A Method for Detecting Windows from Mobile LiDAR Data

One Sentence:

This paper presents a novel method for window detection from mobile LiDAR data.

Ruisheng Wang
Department of Geomatics Engineering
University of Calgary
2500 University Drive NW
Calgary, Alberta, Canada T2N 1N4
ruiswang@ucalgary.ca

Frank P. Ferrie
Centre for Intelligent Machines, McGill
University, 3480 University Street, Montreal,
Quebec, Canada, H3A 2A7
ferrie@cim.mcgill.ca

Jane Macfarlane
Nokia Location & Commerce
425 W Randolph Street
Chicago, IL 60606 USA
jane.macfarlane@nokia.com

Abstract:

Mobile LiDAR (Light Detection And Ranging) data collection is a rapidly emerging technology in which multiple georeferenced sensors (e.g., laser scanners, cameras) are mounted on a moving vehicle to collect real world data. The photorealistic modeling of large-scale real world scenes such as urban environments has become increasingly interesting to the vision, graphics and photogrammetry communities. In this paper, we present an automatic approach to window and façade detection from mobile LiDAR data. The proposed method combines bottom-up with top-down strategies to extract façade planes from noisy LiDAR point clouds. The window detection is achieved through a two-step approach: potential window point detection and window localization. The facade pattern is automatically inferred to enhance the robustness of the window detection. Experimental results on six datasets result in 71.2% and 88.9% in the first two datasets, 100% for the rest four datasets in terms of completeness rate, and 100% correctness rate for all the tested datasets, which demonstrate the effectiveness of the proposed solution for planar façades with rectilinear windows. The application potential includes generation of building facade models with street-level details and texture synthesis for producing realistic occlusion-free façade texture.

Introduction

Creation of 3D models from images has been an intensive research topic in the computer vision, computer graphics, and photogrammetry communities. In recent years, the demand for 3D photorealistic building models has dramatically increased. These needs arise particularly from 3D GPS navigation systems and on-line services such as Google Earth and Nokia maps. Currently, the creation of 3D photorealistic building models still lacks automation and is a labor-intensive process (Van Gool *et al.*, 2007). Automatic solutions could significantly increase productivity, reduce costs, and be of enormous interest to location based service providers such as Nokia and Google. 3D building models can be generated by various techniques such as airborne imaging using active sensors (e.g., LiDAR) and/or passive sensors (e.g., panchromatic camera), ground-based or vehicle-borne sensing techniques (e.g., mobile mapping system). Fully automatic generation of 3D building models from aerial images or LiDAR is now viable (e.g., Zebedin *et al.*, 2006; Poullis and You, 2009; Zhou and Neumann, 2008, 2009). However, these resultant crude building models produced from airborne solutions lack sufficient detail, and are not suitable for applications where street-level representation of cities is required. Ground-based or vehicle-borne laser scanning methods, on the other hand, can produce highly accurate geo-referenced 3D points with sufficient detail due to their close-range data collection. In addition, the capability of capturing millions of 3D points directly in a short time provides laser scanning methods a great potential for fast generation of detailed 3D models. Image-based approaches have also made significant progress in being able to automatically generate impressive 3D models (Pollefeys *et al.*, 2008; Seitz *et al.*, 2006), but they are still inferior to laser-based models. Interactive image-based approaches (Debevec *et al*. 1996) are able to create photorealistic 3D models for a small set of images but cannot be scaled up well for large-scale urban environment modeling.

Windows are important features of facades and play an important role in generating photorealistic building models. Windows often form repeated structures in buildings. In other words, facades exhibit rich redundancy. One efficient way to represent a building is to describe a facade with a few parametric symbols such as window templates in a semantic way. Building facades are often occluded by trees or other objects such as cars or pedestrians in an urban environment. To generate occlusion-free facade textures, one way is to use the detected representative windows to replace occluded windows. In both situations, window detection is an important step to recover facade structures, to produce occlusion-free facade textures, and finally to generate photorealistic building.

Most of the previous range data processing research focused on small objects under controlled laboratory environments (e.g., Curless and Levoy, 1996; Reed and Allen, 1999). Outdoor scenes containing trees, cars, and pedestrians etc. are however more difficult and challenging. This paper addresses the problem of window detection from the LiDAR data collected from a moving vehicle, and makes the following contributions: First, we propose a robust window detection method. The method consists of a potential window detector and a window localization algorithm. Second, we propose a combination of bottom-up with top-down approach to tackle the problem of façade plane detection from LiDAR data. The bottom-up approach is to cluster point clouds into potential façade regions using Principle Component Analysis (PCA). The top-down approach consists in applying the Random Sample Consensus (RANSAC) plane fitting to the potential façade region to ultimately extract the façade. This method effectively avoids the problem of inaccurate normal estimation causing the misclassification of point clouds and increases computational efficiency due to the combination of the bottom-up with top-down approach.

Related Work

To generate detailed geometry of façades, common methods rely on the similarity and repetitive patterns of windows to discover façade structures with the assumption that windows are key elements of façade interpretation (Mueller *et al*., 2007; Mayer and Reznik, 2005; Ali *et al*., 2007; Lee and Nevatia, 2004; Hohmann *et al*., 2009). Some methods use an explicit window model and horizontal and vertical profiling of images (e.g., Lee and Nevatia, 2004) to discover the repetitive pattern. Some use an implicit window model and learning approach to recognize windows (Mayer and Reznik, 2005; Ali *et al*., 2007). Grammar-based methods were first proposed in the field of architecture (Stiny and Gips, 1972), and have been successfully used in the automatic modeling of architecture (Wonka *et al*., 2003). By image similarity detection, shape grammar generation from façade images can be automated (Mueller *et al*., 2007). Typical intensity-based similarity measures include cross-correlation (Lee and Nevatia, 2004) and mutual information (Viola and Wells, 1997; Mueller *et al*., 2007; Russakoff *et al*., 2004). The common assumption in these methods is that windows correspond to a rectangular structure which is fair for most modern box-like buildings except some landmarks. However, in an image of a building, there often exist too many edges, luminance, color and texture variations, occlusions from trees, traffic lights etc., which makes image-based window detection a challenging task.

On the LiDAR side, there is much work on range data segmentation (e.g., Hoover *et al.*, 1996; Jiang *et al.*, 1996) from the vision and robotics communities. Normal computation at each LiDAR point is the first step in most range segmentation algorithms (Stamos and Allen, 2002; Besl and Jain, 1988; Bellon and Silva, 2002) which is also a crucial step for precise region extraction. However, surface normals are often estimated inaccurately for points near boundaries. A good example (Chen and Stamos, 2007) shows cases where inaccurate normal estimation occurs. Without normal computation, planar surfaces can also be estimated directly by using RANSAC plane fitting (Wahl *et al*., 2007). Because of the complexity of the scene in the real world, this method suffers from slow detection correctness rate and convergence of the RANSAC algorithm. To address these problems, in this paper we combine a bottom-up analysis using PCA with a top-down RASANC plane fitting to extract planar façade from noisy LiDAR data.

There are also many methods to detect buildings from aerial LiDAR (e.g., Filin, 2002, Ma, 2005; Sohn *et al.*, 2008; Akel *et al*., 2009; Meng *et al*., 2009a). However, less work has been done on detecting windows from LiDAR data. This is probably because of the nature of LiDAR data: noisy and sparse. Micro-structures like windows are hard to differentiate. In addition, windows without curtains often return signals from the interior of the buildings and the returned signal does not always contain enough valid data representing these surfaces (Pu and Vosselman, 2007). For those windows with curtains, laser points are available but often not on the same plane as façade (Pu and Vosselman, 2007). Frueh *et al*. (2005) created meshes for the entire façade but did not address the window detection problem. To the best of our knowledge, papers by Pu and Vosselman (2007), Ali *et al.* (2008), and Becker and Haala (2009), are the only three research works on window detection from LiDAR so far. Pu and Vosselman (2007) proposed a hole-based window extraction method. Basically this method searches long edges along the Triangular Irregular Network (TIN) of the façade to identify holes, groups points belonging to the same hole, filters out non-window holes heuristically, and finally fits to rectangles. This bottom-up triangular meshing based method suffers from noisy LiDAR data. Ali *et al*. (2008) converted LiDAR data into distance images, and then employed image processing techniques like morphological operations and contour analysis to segment windows. This 3D-to-2D conversion causes information loss. Directly processing 3D LiDAR point clouds is desired. Becker and Haala (2009) extracted windows by cell decomposition (Haala *et al*., 2006) from mobile LiDAR data collected from the StreetMapper (www. streetmapper.net). A so-called facade grammar is inferred from the reconstruction at areas of sufficient LiDAR point densities to support the automatic generation of facade structure in regions where no or only limited LiDAR measurements are available. Our approach to detect windows is based on histogram analysis of window edge points. Different from the existing methods (Lee and Nevatia, 2004, Becker and Haala, 2009), we first extracted building façade planes using a sequential combination of PCA with RANSAC. Then we developed a rule-based operator to detect window edge points and eliminate points from window crossbars. A window localization algorithm is developed and architecture rules such as window size and spacing are automatically inferred to enhance the window detection. After we briefly introduce the data collection device, the details of the proposed method follow.

Data Acquisition

Data is collected by NOKIA Location & Commerce using a data collection vehicle as shown in Figure 1. This mobile mapping system is composed of a 360 degree LiDAR sensor (Velodyne HDL-64E), panoramic camera (Ladybug 3), high definition cameras (Prosilica), GPS, Inertial Measurement Unit (IMU) and Distance Measurement Instrument (DMI). The Velodyne LiDAR sensor consists of 64 lasers mounted on upper and lower blocks of 32 lasers each and the entire unit spins. This design allows for 64 separate lasers to each fire thousands of times per second, generating over one million points per second.

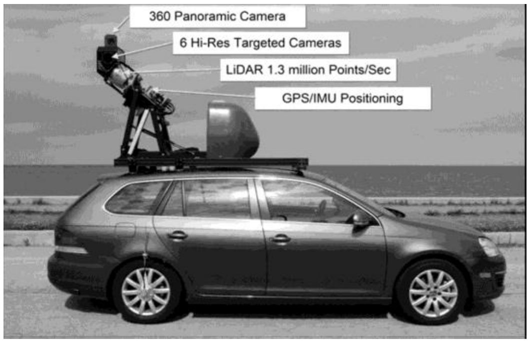


Figure 1 Data collection vehicle “NAVTEQ True”

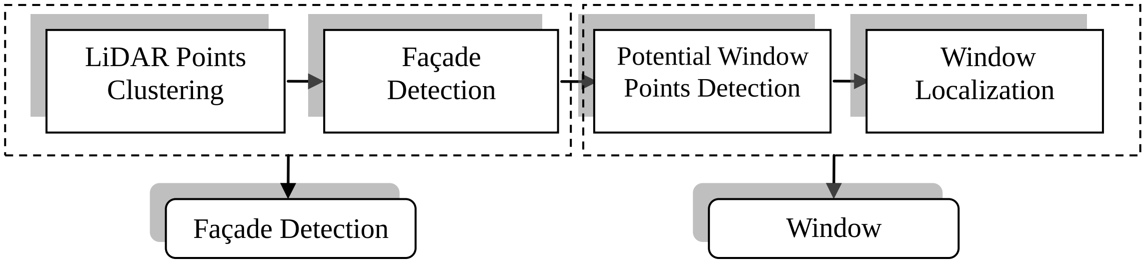


Figure 2: A flowchart of the main steps in our approach

The Method

Our approach consists of multiple stages of processing as illustrated in Figure 2. We first separate ground points from the LiDAR data, and then compute surface normals for the remaining points using PCA. Potential façades can then be identified. A RANSAC plane fitting algorithm is applied to these regions to extract façades. After a façade is found, we extract window point candidates from the façade, and then use a plane-sweep principle to generate horizontal and vertical window profile histograms. Windows are detected through the analysis of these histograms. The façade pattern such as window size and spacing is automatically inferred to form a constraint for the vertical and horizontal window signatures, and thereby enhance the robustness of the window detection. Each of these steps is explained in greater detail in the following subsections.

Ground Point Separation

The volume of LiDAR data is huge. A sub-sampling process is applied through establishing a volumetric representation of LiDAR data. This is to establish a 3D grid in the LiDAR data, and selects only one point from each voxel to discard the remaining points within this voxel. The size of the voxel determines the rate of the sub-sampling, which is specified by the desired resolution or the maximum allowed error tolerance between the samples. We choose 0.2 meter as the size for each dimension of the voxel as the relative accuracy of the Velodyne laser scanner is reported up to 20 cm in this paper. This volumetric representation also facilitates efficient search within the data. The LiDAR points from the ground can be separated to benefit further processing. In this paper, we assume that the elevations of ground points are normally the minimum. We compute a histogram using elevation values of all the points in a Local Tangent Plane (LTP) coordinate system, and select the elevation value corresponding to the first peak of the histogram as the ground height. The points with elevation around this value (i.e. ± 0.2 meter) are regarded as ground points and then eliminated from further processing. Figure 3 shows a typical elevation histogram of a street scene in a flat urban environment using the LiDAR data collected from the "NAVTEQ TURE" mobile mapping system. The points before the first peak are mostly outliers, which is much less than the number of ground points. We always find the elevation, which corresponds to the first peak of the histogram with maximum number in the frequency, as the ground elevation. In situations that do not meet the assumption of a flat terrain, there are many ground filtering algorithms (e.g., Meng *et al*., 2009b) for solving this problem. Here we do not further discuss the filtering algorithms as this is not the main focus of this paper. The yellow points in Plate 1a and 1b, indicate the ground.

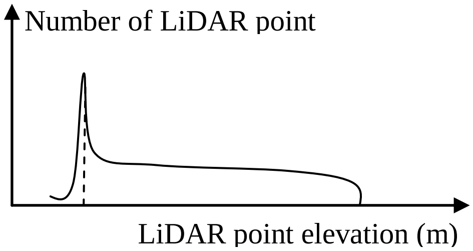


Figure 3 A typical example of elevation histogram of LiDAR points collected along urban streets

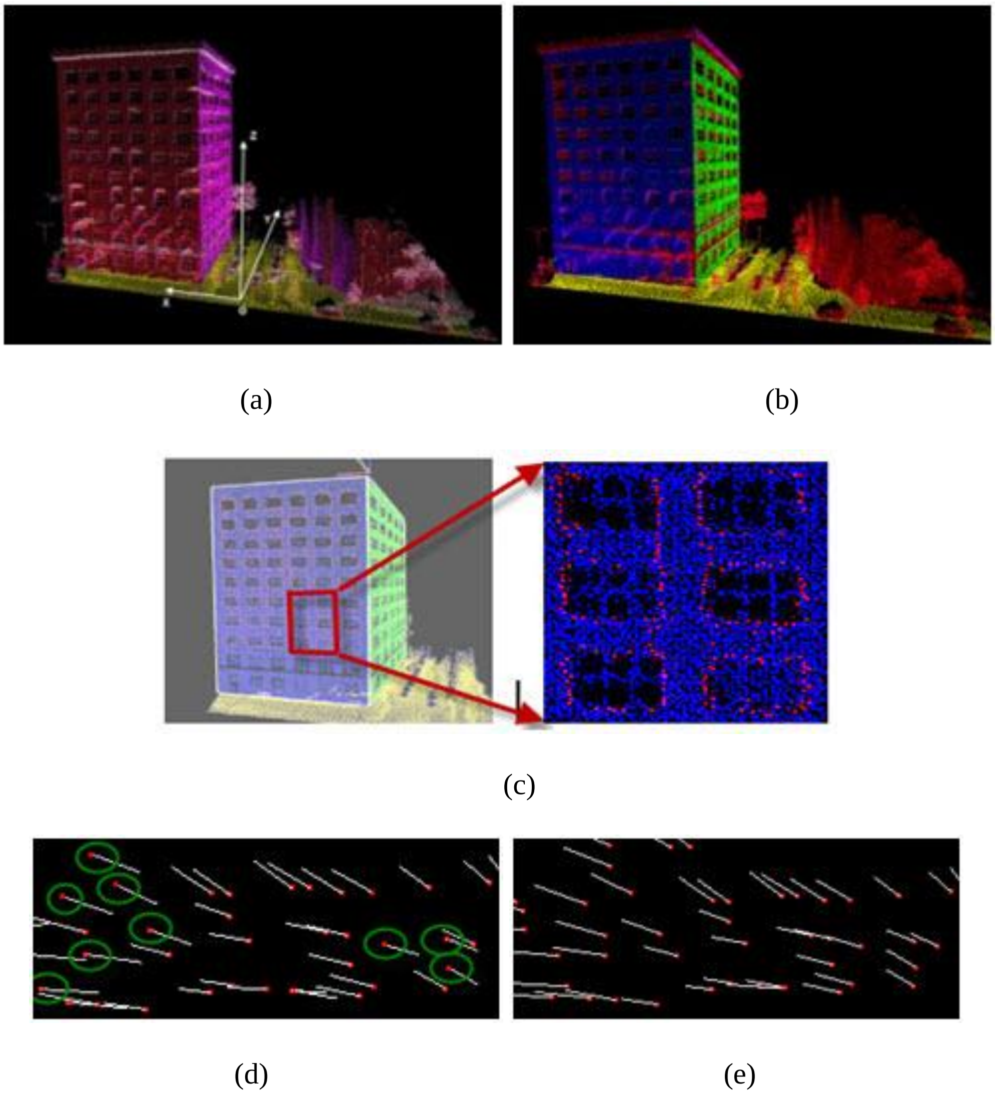
LiDAR Point Clustering

Surface Normal Computation We assume that buildings have rectilinear structure and facades have two major directions, vertical and horizontal, which is true for most buildings except some special landmarks. PCA has found many applications in the areas of remote sensing and computer vision, where it is often employed to transform the data into a new coordinate system for hyper-spectral image classification or face recognition. In this paper, we use PCA for plane fitting. We compute a normal of a point *p* using PCA. Let $\{p_i\}_{\mathrm{i=1:N}}$ be a set of neighboring points of *p*. We form a three by three positive semi-definite matrix (Hoppe *et al.*, 1992)

$$W = \frac{1}{N}\sum_{i=1}^{N}(p_i - \overline{p}) \otimes (p_i - \overline{p}), \tag{1}$$

where $\overline{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$ is the centroid of all the points. If $\lambda_1 \leq \lambda_2 \leq \lambda_3$ denote the corresponding eigenvalues of *W*, the eigenvector v_1 corresponding to the smallest eigenvalue λ_1 has the same direction as the normal of the plane to be fitted. The smaller λ_1 is relative to λ_2 and λ_3 , the flatter the distribution of $\{p_i\}$ is. The neighborhood is defined as a 3×3×3 voxel region centered at *p* in the point clouds. If the number of points within the neighborhood is smaller than three, we do not compute the normal for this point.

Definition of Normal Orientation The normal is often used to determine a surface's orientation toward a light source. The normal orientations computed from PCA are supporting lines whose directions are not defined. For instance, the normal from façade points can be either pointing outward or inward the façade. To achieve consistent normal orientations, we incorporate line-of-sight information from the laser scanner to define normal orientations. For each point *p*, we compute a vector *q* from this point to the laser origin. We choose the normal orientation which always results in a positive value from a dot product of the normal of *p* and *q*. Plate 1d shows an example of incorrect normal orientations indicated by circles. The dots indicate the starting point of the normal. Plate 1e shows correct consistent normal orientations after correction. Because of the noisy LiDAR data and the problem of inaccurate normal estimation around the boundaries, we cannot totally rely on the accuracy of the computed normals. Instead, we roughly classify the points into three categories according to the normal direction ranges: potential façade, potential façade side face, and unknowns as shown in Plate 1a. The red points are potential façade, and the pink points are potential side face. The unknown points are white points, in this case mostly from trees and building/window edges.



(a) (b)

(c)

(d) (e)

Plate 1. The LiDAR data process: (a) LiDAR point clustering using PCA, (b) Segmented LiDAR points, (c) Detected façade and potential window points, (d) Wrong normal orientations indicated by green circles, (e) Correct normal orientations

Façade Detection

The LiDAR data in geodetic coordinates are converted to the Universal Transverse Mercator (UTM) coordinates, and then to a local vehicle heading coordinate system *O_XYZ* as shown in Plate 1a. *X* axis is the vehicle heading direction, *Y* axis is perpendicular to the heading direction, and *Z* axis is vertically up from the ground. Under this coordinate frame, the point normal is manageable. Basically the potential façade regions are those surfaces where the point normal is roughly perpendicular to the *X*-*Z* plane. Potential façade side faces are those regions where the point normal is approximately perpendicular to the *Y*-*Z* plane. The RANSAC plane fitting is applied to the potential façade regions to extract the major plane which is the façade. The side face can be also extracted in road intersections where the side face is visible to the laser scanner. There are existing plane detection methods from sparse point clouds generated from structure from motion algorithms (e.g., Sinha *et al*., 2009), and there are also some variants of RANSAC (Raguram *et al*., 2008), but we use traditional RANSAC in this paper. RANSAC (Fischler and Bolles, 1981) is a general approach to fit a model to a set of data which contains a large portion of outliers. It starts by randomly drawing a minimal set from the point data to estimate the underlying model parameters. A minimal set is the smallest number of points required to uniquely define a given type of geometric primitive, e.g., three sample points define a unique plane. All other data are then tested against the fitted model to determine how many of the points are well approximated by the model. If a point fits well to the estimated model, it is considered as a hypothetical inlier (Here we use a threshold 0.2 meters. The distances of any points to the plane smaller than this threshold will be thought of as inliers). After a given number of iterations, the model which has the largest number of inliers is considered as the best fitted model. This model is then re-estimated from all the hypothetical inliers, because it has only been estimated from the initial set of hypothetical inliers. One key advantage of RANSAC over similar voting schemes like the Hough Transform is that the candidate points for estimating a model can be arbitrarily chosen. Its major deficiency is the considerable computational demand if no further optimizations are applied. We use PCA to compute a normal for each point and then refine the candidate point sets for the RANSAC plane fitting, which increases its computational efficiency. After the façade is extracted, all the façade points are separated from the entire point clouds. The RANSAC plane fitting is then applied to the remaining points to extract the façade side face. Plate 1b shows the segmented LiDAR points in which blue points represent façade, and green ones represent façade side faces. Note that the red points in the areas of windows from the façade are separated from the blue façade points.

Potential Window Points Detection

Some windows leave holes in the façade, but others do not. There are also points available from the crossbars of the low level windows, but no points available from the crossbars of the upper level windows. We first separate the ground floor from the façade because the pattern of the ground floor is often different from the window pattern, e.g. doors and special windows. In this paper, we simply exclude the first 10-30% of data starting from the bottom to the top of the façade depending on the types of the building data. In the future, this process will be automated through façade pattern analysis.

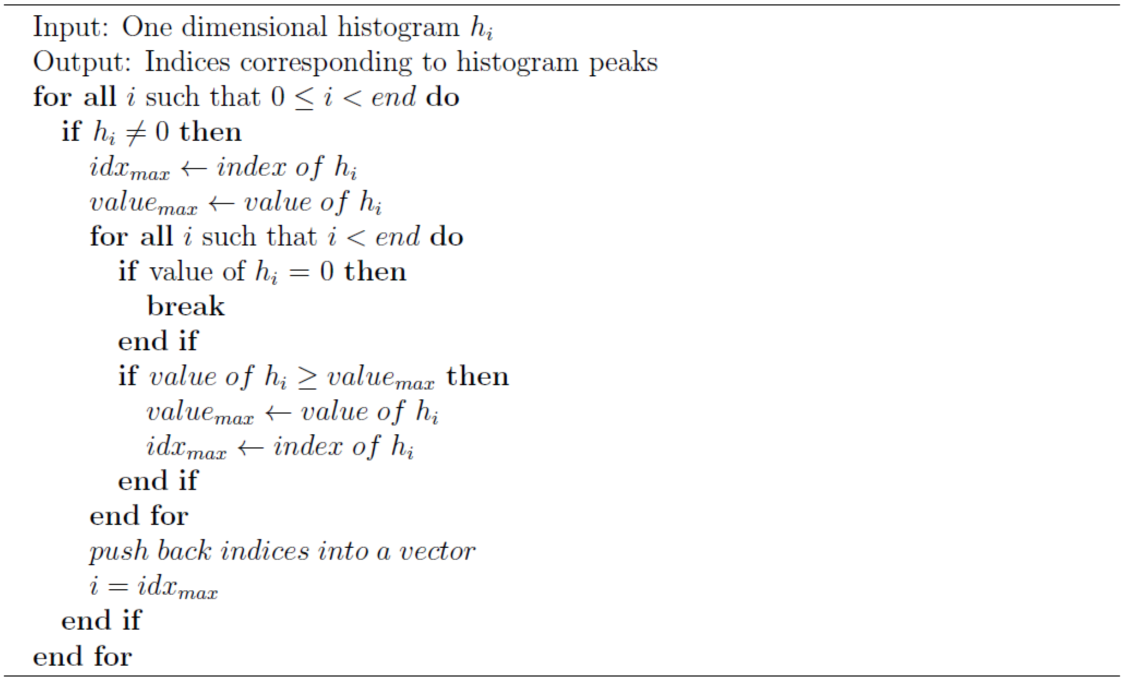
To detect potential window points, we distinguish four different types of window borders: horizontal structures at the top and bottom of the window, and two vertical structures on the left and right sides of the window. The window crossbars are not used to detect potential window points, but in general could provide valuable information about existence of windows. The detected façade is actually a 3D cube whose thickness is determined by the distance threshold (i.e. 0.2) from RANSAC. We first create a volumetric representation of the façade so that the point neighborhood relation can be manipulated. An operator is designed according to the window pattern of the façade to find potential window points excluding those points from window crossbars. Basically, for each point, we examine its neighbors along horizontal and vertical directions respectively. The upper horizontal window edge points are identified if upper neighbor points are found while lower neighbor points are not. A similar rationale is applied to find lower, left, and right window edge points. The window crossbar points are identified if both upper and lower neighbor points are not found, or both left and right neighbor points are not found. For each voxel (*i*, *j*, *k*) in the 3D façade cube, we denote, *f* (*i*, *j*, *k*) = 1, if there is a LiDAR point in this voxel, otherwise, *f* (*i*, *j*, *k*) = 0. Then the operator to find window edge points excluding window crossbars can be described by Equation (1).

$$\begin{cases} \begin{array}{lr} horizontal\ window\ edge\ points, & if\ \{ \sum_{k'=k}^{k'=k+inter} \sum_{i} \sum_{j} f(i,j,k')\} = 0\ \&\&\ \{ \sum_{k'=k}^{k'=k-d} \sum_{i} \sum_{j} f(i,j,k')\} = d \\ & OR\ \ if\ \{ \sum_{k'=k}^{k'=k-inter} \sum_{i} \sum_{j} f(i,j,k')\} = 0\ \&\&\ \{ \sum_{k'=k}^{k'=k+d} \sum_{i} \sum_{j} f(i,j,k')\} = d \\ vertical\ window\ edge\ points, & if\ \{ \sum_{i'=i}^{i'=i+inter} \sum_{j} \sum_{k} f(i',j,k)\} = 0\ \&\&\ \{ \sum_{i'=i}^{i'=i-d} \sum_{j} \sum_{k} f(i',j,k)\} = d \\ & OR\ \ if\ \{ \sum_{i'=i}^{i'=i-inter} \sum_{j} \sum_{k} f(i',j,k)\} = 0\ \&\&\ \{ \sum_{i'=i}^{i'=i+d} \sum_{j} \sum_{k} f(i',j,k)\} = d \\ non-window\ edge\ points, & otherwise \end{array} \end{cases} \qquad (1)$$

The constant variable *inter* is related to the interval between windows, and *d* is related to the width of window crossbars. These two values are experimentally selected according to the pattern of the façade. In this paper, the horizontal interval is 8 (1.6 meter), and the vertical interval 5 (1.0 meter). The *d* is 3 (0.6 meter). The sigma *f* term equal to zero means no points are found in the local neighborhood. The sigma *f* term equal to *d* means points are found at very voxel in the local neighborhood. This is a strong condition which effectively identifies potential window points while excluding points from window crossbars. The red points in Plate 1c show the detected potential window points while window crossbars are eliminated from the potential window points.

Window Localization

Determination of Window Locations To localize the windows, we project horizontal (parallel to the ground) and vertical (perpendicular to the ground) potential window points in horizontal and vertical directions to give a total of two projection profiles: a horizontal projection profile of the horizontal window edge points and a vertical projection profile of the vertical window edge points. To do so, we use the plane-sweep principle to sweep the façade along horizontal and vertical directions individually to count the total number of points in each of these sweeping planes. The profile histograms are generated and the small peaks are suppressed. To accurately localize windows, we develop an algorithm to find the histogram peaks, and the indices of rows or columns corresponding to the peaks are the window locations. Through these indices, we can compute the 3D coordinates of windows in the vehicle heading coordinate system *O_XYZ*. Figure 4 below gives an outline of this method.



```
Input: One dimensional histogram $h_i$
Output: Indices corresponding to histogram peaks
for all $i$ such that $0 \leq i < end$ do
  if $h_i \neq 0$ then
    $idx_{max} \leftarrow index\ of\ h_i$
    $value_{max} \leftarrow value\ of\ h_i$
    for all $i$ such that $i < end$ do
      if value of $h_i = 0$ then
        break
      end if
      if $value\ of\ h_i \geq value_{max}$ then
        $value_{max} \leftarrow value\ of\ h_i$
        $idx_{max} \leftarrow index\ of\ h_i$
      end if
    end for
    $push\ back\ indices\ into\ a\ vector$
    $i = idx_{max}$
  end if
end for
```

Figure 4 Determination of window locations

Pattern Constraints Simple architecture rules such as window size and spacing are automatically inferred from the locations of well detected windows. In this paper, windows from upper levels are always well extracted because data from upper levels of buildings are often cleaner than those from lower levels. This can be partially explained by either less occlusions or longer distance to the laser scanners in the data from upper levels of buildings. These architecture rules are automatically computed, and form a constraint to help identify the window boundary more robustly. For instance, the inferred façade patterns provide rough locations for potential windows, and any histogram peaks outside the rough locations will be considered as outliers and eliminated from further processing.

Experiments and Discussion

We tested the algorithm on six LiDAR datasets containing planar building facades with rectilinear windows. This experiment is limited to office tower like structures and does not reflect the general case. Figure 5a shows LiDAR points in which windows are either holes or planar surfaces. The detected windows and facades from Figure 4a are shown in Figure 5c. Although the LiDAR data are very noisy, these windows are well extracted from LiDAR data excluding some from ground floors (i.e. in row (1)-(2) of Figure 5). We also show corresponding intensity images in Figure 5b, in which facades are occluded by trees, traffic lights or other objects. The glass windows often strongly reflect sunlight which is problematic for data consistency (i.e. in row (2) and (4) of Figure 5b). Detecting windows from these types of optical images will be very challenging. This is yet another motivation to detect windows from LiDAR data. Figure 6 shows a blow up of scene 5 from Figure 5. In Figure 6, the white lines outline the detected windows.

Our method fails on glass buildings where not enough façade points are available (Figure 7a, 7b), and non-planar buildings (Figure 7a, 7b). First of all, the façade detection by RANSAC plane fitting is based on a planar façade assumption. Non-planar façades such as the one in Figure 7a and 7b violate this assumption. Second, not enough points on the façade make the RANSAC plane fitting fail. Our method also fails on buildings containing non-rectilinear windows (Figure 7c, 7d, 7e, 7f). Figure 8a and 8d show the potential window point detection on non-rectilinear windows in Figure 7c and 7e respectively, which turns out that our window edge detector works reasonable well. Figure 8b and 8c show histogram results for the window in Figure 7c. The vertical signatures correctly locate the vertical window boundaries. It's only the horizontal signatures that are problematic. We think a template matching scheme (Brunelli, 2009) that's seeded with the strong signature data will work well here, and leave this particular refinement for future research.

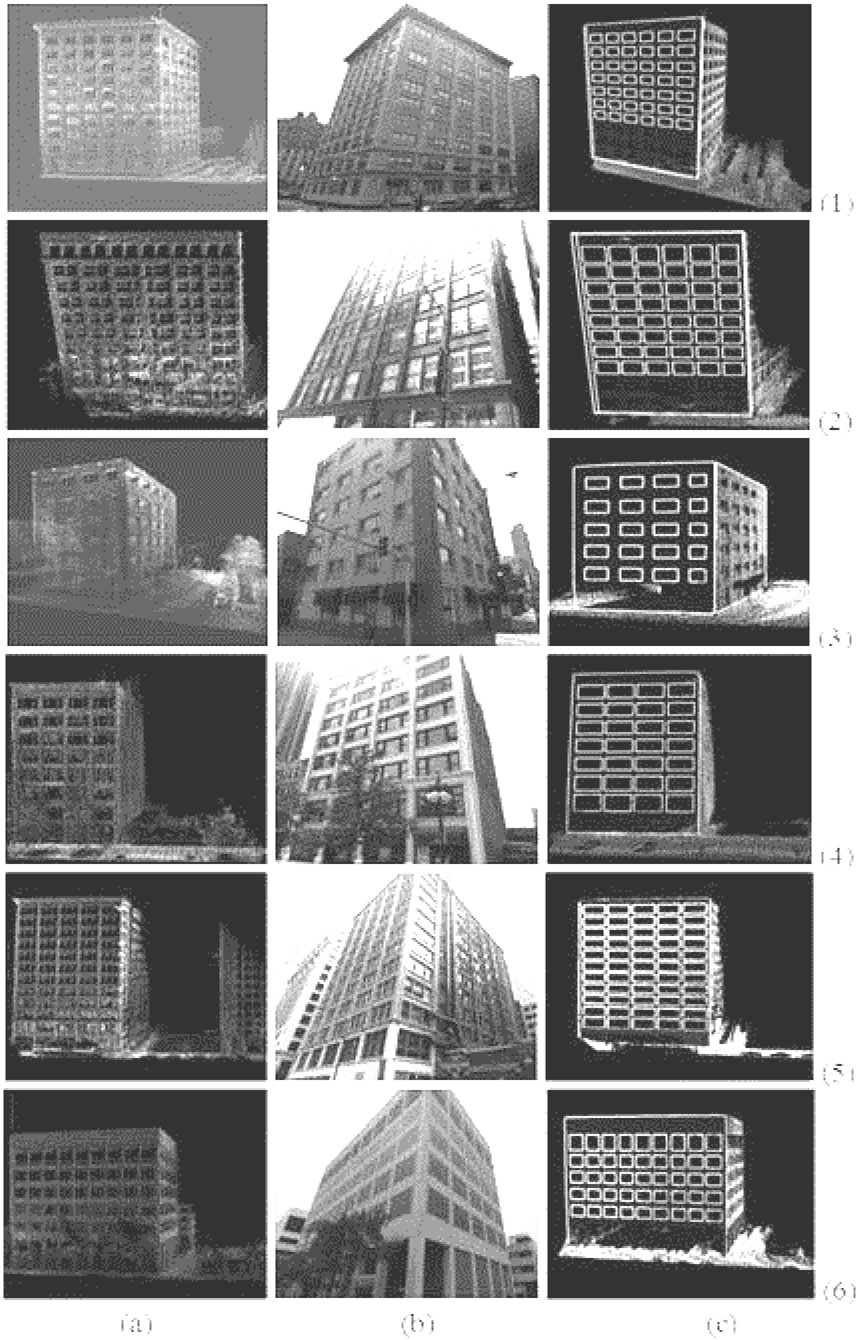


Figure 5 Experimental results (a) shows original colorized LiDAR points; (b) shows corresponding ladybug images; (c) shows detected windows and facades

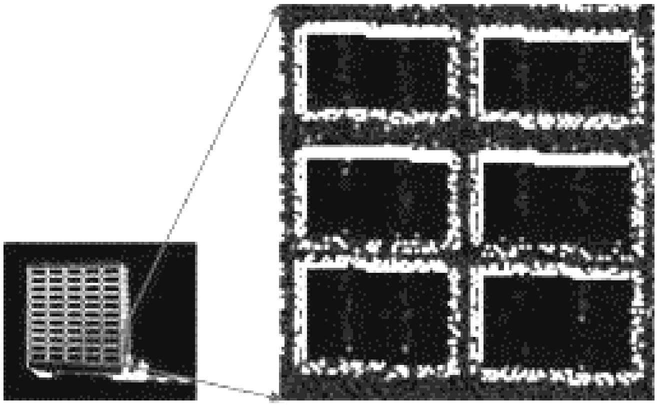
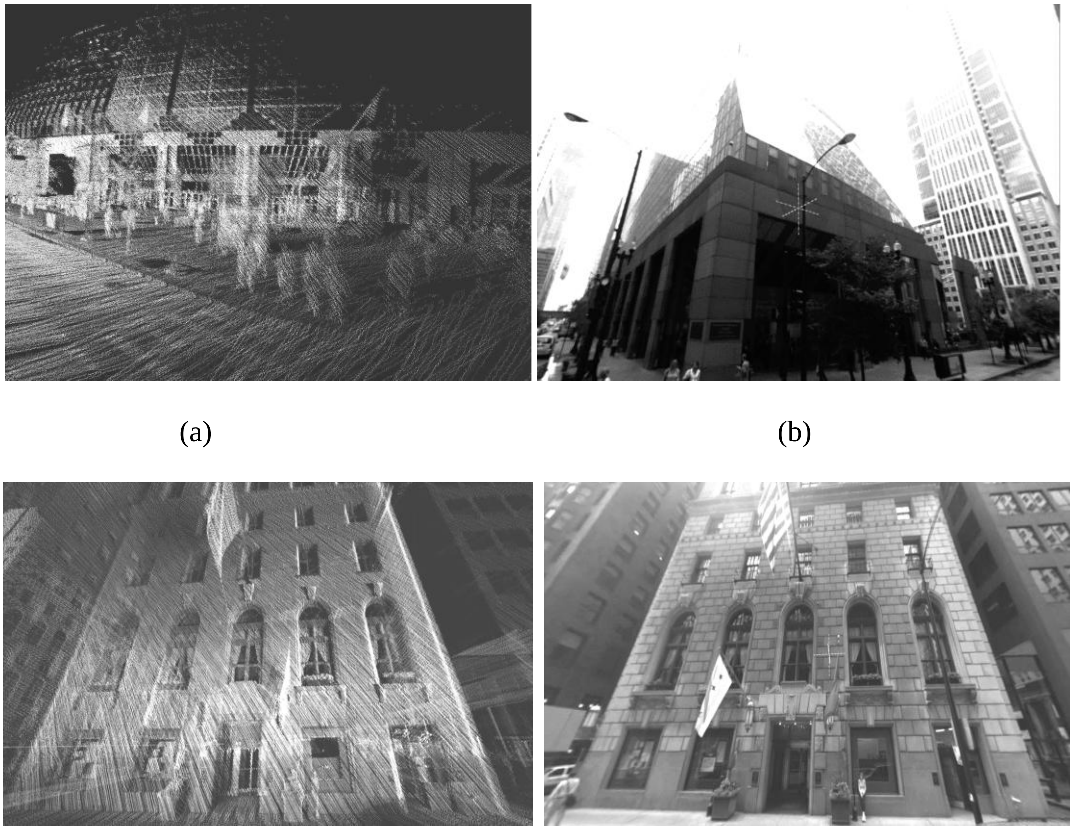


Figure 6 A close up of scene 5 in Figure 4

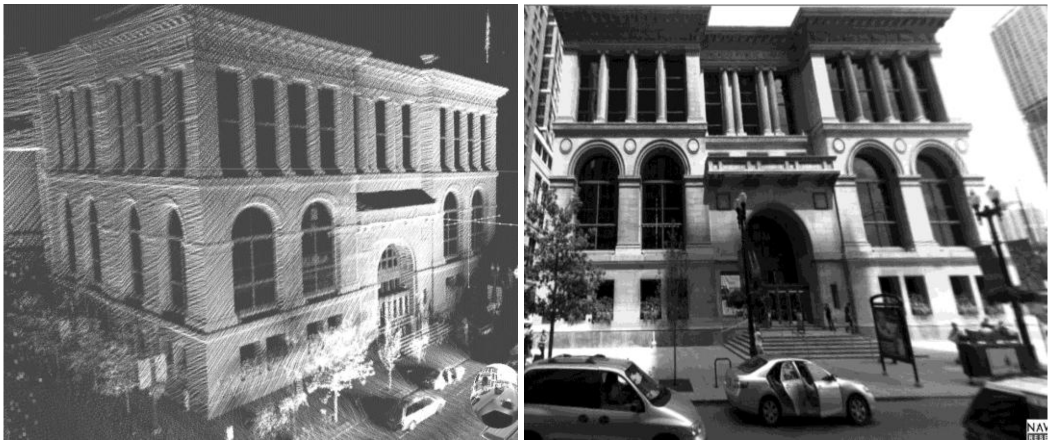


(a)

(b)

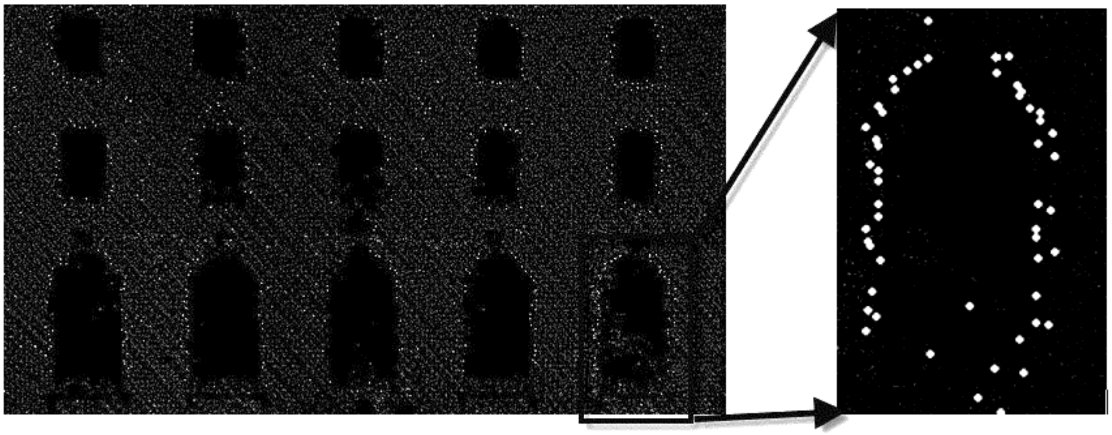
(c)

(d)

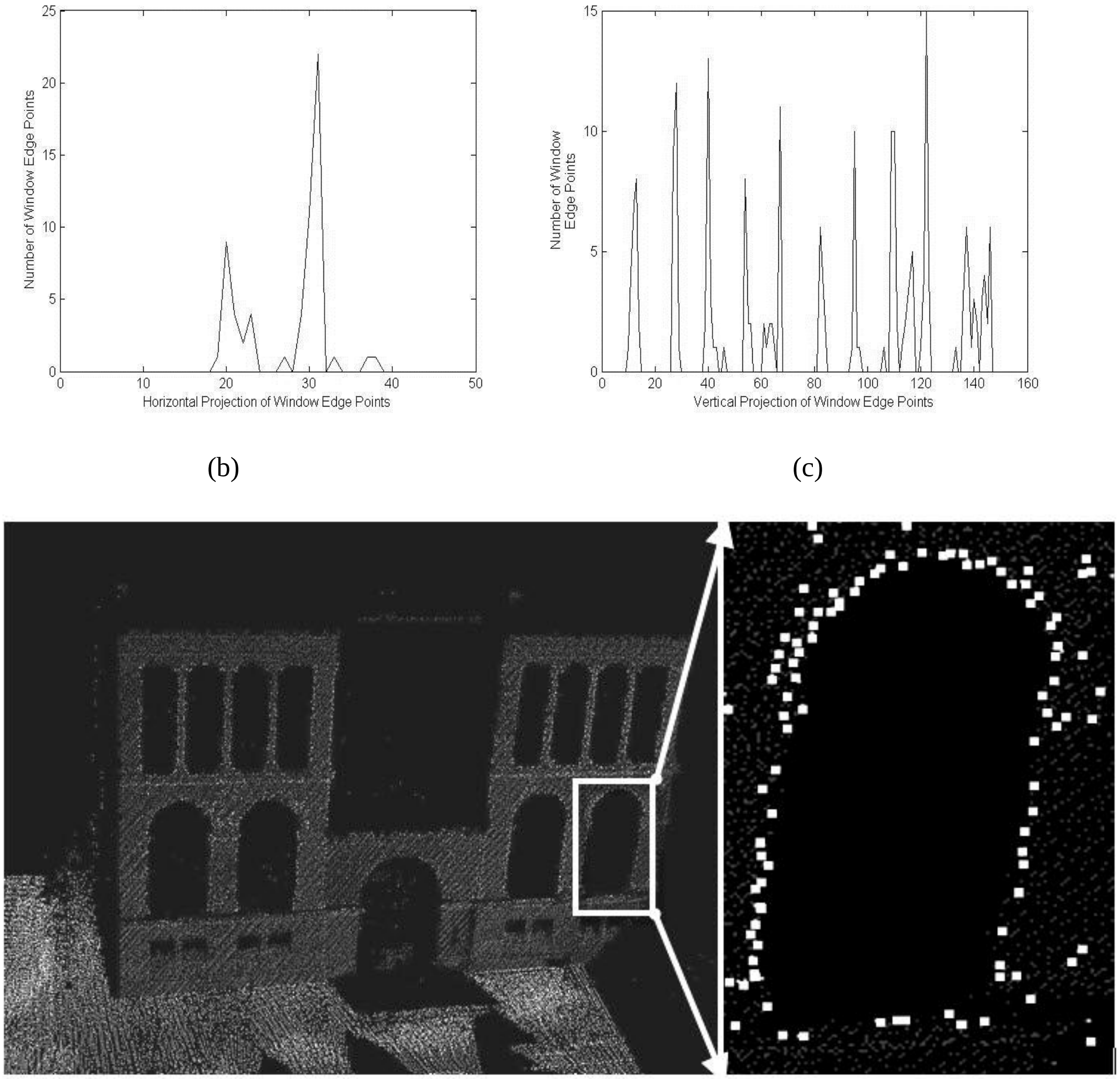


(e) (f)

Figure 7 Failure cases (a) the non-planar glass building; (b) the corresponding image of (a); (c) the building containing non-rectilinear windows; (d) the corresponding image of (c); (e) the building containing non-rectilinear windows; (d) the corresponding image of (e).



(a)



(b)

(c)

(d)

Figure 8 Experiment on arc windows in Figure 7c and 7e (a) Potential window point detection for the window in Figure 7c; (b) Horizontal window edge projection for the window in Figure 7c; (c) Vertical window edge point projection for the window in Figure 7c; (d) Potential window point detection for the window in Figure 7e.

Performance Evaluation

The implementation is in C++ code. The test results are obtained on an Intel Core 2 CPU laptop with 2GB of RAM. Table 1 shows the performance evaluation. All the LiDAR data and intensity images are in binary format, and window extraction was done completely automatically. In the data size column, the numerator in the fraction means the size of the data that was processed to extract windows. The denominator means the size of the data that was manipulated in order to be loaded. We only loaded relevant LiDAR points filtered by laser scanning angle and distance. For instance, in scene 2 (row (2) of Figure 5) we indexed 11 million LiDAR points to load 3.2 million points and filter out irrelevant LiDAR points for the processing. The loading process takes 12 seconds, and the algorithm run time is 13 seconds (both are in debug mode).

To evaluate the performance of the proposed window extraction method, we used two measures to assess the results described in the previous sections. The number of extracted windows was used to compute the completeness and correctness measures (Heipke *et al*., 1997). The completeness denotes the percentage of the reference windows that are extracted by our algorithm, and is defined by

$$completeness = \frac{number\ of\ matched\ reference\ windows}{number\ of\ all\ referenced\ windows}.$$

The correctness represents the percentage of correctly extracted windows with respect to all extracted ones, and is calculated by

$$correctness = \frac{number\ of\ matched\ extracted\ windows}{number\ of\ all\ extracted\ windows}.$$

In Table 1, the total number of reference windows is counted from LiDAR and images. The number of extracted windows is counted from the results. Completeness and correctness are given in the last two columns. For instance, in the first raw of Table 1, the completeness and correctness measures are 71.2% and 100%, which means that 71.2 percent of windows are successfully extracted and 100 percent of extracted windows are correct.

Table 1. Performance evaluation

Scene	Data Size (million)	Loading data Time (sec.)	Algorithm run time (sec.)	Reference Windows	Extracted Windows	Completeness (percentage)	Correctness (percentage)
1	1.8/7.0	7	11	59	42	71.2 %	100%
2	3.2/11	12	13	54	48	88.9 %	100%
3	1.7/6.5	7	10	20	20	100%	100%
4	0.9/3.0	3	13	28	28	100%	100%
5	3.0/10.0	12	14	55	55	100%	100%
6	1.9/6.0	6	12	45	45	100%	100%

One difficulty comparing with existing window detection methods is lack of a common test data set. It also has been very difficult to obtain the data set used in other people's methods. The method by Ali *et al.* (2008) uses a mobile LiDAR data set which is the closest to ours. We attempted to use the similar measure as Ali *et al.* (2008) to evaluate our results. According to the definition of Type I and type II errors, a false positive means if a window was detected where there is no window present; a true positive means if a window was detected where there is a window; a false negative means an existing window was not detected; a true negative means a non-existing window was not detected, namely, correctly rejecting a non-window. There are total 261 windows in our datasets while a total number of 196 windows in that paper. The values in Table 2 are all percentages computed over the total number of each respectively.

Table 2. Evaluation with the definition of Type I and type II errors

	Our method (Percentage)
True Positive	91.2%
False Negative	8.8%
True Negative	100%
False Positive	0%

Limitations

There are limitations in the current implementation of the approach. The first one is the assumption of rectilinear structure. It is hard to have a general model for any type of buildings. The rectilinear structure assumption or Manhattan-word assumption (Coughlan and Yuille, 1999) is quite common for many buildings as well as windows. We found this assumption is sufficient for the type of buildings tested in the paper. Our algorithms cannot perform correctly on the buildings containing non-rectilinear windows such as the building in Figure 6c, 6e.

The second is the data incompleteness. Due to the positioning constraint for any ground-level data acquisition system, it is difficult to always obtain complete and sufficient sampling of all the building surfaces. For instance, the rooftops and back of the buildings cannot be scanned by a vehicle-borne laser scanning system. To obtain a complete model, other data sources such as aerial liDAR/image data or building footprints from GIS are needed. Also the LiDAR scanner used in this paper can only provide usable returns up to 120 meters. Very high buildings may not have sufficient LiDAR points on the upper level of the facade. Another issue is the incomplete data caused by glass buildings such as the building in Figure 6a, 6b.

Conclusions and Future Work

We have proposed an automatic approach to window detection from mobile LiDAR data. The input is a chunk of LiDAR data, and the output is a detected façade and windows. This information can be used to generate a simpler description of the scene or potential texture synthesis. The main contributions of this paper are: a combination of bottom-up and top-down schemes to deal with noisy LiDAR data, and a robust window detection algorithm. There are a few limitations to the current implementation of the approach, but they can be improved. One possible direction is to separate connected buildings into individual buildings. In urban scene, a quite common situation is that many buildings with different heights or patterns are connected together. Efficient methods are needed to separate these connected buildings into individual buildings for a better symmetry pattern analysis. Another direction is to integrate images with LiDAR for better window detection as images normally have much higher resolution and more visual information than LiDAR. Figure 9 shows the overlay between LiDAR data and panoramic images collected from NAVTEQ True. Figure 9a shows the detected windows overlaid with the image and Figure 9b shows the detected windows overlaid with the corresponding edge image. As we mentioned before, window detection alone from images may be a challenging task. But we expect the combination of LiDAR and images will achieve much improvement over those using either LiDAR or images separately. The edge detection from high resolution images can provide more accurate boundary information which can increase the window detection accuracy. To make use of images, LiDAR and images must be accurately registered. In our data, the accuracy of the registration is not satisfied for this requirement. A mutual information based LiDAR-to-image registration algorithm is developed to improve the registration accuracy. We expect that, with the use of panoramic images collected simultaneously with the LiDAR data, a more robust and practical solution will be achieved.

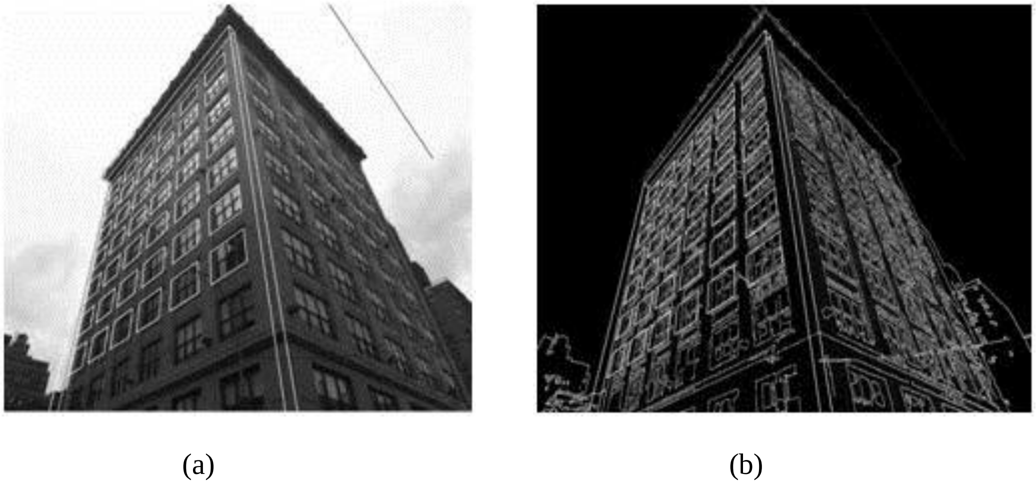


Figure 9 LiDAR and image overlay: (a) Detected windows of scene 1 from Figure 5 overlaid with the panoramic image; (b) overlaid with the corresponding edge image.

Acknowledgement

The first author would like to thank Matei Stroila, Roman Ostrovskiy, and three anonymous reviewers for their valuable comments.

References

Akel, N. A., S. Filin, and Y. Doytsher, 2009. Reconstruction of Complex Shape Buildings from Lidar Data Using Free Form Surfaces, *Photogrammetric Engineering & Remote Sensing*, 75(3), pp. 271-280.

Ali, H., B. Ahmed, and G. Paar, 2008. Robust Window Detection from 3D Laser Scanner Data. *Congress on Image and Signal Processing*, May 28-30 Sayna/Hainan, China.

Ali, H., C. Seifert, N. Jindal, L. Paletta, and G. Paar, 2007. Window Detection in Facades, *Proceedings of the 14th International Conference on Image Analysis and Processing*, pp. 837–842.

Becker, S., and N. Haala, 2009. Grammar supported facade reconstruction from mobile LiDAR mapping. *ISPRS Workshop on Object Extraction for 3D City Models, Road Databases and Traffic Monitoring*, Paris, Sep 3-4.

Bellon, O. R. P., and L. Silva, 2002. New Improvements to Range Image Segmentation by Edge Detection. *IEEE Signal Processing Letters*, Vol. 9, Issue 2, pp. 43-45.

Besl, P. J., and R. C. Jain, 1988. Segmentation Through Variable-Order Surface Fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, Issue 2, pp. 167-192.

Brunelli, R., 2009. Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2.

Chen, C., and I. Stamos, 2007. Range Image Segmentation for Modeling and Object Detection in Urban Scenes, *The 6th International Conference on 3-D Digital Imaging and Modeling*, Montreal, Canada, August 21-23.

Coughlan, J., and A. Yuille, 1999. Manhattan world: compass direction from a single image by bayesian inference. In Proceeding of *IEEE International Conference in computer Vision*, vol.2, 941-947.

Curless, B., and M. Levoy, 1996. A volumetric method for building complex models from range images. *In SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303-312, New York.

Debevec, P., C. Taylor, and J. Malik, 1996. Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach. *Proceedings of ACMSIGGRAPH*, 11–20.

Fischler., M. A., and R. C. Bolles, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395.

Filin, S. 2002. Surface clustering from airborne laser scanning data, in *ISPRS Commission III Symposium, Photogrammetric and Computer Vision*, Graz, Austria, pp. 119-124.

Frueh, C., S. Jain, and A. Zakhor, 2005. Data Processing Algorithms for Generating Textured 3D Building Facade Meshes from Laser Scans and Camera Images, *International Journal of Computer Vision*, Vol. 61, No.2, pp. 159-184.

Haala, N., S., Becker, and M. Kada, 2006. Cell Decomposition for the Generation of Building Models at Multiple Scales. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 36 (3), pp. 19-24.

Heipke, C., H. Mayer, C. Wiedemann, and O. Jamet, 1997. Evaluation of Automatic Road Extraction, *International Archives of Photogrammetry and Remote Sensing*, Vol. 32, No.3, pp.47-56.

Hohmann, B. U. Krispel, S. Havemann, D. Fellner, 2009. CityFit: High-quality urban reconstructions by fitting shape grammars to images and derived textured point clouds. *Proceedings of the 3rd ISPRS International Workshop 3D-ARCH* "3D Virtual Reconstruction and Visualization of Complex Architectures" Trento, Italy.

Hoover, A., R. B. Fisher, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert and A. Fitzgibbon, 1996. An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol: 18, Issue: 7, pp: 673-689.

Hoppe, H., T. Derose, T. Duchamp, J. Mcdonald, and W. Stuetzle, 1992. Surface reconstruction from unorganized points. *Computer Graphics* 26, 2, 71–78.

Jiang, X., U. Meier, and H. Bunke, 1996. Fast Range Image Segmentation Using High-Level Segmentation Primitives, *the 3rd IEEE Workshop on Applications of Computer Vision,* pp 83-88.

Lee, S. C., and R. Nevatia, 2004. Extraction and Integration of Window in a 3D Building Model from Ground View Images, *Proceedings of the IEE computer Society Conference on Computer Vision and Pattern Recognition*.

Ma, R., 2005. DEM Generation and Building Detection from Lidar Data, *Photogrammetric Engineering & Remote Sensing*, 71(7), pp. 847–854.

Mayer, H., and S. Reznik, 2005. Building Facade Interpretation from Image Sequences, *Proceedings of the ISPRS Workshop CMRT*.

Meng, X., L. Wang, and N. Currit, 2009a. Morphology-based building detection from airborne lidar data, *Photogrammetric Engineering and Remote Sensing*, 75(4):437-442.

Meng, X., L. Wang, J.L. Silván, and N. Currit. 2009b. A multi-directional ground filtering algorithm for airborne LIDAR. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(1): 117-124.

Mueller, P., G. Zeng, P. Wonka, and L. Van Gool, 2007. Image-based procedural modeling of facades. *ACM Transactions on Graphics* (TOG), Volume 26, Issue 3.

Pollefeys, M., D. Nister, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles, 2008. Detailed real-time urban 3d reconstruction from video, *International Journal of Computer Vision*, 78(2–3), pp. 143–167.

Poullis, C., and S. You, 2009. Automatic Reconstruction of Cities from Remote Sensor Data, *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), Miami, USA, June 2009.

Pu, S., and G. Vosselman, 2007. Extracting windows from terrestrial laser scanning, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, part 3/W52, Espoo, Finland, Sep. 12-14.

Raguram R., J. M. Frahm, and M. Pollefeys, 2008. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus, *European Conference on Computer Vision* (ECCV).

Reed, M. K., and P.K. Allen, 1999. 3-D modeling from range imagery: An incremental method with a planning component. *Image and Vision Computing*, 17:99-111.

Seitz, S. M., B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms, In: IEEE *Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 519–526.

Sinha, S. N., D. Steedly, and R. Szeliski, 2009. Piecewise planar stereo for image-based rendering. *In International Conference on Computer Vision* (ICCV).

Sohn, G., X. Huang, and V. Tao, 2008. Using Binary Space Partitioning Tree for Reconstructing 3D Building Models from Airborne LiDAR Data. *Photogrammetric Engineering & Remote Sensing*, 74(11):1425-1440.

Stamos, I., and P. K. Allen, 2002. Geometry and Texture Recovery of Scenes of Large Scale. *Journal of Computer Vision and Image Understanding*, Vol.88, No. 2, pp. 94-118.

Stiny, G., and J. Gips, 1972. Shape Grammars and the Generative Specification of Painting and Sculpture, *Proceedings of IFIP Congress* 71, pp. 1460–1465. North-Holland, Amsterdam.

Russakoff, D., C. Tomasi, T. Rohlfing, G. Maurer, 2004. Image Similarity Using Mutual Information of Regions, *Proceedings of the European Conference on Computer Vision*, Prague, Czech Republic, pp. 596–607.

Van Gool, L., G. Zeng, F. V. den Borre, and P. Müller, 2007. Towards Mass-produced Building Models, *Photogrammetric Image Analysis*, Munich, Germany, September 2007.

Viola, P., and W. M. Wells, 1997. Alignment by maximization of mutual information, *International Journal of Computer Vision*, 24:137-154.

Wahl, R., R. Klein, and R. Schnabel, 2007. Efficient RANSAC for Point-Cloud Shape Detection, In: Computer Graphics Forum, 26:2, pp. 214-226.

Wonka, P., M. Wimmer, F. Sillion, and W. Ribasky, 2003. Instant Architecture. *ACM Transactions on Graphics*, 22(3), 669–677.

Zebedin, L., A. Klaus, B. Gruber-Geymayer, and K. Karner, 2006. Towards 3d map generation from digital aerial images, *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6), pp. 413–427.

Zhou, Q, and U. Neumann, 2008. Fast and Extensible Building Modeling from Airborne LiDAR Data, *ACM SIGSPATIAL GIS*, 2008.

Zhou, Q, and U. Neumann, 2009. A streaming framework for seamless building reconstruction from large-scale aerial lidar data, *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2759-2766.