COMPLEX SCENE ANALYSIS IN URBAN AREAS BASED ON AN ENSEMBLE CLUSTERING METHOD APPLIED ON LIDAR DATA

P. Ramzi*, F. Samadzadegan

Dept. of Geomatics Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran -
(samadz, pramzi)@ut.ac.ir

Commission III, WG III/4

KEY WORDS: LIDAR, Feature, Object, Extraction, Training, Fusion, Urban, Building

ABSTRACT:

3D object extraction is one of the main interests and has lots of applications in photogrammetry and computer vision. In recent years, airborne laser-scanning has been accepted as an effective 3D data collection technique for extracting spatial object models such as digital terrain models (DTM) and building models. Data clustering, also known as unsupervised learning is one of the key techniques in object extraction and is used to understand structure of unlabeled data. Classical clustering methods such as k-means attempt to subdivide a data set into subsets or clusters. A large number of recent researches have attempted to improve the performance of clustering. In this paper, the boost-clustering algorithm which is a novel clustering methodology that exploits the general principles of boosting is implemented and evaluated on features extracted from LiDAR data. This method is a multi-clustering technique in which At each iteration, a new training set is created using weighted random sampling from the original dataset and a simple clustering algorithm such as k-means is applied to provide a new data partitioning. The final clustering solution is produced by aggregating the weighted multiple clustering results. This clustering methodology is used for the analysis of complex scenes in urban areas by extracting three different object classes of buildings, trees and ground, using LiDAR datasets. Experimental results indicate that boost clustering using k-means as its underlying training method provides improved performance and accuracy comparing to simple k-means algorithm.

1. INTRODUCTION

Airborne laser scanning also known as LiDAR has proven to be a suitable technique for collecting 3D information of the ground surface. The high density and accuracy of these surface points have encouraged research in processing and analyzing the data to develop automated processes for feature extraction, DEM generation, object recognition and object reconstruction. In LiDAR systems, data is collected strip wise and usually in four bands of first and last pulse range and intensity (Arefi et al, 2004). Clustering is a method of object extraction and its goal is to reduce the amount of data by categorizing or grouping similar data items together. It is known as an instance of unsupervised learning (Dulyakarn and Rangsanseri, 2001). The grouping of the patterns is accomplished through clustering by defining and quantifying similarities between the individual data points or patterns. The patterns that are similar to the highest extent are assigned to the same cluster. Generally, clustering algorithms can be categorized into iterative square-error partitional clustering, hierarchical clustering, grid-based clustering and density-based clustering (Pedrycz, 1997; Jain et al., 2000).

The most well-known partitioning algorithm is the k-means which is a partitional clustering method so that the data set is partitioned into k subsets in a manner that all points in a given subset are closest to the same center. In other words, it randomly selects k of the instances to represent the clusters. Based on the selected attributes, all remaining instances are assigned to their closer center. K-means then computes the new centers by taking the mean of all data points belonging to the same cluster. The operation is iterated until there is no change in the gravity centers. If k cannot be known ahead of time, various values of k can be evaluated until the most suitable one is found. The effectiveness of this method as well as of others relies heavily on the objective function used in measuring the distance between instances. The difficulty is in finding a distance measure that works well with all types of data (Jane and Dubes, 1995). Some attempts have been carried out to improve the performance of the k-means algorithm such as using the Mahalanobis distance to detect hyper-ellipsoidal shaped clusters or using a fuzzy criterion function resulting in a fuzzy c-means algorithm (Bezdek and Pal, 1992). A few authors have provided methods using the idea of boosting in clustering (Frossyniotis et al., 2004; Saffari and Bischof, 2007; Liu et al., 2008).

1.1 Related Work

Boosting is a general and provably effective method which attempts to boost the accuracy of any given learning algorithm by combining rough and moderately inaccurate classifiers (Freund and Schapire, 1999). The difficulty of using boosting in clustering is that in the classification case it is straightforward whether a basic classifier performs well with respect to a training point, while in the clustering case this task is difficult since there is a lack of knowledge concerning the label of the cluster to which a training point actually belongs (Frossyniotis et al., 2004). The authors in (Frossyniotis et al., 2004) used the same concept, by using two different performance measures for assessing the clustering quality. They incorporated a very similar approach used in the original Discrete AdaBoost

* Corresponding author.

(Freund and Schapire, 1996) for updating the weights and compared the performance of k-means and fuzzy c-means to their boosted versions, and showed better clustering results on a variety of datasets. (Saffari and Bischof, 2007) provided a boosting-based clustering algorithm which builds forward stage-wise additive models for data partitioning and claimed this algorithm overcomes some problems of Frossyniotis et al algorithm (Frossyniotis et al., 2004). It should be noted that the boost-clustering algorithm does not make any assumption about the underlying clustering algorithm, and so is applicable to any clustering algorithm.

However, most of the above methods are provided and evaluated on artificial or standard datasets with small sizes and the significance of improvement in object extraction using this method is not evaluated in urban areas. In this paper, the boost-clustering method is implemented and evaluated on two subsets of LiDAR data in an urban area. The results are then provided in the form of error matrix and some quality analysis factors used for the analysis of classification performance, and compared to the results of the core algorithm in boosting, simple k-means.

2. BOOSTING ALGORITHM

Boosting is a general method for improving the classification accuracy of any classification algorithm. The original idea of boosting was introduced by (Kearns and Valiant, 1998). Boosting directly converts a weak learning model, which performs just slightly better than randomly guessing, into a strong learning model that can be arbitrarily accurate. In boosting, after each weak learning iteration, misclassified training samples are adaptively given high weights in the next iteration. This forces the next weak learner to focus more on the misclassified training data. Because of the good classification performance of AdaBoost, it is widely used in many computer vision problems and some promising results have been obtained (Li et al., 2004). A few attempts have been accomplished to bring the same idea to the clustering domain.

2.1 Boosting Clustering

Boost-clustering is an ensemble clustering approach that iteratively recycles the training examples providing multiple clusterings and resulting in a common partition (Frossyniotis et al., 2004). In ensemble approaches, any member of the ensemble of classifiers are trained sequentially to compensate the drawbacks of the previously trained models, usually using the concept of sample weights. It is sometimes considered as a classifier fusion method in decision level. At each iteration, a distribution over the training points is computed and a new training set is constructed using random sampling from the original dataset. Then a basic clustering algorithm is applied to partition the new training set. The final clustering solution is produced by aggregating the obtained partitions using weighted voting, where the weight of each partition is a measure of its quality (Frossyniotis et al., 2004). Another major advantage of boost clustering is that its performance is not influenced by the randomness of initialization or by the specific type of the basic clustering algorithm used. In addition, it has the great advantage of providing clustering solutions of arbitrary shape though using weak learning algorithms that provide spherical clusters, such as the k-means. It is because the basic clustering method (k-means) is parametric, while the boost-clustering method is nonparametric in the sense that the final partitioning is specified in terms of the membership degrees $h_{i,j}$ and not through the specification of some model parameters.

This fact gives the flexibility to define arbitrarily shaped data partitions (Frossyniotis et al., 2004).

The utilized algorithm is summarized below (Frossyniotis et al., 2004):

1. Input: Dataset $(x_1,...,x_N), x_i \in \Re^d$, number of clusters (C) and maximum number of Iterations (T), Initialize $w_i^1 = {}^1\!/_N$
2. for t=1to T
    a. produce a bootstrap replicate of original dataset
    b. apply the k-means algorithm on dataset to produce the cluster hypothesis $H_i^t = (h_{i,1}^t, h_{i,2}^t, ..., h_{i,C}^t)$ where $h_{i,1}$ is the membership of instance i to cluster j
    c. if t>1, renumber the cluster indices of H^t according to the results of previous iteration
    d. calculate the pseudo-loss
    $$\varepsilon_t = \frac{1}{2}\sum_{i=1}^{N} w_i^t CQ_i^t \qquad (1)$$
    e. set $\beta = \frac{1-\varepsilon_t}{\varepsilon_t}$
    f. if $\varepsilon_t > 0.5$, go to step 3
    g. update distribution W:
    $$W_i^{t+1} = \frac{w_i^t \beta_t^{CQ_i^t}}{Z_t} \qquad (2)$$
    h. compute the aggregate cluster hypothesis:
    $$h_{ag}^t = \arg\max_{k=1,...,C} \sum_{\tau=1}^{t} \left[\frac{\log(\beta_\tau)}{\sum_{j=1}^{t} \log(\beta_j)} h_{i,k}^\tau \right] \qquad (3)$$
3. Output the final cluster hypothesis $H^f = H_{ag}^T$

In the above algorithm, a set X of N dimensional instances x_i, a basic clustering algorithm (k-means) and the desired number of clusters C are first assumed. At each iteration t, the clustering result will be denoted as H^t, while H_{ag}^T is the aggregate partitioning obtained using clustering of previous iteration. Consequently, at the final step, H^f is will be equal to H_{ag}^T. In this algorithm, at each iteration t, a weight w_i^t is computed for each instance x_i such that the higher the weight the more difficult is for x_i to be clustered. At each iteration t, first a dataset X^t is constructed by sampling from X using the distribution W^t and then a partitioning result H^t is produced using the basic clustering algorithm. In the above methodology an index CQ_i^t is used to evaluate the clustering quality of an instance x_i for the partition H^t. In our implementation, index CQ is computed using equation 4.

$$CQ_i^t = 1 - h_{i,good}^t - h_{i,bad}^t \qquad (4)$$

where

$h_{i,good}^t$ = the maximum membership degree of x_i to a cluster.

$h_{i,bad}^t$ = the minimum membership degree to a cluster.

Here, the membership degree $h_{i,j}$ for every instance x_i to cluster j, is produced based on the Euclidean distance d:

$$h_{i,j} = \frac{1}{\dfrac{d(x_i, \mu_j)}{\sum_{k=1}^{C} d(x_i, \mu_k)}} \tag{5}$$

where
$\mu_j \in \Re^d$ = cluster center.

At each iteration, the boost-clustering algorithm clusters data points that were hard to cluster in previous iterations. An important issue to be addressed here and that is the cluster correspondence problem between the clustering results of different iterations (Frossyniotis et al., 2004).

2.2 Feature Extraction

The first step in every clustering process is to extract the feature image bands. These features must contain useful information to discriminate between different regions of the surface. In our experiment we have used two types of features:

- The filtered first pulse range image using gradient
- Opening filtered last pulse range image

By our experiments, these two features have enough information to extract our objects of interest.

The normalized difference of the first and last pulse range images (NDDI) is usually used as the major feature band for discrimination of the vegetation pixels from the others. However, building boundaries also show a large value in this image feature. It is because when the laser beam hits the exposed surface it will have a footprint with a size in the range of 15-30 cm or more. So, if the laser beam hits the edge of a building, then part of the beam footprint will be reflected from the top roof of the building and the other part might reach the ground (Alharthy and Bethel, 2002). The high gradient response on building edges was utilized to filter out the NDDI image using equation 6.

$$NDDI = \frac{FPR - LPR}{FPR + LPR} \tag{6}$$

if gradient $\geq$ threshold, then (FPR-LPR) = 0.0

where
FPR = first-pulse range image data
LPR = last-pulse range image data

The gradient of an image is calculated using equation 7:

$$G(image) = \sqrt{G_x(image)^2 + G_y(image)^2} \tag{7}$$

where
G_x = gradient operators in x direction.
G_y = gradient operators in y direction.

The morphology Opening operator is utilized to filter elevation space. This operator with a flat structuring element eliminates the trend surface of the terrain. The main problem of using this filter is to define the proper size of the structuring element which should be big enough to cover all 3D objects which can be found on the terrain surface. The Opening operation is defined by:

$$A \circ B = (A \Theta B) \oplus B \tag{8}$$

where

$$A \oplus B = \left\{ x \mid \left(\hat{B}_x \cap A \right) \subseteq A \right\} \tag{9}$$

is the morphological Dilation of set A with structure element B. And

$$A \Theta B = \left\{ x \mid B_x \subseteq A \right\} \tag{10}$$

is the morphological Erosion of set A with structure element B (Gonzalez and Woods, 2006).

2.3 Quality Analysis

Comparative studies on clustering algorithms are difficult due to lack of universally agreed upon quantitative performance evaluation measures (Jain et al., 1999). Many similar works in the clustering area use the classification error as the final quality measurement; so in this research, we adopt a similar approach.

Here, we use error matrix as main evaluation method of interpretation result. Each column of this matrix indicates the instances in a predicted class. Each row represents the instances in an actual class. All the diagonal variants refer to the correct interpreted numbers of different classes found in reality. Some measures can be derived from the error matrix, such as producer accuracy, user accuracy and overall accuracy (Liu et al, 2007).

Producer Accuracy (PA) is the probability that a sampled unit in the image is in that particular class. User Accuracy (UA) is the probability that a certain reference class has also been labelled that class. Producer accuracy and user accuracy measures of each class indicate the interpretability of each feature class. We can see the producer accuracy and user accuracy of all the classes in the measures of “producer overall accuracy” and “user overall accuracy”.

$$PA_i = \left(\frac{N_{i,i}}{N_{.i}} \right) * 100\% \quad , \quad UA_i = \left(\frac{N_{i,i}}{N_{i.}} \right) * 100\% \tag{11}$$

where
$N_{i,j}$ = (i,j)th entry in confusion matrix
$N_{i.}$ = the sum of all columns for row i
$N_{.j}$ is the sum of all rows for column i.

“Overall accuracy” considers all the producer accuracy and user accuracy of all the feature classes. Overall accuracy yields one number of the whole error matrix. It‘s the sum of correctly classified samples divided by the total sample number from user set and reference set (Liu et al, 2007).

$$OA = \frac{\sum_{i=1}^{k} N_{i,i}}{\left(\sum_{i=1}^{k} N_{.i} + \sum_{i=1}^{k} N_{i.} \right)} * 100\% \tag{12}$$

Another factor can be also extracted from confusion matrix to evaluate the quality of classification algorithms, which is Kqualifier used to quantify the suitability of the whole clustering method.

$$K = \frac{\sum_{i=1}^{k}\sum_{i=1}^{k} N_{i,j} . \sum_{i=1}^{k} N_{i,i} - \sum_{i=1}^{k}\left(N_{i.}.N_{.i}\right)}{\left(\sum_{i=1}^{k}\sum_{j=1}^{k} N_{i,j}\right)^{2} - \sum_{i=1}^{k}\left(N_{i.}.N_{.i}\right)} \qquad (13)$$

where
k = number of clusters

3. EXPERIMENTAL RESULTS

In this research, we have used two subsets of LiDAR data recorded from the city of Stuttgart, Germany. This data is recorded in four bands of first and last pulse range and intensity. The pixel size of this data is 30 cm. This means the average density of the recorded 3D points which is close to 9 per meter.

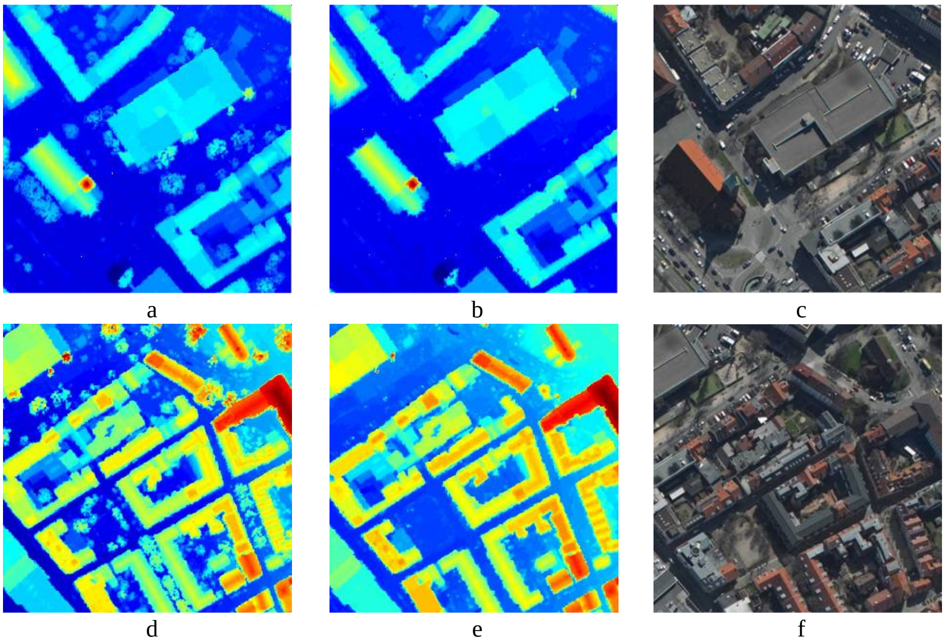

Figure 1. Datasets used in our research. a) first pulse range of the first dataset, b) last pulse range of the first dataset, c) digital aerial image of the first dataset, d) first pulse range of the second dataset, e) last pulse range of the second dataset, f) digital aerial image of the second dataset

For better understanding of the objects, digital color (RGB) images have been also captured from this area using a medium format digital areal camera. In figure 1, color-coded first and last pulse images and also the RGB images of the investigated areas are illustrated. The trees can be distinguishes by comparing first and last pulse images.

3.1 Results of Feature Extraction Algorithms

The level of the discrepancy between first and last return heights before and after applying the gradient filter is shown in figures 2, 3 for our two datasets. The discrepancy was larger than zero in the tree regions as expected.

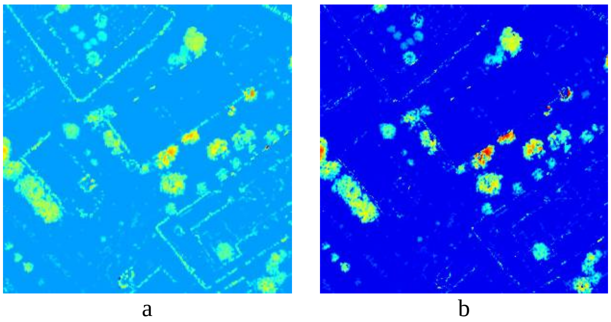

Figure 2. The normalized difference of the first and last pulse range images for our first dataset. a) before gradient filtering, b) after gradient filtering

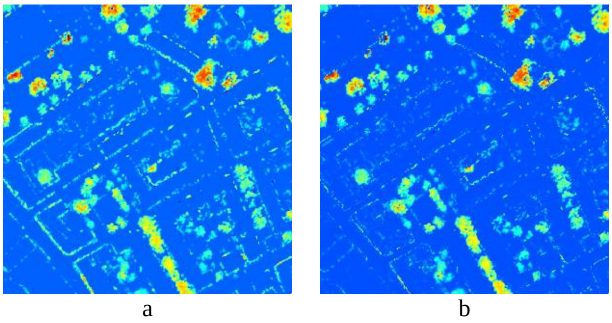

Figure 3. The normalized difference of the first and last pulse range images for our second dataset. a) before gradient filtering, b) after gradient filtering

The feature image of applying the morphological operator on last pulse range image with 5*5 structuring element is illustrated in figure 4. Here, the size of structuring element is selected by experiments on these two datasets.

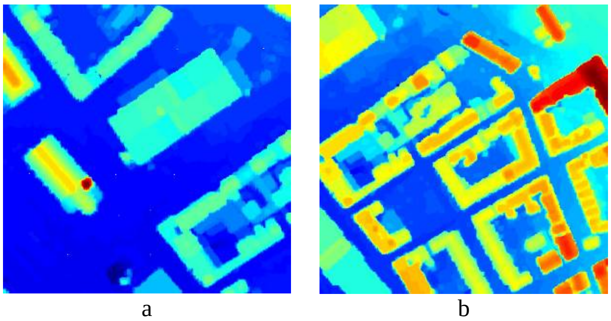

Figure 4. Applying morphological opening operator with structuring element of size 5*5 to last pulse range images. a) the first dataset, b) the second dataset

3.2 Evaluation of the Clustering Results

The results of k-means and boost k-means clustering algorithms applied to features of our two datasets are shown in figure 5 and figure 6. In our experiments the cluster number is considered fixed and equal to 3 because our objects of interest in urban areas are bare earth (blue), vegetation (green) and buildings (red). For the creation of confusion (error) matrix, first, the ground truth (also known as reference clustering results) should be defined. For this, 3D vectors of these areas consist of vegetation and building areas are used. The areas of polygons in pixel unit (number of pixels in the vector polygons of objects) are used as the values of reference clusters in error matrices. The user values are computed by counting the number of truly clustered patterns inside the polygons.

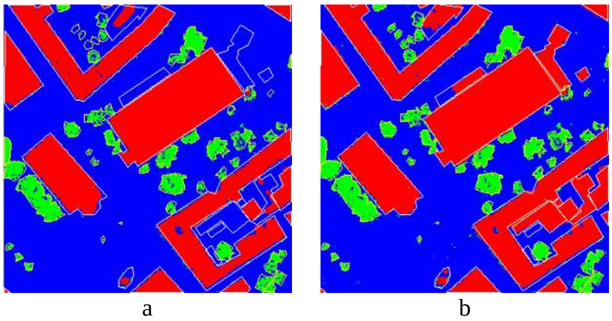

Figure 5. Overlay of reference vectors on clustering results of first dataset. a) result of k-means algorithm, b) result of boost k-means algorithm.

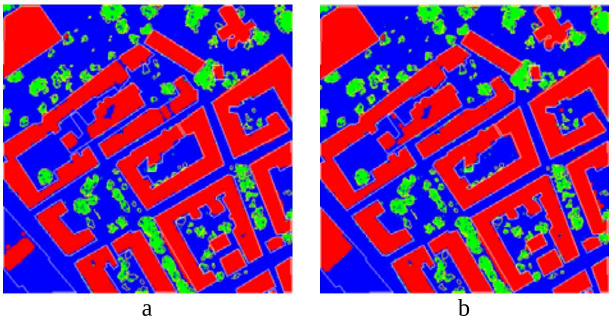

Figure 6. Overlay of reference vectors on clustering results of second dataset. a) result of k-means algorithm, b) result of boost k-means algorithm.

On the first view, both clustering algorithms provide reasonable classes of vegetation, buildings and ground but an accurate and numerical comparison will be carried out comparing the true object elements in the areas of interest.

In Tables 1, 2, the confusion matrices contain the number of pixels assigned to each cluster in the results of k-means clustering is provided. The confusion matrices and NMI factor of the results of boost k-means algorithm are also given in Tables 3, 4.

Table 1. Error matrix and quality factors of k-means clustering applied to first dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	34077	41	975
	Tree	205	6844	1178
	Ground	7946	2197	65607
Producer Accuracy		80.7%	75.4%	96.8%
Producer Accuracy		97.1%	83.2%	86.6%
Overal Accuracy		89.5%		
K-factor		0.801		

Table 2. Error matrix and quality factors of k-means clustering applied to second dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	58393	120	1858
	Tree	261	9808	1810
	Ground	10025	3809	68570
Producer Accuracy		85.0%	71.4%	94.9%
Producer Accuracy		96.7%	82.6%	83.2%
Overal Accuracy		88.4%		
K-factor		0.798		

It should be noted that the confusion matrix is should be diagonal in the ideal case. According to the above confusion matrices and NMI factors and also visual interpretation, improvement in results of clustering using boosting method is obvious for our classes of interest in theses datasets.

Table 3. Error matrix and quality factors of boost k-means clustering applied to first dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	39378	77	1895
	Tree	303	7757	1997
	Ground	2547	1248	63868
Producer Accuracy		93.2%	85.4%	94.2%
Producer Accuracy		95.2%	77.1%	94.4%
Overal Accuracy		93.2%		
K-factor		0.876		

Table 4. Error matrix and quality factors of boost k-means clustering applied to second dataset.

Error Matrix		Reference Map		
		Building	Tree	Ground
Results	Building	61027	212	1393
	Tree	428	10701	1178
	Ground	7224	2824	69667
Producer Accuracy		88.9%	77.9%	96.45
Producer Accuracy		97.4%	86.9%	87.4%
Overal Accuracy		91.4%		
K-factor		0.850		

4. SUMMARY

In this research a boost clustering methodology was applied on two datasets of LiDAR data in an urban area. The proposed method is a multiple clustering method based on the iterative application of a basic clustering algorithm. We evaluated this algorithm using two datasets, to investigate if this algorithm can lead to improved quality and robustness of performance. For the quality analysis of data clustering we used Some quality analysis factors such as produces, user and overall accuracy between the true labels and the labels returned by the clustering algorithms as the quality assessment measure. The experimental results on LiDAR datasets have shown that boost clustering algorithm can lead to better results compared to the solution obtained from the basic algorithm. The usefulness of the two feature channels Gradient Filtered NDDI and Opening of Last Pulse Range image for separating vegetation region with 3D extend and building regions from background has been also shown by the experiments.

There are also several directions for future work in this area. The most important is to determine the optimal number of clusters existing in the dataset. Other interesting future research topics concern the definition of best features of LiDAR data for data clustering and also using digital aerial and intensity images as well as the experimentation with other types of basic clustering algorithms and comparing the results of boost clustering with other strong clustering methods such as fuzzy k-means and neural networks or other multiple clustering based approaches.

REFERENCES

Alharthy,A., Bethel, J., 2002. Heuristic filtering and 3D feature extraction from LiDAR data. *IAPRS*, Graz, Austria. vol. XXXIII, pp. 29-35.

Bezdek, J., Pal, S., 1992. *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*. IEEE Press, New York, NY. 539 pages.

Dulyakarn, P., Rangsanseri, Y., 2001. Fuzzy c-means clustering using spatial information with application to remote sensing. In: *22nd Asian Conference on Remote Sensing, Singapore*. pp. 5-9.

Freund, Y., Schapire, Y., 1996. Experiments with a new boosting algorithm. In: *Proceedings of the Thirteenth International Conference on Machine Learning (ICML),* pp: 148–156.

Freund, Y., Schapire, E., 1999. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5), pp. 771-780.

Frossyniotis , D., Likas, A., Stafylopatis, A., 2004. A clustering method based on boosting. *Pattern Recognition Letters,* 25, pp. 641–654.

Gonzalez R. C., Woods, R. E. 2006. *Digital Image Processing* (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A.K., Dubes, R.C., 1988. *Algorithms for Clustering Data*, Prentice Hall, New Jersey.

Jain, A. K., Murty, M. N., Flynn, P.J., 1999. Data clustering: a review. *ACM Computing Surveys*, 31 (3). pp. 264-323

Kearns, M., Valiant. L. G., 1998. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory.

Kuncheva, L. I., 2004. *Combining Pattern Classifiers, Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey. pp. 251-253.

Li, X., Wang, L., Sung, E., 2004. Improving adaBoost for classification on small training sample sets with active learning, In: *The Sixth Asian Conference on Computer Vision (ACCV)*, Korea.

Liu, C., Frazier, P., Kumar, L., 2007. Comparative assessment of the measures of thematic classification accuracy. *Remote Sensing of Environment*, 107, pp. 606-616.

Liu, Y., Jin, R., Jain, A. K., 2007. BoostCluster: boosting clustering by pairwise constraints. In: *KDD'07, the 13th International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA. pp: 450-459.

Pedrycz, W., 1997. Fuzzy clustering with partial supervision. In: *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, 27(5), pp. 787-795.

Saffari, A., Bischof, H., 2007. Clustering in a boosting framework. In: *Proceedings of Computer Vision Winter Workshop*, St. Lambrecht, Austria.

Zhong, S., Ghosh, D., 2003. A Unified framework for model-based clustering. *Journal of Machine Learning Research.* 4, pp. 1001-1037.