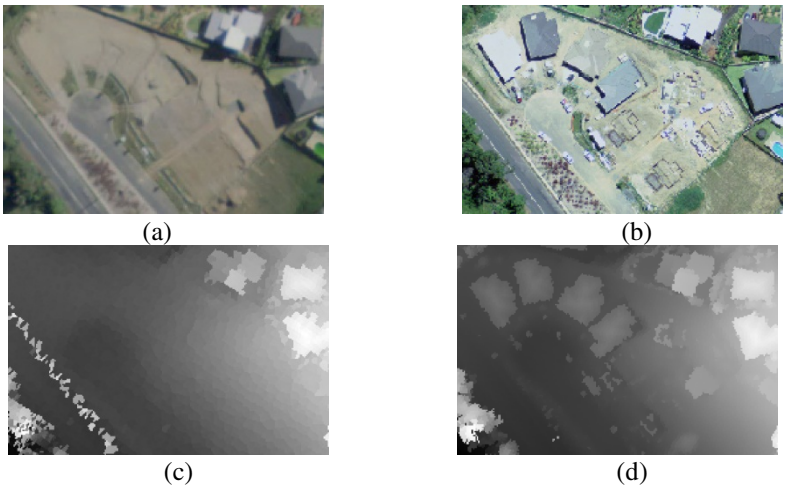

Fig. 1. Sample sub-image (a) The old aerial image; (b) The new aerial image; (c) The old Lidar DSM; (d) The new Lidar DSM

3.2 Experimental Design

Three experiments were conducted for each algorithm based on different training sets. We used 10 times 10-fold CV in all three experiments. For the first experiment, 8 sub-images were selected manually from the whole dataset and combined together as the training set. Among them were 5 sub-images with both changed and unchanged contents, and 3 other sub-images without changes. No sampling process was used in this part. Therefore, the training set in *Experiment 1* is highly unbalanced. Many more unchanged instances were found in the dataset due to the small time interval between the acquisition of two images. This makes the classifier more likely to label a test instance as the majority class in the training set [10]. To address this problem, random sampling was used in *Experiment 2* to decrease the size of the unchanged class. Due to the relatively small number of changed instances, all pixels belonging to this class were retained. The unchanged class was randomly sampled to obtain twice the number of samples of the changed class, in order to tradeoff between the imbalance and representativeness of the training data. In *Experiment 3*, we manually selected some patches from the 8 sub-images to create the training set. Changed patches were selected from all 5 sub-images that contain changes, and unchanged patches were selected from 7 of them. Those instances with outlier values were removed in this step. For further testing of the performance of the three algorithms, all methods were tested on all 8 original sub-images using the best models built in each experiment.

For the neural network, the learning rate was set to 0.01 and momentum was 0.1 in all experiments. Only one hidden layer was constructed with the number of hidden units equals to the average of the number of input and output units. Two output units were used to represent changed and unchanged classes. Weights in the network were updated based on the back propagation algorithm. The second method was SVM,

implemented in the "libsvm" package [11]. All attributes were scaled into the range between -1 and 1 before feeding into the classifier. A polynomial kernel function was chosen for nonlinear mapping and the penalty term was also considered. The best parameter values of degree d and coefficient C in the optimization function were determined based on grid search and 5-fold cross-validation. For Logitboost algorithm, the only parameter that needs to be defined is the maximum iteration number for boosting, which was set to 500 in all experiments. Cross validation was used for choosing the best stopping point and avoiding overfitting. All detection results are compared against the manually created ground truth map and numerical comparisons are carried out based on the confusion matrix (see Table 2). The following criteria (derived from confusion matrix) are used in each experiment, namely the overall accuracy, true negative (TN) rate, false negative (FN) rate, completeness and false alarm rate, and their definitions are listed below.

Table 2. Confusion matrix

	Predicted Class	
Actual Class	Changed	Unchanged
Changed	True positive (TP)	False negative (FN)
Unchanged	False positive (FP)	True negative (TN)

$$Overall\_accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Completeness = \frac{TP}{TP + FN} \quad FN\_rate = \frac{FN}{TP + FN} \tag{3}$$

$$TN\_rate = \frac{TN}{TN + FP} \quad False\_alarm\_rate = \frac{FP}{TN + FP}$$

3.3 Results and Analysis

The averaged results of different learning algorithms in the three experiments are listed in Tables 3, 4 and 5 and the averaged results of testing the best models of all three methods on the 8 original sub-images are summarized in Tables 6, 7 and 8.

Table 3. Averaged performance of three learning algorithms in Experiment 1 using 10 times 10-fold CV

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9913	0.9914	0.972
Completeness	0.9048	0.9	0.5844
FN rate	0.0952	0.1	0.4156
TN rate	0.996	0.996	0.992
False alarm rate	0.004	0.004	0.008

Table 4. Averaged performance of three learning algorithms in Experiment 2 using 10 times 10-fold CV

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9771	0.9758	0.94
Completeness	0.9614	0.9588	0.8939
FN rate	0.0386	0.0412	0.1061
TN rate	0.985	0.9843	0.9631
False alarm rate	0.015	0.0157	0.0369

Table 5. Averaged performance of three learning algorithms in Experiment 3 using 10 times 10-fold CV

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9997	0.9996	0.9991
Completeness	0.9988	0.998	0.9942
FN rate	0.0012	0.002	0.0058
TN rate	1	1	1
False alarm rate	0	0	0

Table 6. Averaged test results of three learning algorithms using their corresponding best models built in Experiment 1

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9921	0.9925	0.9752
Completeness	0.8206	0.795	0.3706
FN rate	0.1794	0.205	0.6294
TN rate	0.9956	0.9966	0.9925
False alarm rate	0.0044	0.0034	0.0075

Table 7. Averaged test results of three learning algorithms using their corresponding best models built in Experiment 2

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9843	0.9848	0.9623
Completeness	0.9228	0.9094	0.7854
FN rate	0.0772	0.0906	0.2146
TN rate	0.9852	0.986	0.965
False alarm rate	0.0147	0.014	0.035

Table 8. Averaged test results of three learning algorithms using their corresponding best models built in Experiment 3

Method	ANN	SVM	Logitboost
Overall Accuracy	0.9421	0.939	0.9424
Completeness	0.9214	0.93	0.8978
FN rate	0.0786	0.07	0.1022
TN rate	0.941	0.939	0.9418
False alarm rate	0.059	0.061	0.0583

From Table 3 one can see that both ANN and SVM achieved good detection results with overall accuracy of 99.13% and 99.14% respectively. More than 90% of changes (completeness) can be detected using either method. In contrast, although Logitboost achieved 97.2% overall accuracy, only 58.44% of real change was detected; this shows that overall accuracy by itself is not a good performance evaluation criterion for unbalanced datasets. When testing the best model of the three algorithms built in Experiment 1 on all sub-images (see Table 6), all the overall accuracies were higher than 97%. However, the completeness decreased to 82.06%, 79.5% and 37.06% when using ANN, SVM and Logitboost respectively.

The detection results using a randomly sampled training set are shown in Table 4. From Tables 3 and 4, one can claim that random sampling of training instances improves the completeness markedly, while the TN rate drops just a little. When testing on sub-images with the best model built on a randomly sampled training set (see Table 7), both ANN and SVM detected more than 90% of changes and the completeness of Logitboost is 78.54%, which is an improvement by more than 40% compared with the 37.06% in Table 6. This is because the imbalance in the training dataset was reduced during the sampling process, and the performance of all three algorithms improved.

In Experiment 3, we manually selected some patches for creating the training dataset. This can be viewed as a kind of spatial sampling. As can be seen from Tables 7 and 8, compared with using a randomly sampled dataset, the completeness for SVM and Logitboost were further enhanced by 2.1% and 11.2% respectively, while remaining the same for neural network. This improvement is due to the data cleaning during the manual selection process. On the patch based training set (see Table 8), SVM outperformed the other two methods in terms of completeness, although its overall accuracy and TN rate were lower.

For comparison purposes, traditional PCA method was tested on the same datasets. Bi-temporal datasets were stacked together (with 8 attributes for each pixel) followed by PCA transformation to detect changed parts. The component with the best contrast was selected manually and changes were separated from the background using thresholds. The averaged results of PCA are listed in Table 9. The testing results of the three algorithms using their best model built in each experiment on the sub-image shown in Fig. 1 are listed in Fig. 2 (a) ~ (i) and the result using traditional PCA method is shown in Fig. 2 (j).

Table 9. Averaged detection results of PCA method

Method	Overall Accuracy	Completeness	FN rate	TN rate	False alarm rate
PCA	0.8753	0.7742	0.2258	0.8838	0.1162

To sum up, both ANN and SVM outperform Logitboost, but it is difficult to choose between ANN and SVM. Moreover, Logitboost is more sensitive to the unbalanced and noisy datasets compared with ANN and SVM. When fusing high resolution images with lidar data, both ANN and SVM exceed PCA even on highly unbalanced and noisy datasets and Logitboost can also outperform PCA as long as the dataset is relatively clean and balanced. This may be explained by the underlying mechanism of

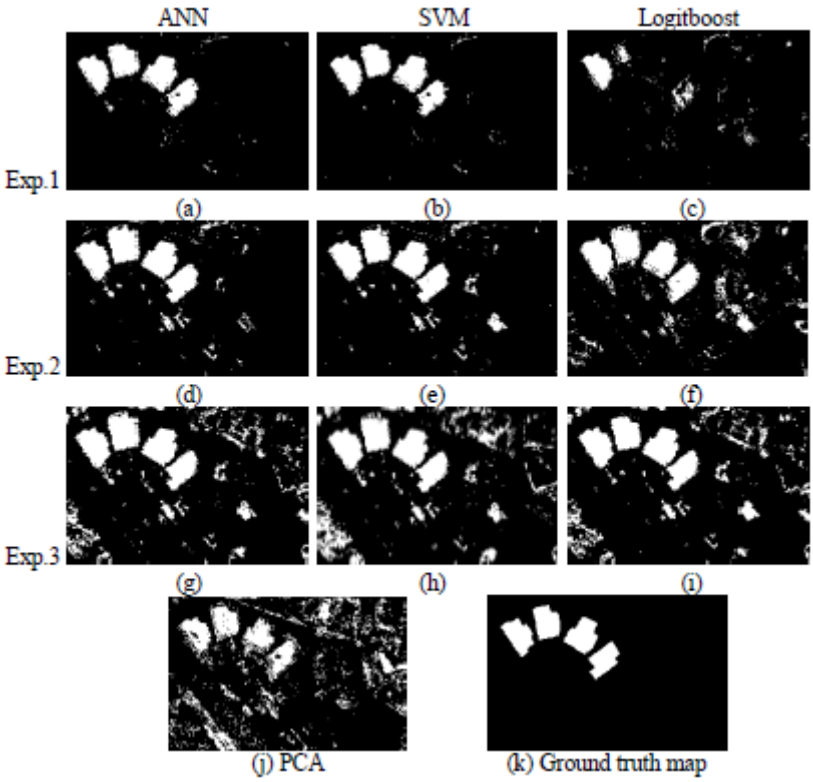

Fig. 1. Testing results of three algorithms using the best model built in each experiment on the sub-image shown in Fig. 1

each method. SVM is designed to maximize the separation margin between two classes, and is thus robust to noise, and the global optimum solution is found by solving the quadratic programming problem. A nonlinear boundary is achieved using the kernel trick and performance is further improved after adding a regularization term. For ANN, as a nonlinear mapping function is used in the network, it has the ability to model complex relationships. Although it cannot guarantee convergence to the global optima, it still works well on many real problems, and this drawback can be partially solved by adding a momentum term. Benefitting from the boosting process, Logitboost is able to model the nonlinear mapping even using a single linear base learner. However, since AdaBoost is sensitive to noise, Logitboost reveals weaker robustness compared with ANN and SVM. In contrast, PCA is based on a linear transformation and cannot handle nonlinear models well. Besides, covariance matrix estimation in PCA is based on the whole image and the noise in the process is known to have impacts. All of these factors together determine the better performance of the supervised learning algorithms.

4 Conclusions

In this paper, the capabilities of three supervised learning algorithms for urban change detection were evaluated. Experimental results illustrate that the performance of all algorithms can be enhanced using random sampling to improve the imbalance in training instances. Moreover, when spatial sampling of training samples is employed, both SVM and Logitboost algorithms work even better. Logitboost shows more sensitivity to imbalance and noise and is not as robust as the other two supervised methods. When compared with PCA, substantial improvements can be found and results can be interpreted from a theoretical point of view. In future, other kinds of features can be considered and cost-sensitive methods for handling unbalanced datasets (e.g. cost-sensitive SVM) can be developed as well. Besides, although the TP rate of changed class increases after sampling, more pseudo detections also appeared (see Fig. 2). This problem should be addressed in future to further improve the result. Finally, due to the success of spatial sampling, we will move from pixel to region based recognition in future research.

Acknowledgments. The authors wish to thank NSW Department of Land and Property Information for the images and lidar data.

References

1. Singh, A.: Digital Change Detection Techniques Using Remotely-sensed Data. International Journal of Remote Sensing 10, 989–1004 (1989)
2. Lu, D., Mausel, P., Brondízio, E., Moran, E.: Change Detection Techniques. International Journal of Remote Sensing 25, 2365–2401 (2004)
3. Luo, B., Chanussot, J.: Geometrical Features for the Classification of Very High Resolution Multispectral Remote-sensing Images. In: 17th IEEE International Conference on Image Processing, pp. 1045–1048. IEEE Press, Hong Kong (2010)
4. Trinder, J., Salah, M.: Aerial Images and Lidar Data Fusion for Disaster Change Detection. In: The XXII Congress of the International Society for Photogrammetry and Remote Sensing, Melbourne, pp. 227–232 (2012)
5. Li, M.C., Cheng, L., Gong, J.Y., Liu, Y.X., Chen, Z.J., Li, F.X., Chen, G., Song, X.G.: Post-earthquake Assessment of Building Damage Degree Using Lidar Data and Imagery. Science in China, Series E: Technological Sciences 51, 133–143 (2008)
6. Chen, L.C., Lin, L.J.: Detection of Building Changes from Aerial Images and Light Detection and Ranging (LIDAR) Data. Journal of Applied Remote Sensing 4, art. no. 041870 (2010)
7. Zometzer, S.F., Davis, J.L., Lau, C.: An Introduction to Neural and Electronic Networks, 2nd edn. Academic Press, New York (1994)
8. Vapnik, V.N.: Universal Learning Technology: Support Vector Machines. NEC Journal of Advanced Technology 2, 137–144 (2005)

9. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. Annals of Statistics 28, 337–407 (2000)
10. Ng, W., Dash, M.: An Evaluation of Progressive Sampling for Unbalanced Data Sets. In: 6th IEEE International Conference on Data Mining Workshops, pp. 657–661. IEEE Press, Hong Kong (2006)
11. LIBSVM: A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm