2010 International Conference on Pattern Recognition

Dense Structure Inference for Object Classification in Aerial LIDAR Dataset

Eunyoung Kim and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA, USA
{*kimeunyo* | *medioni*}*@usc.edu*

***Abstract*—We present a framework to classify small free-form objects in 3D aerial scans of a large urban area.**

The system first identifies large structures such as the ground surface and roofs of buildings densely built in the scene, by fitting planar patches and grouping adjacent patches similar in pose together. Then, it segments initial object candidates which represent the visible surface of an object using the identified structures.

To deal with sparse density in points representing each candidate, we also propose a novel method to infer a dense 3D structure from the given sparse and noisy points without any meshes and iterations.

To label object candidates, we build a tree-structure database of object classes, which captures latent patterns in shape of 3D objects in a hierarchical manner.

We demonstrate our system on the aerial LIDAR dataset acquired from a few square kilometers of Ottawa.

***Keywords*-Object classification; Range image; LIDAR; Densification;**

I. Introduction

As recent advances in light detection and ranging(LIDAR) technology allow the ability to collect 3D data over vast urban areas with excellent resolution and accuracy, the automatic recognition of 3D small objects in the scene becomes important for various applications such as environment monitoring and autonomous robotic navigation. Our goal is thus to develop a system that automatically categorizes small free-form 3D objects(*e.g.* cars) in aerial range images, whereas the majority of existing works on large-scale range image processing have focused on identifying terrain and buildings or linear structures such as poles.

The first image of Fig. 1 shows an example of typical urban region containing buildings, houses, vegetation and small objects. The region covers $2,196 \times 2,997m^2$ and produces the number of 3D points(2.5GB in binary). Specifically, there are up to a few hundreds of thousands of 3D points in $50 \times 50m^2$ region.

Labeling small objects in an aerial LIDAR data is challenging because of irregularity and sparsity in point clouds representing the visible surfaces of 3D objects(approximately, 1 point in $20 \times 20cm^2$ region). Fig.2(a) shows the original point cloud from a car, which has sparse 3D points and a big hole caused by a front window. This may result in misinterpreting the shape of an object and, consequently, poor classification performance. Therefore, our system also includes dense structure inference from given sparse 3D points.

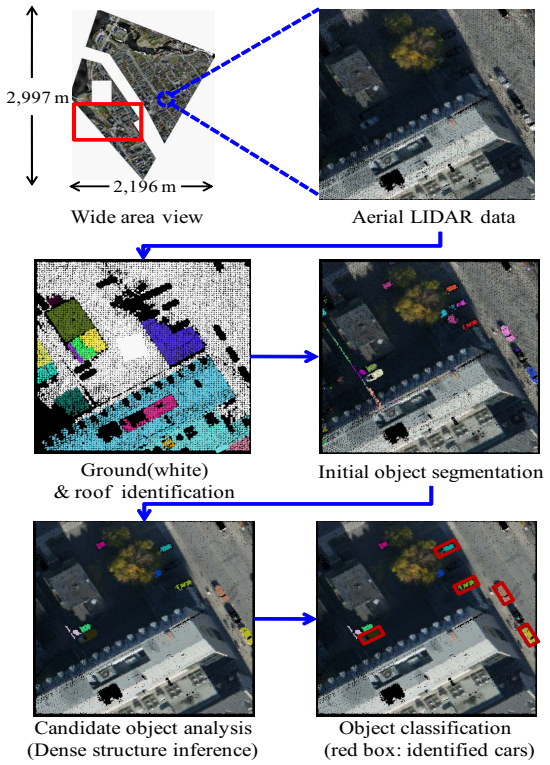


Figure 1. System overview

Fig. 1 shows a flow chart of our system. Given an aerial LIDAR data, the system starts by identifying the terrain and the roof surfaces of the buildings(houses) using planar primitives, and delineates the initial object candidates using spatial contexts between the large structures and small objects.

Then, for every candidate, we infer uniformly-sampled dense 3D points smoothly continuous with the existing surface from the given sparse point cloud.

Finally, the system labels each resultant point cloud which represents the visible surface of a 3D object.

Our contribution is two-fold:

1) We develop a generic system that recognizes free-form

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.747

3D objects in aerial range images
2) We also propose an approach to dense structure inference for better shape analysis.

II. Related work

Research in object categorization in range images of urban area has been mostly focused on identifying structured objects such as buildings and poles. Anguelov *et al.*[1] introduced a framework to segment 3D data into four classes(ground, building, tree and shrubbery), that maps a 3D point as a node, and finds 3D point cluster representing the object by efficient graph-cut inference under MRFs. [2] also showed a point classification approach using Max-Margin Markov Network that incorporates multiple levels of contextual information. Recently, Golovinskiy *et al.* combined spin images with other contextual features to classify some of free-form objects in 3D point clouds of urban environment[3]. The benefit of our system over this approach includes well-defined hierarchical candidate segmentation using spatial context between large structures and small objects.

III. Tensor Voting in 3D

Since tensor voting(TV) is a main tool to infer surface information from 3D points in our system, this section provides an overview of TV in 3D. TV[4] is a perceptual organization framework that is able to infer salient geometric structures such as point, curve and surfaces based on the support the tokens which comprise them receive from their neighbors in *N*D, without predefined model and iteration. It only requires one parameter to define the scale of voting.

All input tokens, a set of unoriented or oriented points, are encoded as a second order symmetric tensor. Then, each input tensor collects the tensor vote cast from its neighbors by aligning the voting fields along the orientation of tensors. The voting fields are well defined with saliency decay function that attenuates the saliency with length and *curveness* of smoothest path connecting the receiver and voter.

After voting, the tensor can be decomposed as:

$$\begin{aligned} T &= \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T + \lambda_3 e_3 e_3^T \\ &= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) \\ &\quad + \lambda_3 (e_1 e_1^T + e_2 e_2^T + e_3 e_3^T), \end{aligned}$$

where λ_i are the eigenvalues in decreasing order and e_i are the corresponding eigenvectors. From the equation, we can infer three types of information as follows:

- *Pointness*: no orientation and saliency
- *Curveness*: orientation is e_3 and saliency is $\lambda_2 - \lambda_3$
- *Surfaceness*: orientation is $e_1$and saliency is $\lambda_1 - \lambda_2$

IV. Object segmentation

Segmenting object candidates in large-scale range images is a necessary step prior to efficient recognition, as an aerial range image covers a huge urban area and it is inefficient to classify objects by window search.

This section thus describes our method[5] that hierarchically segments point clouds corresponding to small 3D objects using large structures.

A. Large structure identification

From the observation that the majority of large structures in the urban area are nearly planar, we can infer these structures by recognizing smoothly continuous planar patches in the scene.

Given a range image, we first partition it into uniform voxels, and then identify a well-fitted planar patch for every voxel. Because of noise and uncertainty in 3D depth data, the TV process serves to infer surface orientation. Every occupied voxel selects a representative as an input token for the TV process, and then the TV process infers surface orientation(e_1) of each token from its neighboring regions. The inferred orientation is used to estimate a planar patch in the voxel and the patch is validated, if all 3D points in the voxel agree with it.

After identifying a planar patch for every voxel, we infer smooth planar surfaces by aggregating adjacent planar patches with consistent pose.

Finally, the ground surface is identified based on the height and size of the region. Every region that has larger area than threshold and no 3D points below non-boundary region is classified as a roof of a building.

B. Initial object segmentation

With the reasonable assumption that every small object rests on a large structure, the system is able to recognize a point cloud representing the visible surface of an object by inferring the possible pose of the object from the identified large structure.

Because an aerial image only contains a top surface of an object, for every structure, we identify the point clusters by grouping the farthest 3D points from the surface, and then define the volume of each cluster using the spatial extent(width and depth) of the cluster and the height of the cluster from the surface. Finally, the 3D points in the estimated volume are classified as an initial object candidate on the roofs or the ground surface.

V. Dense structure inference

Since the initial object candidate is segmented based on contiguity of 3D points without consideration of the surface that these points intend to represent, each candidate should be validated and refined by enforcing the surface property for better labeling.

Initially, we discard non-object point clouds, *e.g.* a point cloud from trees, in terms of the number of 3D points(N_P), dimension (D_P) and *surfaceness*(S_P) of the candidate. In our experiments, a point cloud P satisfying $100 \leqq N_P$, $1m \leqq D_P \leqq 5m$ and $4 \leqq S_P$, where S_P is the maximum surface saliency in P, is considered as the valid visible surface of a free-form 3D object.

As shown in Fig.2(a), the point cloud also has irregular holes and noisy 3D points, which may weaken the performance of our shape-based recognition process. We thus propose a novel method that infers a dense depth map from the given noisy and sparse 3D points.

Many works on filling holes in meshes have been proposed to obtain the complete shape of 3D objects from range sensors: simple point interpolation[6] and interpolation using neighboring surface information based on Partial Differential Equations(PDEs)[7], Finite Element Methods(FEMs)[8] and Signed Distance Field(SDF)[9][10]. Unlike these approaches, our method is able to infer dense points without meshes, iterations and any initial states.

Our module aims to measure a new 3D point that most highly agrees with the surface having the most supports from its neighbors. That is, a 3D point with the highest surface saliency is chosen.

Given an object candidate P, all the points in P are mapped to the uniform 2D grid. The resolution of the grid determines the resolution of the resultant point cloud. In our experiments, each cell size was $10cm$. Note that, in our application, the process is simplified to infer the depth of holes in 2D due to the fact that a range image only provides a top-view of objects, but the same idea can be applied to 3D as well.

Then, to delineate the boundary of the object surface, we also use a neighboring point cloud P_G of the candidate, including the points already classified as the ground or roofs.

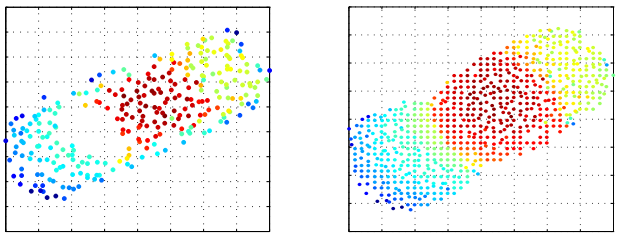
After projecting the point sets P and P_G to the grid, we densify the structure by assigning a new 3D point to each unoccupied cell.

For every unoccupied cell, we generate a set of depth candidates P_C, and then apply the TV process to compute a surface information of every point in P_C by collecting the information from its adjacent tensors in P and P_G. Finally, a point p_{new} occupying the cell is:

$$p_{new} = \arg\max_i (\lambda_1^{p_i} - \lambda_2^{p_i}), \quad p_i \in P_C,$$

where $\lambda_1^{p_i}$ - $\lambda_2^{p_i}$ is the surface saliency of p_i.(See Sec.III)

An example of the resultant dense point cloud(688 points) inferred from the original points(289 points) is shown in Fig. 2(b). This process can be parallel, as our method infers a new point regardless of the result of other neighboring cells.



(a) Original point cloud (grid size: 50cm)

(b) Dense point cloud

Figure 2. Dense structure inference(Color-coded depth)

VI. Object categorization

To label object candidates, we use a shape-based classification approach using a hierarchical variant of Latent Dirichlet Allocation(LDA).

In this framework, a range image is represented by a group of visual words that contain local and global description of a given point. The database has tree structure. Each range image is assigned to one of existing path in the tree, where each node(topic) is represented by a multinomial probabilistic distribution over topics(visual words). These probabilistic distributions are based on the Dirichlet distribution and capture the frequencies of co-occurrence of visual words and topics, which correspond to latent shape patterns.

During the training process, the structure and distributions of the hierarchical structured database(HSD) are inferred from range images by Gibbs sampling. In the online phase, given an object candidate, the classification process identifies its label by first identifying the paths having similar shape patterns in the tree and comparing the shape similarity between each training image under the paths and the candidate. In our system, the candidate is categorized into one of the existing classes or a new class.

The context information is also imposed on the labeling process. Since the system already knows a type of the large structure the object rests on(either the ground or roofs), it can specify the possible classes that the candidate might belong to. For example, a candidate inferred from the ground should be labeled as one of the classes which can be on the ground, not a dish class.

VII. Experimental results

We have tested the proposed system on the aerial LIDAR dataset that contains approximately several hundreds of millions 3D points captured from Ottawa shown in Fig. 1[11]. Some regions are hidden as sequestered data for independent evaluation. Ground truth for some urban objects in the red boxed region is also available. Since the aerial range image is too large to be processed at once, we partition the volume into small blocks ($50\times50m^2$ in our experiments). Each block is processed in parallel and then, we merge the consistent results in the overlapping regions.

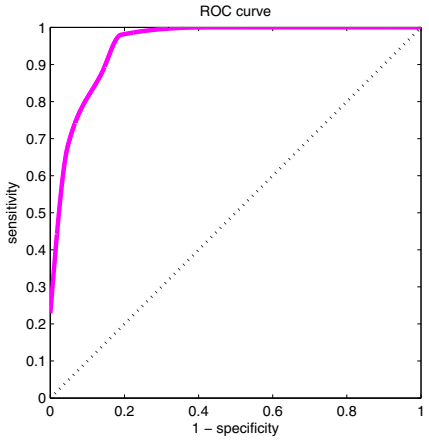


Figure 3. ROC curve for "car" recognition

Table I
CLASSIFICATION PERFORMANCE

	GT	SOC	SOCC	OC	FA
dense	108	120	81 (75%)	79 (73%)	0
sparse				57 (54%)	1

Figure 4. Ambiguous boundaries between cars(red patches: roof)

In our experiments, we identify a free-form object class, "cars", a only visible class in aerial LIDAR images. To build the database, we use 20 synthetic range images generated from 3D car models freely available on the Internet, and 3 car images extracted from non-ground truth region.

The recognition performance of our system is demonstrated in Fig. 3 and Table I. GT is the number of true instances, SOC is the number of total object candidates segmented, SOCC is the number of object candidates that really correspond to the truth instance, OC is the number of object candidate classified correctly, and FA is the number of false alarms. Based on the quantitative analysis, we can infer that, 1)using dense point clouds improves the recognition performance, 2) for the candidate objects, the system labels them correctly(*e.g.* 97% of SOCC). That is, the overall performance highly depends on candidate segmentation, which is hard due to ambiguities in 3D points as shown in Fig. 4.

On a PC with two 3.0Hz CPUs and 8GB of RAM, the average processing time for densification and classification is 26.6 secs and 0.57 secs, respectively, for each object candidate, without any optimization. Because the dense structure inference process can be parallelized, it can be much faster by implementing it on GPU.

Note that our framework can identify any type of free-form objects.

VIII. Conclusion

We present a shape-based recognition system that categorizes small urban objects in aerial range images. The system first identifies large structures and initial object candidates using the identified structures. Then, each candidate infers the dense 3D structure and is labeled using a hierarchical structure database.

In the future, we will improve the segmentation process.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of markov random fields for segmentation of 3D scan data," in *CVPR*, 2005, pp. 169–176.

[2] D. Munoz, A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *CVPR*, 2009.

[3] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environment," in *ICCV*, 2009.

[4] G. Medioni, M.-S. Lee, and C.-K. Tang, *A Computational Framework for Segmentation and Grouping*. Elsevier Science Inc., 2000.

[5] E. Kim and G. Medioni, "Urban scene understanding from aerial and ground LIDAR data," *Machine Vision and Application*, to appear.

[6] P. Liepa, "Filling holes in meshes," in *Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, 2003.

[7] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro, "Inpainting surface holes," in *ICIP*, 2003.

[8] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu, "A finite element method for surface restoration with smooth boundary conditions," *Computer Aided Geometric Design*, vol. 21, no. 5, pp. 427–445, 2004.

[9] R. Sagawa and K. Ikeuchi, "Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution," *TPAMI*, vol. 30, no. 4, pp. 686–699, 2008.

[10] T.Masuda, "Filling the signed distance field by fitting local quadrics," in *3DPVT*, 2004.

[11] "http://daytaohio.com/Wright_State100.php."