Aerial Lidar Data Classification using Expectation-Maximization

Suresh K. Lodha Darren M. Fitzpatrick David P. Helmbold
School of Engineering
University of California, Santa Cruz, CA USA
{lodha,darrenf,dph}@soe.ucsc.edu

Abstract

We use the Expectation-Maximization (EM) algorithm to classify 3D aerial lidar scattered height data into four categories: road, grass, buildings, and trees. To do so we use five features: height, height variation, normal variation, lidar return intensity, and image intensity. We also use only lidar-derived features to organize the data into three classes (the road and grass classes are merged). We apply and test our results using ten regions taken from lidar data collected over an area of approximately eight square miles, obtaining higher than 94% accuracy. We also apply our classifier to our entire dataset, and present visual classification results both with and without uncertainty. We use several approaches to evaluate the parameter and model choices possible when applying EM to our data. We observe that our classification results are stable and robust over the various subregions of our data which we tested. We also compare our results here with previous classification efforts using this data.

1. Introduction

Aerial and ground-based lidar data is being used to create virtual cities [9, 6, 13], terrain models [16], and to classify different vegetation types [2]. Typically, these datasets are quite large and require some sort of automatic processing. The standard technique is to first normalize the height data (subtract a ground model), then use a threshold to classify data into into low- and high-height data. In relatively flat regions which contain few trees this may yield reasonable results, *e.g*. the USC campus [17]; however in areas which are forested or highly-sloped, manual input and correction is essential with current methods in order to obtain an useful classification.

This work presents an algorithm for automatic classification of aerial lidar data into 4 groups – buildings, trees, roads, and grass – using the lidar data registered with aerial imagery. When aerial imagery is not available, our algorithm classifies aerial lidar data automatically into 3 classes: buildings, trees, and road-grass. We use 5 features: height, height variation, normal variation, lidar return intensity, and image intensity. We do not use the last feature for 3-way classification.

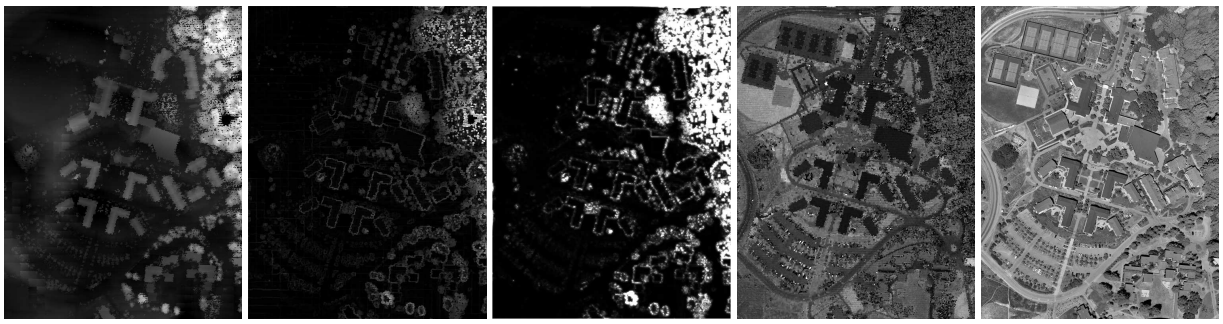
For our classification we use the well-known Expectation-Maximization algorithm to fit a probabilistic model to our data. Specifically, we use EM to find maximum-likelihood estimates (MLEs) of the parameters in a Mixture of Gaussians (MoG) model for each class. When applying EM, one must choose a model and also any model parameters which are not automatically determined through MLE. We discuss the semi-automatic methodology we used to determine these remaining model parameters.

We apply our algorithm to aerial lidar data collected over a region which has highly undulating terrain and is well-forested. We present our results in Section 5. We measure the classification accuracy within ten subregions of our dataset (portions of which were manually labeled for training and accuracy assessment) and find that the results are quite stable. We also gauge the robustness of our classifier by applying it to our entire dataset. We present the classifications visually both with and without uncertainty information. We also provide a comparison to our previous classification efforts using this data.

2. Related Work

Previous work on aerial lidar data classification can be broadly put into two categories: (i) classification of aerial lidar data into terrain and non-terrain points, and (ii) classification of aerial lidar data into features such as trees, buildings, etc.

Several algorithms for classifying data into terrain and non-terrain points have been presented, including those by Kraus and Pfeifer [10] using an iterative linear prediction scheme, by Vosselman *et al*. [16] using slope-based techniques, and by Axelsson [1] using adaptive irregular triangular networks. Sithole and Vosselman [14] present a comparison of these algorithms. We have used a variation of these standard algorithms to create a terrain model from aerial lidar data in order to compute normalized height.



(a) Normalized Height (H) (b) Height Variation (HV) (c) Normal Variation (NV) (d) Lidar Return Intensity (LRI) (e) Image Intensity (I)

Figure 1. The five feature images computed for Region 2.

Previous multiclass classification efforts include research by Axelsson [1], Maas [12], Filin [5], and Haala and Brenner [7]. Most of these approaches are ad hoc and based on heuristics. In addition, in most cases, results are presented for small regions without much discussion on the overall quality of results obtained. Furthermore, most of the previous work in classification of aerial lidar data has concentrated on unsupervised clustering into a smaller number of classes often resulting in coarse classification. Finally, these approaches often require substantial manual input.

The most relevant previous work uses the Expectation-Maximization (EM) algorithm with a Mixture of Gaussian Models to classify lidar data into four categories [3]. There are several differences between this previous work and the current. First, the previous work requires the use of additional information – DEM data, co-registered with aerial lidar data. Second, the previous implementation of EM algorithm suffers from instability due to dependencies of the algorithm on user-selected parameters, specifically, K, the number of components in the mixture model. We provide data to justify a semi-automatic method for determining this value. Third, the previous work did not apply the result to a large, untrained dataset (such as our entire dataset classification). Fourth, we use a different feature set. We introduce normal variation, which we will later find is very useful in our classification. Multiple return difference [3], which was used as a feature in the previous work, is discarded as it was not found to be useful. Finally, as reported in Section 5, we obtain higher accuracy (better than 94%) in comparison to the 66-84% accuracy reported in the previous work. We also compare our results to previous results using the SVM algorithm to classify this data [11].

3. Data Processing

Our lidar dataset was acquired by Airborne1 Inc. The data was collected for approximately eight square miles of target region using a 1064 nm laser at a pulse rate of 25 KHz. The raw data consists of about 36 million points, with an average point spacing of 0.26 meters. We resampled this irregular lidar point cloud onto a regular grid with a spacing of 0.5m using nearest-neighbor interpolation.

In addition, we use high-resolution (0.5ft/pixel) orthorectified gray-scale aerial imagery. We downsampled the aerial imagery to the same 0.5m/pixel resolution as the lidar data and registered the two.

3.1. Features

We have identified five features to be used for data classification purposes: normalized height, height variation, normal variation, lidar return intensity, and image intensity.

- *Normalized Height (H)*: We computed the terrain elevation data automatically from the aerial lidar data using a variation of the standard DEM extraction algorithms [14]. The lidar data is normalized by subtracting terrain elevations from the lidar data.

- *Height Variation (HV)*: Height variation is measured within a 3×3 pixel $(2.25m^2)$ window and is calculated as the absolute difference between the min and max values of the normalized height within the window.

- *Normal Variation (NV)*: We first compute the normal at each grid point using finite differences. Normal variation is the average dot product of each normal with other normals within a 11×11 pixel $(30.25m^2)$ window. This value gives a measure of planarity within the window.

- *Lidar Return Intensity (LRI)*: Along with the height values, aerial lidar data contains the amplitude of the response reflected back to the laser scanner. We refer to this amplitude as LRI.

- *Image Intensity (I)*: Image intensity corresponds to the response of the terrain and non-terrain surfaces to visible light. This is obtained from the gray-scale aerial images.

All the five features have been normalized to integer values between 0 and 255. Figure 1 shows images of these five features computed on Region 2.

3.2. Classes and Training

We classify the dataset into four groups: buildings (rooftops), trees or high vegetation (includes coniferous and deciduous trees), grass or low vegetation (includes green and dry grass), and road (asphalt roads, concrete pathways and soil). We also combine the road and grass classes to perform a 3-class labeling using only lidar-derived features. Ten different regions of the original dataset were segmented and subregions of these were manually labeled for training and validation. The sizes of these ten regions vary from 100,000 to 150,000 points; together they comprise about 7% of our entire dataset. Roughly 25-30% of these regions were manually labeled using a graphical user interface to cover the four classes adequately. However, the proportions of the classes vary significantly from region to region.

4. The Expectation-Maximization Technique

Expectation-maximization (EM) [4] is a general method for fitting probability distributions which may contain unobservable latent variables. Specifically, EM is often used to compute maximum likelihood estimates (MLEs) of parameters to probabilistic models being fit to data where some unobservable information prevents direct MLE. Within the context of probabilistic data classification by mixture models, EM can be used to compute an MLE of the mixture parameters for data when it is unclear from which component of the mixture each datum was drawn from. EM alternates between using the current parameter estimates to compute the likelihoods of each datum belonging to each component, and using these likelihoods to update the parameter estimates. Thus one can obtain an useful estimate of the probability that a datum was drawn from some mixture of distributions, even where it is impossible to know from exactly which distribution the datum was sampled from.

4.1. Supervised Classification and EM

We have applied EM to a supervised classification problem. Supervised classification requires a training set of measurement (feature value) vectors as well as an associated labeling assigning each feature vector to a particular class. This training dataset is used to create a classifier with the hope that it will perform well on unseen data which is similar to the training data. Of course, an important assumption here is that unseen data will be drawn from a similar distribution to the training set; hence one should make all practical efforts to ensure that this is the case.

Considering the task of assigning a class label c, taken from a set of possible labels C, to a feature vector x, we may compute the posterior probability of x belonging to c using Bayes' rule:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{1}$$

where $P(x) = \sum_{c \in C} P(x \mid c)P(c)$ and $P(c)$ is the prior probability of class c. Assuming no compelling prior information otherwise, it is common to assume that $P(c)$ for each class is equal to $1/|C|$. Although we do indeed have some *a priori* knowledge of these values by our labeled training data, we have opted to leave them uniform. This more evenly distributes classifier accuracy over each class, yielding an increase in average per-class accuracy at the expense of overall accuracy. The effect upon the three-way classification was small: an increase of 0.41% in mean per-class accuracy with a 0.16% decrease in overall accuracy. In the four-class instance this gave an increase in average per-class accuracy of 0.96% with a tradeoff of 1.04% in overall accuracy.

For multiclass labeling, the label assigned to x will simply be the value c for which $P(c \mid x)$ is the largest. One may observe that in fact this implies that our algorithm apply EM to the data independently for each class, associating x with the class for which the trained classifier fits best. The only information still needed in order to perform classification are the class-conditional densities $P(x \mid c)$. These will be estimated, as is often the done, by using a mixture model; the model will be fit automatically by EM.

4.2. Model Choices and Parameters

As EM is a generic algorithm, we must choose an underlying model with which to describe our data as well as choose all parameters for said model which are not automatically determined. For our Mixture of Gaussians (MoG) model, this amounts to choosing K, the number of Gaussians in the mixture. Modeling with a mixture of Gaussians is an appropriate choice for data which is created by randomly perturbing prototypes. In this way a complex feature distribution can be usefully approximated by a simpler, finite mixture. Analysis of our data through marginal and joint histograms of feature values have convinced us this model is a reasonable choice for our task. We further restrict our model to independent Gaussians over each feature (zero-covariance) in order to circumvent the numerical precision problems inherent in fitting an arbitrary MoG model over all features. This restriction had no appreciable effect on measured performance: comparing the results of our leave-one-out test using independent as opposed to arbitrary

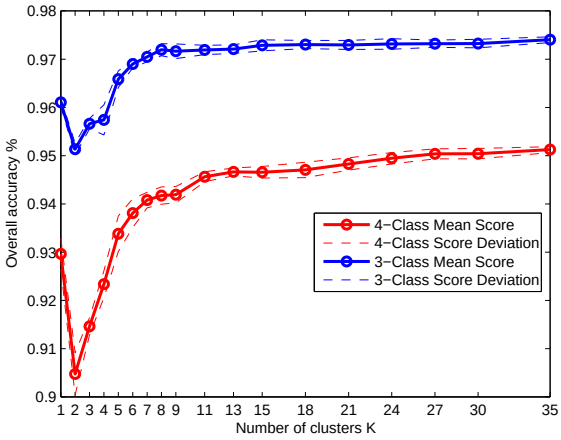


Figure 2. Mean overall MCCV classification accuracy over 10 random partitionings of the training data as different values of K are selected.

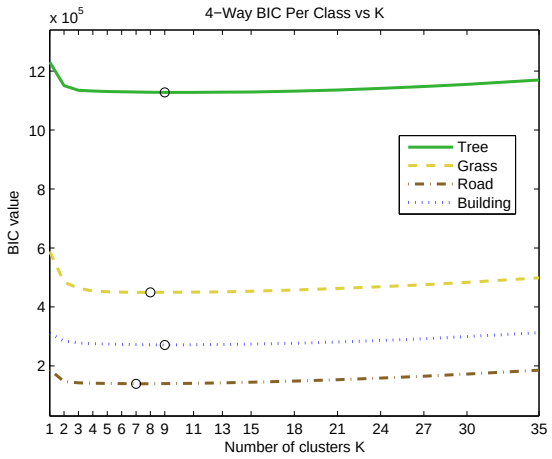


Figure 3. Mean BIC value calculated using 10 random partitionings of the training data for each class as different values of K are selected. The local minima have been marked.

distributions using identical K values showed an overall difference less than 0.2% in overall classification accuracy in both three- and four-way classification. Furthermore, we restricted the variance values of these Gaussians to at least .25 for each feature (recall that our features have integer values on the range $[0,255]$). This is to avoid the case where a Gaussian is fit to a very few feature values with variance approaching zero, leading to degenerate (infinite) likelihoods.

We model the probability of observing a set of measurements x given labeling c as

$$p(x \mid c) = \sum_{j=1}^{K} P_j G(x \mid \mu_j, \Sigma_j) \tag{2}$$

The parameters for each of the Gaussians, (μ_j, Σ_j), as well as their respective mixing coefficients P_j will be iteratively determined by EM automatically after manually choosing the remaining model parameter K, the number of components in the mixture.

Unfortunately the choice of K is not a completely straightforward task. The model needs to describe the data as accurately as is practical, yet one may always add additional Gaussians to create a new model which performs at least as well. At some point the overfitting of an overly complex model must outweigh the benefit gained from a closer fit. Although research is ongoing in the area of automatically determining an optimal value for K, we have chosen one strong candidate criteria, the *Bayesian Information Criterion* (BIC), to aid in deciding the best value as described in [8]. Visual and numerical classification results justify this decision. BIC, like similar metrics such as the Akaike Information Criterion, weigh increase in training data log-likelihood against increased model complexity to yield a single function which may be minimized in order to automatically determine appropriate parameter values.

Training data log-likelihoods were taken from tests using *Monte Carlo Cross-Validation* (MCCV) [15]. MCCV is a technique for cross-validating training data using the average of a series of tests over random partitions of the data. This method of testing is particularly suited to evaluating model parameters, as overfitting problems should be more readily apparent relative to other standard train-test methods such as v-fold Cross-Validation. Although Smyth reports successfully using MCCV to identify optimal values for K [15], we found that MCCV alone was unable to identify a reasonable value. Instead, we used the log-likelihood values resulting from the MCCV tests in our calculation of BIC, which yielded quite reasonable results.

Shown in Figure 2 are the results from MCCV testing using equal values of K for each class. As can be seen, even at large values of K, slight increases in accuracy are observed. Thus we cannot rely upon MCCV accuracy results alone. Instead, we use the log-likelihoods resulting from these tests to compute the per-class BIC values plotted in Figure 3. Minimizing the BIC gave reasonable values of K for each class which we used for all results discussed in Section 5. Finally, we may compare various values of K by examining the resulting fit when trained upon our data. Figure 4 shows the resulting distribution upon the lidar return intensity feature after training the building class with various values of K. Clearly, the higher values of K fit the data better than the lowest, although the difference between the highest values is much less significant. In the 4-class case, the values of K selected were 9, 9, 7, and 8 for buildings, trees, road, and grass, respectively. In the 3-

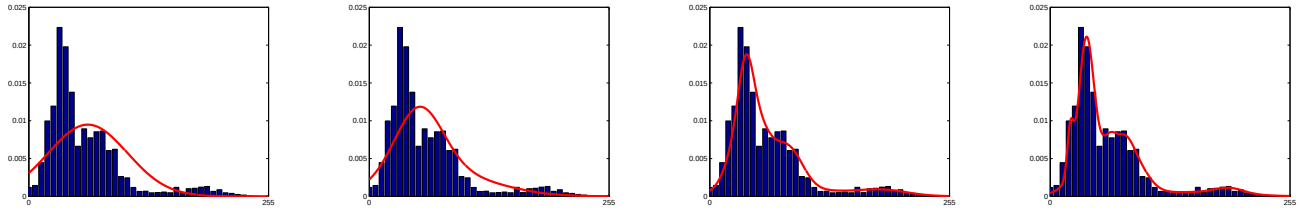


Figure 4. The fit obtained upon the lidar return intensity feature for the building class. Left to right: K equal to 1, 3, 9, and 30.

class case, values of 9, 9, and 8 for buildings, trees, and road-grass were used.

5. Results and Analysis

All accuracy results presented in this section are created using leave-one-out testing. For leave-one-out tests, 9 of our 10 labeled regions serve as the source of training samples used as input to our EM algorithm. The accuracy of the resulting classifier is then measured upon all labeled points within the remaining (held out) region. For labeled data in the 9 training regions, $1/8$ of the points are randomly sampled as training input. We have found that in practice downsampling the training data to as little as one tenth the available size has no significant effect upon our results. When calculating the overall leave-one-out accuracy for all 10 regions, we take the sum of the correctly classified points in each region and divide it by the total number of points in all regions, *i.e.* smaller regions contribute less to the overall score.

We report two types of accuracy: sample- and class-weighted. Sample-weighted accuracy is simply the percentage of correctly classified points. Class-weighted accuracy is the mean percentage of correctly classified points from each class. That is, per-class accuracies are computed and then averaged to compute class-weighted accuracy. Sometimes a certain class is the most interesting yet it is relatively uncommon in the data. Class-weighted accuracy is useful in identifying cases where a rare classes is poorly classified yet overall accuracy is still high, which might be undesirable.

We now present our results both visually and numerically. Recall that for these tests 9, 9, 7 and 8 Gaussians were fit to the building, tree, road, and grass classes respectively (9, 9 and 8 Gaussians for building, tree and road-grass in the 3-class case). Table 1 presents leave-one-out test accuracy for each region and overall. We achieve an overall sample-weighted accuracy of better than 94% in the 4-class case and better than 97% in the 3-class case. The overall class-weighted accuracy results are similar, however some of the test regions performed significantly poorer than others. This can be attributed to localized differences in features in these two regions. Luminance and lidar return intensity both varied significantly in some classes over various areas. Particularly, the grass class (in the 4-way classification) and

Region	5 features, 4 classes		3 features, 3 classes	
	class-weighted	sample-weighted	class-weighted	sample-weighted
baskin	74.21 %	92.95 %	86.17 %	95.21 %
c8	93.26 %	94.62 %	98.02 %	97.80 %
crown	91.60 %	95.47 %	95.88 %	95.62 %
efh	94.32 %	96.27 %	96.52 %	97.95 %
fsh	93.60 %	94.52 %	96.73 %	96.59 %
oakes	93.82 %	93.87 %	96.35 %	98.44 %
pc	89.39 %	88.25 %	96.08 %	97.08 %
porter	95.48 %	95.75 %	97.44 %	96.21 %
sh	76.25 %	93.83 %	86.44 %	95.25 %
ta	94.46 %	94.35 %	98.66 %	98.54 %
Overall	93.83 %	94.31 %	96.98 %	97.12 %

Table 1. Leave-one-out test accuracy: overall and for each of the 10 regions.

5/4	Tree	Grass	Road	Bldg.	Error I
Tree	96.61	0.24	1.09	2.06	3.39
Grass	0.35	91.44	6.92	1.30	8.57
Road	0.59	7.41	91.39	0.61	8.61
Bldng.	3.81	0.06	0.26	95.87	4.13
Error II	4.75	7.71	8.27	3.97	

3/3	Tree	Bldng.	Road-Grass	Error I
Tree	96.36	2.91	0.73	3.64
Bldng	3.47	96.44	0.09	3.56
Road-Grass	0.59	1.27	98.14	1.86
Error II	4.06	4.18	0.82	

Table 2. Classification accuracy and Type I/II error results organized by class, for 5 features/4 classes and 3 features/3 classes. These are the cumulative results over all 10 test regions.

the merged road-grass class (in the 3-way classification) were poorly classified in the two regions due to these differences. This is understandable as the grass points in these two regions were significantly darker than in other areas. Sample-weighted accuracy also dropped to 88.25% in one of the ten regions during the 4-way classification. This is due to a relative abundance of the most commonly confused classes, road and grass, in this region. Since these classes were merged in the 3-class instance, this disparity is not present in the 3-way results.

Table 2 presents the confusion matrix for the overall leave-one-out results. As mentioned, in the 4-class case road and grass are the most commonly confused classes.

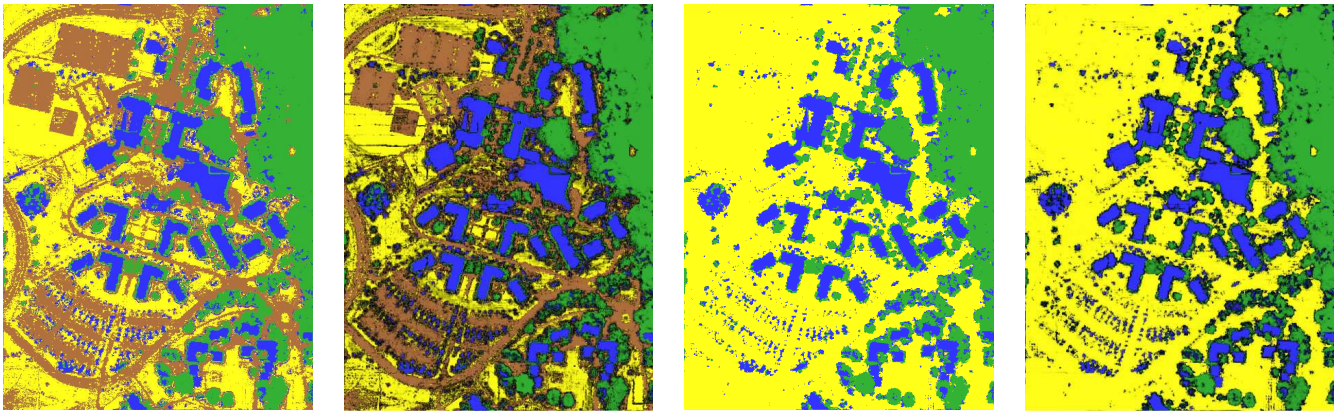


Figure 5. Leave-one-out classification results for the College 8 region: (far left) 4-way classification using 5 features; (2nd from left) 4-way confidence-rated classification using 5 features; (2nd from right) 3-way classification using 4 features; (far right) 3-way confidence-rated classification using 4 features.

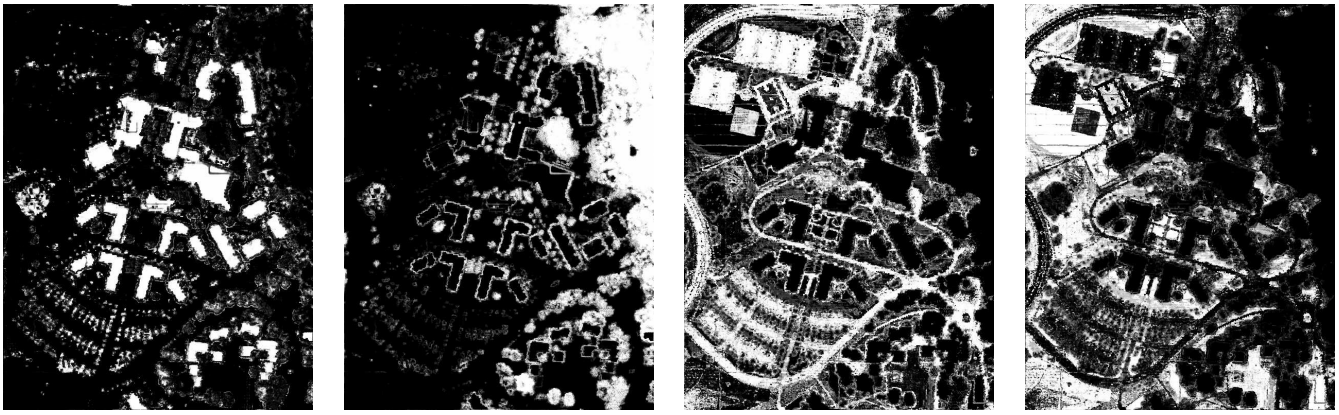


Figure 6. Per-class density images for each class on the College 8 region using the leave-one-out test with 4 classes; left to right the images show the probabilities for building, tree, road and grass.

Another significant source of error in the 4-way classification were building points misclassified as tree. Buildings were much easier to discern from the cumulative road-grass class than from the road and grass classes independently. This leaves confusion between buildings and trees as the remaining significant source of error in the 3-way classification.

As mentioned in Section 2, these accuracies are significantly higher than our previously reported results using the EM algorithm [3]. Previously reported accuracies ranged from 66% to 84% over the 10 testing regions, we now obtain higher than 94% overall accuracy on these regions. Also, when applying the previous EM classifiers to newer feature data, we noticed that the results were unstable and had difficulty reobtaining similar accuracy rates. Our newer EM classifier is more stable and accurate. This increase in stability and accuracy can primarily be attributed to our improved feature data and automatic parameter selection. In comparing our current EM results to our previous results using SVM [11] we observe that both accuracy rates and sources of error are similar. Specifically, we observe that in both cases confusion between road and grass is the primary source of error, with building points misclassified as tree being another significant source of error. This leads us to conclude that classification using EM is not our only viable choice. From this comparison we also conclude that the most promising route to improving classification accuracy is in further refinement of our feature data. Particularly, our current height-based features are somewhat ineffectual in discerning between road and grass points, and, in certain cases, between tree and building points.

We also present visual classification results for one of our 10 subregions as well as for our entire dataset, with and without uncertainty. In these figures points classified as building are shown in blue, tree points are shown in green, and road and grass points are shown in brown and yellow, respectively. For 3-way output the road-grass points are shown in yellow. Also, in confidence-rated output, colors are scaled by their classification certainty, that is, darker points are to be trusted less. This confidence rating is taken

as the ratio of the difference between the most and next most probable to the most probable classification. Figure 5 shows visual classification results for one of our ten subregions. Recall that less than 5% of points in the remaining 9 subregions was used for training. Buildings are clearly distinguishable, as are trees and the surrounding roads. The most evident source of error, visually, are building edges labeled as tree, as well as tree points mislabeled as building. In the images with uncertainty output these are typically marked by lower confidence. This visual evaluation corresponds well with the insight gained from the confusion matrix presented in Table 2. Also evident is that in certain areas no good labeling for our algorithm exists, *e.g.* cars in parking lots which are classified as buildings and tennis courts which are classified as roads. These points, too, are assigned low classification confidence. Further research is necessary in order to elegantly handle these cases. One possibility might be to assign an 'other' labeling to all objects for which no useful classification exists.

Figure 7 presents a visualization of the classification results for all the points within our entire dataset. We observe the same level of visual quality in these results as obtained in the testing subregions. Specifically, roads, buildings and areas of forest are clearly visible, with the confidence-rated output guiding the eye toward the most reliable portions of the classification. We also notice some spatio-temporal inconsistencies in these results. For example, in the upper-left there is an area of mixed building and tree points with low confidence. This area contained trailers during lidar collection which were not present during aerial image collection. The classifier used for this output was trained upon a random sampling of $1/8$ the labeled data in all 10 subregions. This training data comprised less than 1% of the total number of classified points. Clearly our results extend from the testing subregions into nearby geographical areas quite successfully. Only further comparison with additional datasets will tell the limits of the extensibility of classifiers from one region to another.

6. Conclusions and Future Directions

We have presented a method of applying the EM algorithm to the classification of aerial lidar data. We do so quite successfully with an overall accuracy rate of greater than 94%, which is better than our previously reported results. Our visual classification results demonstrate intuitively the effectiveness of our technique. The confidence-rated visual output clearly demonstrates the relative strengths and weaknesses in our classification. Comparing our results reported here with previous classification efforts upon the same data lends valuable insight into the relative strengths and weaknesses of these approaches. There are many future possibilities for extending our work. These include the use of additional features, extension into different sets of classes, the enforcement of spatial coherency constraints, as well as many post-processing possibilities such as building footprint extraction and virtual walk-throughs.

Acknowledgements

This research is partially supported by the Multidisciplinary Research Initiative (MURI) grant by U.S. Army Research Office under Agreement Number DAAD19-00-1-0352, the NSF grant ACI-0222900, and the NSF-REU grant supplement CCF-0222900.

References

[1] P. Axelsson. Processing of laser scanner data -algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):138–147, 1999. 1, 2

[2] J. B. Blair, D. L. Rabine, and M. A. Hofton. The laser vegetation imaging sensor: a medium altitude, digitsation-only, airborne laser altimeter for mapping vegetaion and topography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:115–122, 1999. 1

[3] A. P. Charaniya, R. Manduchi, and S. K. Lodha. Supervised parametric classification of aerial lidar data. *IEEE workshop on Real-Time 3D Sensors*, pages 25–32, June 2004. 2, 6

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39(1):38, 1977. 3

[5] S. Filin. Surface clustering from airborne laser scanning. In *ISPRS Commission III, Symposium 2002 September 9 - 13, 2002, Graz, Austria*, 2002. 2

[6] C. Frueh and A. Zakhor. Constructing 3D city models by merging ground-based and airborne views. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003. 1

[7] N. Haala and C. Brenner. Extraction of buildings and trees in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):130–137, 1999. 2

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001. 4

[9] J. Hu, S. You, and U. Neumann. Approaches to large scale urban modeling. *IEEE Computer Graphics and Applications*, 23(6):62–69, November/December 2003. 1

[10] K. Kraus and N. Pfeifer. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:193–203, 1998. 1

[11] S. Lodha, E. Kreps, D. Helmbold, and D. Fitzpatrick. Aerial LiDAR Data Classification using Support Vector Machines (SVM). In *Third International Symposium on 3D Data Processing, Visualization and Transmission*, 2006. 2, 6

[12] H.-G. Maas. The potential of height texture measures for the segmentation of airborne laserscanner data. *Fourth International Airborne Remote Sensing Conference and Exhibition*,

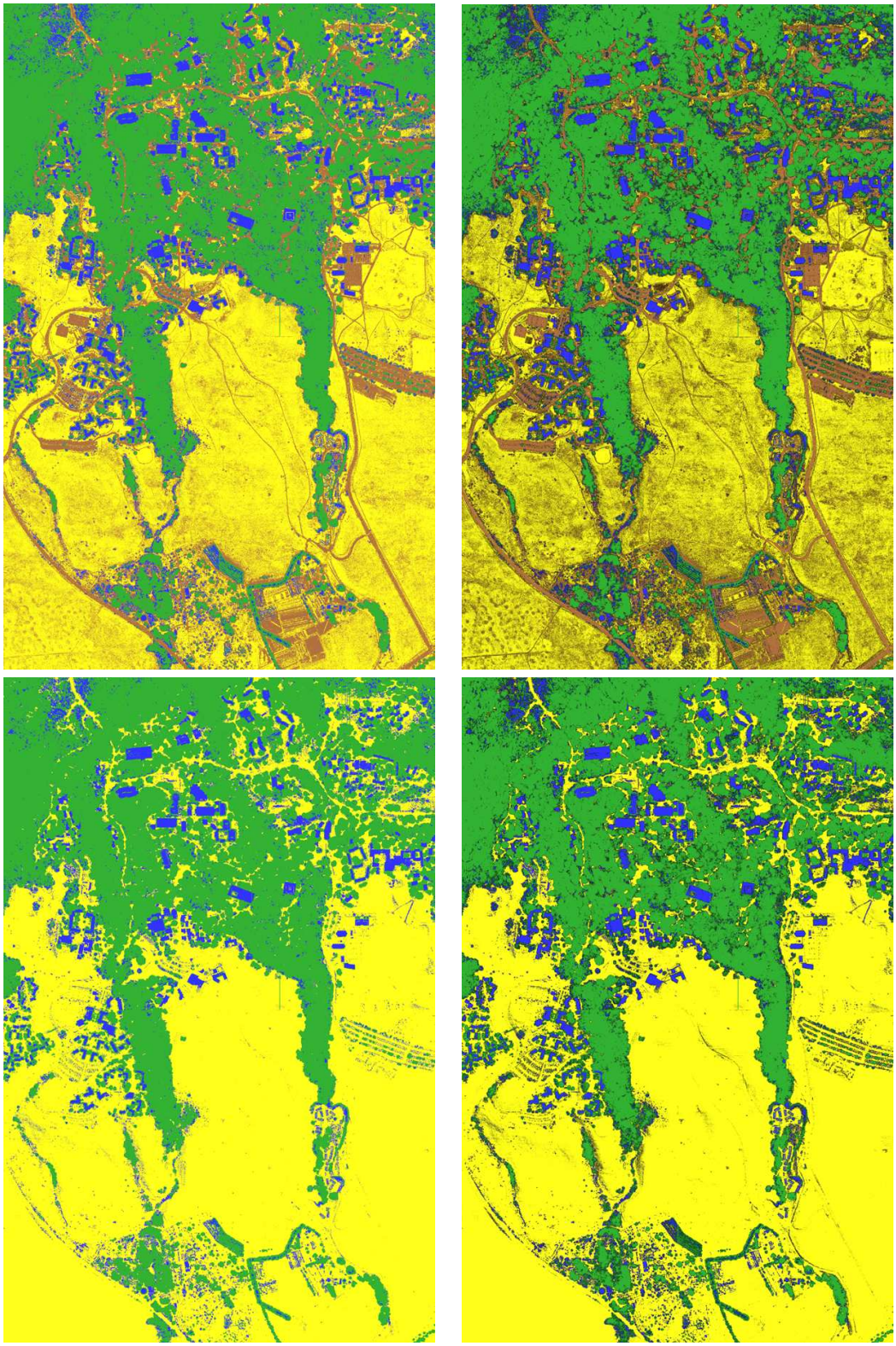


Figure 7. Classification of the entire dataset using (upper half) 4 classes and 5 features and (lower half) 3 classes and 3 features, without (left) and with (right) confidence rating.

21st Canadian Symposium on Remote Sensing:Ottawa, Ontario, Canada, 1999. 2

[13] W. Ribarsky, T. Wasilewski, and N. Faust. From urban terrain models to visible citites. *IEEE Computer Graphics and Applications*, 22(4):10–15, July 2002. 1

[14] G. Sithole and G. Vosselman. Comparison of filtering algorithms. In *ISPRS Commission III, Symposium 2002 September 9 - 13, 2002, Graz, Austria*, 2002. 1, 2

[15] P. Smyth. Clustering using monte carlo cross-validation. In *Knowledge Discovery and Data Mining*, pages 126–133, 1996. 4

[16] G. Vosselman. Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing*, XXXIII, Amsterdam, 2000, 2000. 1

[17] S. You, J. Hu, U. Neumann, and P. Fox. Urban site modeling from lidar. In *Second International Workshop on Computer Graphics and Geometric Modeling*, 2003. 1