AERIAL LASER SCANNING AND IMAGERY DATA FUSION FOR ROAD DETECTION IN CITY SCALE

Anh-Vu Vo, Linh Truong-Hong and Debra F. Laefer
Urban Modelling Group
School of Civil, Structural and Environmental Engineering & Earth Institute
University College Dublin, Ireland

ABSTRACT

This paper presents a workflow including a novel algorithm for road detection from dense LiDAR fused with high-resolution aerial imagery data. Using a supervised machine learning approach point clouds are firstly classified into one of three groups: building, ground, or unassigned. Ground points are further processed by a novel algorithm to extract a road network. The algorithm exploits the high variance of slope and height of the point data in the direction orthogonal to the road boundaries. Applying the proposed approach on a 40 million point dataset successfully extracted a complex road network with an F-measure of 76.9%.

Index Terms: aerial laser scanning, aerial imagery, data fusion, road detection, machine learning, hybrid indexing

1. INTRODUCTION

Automatic road detection from remote sensing data is useful for many real world problems such as autonomous navigation. Traditionally road detection has relied heavily on satellite or aerial image interpretation. More recently aerial laser scanning (ALS) has emerged as an alternative to the more mature photogrammetry technology. Laser scanning, also known as Light Detection And Ranging (LiDAR), is capable of accurately documenting real world geometries in three dimensions, whereas the input for photogrammetry is two-dimensional data. Imagery data can be fused with LiDAR point clouds to generate integrated datasets (e.g. [1]). In this paper, road detection from ALS point clouds fused with orthophotos is addressed with a novel approach that involves an innovative data management strategy essential to the workflow for high-resolution data.

2. RELATED WORKS

Various methods have been used for road extraction from laser scanning data. They can be categorized as either (1) filtering, (2) clustering/segmentation, or (3) machine learning. Most methods exploit laser reflectance, height, and its derivatives (e.g. height variance) as the most important means for distinguishing road versus other data. Furthermore, while not always used, the added benefit of combining LiDAR with aerial photogrammetry data has long been recognized (e.g. [2]).

Prominent amongst the filter-based techniques for road extraction are those by Alharthy & Bethel [3], Clode et al. [4], and Clode et al. [5]. Alharthy & Bethel [3] filtered point data sequentially use a point's distance to a Digital Terrain Model (DTM) and laser reflectance strength before a connected component labelling is performed to group adjacent, filtered points. Similarly, Clode et al. [4] utilized a hierarchical rule-based classification. In addition to the two features used in [3], a minimum limit of point density and morphological filtering were employed. That research was later extended by Clode et al. [5] with a vectorization process performed on the extracted road points.

As part of the segmenting/clustering category, Choi et al. [6] grouped adjacent points incrementally based on the difference in the height and laser reflectance between the points and their neighbours. Additionally road slope was considered for detecting erroneous clusters. Another application of segmentation for road detection was presented by Hu and Tao [2] where the ALS and aerial imagery data were segmented separately. Optical imagery was more successful distinguishing roads from trees and grasslands, while low vegetated areas were distinguishable by their high laser reflectance. Buildings were recognized by their relative height. Post-segmentation, parking lots remained attached to road areas and required a Hough transformation to detect the straight road strips. Notably, this approach is applicable only for grid-shaped road networks.

Samadzadegan et al. [7] published one of the few examples of a machine learning based road detection approach resulting in a correctness of 87.37% by fusing multiple classifiers. From height and intensity information acquired by laser scanning, the authors generated several combinations of classifiers. The best performing combination was selected via a genetic algorithm.

In comparison to other fields such as building detection and tree species classification from laser scanning data, automatic road extraction is a less active research topic. The problem is still far from completely solved. This paper contributes to the field by introducing a point cloud processing workflow and a new algorithm for extracting points on road boundaries from ALS fused with aerial imagery data.

3. PROCESSING WORKFLOW

3.1. Data processing workflow

The proposed road extraction workflow is depicted in Fig. 1. ALS point clouds and orthophotos are fused together to form coloured point clouds, which are then loaded into an Oracle database and indexed by a hybrid quadtree-kdtree indexing. Next, additional features such as normal vector, local surface roughness, and HSL (Hue – Saturation – Lightness) colour are computed. A supervised classification continues the workflow. The process classifies points as (1) ground, (2) building, or (3) unassigned. Road curbs and other obstacles bounding road regions are detected before a quadtree based region growing algorithm is applied, which connects an initial seed point to other ground points, if an obstacle-free path exists that connects the seed to the new points.

3.2. Data and data management

For the purpose of this paper, 0.5 km^2 in the port of Zeebrugge, Belgium was selected as represented by ALS data with a nominal density of 65 points/m^2 (Fig. 1a&c) and associated colour orthophotos with a 5 cm resolution [originally presented in 3 bands red-green-blue (RGB)] (Fig. 2b). The imagery and ALS data were ac-

quired simultaneously so that they match when fused together (Fig. 2d), except for some fast moving objects such as cars on highways. As per the normal practice, the ALS and orthophotos were provided as separate tiles, (each 500 m×500 m). The ALS data were in an ASCII format containing only laser intensity and point coordinates. Other point attributes such as the time stamp and return number (usually associated with ALS data) were not available. The orthophotos were given in geo-referenced tiff format.

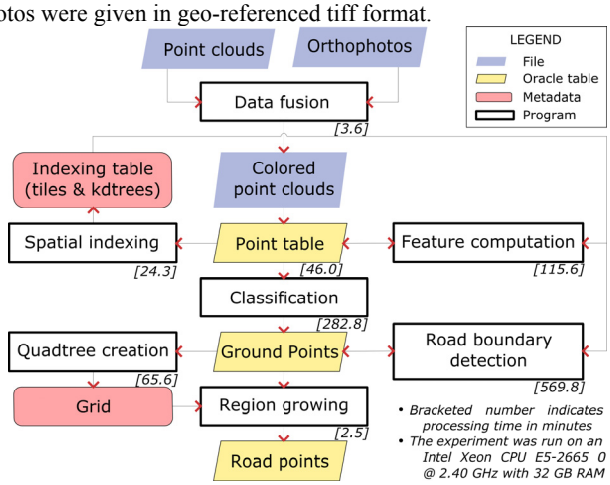
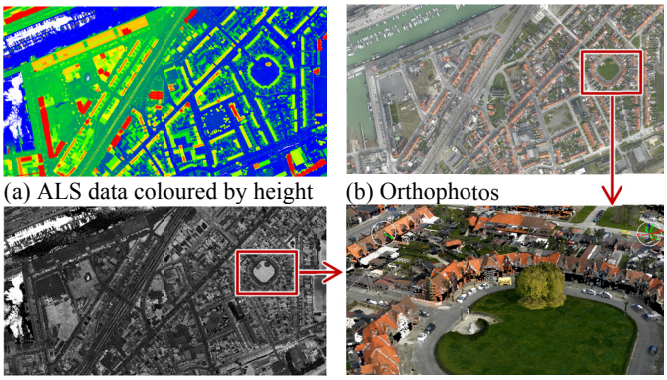


Fig. 1. General workflow



(a) ALS data coloured by height (b) Orthophotos

(c) ALS data coloured by intensity (d) 3D point cloud with colours derived from orthophotos

Fig. 2. ALS, aerial imagery data and the fusion result

The large nature of the data precludes storing the entire point cloud within the main memory for most conventional computers. So an out-of-core data management solution coordinating data input and output between main memory and disk is required. Handling spatial objects is necessary since data retrieval based on spatial conditions (e.g. range search and neighbour search) is typical, frequent, and computationally demanding. In this study, all steps (except the file-based data fusion) were conducted within a database environment. A hybrid quadtree–kdtree indexing (Fig. 3) was implemented atop a flat table storing LiDAR points to enable fast all-nearest-neighbour (ANN) computation on the points. At the top level, point data were partitioned into multiple 125m×125m tiles using a Hilbert code implementation. Tile size was selected based on the amount of memory dedicated for the index (e.g. approximately 120 megabytes/tile). Under each tile, a 3D kd-tree was built. The kd-tree includes points in a buffer around the tile, plus the points within this tile itself to avoid discontinuity around the tile boundaries. The kd-trees were made reusable by being serialized and stored as binary large objects in an indexing table. Each was retrieved and de-serialized back to the main memory, once a search within its extent was invoked. This hybrid indexing well adapts to the spatial distribution of ALS data [i.e. dominantly horizontal (2D) at the global level and fully 3D at the local level]. Within a tile, ANN queries are fast because its associated 3D index resides in the main memory. This implementation is not yet generic, nor optimal, but did sufficiently support all spatial queries performed in this study.

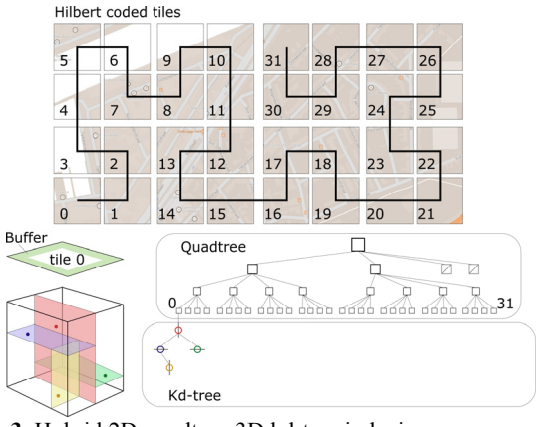


Fig. 3. Hybrid 2D quadtree-3D kd-tree indexing

4. POINT CLOUD CLASSIFICATION

To reduce computational efforts, a less expensive point classification is performed on the entire cloud before a more demanding road extraction process is applied only to the ground points. A supervised approach is employed to classify the point cloud into one of three groups: building, ground or unassigned. The classification process involves three main steps: (1) compute point features, (2) select optimal feature vector, and (3) classify the entire data with the best performing classifier. The Weka toolkit was utilized to create classification models in this study [11].

After being fused with orthophotos, the point cloud possesses two raw (i.e. from sensor) attributes: intensity and RGB colour values. Additionally, there are several secondary features possibly derived from the point coordinates and the two initial attributes that can be beneficial for point classification. In this study, the following features were investigated: (1) height: z value of the point, (2) imagery intensity described in two colour spaces, (3) laser intensity, (4) height variation – maximum variation of z values within a spherical neighbourhood N of the given point (5) surface roughness – indirectly represented by the quadratic mean of orthogonal distances of all points in N to a plane P fitting to all points in N, and (6) normal vector of P, represented as (n_x, n_y, n_z) in a Cartesian coordinate system or (θ,ϕ) in a radial coordinate system. An iterative Principle Component Analysis [12] was implemented with a weighting factor inversely proportional to the point-to-plane distance to improve plane fitting.

To analyse the influence of the above features, several combinations (termed feature vectors) were investigated (FV0 to FV5 on Table 1). FV6 and FV7 compared the differences caused by the various ways of representing normal vectors and colours. Performance of each feature vector was evaluated by a training-and-evaluating process. Sixteen different regions selected from the original data covering approximately 16% of the study area, were manually labelled. Two-thirds of the labelled data were used to build a J48 decision tree classifier with a Weka machine learning toolkit [11]. Classifier accuracy was estimated against the remaining labelled data. The classification performance of each feature vector is plotted on Fig. 4 with 4 measures: F1 score for ground

(blue), building (red), and unassigned points (green) and the number of correctly classified instances (CCI) including all three classes (purple). The F1 scores computed for each class are the harmonic means of precision and recall while evaluated against the testing sets, $F1 = 2\ (precision \times recall) / (precision + recall)$.

Table 1 – Combinations of point features for classification

	H	NV	SR	II	LI	HV	Comments
FV0	•	•	•				Most important features
FV1				•	•	•	Exclude core features
FV2	•	•	•	•	•	•	**Best combination**
FV3	•	•	•		•	•	Influence of II
FV4	•	•	•	•		•	Influence of LI
FV5	•	•	•	•	•		Influence of HV
FV6	•	○	•	•	•	•	(n_x,n_y,n_z) vs. (θ,ϕ) NV
FV7	•	•	•	◊	•	•	RGB vs. HSL colour

• feature included for classification
◊ switch normal vector from spherical to Descartes coordinate system
○ switch colour from HSL to RGB space
height (H), normal vector (NV), surface roughness (SR), image intensity (II), laser intensity (LI), height variation (HV), point density (PD)

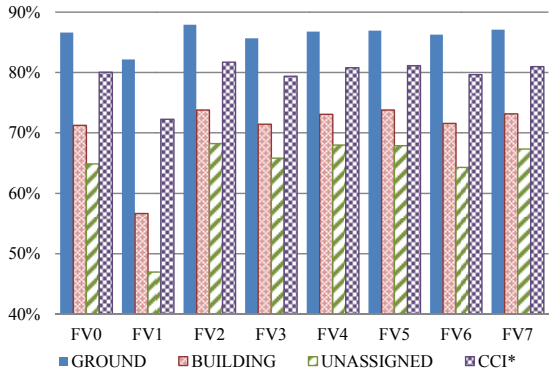


Fig. 4. Performance of the feature vectors in Table 1
(*): CCI - number of correctly classified instances

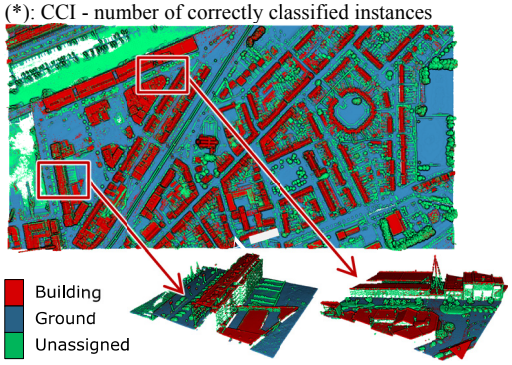


Fig. 5. Classification result

Based on Fig. 4, the most important features for point classification were height, normal vector, and local surface roughness. The CCI was 80.1% when the three core features were used (FV0) but dropped to 72.3% when excluded (FV1). The absence of image intensity, laser intensity and height variation reduced the CCI by 2.3%, 0.8% and 0.9%, respectively (FV3, 4 & 5). Representing normal vectors in a radial form (θ, ϕ) was better for classification than Cartesian coordinates (n_x, n_y, n_z) (FV2 vs. FV6) due to the trivial lengths of normal vectors, which can be discarded when the vectors are represented in a spherical system. In Cartesian coordinates, the lengths are blended into all three variables (n_x, n_y, n_z), which complicates the problem without improving classification levels. The more robust HSL colour was better than RGB (81.7% in FV7 vs. 79.7% in FV2) as previously noted by Sithole [10]. Ground point classification rates were significantly higher than the other two classes (blue vs. red and green columns in Fig. 4), because of feature consistency. The best performing feature vector was identified as height, normal vector (ϕ, θ), roughness, HSL colour, intensity and height variation. To exploit all manually labelled data, all are used to build a new classifier. The final classifier classified the entire dataset (Fig. 5). Points classified as ground were further processed for road extraction.

5. ROAD EXTRACTION

Laser intensity has been used successfully for distinguishing road surfaces from other materials (e.g. [2]-[4]). Asphalt, appears within a very distinctive range in the laser intensity spectrum. However, in this study the roads were made from various materials without sufficiently distinguishable intensities (Fig. 2c). Thus, a new method was needed. The proposed method has two main steps: (1) identifying road curbs and obstacles bounding road regions based on spatial distribution of point data, and (2) extracting road points using a quadtree-based region growing algorithm considering intensity and colour conditions.

5.1. Detection of road curbs and obstacles

Road curbs and obstacles were defined as objects preventing vehicle progression due to height or slope variation within a finite spatial extent (e.g. a 1 m radius circle). Fig. 6a shows an example along a road cross-section at position A within Fig. 5. Small features (e.g. curbs) are visible due to the high data density. The method computes two features: directional slope variation, $\nabla_\beta slope(p_i)$, and directional height variation, $\nabla_\beta height(p_i)$. Each are computed for every single point p_i within the ground points.

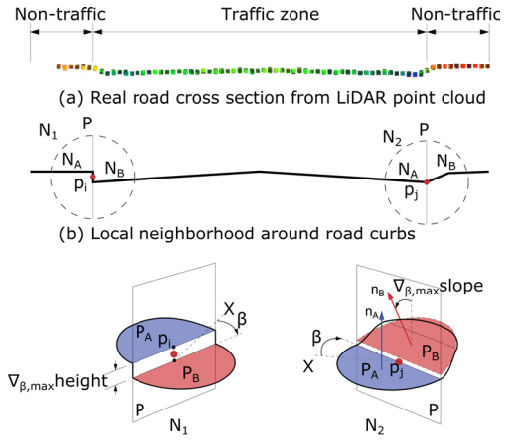


Fig. 6. Directional slope and height variation computation

Firstly neighbouring points N within a spherical neighbourhood of p_i are partitioned into two point groups N_A and N_B by a vertical plane P containing p_i and making an angle of β with the X axis (Fig. 6). P_A and P_B are defined as the best fit planes to the points in N_A and N_B. With a given value of β, $\nabla_\beta slope(p_i)$ is defined as the angle between P_A and P_B, whereas $\nabla_\beta height(p_i)$ is the height difference between the vertical projections of p_i on P_A and P_B. Finally, the maximum directional slope and height variations over β, $\nabla_{\beta,max} slope(p_i)$ and $\nabla_{\beta,max} height(p_i)$ are determined.

Figure 7 presents the results of $\nabla_{\beta,max} slope$ and $\nabla_{\beta,max} height$ for a segment of ground points. Road boundaries are clearly distinguishable and better than using the residual value (see 3.1) (Fig. 7c) or the non-directional height variation approaches (Fig. 7d). Points having $\nabla_{\beta,max} slope > 8^o$ and $\nabla_{\beta,max} height > 5$ cm were con-

sidered as obstacles and were used as input to a region growing road extraction as presented in the next section. Thresholds were selected empirically.

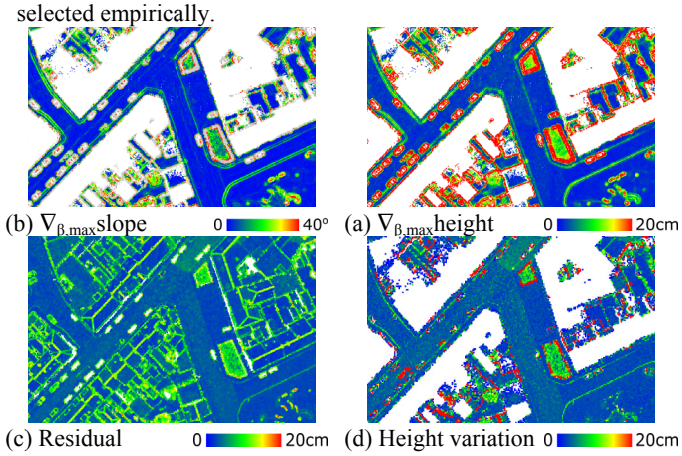


Fig. 7. Directional slope and height variation versus conventional height variation and residual

5.2. Quadtree-based region growing

A region growing algorithm was combined with a rasterization for performance enhancement as seen in [11] to extract road points (Fig. 8). An initial seed, namely any pixel (i.e. quadtree node) located inside the road network, is manually specified (e.g. the plus mark in Fig 8b). Around the seed, a buffer approximating a required clearance for one vehicle is constructed (e.g. 0.75 m radius circle). Every pixel within the buffer, as well as all points enclosed in the pixel, is labelled as road, if the buffer is obstacle free. The newly detected road pixels are set as new seeds for the next iterations, if they satisfy additional intensity and colour conditions (i.e. intensity < 550 and hue ∈ (0.18, 0.3)). Intensity and colour of a pixel are set as the maximum intensity and the average HSL of all points contained in the pixel. These criteria assist in distinguishing road from grass, even though they are insufficient by themselves for direct road extraction.

6. DISCUSSIONS AND CONCLUDING REMARKS

Most of major road segments were identified, except those blocked by speed bumps or other obstacles (e.g. location C&D in Fig. 8). Even though relaxing the slope and height variation thresholds would extend the region growing beyond these obstructions, the potential for false detection might increase. Most misclassification was attributable to large parking lots (e.g. location A in Fig. 8). Evaluation against a manual result showed a precision of 66.1%, recall of 91.9% and an F1 score of 76.9%.While the LiDAR point cloud provided the highly accurate dense 3D data, enabling detection of fine features such as road curbs or barriers, colour from the orthophotos increased the point cloud classification accuracy by 2.3%, equivalent to adding approximately 920,000 points. Orthophoto based colour and laser intensity also helped exclude grassy areas (e.g. B1 and B2 in Fig. 8a).

While minimizing computational cost is not the aim of this study, the processing time for each single module in the chain is presented in Fig. 1 (with the total of 18.5 hours). Notably, the classification step improves performance but is not compulsory, and further optimization alleviating the costs is possible. The proposed approach is more widely applicable than many other studies in this field, as it does not require a digital elevation model or a two-dimensional road map as input. The initial seeding point could be automated, as well as thresholds for directional slope and height variation. The results showed all locations accessible from the initial seed point. Such map would be more useful for autonomous navigation than traditional maps.

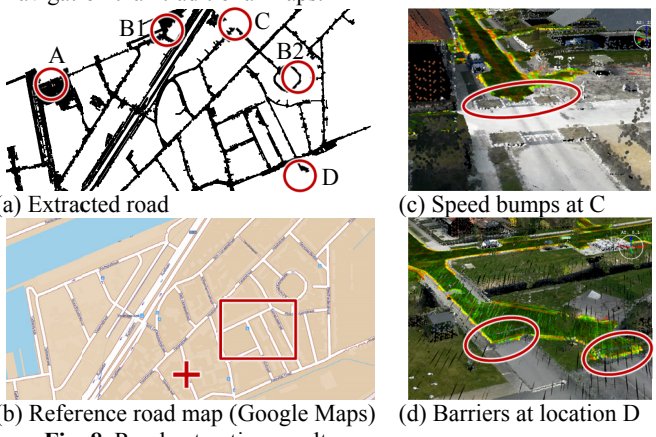


Fig. 8. Road extraction result

7. ACKNOWLEDGMENTS

This work was funded by European Research Council grant ERC-2012-StG_20111012 Project 397836. The authors would like to thank the Belgian Royal Military Academy for acquiring and providing the data used in this study, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee.

8. REFERENCES

[1] M. Dalponte, L. Bruzzone, and D. Gianelle, "Fusion of Hyperspectral and LIDAR Remote Sensing Data for Classification of Complex Forest Areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1416–1427, 2008.

[2] X. Hu and C. V. Tao, "Automatic road extraction from DENSE urban area by integrate processing of high resolution imagery and Lidar data," in *Int'l Arc. Photo., Rem. Sens. & Spatial Info. Sci.*, 2004, p. 35:B3.

[3] A. Alharthy and J. Bethel, "Automated road extraction from LIDAR data," in *Proc. ASPRS Ann. Conf.*, 2003, pp. 05–09.

[4] S. Clode, P. Kootsookos, and F. Rottensteiner, "The automatic extraction of roads from LiDAR data," in *Int'l Soc. Photo. & Rem. Sens. 20th Ann. Cong.*, 2004, vol. 35, pp. 231–237.

[5] S. Clode, F. Rottensteiner, P. Kootsookos, and E. Zelniker, "Detection and Vectorization of Roads from Lidar Data," *Photo. Eng. Rem. Sens.*, vol. 73, no. 5, pp. 517–535, 2007.

[6] Y. W. Choi, Y. W. Jang, H. J. Lee, and G. S. Cho, "Heuristic road extraction," in *Proc. - 2007 Int't Symp. Info. Tech. Convergence, ISITC*, pp. 338–342, 2007.

[7] F. Samadzadegan, M. Hahn, and B. Bigdeli, "Automatic road extraction from LIDAR data based on classifier fusion," 2009 *Joint Urban Rem. Sens. Event*, 2009.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Exploration*, vol. 11, no. 1, pp. 10–18, 2009.

[9] J. Deschaud and F. Goulette, "A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing," in *Proc. of 3D Processing, Visualization and Transmission Conf.*, 2010.

[10] G. Sithole, "Detection of Bricks in a Masonry Wall," *Int'l Arch.Photo. Rem. Sens. &Spatial Info. Sci.*, pp. 567–572, 2008.

[11] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation" *ISPRS J. Photo. Rem. Sens.*, vol. 104, pp. 88–100, 2015.