Coupled CRFs for estimating the underlying ground surface from airborne LiDAR data

Wei-Lwun Lu, Kevin Murphy, James J. Little, Alla Sheffer, Hongbo Fu
University of British Columbia, Computer Science Technical Report TR-2008-05

Abstract—Airborne laser scanners (LiDAR) return point clouds of millions of points imaging large regions. It is very challenging to recover the bare earth, i.e., the surface remaining after the buildings and vegetative cover have been identified and removed; manual correction of the recovered surface is very costly. Our solution combines classification into ground and non-ground with reconstruction of the continuous underlying surface. We define a joint model on the class labels and estimated surface, $p(\mathbf{c}, \mathbf{z}|\mathbf{x})$, where $c_i \in \{0, 1\}$ is the label of point i (ground or non-ground), z_i is the estimated bare-earth surface at point i, and x_i is the observed height of point i. We learn the parameters of this CRF using supervised learning. The graph structure is obtained by triangulating the point clouds. Given the model, we compute a MAP estimate of the surface, $\arg\max p(\mathbf{z}|\mathbf{x})$, using the EM algorithm, treating the labels $\mathbf{c}$ as missing data. Extensive testing shows that the recovered surfaces agree very well with those reconstructed from manually corrected data. Moreover, the resulting classification of points is competitive with the best in the literature.

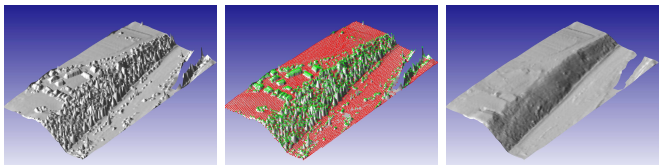
1 Introduction

It is now possible to acquire large amounts of high quality 3D range data, in the form of point clouds, using a variety of devices, from laser range scanners to LiDAR (Light Detection and Ranging). When applied to outdoor scenes, the most widely studied problem is to to classify the points into object classes, such as "ground", "building", "tree" or "shrubbery" [1], or "surface", "linear structure" or "porous" [2]; this is useful for mobile robot navigation. We are interested in a slightly different problem, namely estimating the underlying surface of the earth from an airborne laser scanner. This involves classifying points as ground or non-ground, and then estimating the height of the surface underneath the non-ground points. This is sometimes described as estimating the "bare earth" surface [3]. See Figure 1 for an illustration.

There is considerable commercial interest in the bare earth problem. However, it is quite challenging. As illustrated in Figure 2, we cannot simply declare points with high elevation to be non-ground, because of hills. Similarly, we cannot simply declare points in flat regions to be ground, because of roof-tops. A variety of heuristics have therefore been developed to tackle the problem [3].

In this paper, we propose to use machine learning methods to solve the problem. More precisely, we propose a joint probabilistic model for the labels c_j and the estimated underlying surface heights z_j of each point j, conditional on the observed data x_j: i.e., we learn a model of $p(\mathbf{c}, \mathbf{z}|\mathbf{x})$ from labeled training data. We model correlation between the neighboring points using a random field. Since this is conditional on the observed data (i.e., a discriminative model), it is often called a conditional random field (CRF) [4], [5].

Most previous work on CRFs has focused on modeling the correlation between discrete labels (i.e., the c_j's). We extend this by also modeling the correlation between



(a) (b) (c)

Fig. 1. (a) The original 3D mesh, the Delaunay triangulation of the original 3D point cloud. (b) The classification of 3D points to ground/non-ground (red/green). (c) The estimated bare-earth surface.

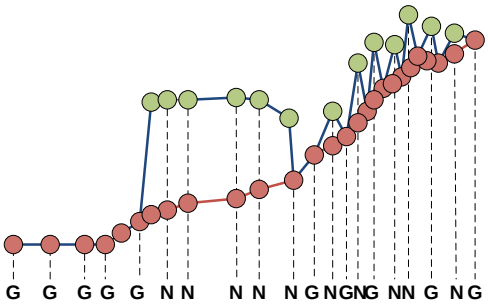


Fig. 2. Illustration of the problem along a 1D slice from our data, illustrating a building and some trees (note that these semantic labels are not present in the data and are not used by the model). Red is the true bare earth surface which we wish to estimate, z_i; green is the surface which LiDAR samples, x_i. (In places where $z_i = x_i$, we just show a red dot.) Letters are the estimated states c_i, either ground or non-ground.

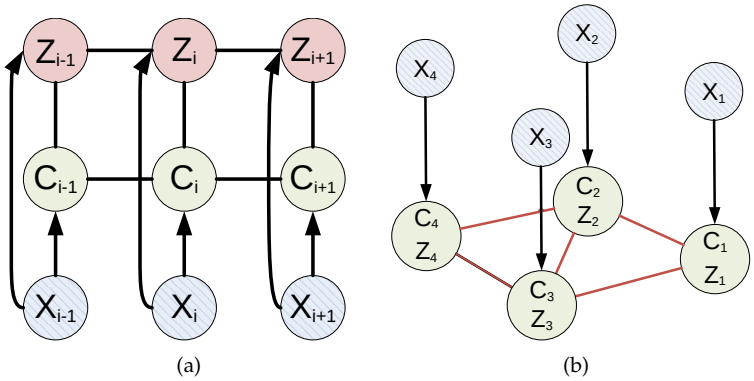


Fig. 3. Sketch of CRF. (a) 1D version. Note that each c_i can be estimated based on the x_i's in some region around i, but we only show a direct connection from x_i for simplicity. (b) 2D version. Green dots represent unknown z_i or c_i's. Red circles represent known x_i's. We connect each latent point to its nearest neighbors, using a Delaunay triangulation.

continuous heights (i.e., the z_j's). In particular, we enforce the underlying heights of neighboring non-ground points to be similar (representing a smooth surface), whereas points labeled as ground should be close to the observed data. Since the $\mathbf{z}$ and $\mathbf{c}$ fields are coupled, we perform MAP estimation using EM. In the E step, we estimate the labels $\mathbf{c}$, and in the M step, we estimate the heights $\mathbf{z}$. We demonstrate that this learning-based approach outperforms the previous best systems (based on manually tuned heuristics) at both the classification problem, and at the surface estimation problem. Furthermore, our solution is fast, and easy to port to other data sets.

2 Related Work

[6] considered the problem of segmenting 3D point clouds using a generative model, where each region consists of a parametric model of the surface, such as planes, cylinders, cones, etc. Although we use segmentation (specifically, the method of [7]) in our approach (in order to derive segment-based features), we are not interested in segmentation per se.

The problem of classifying 3D point clouds was studied in [1] using a max-margin Markov network. This was trained in a discriminative way to classify 3D points into 4 classes: ground, trees, grass and building. The features used are somewhat like SIFT, in that they consist of histograms which count the number of points in certain regions around the point of interest. The features were combined with a quadratic kernel. The approach of [2] was simpler, due to the need to satisfy real-time constraints. They used a Gaussian mixture model to classify points based on simple local statistics. However, neither paper deals with the problem of estimating the underlying surface (the bare earth problem).

Many heuristic approaches to the bare earth problem are reviewed and compared in [3]. This paper evaluated many methods (in terms of their ability to classify ground vs non-ground) on a benchmark dataset, which we will call the "Sithole data". We will compare our method to these in Section 5. We will also compare to the more recent approach of [8]. This uses a CRF on the discrete label field, but does not model correlations amongst the heights. Also, it uses manually-tuned methods, rather than machine learning.

3 Model

3.1 Overview

Let $c_i \in \{0, 1\}$ represent the class label (ground and non-ground) at location i, $z_i \in \mathbb{R}^+$ represent the estimated bare earth height, and $x_i \in \mathbb{R}^+$ represent the observed height. Our proposed model is illustrated in Figure 3. This corresponds to the joint conditional density model:

$$p(\mathbf{z}, \mathbf{c}|\mathbf{x}) \propto \exp(-J(\mathbf{z}, \mathbf{c}, \mathbf{x})) \tag{1}$$

where $J(\mathbf{z}, \mathbf{c}, \mathbf{x})$ is the "energy" or cost of configuration $\mathbf{z}, \mathbf{c}$ when the data is $\mathbf{x}$. (We formulate the problem in terms of energy rather than probability to avoid the need to explicitly normalize the distribution.) We define the energy as follows:

$$J(\mathbf{z}, \mathbf{c}, \mathbf{x}) = J_z(\mathbf{z}, \mathbf{c}, \mathbf{x}) + J_c(\mathbf{c}, \mathbf{x}) \tag{2}$$

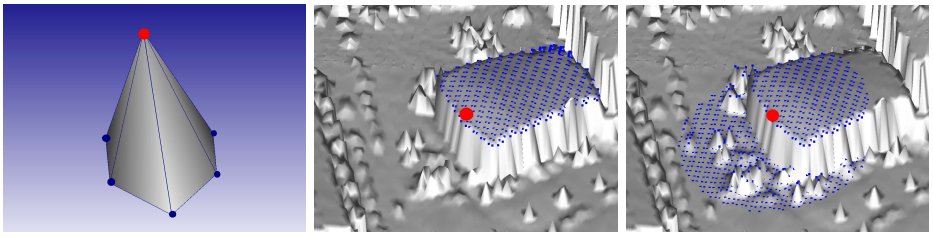
where $J_z \propto -\log p(\mathbf{z}|\mathbf{c}, \mathbf{x})$ and $J_c \propto -\log p(\mathbf{c}|\mathbf{x})$. We explain each of these terms below.

3.2 Height field

We define the energy on the estimated heights as follows:

$$J_z(\mathbf{z}, \mathbf{c}, \mathbf{x}) = \sum_i \lambda_1 I(c_i = 1)(z_i - \frac{1}{|n_i|} \sum_{j \in n_i} z_j)^2 + \lambda_0 I(c_i = 0)(z_i - x_i)^2 \tag{3}$$

where n_i are the neighbors of node i in the graph constructed from the Delaunay triangulation of the points. If point i is non-ground, the first term enforces that z_i is close to the average of its neighbors; if i is ground, the second term enforces that z_i is close to x_i. The relative strengths of these two terms are controlled by λ_1 and λ_0;



(a) vertex-based (b) segment-based (c) disc-based

Fig. 4. Features with different neighborhood: (a) vertex-based features. (b) segment-based features. (c) disc-based features. In all figures, the red point represents the current vertex v_i and the blue points represent its supporting neighborhood.

No.	Type	Description
1	Vertex	average arc tangent to neighbors
2	Vertex	minimum arc tangent to neighbors
3	Vertex	maximum arc tangent to neighbors
4	Vertex	z difference to global z mean
5	Vertex	outlier confidence
6	Segment	segment z variance
7	Segment	segment relative height
8	Segment	segment average z difference to higher region neighbors
9	Segment	segment average z difference to lower region neighbors
10	Segment	percentage of higher neighbors
11	Segment	percentage of lower neighbors
12	Segment	number of vertices
13	Disc	z difference to lowest neighbors

TABLE 1

these correspond to the smoothness of the field and the fidelity to the data, respectively, and are estimated using cross validation.

It will be helpful for later if we rewrite the above equation in matrix-vector notation. Let $\mathbf{L}$ be the $n \times n$ Laplacian matrix defined as

$$L_{i,j} = \begin{cases} -1 & \text{if } i = j \\ \frac{1}{n_i} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Let $\mathbf{C} = \text{diag}(\mathbf{c})$. Then the above cost function becomes

$$J_z(\mathbf{z}, \mathbf{c}, \mathbf{x}) = \lambda_1 \mathbf{z}^T (\mathbf{L}^T \mathbf{C} \mathbf{L}) \mathbf{z} + \lambda_0 (\mathbf{z} - \mathbf{x})^T (\mathbf{I} - \mathbf{C})(\mathbf{z} - \mathbf{x}) \quad (5)$$

This model is similar to a "switching Gaussian MRF" model, since the edge potentials are turned "on" and "off" based on the labels $\mathbf{c}$.

3.3 Label field

We can model correlation between the labels using a binary (two-state) CRF as follows:

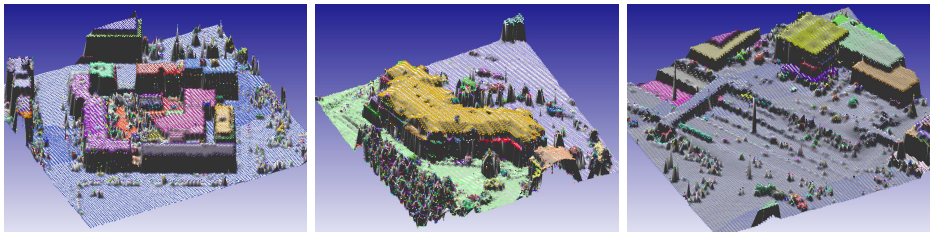
$$J_c(\mathbf{c}, \mathbf{z}) = \sum_i \phi_i(c_i, \mathbf{x}) + \sum_{j \in n_i} \phi_{ij}(c_i, c_j, \mathbf{x}) \quad (6)$$

where ϕ_i is the local evidence potential and ϕ_{ij} is an edge potential. By imposing appropriate constraints on the edge potentials ϕ_{ij}, we can ensure that we can compute the exact MAP estimate $\arg\max p(\mathbf{c}|\mathbf{x})$ using graph cuts [9]. This was the approach adopted in [1], [8]. However, we have found that we can achieve equally good performance using just the local evidence terms, i.e., using an iid classifier, and ignoring the edge potentials. This is probably because correlation amongst the c_i's is indirectly modeled via correlation of the z_i's.

The local evidence term $\phi_i(c_i|\mathbf{x})$ is just a binary classifier. We learned this using standard supervised learning techniques. In particular, we used the GentleBoost algorithm [10] with decision trees as the weak classifiers. The 13 features are shown in Table 1. We use 30 decision trees, and each tree can have up to 5 splits. (These parameters were chosen by cross validation.) We convert the output of the boosted classifier to a probability using a logistic regression model, trained on validation data [11]: $p(c_i = 1|\mathbf{x}) = \sigma(a\phi_i + b)$, where $\sigma(u) = 1/(1 + e^{-u})$ is the sigmoid function. This is helpful when we consider EM (see Section 4).

The features we used can be classified into three categories: vertex-based, segment-based, and disc-based features. Vertex-based features are computed by using the current vertex v_i and its neighboring vertices v_j based on the Delaunay triangulation (Figure 4 (a)). These features are used to capture local information around a vertex. For example, the vertices of tree-tops usually have larger average arc tangent to its neighbors than vertices of ground points. Vertex-based features can effectively distinguish between tree and ground vertices, but cannot distinguish between buildings and ground.

In order to capture information provided by a larger context, we also use segment-based features. We use the graph-based segmentation algorithm introduced by



(a) (b) (c)

Fig. 5. Segmentation of point cloud. (a) complex buildings. (b) a large building. (c) buildings and a bridge. We can observe that the segmentation algorithm works in most of the cases except that it partitions the ground and bridge into the same group in (c).

Felzenszwalb and Huttenlocher [7]. The graph is constructed by Delaunay triangulation, and the edge weight of two neighboring vertices is defined as the slope between them. Figure 5 provides some examples of the segmentation. We can see that this algorithm does a good job at partitioning the point cloud into meaningful groups. Furthemore, this algorithm is very fast. After obtaining the segmentation, various features are computed: see Figure 4 (b). For instance, the relative height of a segment is the average elevation difference between its boundary vertices to the lowest neighboring vertices within a specified distance. Note that all vertices of the same segment have the same segment-based features.

The segmentation algorithm sometimes does not partition different man-made structures into different groups. For example, in Figure 5(c), the segmentation algorithm mis-partitions the bridge and ground into the same group. In order to tackle this problem, we introduce the disc-based features. These are computed by using a disc-shaped neighborhood with a specified radius (Figure 4 (c)). Currently, the only disc-based feature we use is the elevation difference between the current vertex v_i and the lowest vertex within a range; this feature can effectively distinguish between points on the ground and on bridges.

4 Inference

We now discuss how to compute a MAP estimate of $p(\mathbf{z}|\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{z}, \mathbf{c}|\mathbf{x})$. We use the EM algorithm for this, treating the labels as missing data. We initialize by running the classifier $p(\mathbf{c}|\mathbf{x})$.

4.1 M step: estimating the surface

From Equation 5, the expected negative complete data log likelihood is given by

$$\begin{aligned} -E_{\mathbf{C}} \log p(\mathbf{z}, \mathbf{c}|\mathbf{x}) &= E_{\mathbf{C}} J_z(\mathbf{z}, \mathbf{c}, \mathbf{z}) \\ &= \lambda_1 \mathbf{z}^T (\mathbf{L}^T \mathbf{W} \mathbf{L}) \mathbf{z} + \lambda_0 (\mathbf{z} - \mathbf{x})^T (\mathbf{I} - \mathbf{W})(\mathbf{z} - \mathbf{x}) \end{aligned} \quad (7)$$

where $\mathbf{W} = E[\mathbf{C}] = \text{diag}(p(c_1 = 1), \ldots, p(c_n = 1))$. Maximizing this is a weighted least squares problem, which can be efficiently solved using Laplacian mesh smoothing [12]. Specifically, we just solve the following system of linear equations:

$$w_i \lambda_1 \left(z_i - \frac{1}{n_i} \sum_{j \in \mathcal{N}_i} z_j \right) = 0 \quad (8)$$

$$(1 - w_i) \lambda_0 (z_i - x_i) = 0 \quad (9)$$

where $w_i = p(c_i = 1|\mathbf{x})$ is the probability that vertex i belongs to non-ground. In matrix-vector form this becomes

$$\begin{pmatrix} \mathbf{WL} \\ \mathbf{I} - \mathbf{W} \end{pmatrix} \mathbf{z} = \begin{pmatrix} \mathbf{0} \\ (\mathbf{I} - \mathbf{W})\mathbf{x} \end{pmatrix} \quad (10)$$

4.2 E step: Re-classifying Vertices

We now discuss how to reestimate the labels $\mathbf{c}$ from the heights $\mathbf{z}$. The cost function factorizes into a sum of terms, one per c_i:

$$J(\mathbf{c}) = \sum_i \psi_i(c_i) \quad (11)$$

where

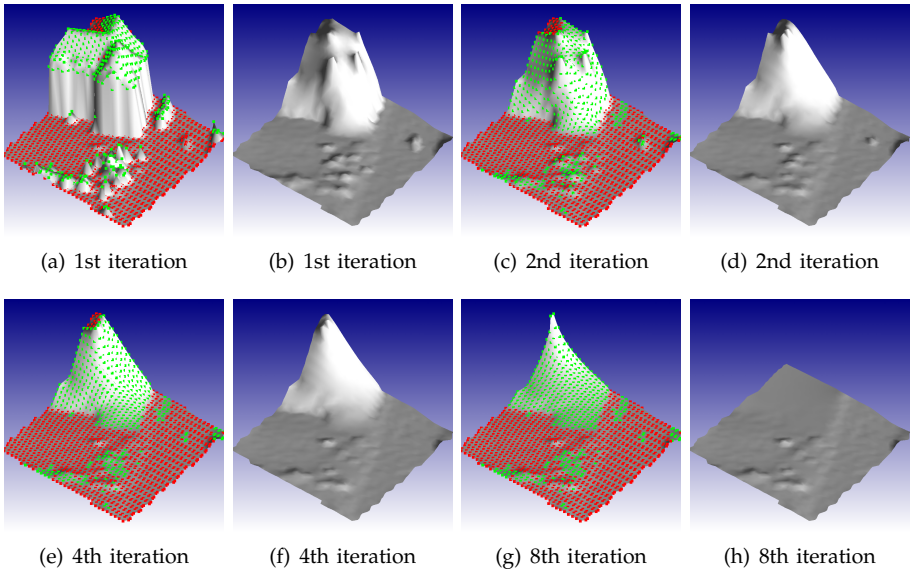
$$\psi_i = \begin{pmatrix} \lambda_1 (z_i - \frac{1}{|n_i|} \sum_{j \in n_i} z_j)^2 + \phi_i(1) \\ \lambda_0 (z_i - x_i)^2 + \phi_i(0) \end{pmatrix} \quad (12)$$

Hence $p(\mathbf{c}|\mathbf{x}, \mathbf{z}) = \prod_i p(c_i|z_i, x_i)$ where

$$\begin{aligned} p(c_i = 1|\mathbf{x}, \mathbf{z}) &= \frac{e^{-\psi_i(1)}}{e^{-\psi_i(1)} + e^{-\psi_i(0)}} \\ &= \frac{1}{1 + e^{\psi_i(1) - \psi_i(0)}} \\ &= \sigma(\psi_i(0) - \psi_i(1)) \end{aligned} \quad (13)$$

where $\sigma(u) = 1/(1 + e^{-u})$ is the logistic function. Essentially we are performing a soft classification of c_i based on whether z_i is more similar to its neighbors, or to the data, and also based on the local evidence ϕ_i. This is similar to the ad hoc weighting scheme proposed by Kraus and Pfeifer [13].

Although this reweighting scheme works quite well, we got better performance by using a learning scheme to model $p(c_i|z_i, x_i)$, rather than relying on exact inference in a possibly incorrect model. Specifically, we use boosted decision trees to train a classifier to predict the labels. (A similar approach was used in [14], which used



(a) 1st iteration (b) 1st iteration (c) 2nd iteration (d) 2nd iteration

(e) 4th iteration (f) 4th iteration (g) 8th iteration (h) 8th iteration

Fig. 6. This figure illustrates the EM algorithm. (a) is the initial classification results obtained from $p(\mathbf{c}|\mathbf{x})$. Red dots represent ground points and green dots represent non-ground points. We can observe that some points on the roof-top are mis-classified as ground points. (b) shows the estimated $\mathbf{z}$ after the first M-step. Since there are some mis-classification on the roof-top, the estimated surface is not correct. (c) shows the results of $p(\mathbf{c}|\mathbf{x}, \mathbf{z})$ of the first E-step. Observe that some of the mis-classified points are re-classified as non-ground because they are not locally smooth. (d) shows the second M-step. (e) shows the 4th E-step. (f) shows the 4th M-step. (g) shows the 8th E-step. (h) shows the 8th M-step. We can see that after 8 iterations of EM, the mis-classification on the roof-top has been corrected.

boosting to do approximate inference in an intractable CRF.) We trained this classifier to map the $\mathbf{z}$ estimated at the first step of the EM algorithm to the correct labels $\mathbf{c}$. We use the same classifier at each subsequent iteration. EM typically converges in 10-20 steps. See Figure 6 for an illustration of the algorithm in action.

5 Experimental Results

In order to evaluate the performance of the proposed algorithm, we test our system on two datasets: the Sithole *et al.* [3] dataset and the Terrapoint dataset [15]. We evaluate both the quantitative and qualitative performance of the proposed algorithm.

We evaluate the quantitative performance of our system by the classification errors and the distance between the extracted and ground-truth bare-earth surface. To measure the distance between the estimated and ground-truth bare-earth surface, we define the distance $dist(p, S)$ between a point p and a surface S as:

$$dist(p, S) = \min_{p' \in S} \|p - p'\| \tag{14}$$

The average distance between surfaces S_1 and S_2 thus can be defined as

$$dist_{avg}(S_1, S_2) = \frac{1}{|S_1|} \int_{p \in S_1} dist(p, S_2) dp \tag{15}$$

where $1/|S_1|$ is the area of S_1. In particular, we use a standard package named Metro [16] to compute the average distance between two 3D meshes.

5.1 Sithole *et al.* Dataset

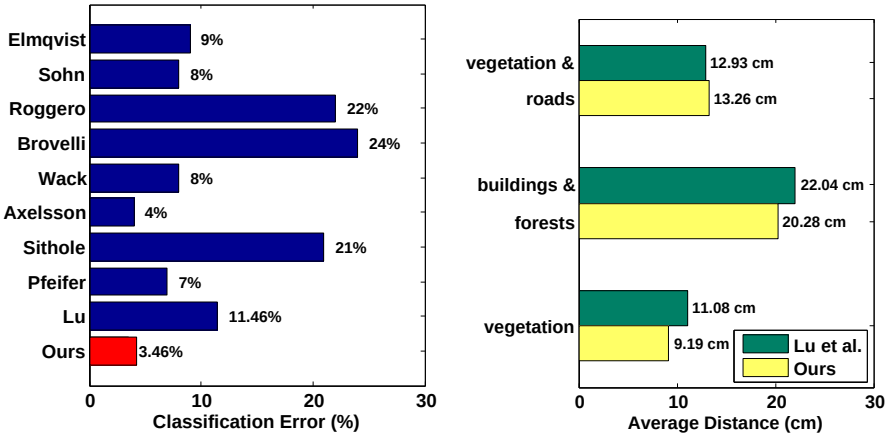
The Sithole *et al.* dataset [3] consists of 15 sites with various terrain characteristics including buildings, steep slopes, bridges, terrain discontinuities, ramps, vegetation on slopes and many others . Sithole *et al.* manually classified each data point and thus the ground-truth labellings are very accurate. The Sithole dataset consists of 384325 3D points.

In Figure 7 (a), we compare our classification accuracy with the state-of-the-art filtering algorithms evaluated by Sithole *et al.* [3]. Since we do not have another independent training set, we run 10-fold cross-validation on the Sithole dataset to produce the classification accuracy. Our classification error percentage, 3.46%, is slightly better than the state-of-the-art hand-tuned algorithms.

Figure 8 (a)-(c) shows the qualitative results of the Sithole dataset. We can observe that the proposed algorithm is robust under different kinds of terrains, including (a) buildings on steep slope, (b) bridge, (c) building. Due to the use of segment-based and radius-disc-based features, the proposed algorithm can reliably dealing with terrains with bridges and buildings on slope, which ware notorious in the Sithole dataset [3].

5.2 Terrapoint Dataset

The Terrapoint data [15] consists of three huge sites with 3 millions of data points. The first and third sites contain vegetation and roads, while the second site is composed of forests, buildings, and cars. Unfortunately, Terrapoint classifies dataset in a conservative way, i.e., they mark



(a) Sithole dataset (b) Terrapoint dataset

Fig. 7. (a) Errors in estimating labels $\mathbf{c}$ on the Sithole dataset, compared with other methods evaluated in [3]. (b) Errors in estimating surface $\mathbf{z}$ on the Terrapoint dataset, compared with Lu *et al*. [8].

few points as ground in order to increase the quality of the bare-earth extraction. As a result, the ground-truth classification is not accurate because many ground points are classified as non-ground.

We use the Terrapoint dataset as the independent test case to evaluate the performance of our classifier trained by using the Sithole *et al*. dataset. We partition the Terrapoint dataset into patches of 50000 points and run our algorithm on each of them. Figure 7 (b) shows the quantitative performance of our algorithm compared to Lu *et al*. [8]. Our algorithm outperforms Lu *et al*. [8] in two of the three dataset, and has very similar performance in the other one.

Figure 8 (d) and (e) shows some qualitative results of our algorithm on the Terrapoint dataset. We can observe that our algorithm can reliable extract the bare-earth surface in both urban and rural areas.

6 Summary

We presented a coupled CRF method for recovering bare earth models from airborne LiDAR data. One CRF models the discrete labels of ground and non-ground, while the other models the smooth underlying continuous height field. Using EM, we recover the bare earth surface while re-classifying the points into ground and non-ground using a model that compares our initial estimate of z to the correct labels c. We initialize the EM process with the output of a boosted classifier learned from our labeled data.

Extensive testing against hand-classified data and commercial data shows that our results improve upon manually tuned classifiers and produce surfaces that better the results of previous learning-based bare earth systems. As yet, our system does not include models of sensing errors due to scattering and multiple returns, so the system could be improved by taking such errors into account. Despite the fact that the method does not use any knowledge about regularities of man-made structures, it is effective at isolating them in our data.

References

[1] Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.: Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In: CVPR. (2005)

[2] Lalonde, J.F., Vandapel, N., Huber, D., Hebert, M.: Natural Terrain Classification Using Three-Dimensional Ladar Data for Ground Robot Mobility. J. Field Robotics **23** (2006) 839–861

[3] Sithole, G., Vosselman, G.: Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry & Remote Sensing **59** (2004) 85–101

[4] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Intl. Conf. on Machine Learning. (2001)

[5] Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2003)

[6] Han, F., Tu, Z., Zhu, S.C.: Range Image Segmentation by an Effective Jump-Diffusion Method. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 1138–1153

[7] Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59** (2004) 167–181

[8] Lu, W.L., Little, J.J., Sheffer, A., Fu, H.: Deforestation: Extracting 3d bare-earth surface from airborne lidar data. In: The Fifth Canadian Conference on Computer and Robot Vision. (2008)

[9] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. In: ICCV (1). (1999) 377–384

[10] Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics **28** (2000) 337–407

[11] Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola, A., Bartlett, P., Schoelkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press (1999)

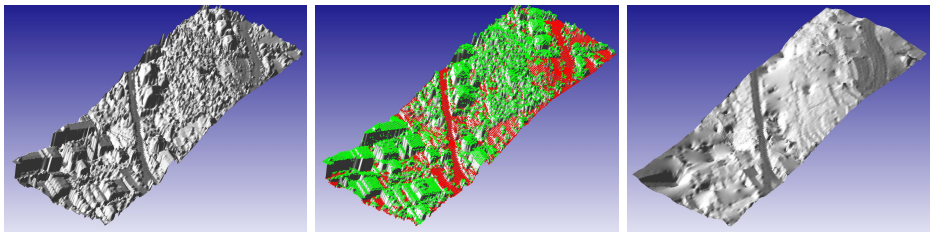
[12] Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: Laplacian Mesh Optimization. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques. (2006) 381–389

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

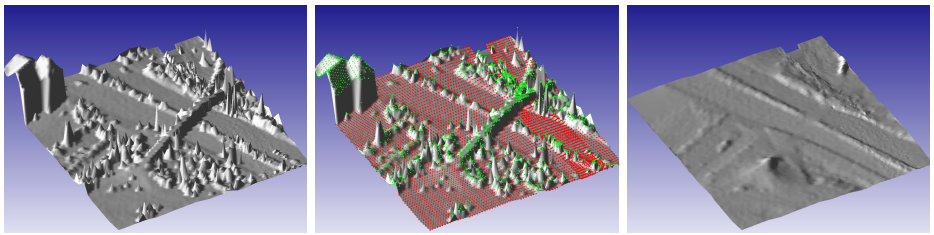
[14] Torralba, A., Murphy, K., Freeman, W.: Contextual models for object detection using boosted random fields. In: NIPS. (2004)

[15] Terrapoint: (http://www.terrapoint.com/)

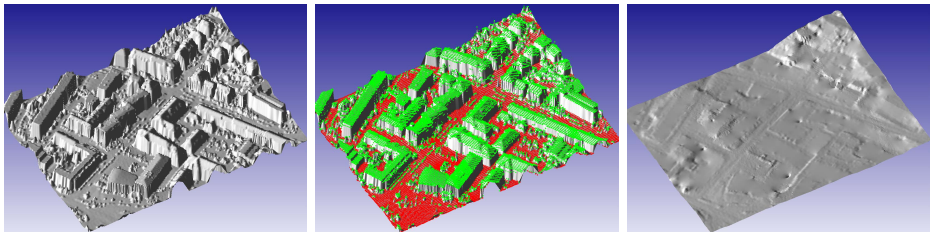
[16] Cignoni, P., Rocchini, C., Scopigno, R.: Metro: measuring error on simplified surfaces. In: Computer Graphics Forum. Volume 17. (1998) 167–174



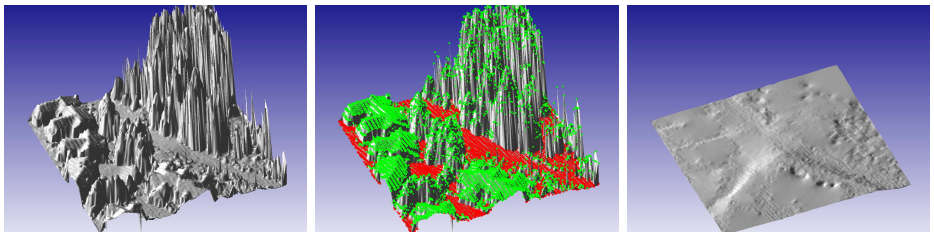
(a) vegetation and buildings on steep slope



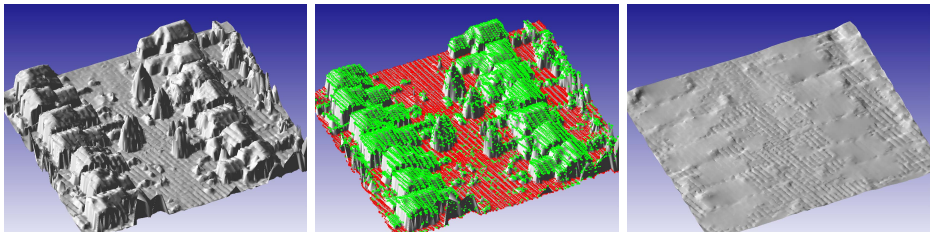
(b) bridge and rural area



(c) buildings



(d) building and trees



(e) residential area

Fig. 8. Qualitative evaluation of the Sithole *et al*. dataset [3] and the Terrapoint dataset [15]. The first column is the original point cloud after Delaunay triagulation. The second column is the classification results. The red points are classified as ground while the green points are classified as non-ground. The third column is the estimated bare-earth surface. (a)-(c) are results from the Sithole dataset, and (d)(e) are results from the Terrapoint dataset.