Abstract

We present an automatic approach to window and façade detection from LiDAR (Light Detection And Ranging) data collected from a moving vehicle along streets in urban environments. 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. The application potential includes generation of building façade models with street-level details and texture synthesis for producing realistic occlusion-free façade texture.

1. Introduction

Creation of 3D models from images has been an intensive research topic in the vision, graphics, and photogrammetry communities. With the rapid growth in geo-applications like Microsoft Bing Maps or Google Earth, and the fast evolution towards 3D GPS navigation systems, there is a clear need for highly realistic geometric models of the world, especially 3D building models. Currently, the creation of 3D building models is still a very interactive procedure. Any advance in productivity for the creation of such models would be extremely timely [1].

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. aerial 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 [2, 25]. However, the resultant crude building models 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.

Windows are important feature of façades, and often form repeated structures in buildings. In other words, façades exhibit rich redundancy. If all identical windows can be represented by a single window symbol, then the façade can be described with a few parametric symbols in a semantic way. Window detection is an important step to recover façade structures. Automatically inferring these structures and encoding them into grammar rules are desired.

1.1. 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 [3, 4, 5, 6, 7]. Methods can use either an explicit window model [6] and horizontal and vertical profiling of images to discover the repetitive pattern, or use an implicit window model and learning approach to recognize windows [4, 5]. Grammar-based methods were first proposed in the field of architecture [8], and have been successfully used in [9]. By image similarity detection, shape grammar generation from façade images can be automated [3]. Typical intensity-based similarity measures include cross-correlation [6] and mutual information [3, 18, 19, 20, 21]. The common assumption in these methods is that windows are 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 data side, there is much work on range data segmentation [e.g. 22, 23] from the vision and robotics communities. Normal computation at each LiDAR point is the first step in most range segmentation
algorithms [15, 16, 17], which is also a crucial step for precise region extraction. However, surface normals are often estimated inaccurately for points near boundaries. A good example in [14] shows cases where inaccurate normal estimation occurs. Without normal computation, planar surfaces can also be estimated directly by using RANSAC plane fitting [18]. 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.

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. For those windows with curtains, laser points are available but often not on the same plane as façade. To the best of our knowledge, papers [11, 12] are the only two research works on window detection from LiDAR so far. A hole-based extraction method is presented in [11]. 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. However, this bottom-up triangular meshing based method suffers from noisy LiDAR data. The method in [12] converts LiDAR data into distance images, and then employs image processing techniques like morphological operations and contour analysis to segment windows. This 3D-to-2D conversion causes information loss. Directly processing 3D LiDAR points is desired. The method in [10] creates meshes for the whole façade but without addressing the window detection issue.

1.2. Our Approach

This paper 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 approaches 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. The advantages of this two-step approach are (1) avoiding the problem of inaccurate normal estimation causing the misclassification of points, (2) greatly reducing wrong façade detections due to the complexity of the scene in the real world and very noisy LiDAR data, and (3) computational efficiency because of the fast convergence of the RANSAC algorithm.

The rest of the paper is organized as follows. The data acquisition is described in Section 2. Section 3 discusses the proposed method. Experimental results and discussions are presented in Section 4. Conclusions and future work are given in Section 5.

2. Data Acquisition

Data is collected by NAVTEQ using the data collection vehicle shown in Fig. 1 (left). This mobile mapping system is composed of a 360 degree LiDAR sensor (Velodyne HDL-64E), cameras, 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. All of these sensors are geo-referenced through a GPS and IMU.

Figure 1 Data collection vehicle “NAVTEQ True”

3. 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 is automatically inferred to form a constraint to enhance the robustness of the window detection. Each of these steps is explained in greater detail in the following subsections.
3.1 Ground Point Separation

The volume of LiDAR data is huge. A sub-ramping process is applied through establishing a volumetric representation of LiDAR data. The dimensions of the grid are computed based on the sampling rate which is automatically calculated either by specifying the desired resolution or the maximum allowed error tolerance between the samples. 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 histogram using elevation values of all the points in a Local Tangent Plane (LCP) 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. The yellow points in Figure 3a, 3b, 3c indicate the ground.

3.1. 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. We compute a normal of a point \( p \) using PCA.

Let \( \{p_i\}_{i=1}^N \) be a set of neighboring points of \( p \). We form a three by three semi-definite matrix [13]

\[
W = \frac{1}{N} \sum_{i=1}^N (p_i - \bar{p}) \cdot (p_i - \bar{p}),
\]

where \( \bar{p} = \frac{1}{N} \sum_{i=1}^N p_i \) 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 \( \lambda_1 \) has the same direction as the normal of the plane to be fitted. The smaller \( \lambda_1 \) is relative to \( \lambda_2 \) and \( \lambda_3 \), the flatter the distribution of \( \{p_i\} \) is. The neighborhood is defined as \( 3 \times 3 \times 3 \) voxel region centered at \( p \) in the point clouds.

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 scanning 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 \( q \). Figure 3d shows incorrect normal orientations indicated by green circles. The red dots show the starting point of the normal. Figure 3e shows correct consistent normal orientations.

Figure 3 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
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 the Figure 3a. 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.

3.2. Façade Detection

We convert the Universal Transverse Mercator (UTM) coordinates to a local vehicle heading coordinate system O_XYZ as shown in Figure 3a. 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 as shown in Figure 3b. 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 algorithm (e.g. [29]), and there are also some variants of RANSAC [24], but we use traditional RANSAC in this paper. Figure 3b 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.

3.3. 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 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 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). The constant variable \(\text{inter}\) 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. 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 Figure 3c show the detected potential window points while window crossbars are eliminated from the potential window points.

3.4. 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. Algorithm (1) below gives an outline of this method.
Pattern Constraints

Simple architecture rules 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. Architecture rules like window size and spacing are automatically computed, and form a constraint to help identify the window boundary more robustly.

4. Experiments and Discussions

We tested the algorithm on six LiDAR datasets. Figure 4a shows colorized LiDAR points in which windows are either holes or planar surfaces. The detected windows and facades from Figure 4a are shown in Figure 4c. 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 4). We also show corresponding intensity images in Figure 6b, 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 4b). Detecting windows from these types of optical images will be very challenging.

This is yet another motivation to detect windows from LiDAR data. Figure 5 shows a blow up of scene 5 from Figure 4. In Figure 5, the blue dots are façade, red and green dots are potential vertical and horizontal window points respectively. The white lines outline the detected windows.

### Algorithm 1: Determination of Window Locations

```
Data: One dimensional histogram
Result: Indices corresponding to histogram peaks

1. for each histogram bin \( h_i \) to end do
2.   if \( h_i \) not equal to 0 then
3.     max_idx = index of \( h_i \)
4.     max_value = value of \( h_i \)
5.   for each \( h_i \) to end do
6.     if value of \( h_i \) = 0 break
7.     if value of \( h_i \) >= max_value then
8.       max_value = value of \( h_i \)
9.       max_idx = index of \( h_i \)
10.  push_back indices into a vector
11.  i = max_idx
```

**Fig. 4.** Experimental results. (a) shows original colorized LiDAR points. (b) shows corresponding ladybug images. (c) shows detected windows and facades.

---

### Formula (1)

Horizontal window edge points,

\[
\begin{align*}
\text{if } \sum_{k' \neq k} \sum_{j} f(i, j, k') &= 0 \text{ \&\& } \sum_{k} \sum_{j} f(i, j, k') \\
\text{OR if } \sum_{k' \neq k} \sum_{j} f(i, j, k') &= 0 \text{ \&\& } \sum_{k} \sum_{j} f(i, j, k') \\
\text{vertical window edge points,} &
\end{align*}
\]

\[
\begin{align*}
\text{if } \sum_{i' \neq i} \sum_{k} f(i', j, k) &= 0 \text{ \&\& } \sum_{i} \sum_{k} f(i', j, k) \\
\text{OR if } \sum_{i' \neq i} \sum_{k} f(i', j, k) &= 0 \text{ \&\& } \sum_{i} \sum_{k} f(i', j, k) \\
\text{non - window edge points,} &
\end{align*}
\]
4.1. Implementation and Performance

The implementation is in C++ code. We use the library from the Mobile Robot Programming Toolkit [22] for RANSAC implementation, and the GNU scientific library for eigenvector and other linear algebra calculations. 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 6) 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.

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 [28]. The completeness denotes the percentage of the reference windows that are extracted by our algorithm, and is defined by

\[
\text{completeness} = \frac{\text{number of matched reference windows}}{\text{number of all referenced windows}}
\]

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

\[
\text{correctness} = \frac{\text{number of matched extracted windows}}{\text{number of all extracted windows}}
\]

In Table 1, the total number of reference windows is counted from the results. Completeness and correctness are given in the last two columns. For instance, in the first row 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.

To compare with existing methods [12, 11] on window extraction from LiDAR data, we also used another measure to evaluate our results. We counted a true positive if a window was correctly detected, false negative if an existing window was not detected, true negative if the window is present but not detected as a window, and false positive if a window was detected where there is no window present. Table 2 shows the comparison with the existing method [12]. There are total 261 windows in our datasets while a total number of 196 windows in the paper [12]. The values in Table 2 are all percentages computed over the total number of each respectively.

Table 2. Comparison with the existing method

<table>
<thead>
<tr>
<th></th>
<th>Our method (Percentage)</th>
<th>Method [12] (Percentage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive</td>
<td>91.2%</td>
<td>68.9%</td>
</tr>
<tr>
<td>False Negative</td>
<td>8.8%</td>
<td>10.7%</td>
</tr>
<tr>
<td>True Negative</td>
<td>0%</td>
<td>13.8%</td>
</tr>
<tr>
<td>False Positive</td>
<td>0%</td>
<td>6.6%</td>
</tr>
</tbody>
</table>

The method [11] doesn’t provide such statistics, and only reported that more than 90% of the windows can be accurately recognized, which is similar to the true positive rate in our paper. But we have much lower false positive and true negative rate than the method [11].
4.2. Limitations

There are limitations in the current implementation of the approach.

**Rectilinear structure assumption** It is hard for any problem to have a flexible model with many degrees of freedom. Imposing reasonable priors on building and window regularity is a compromise for robustness and automation of the algorithm. The rectilinear structure assumption or Manhattan-word assumption [27], is quite common for many buildings as well as windows. For those buildings with complicated shapes, the rectilinear structure assumption can still be a first-level approximation of an arbitrary surface [26]. We found this assumption is sufficient for the type of buildings tested in the paper.

**Incomplete Data** 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.

Figure 6 LiDAR and image overlay: (a) Detected windows of scene 1 from Fig.4 overlaid with the panoramic image (b) overlaid with the corresponding edge image

5. 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. Figure 6 shows the overlay between LiDAR data and panoramic images collected from NAVTEQ True. Figure 6a shows the detected windows overlaid with the image and Figure 6b shows the detected windows overlaid with the corresponding edge image. Next step, we plan to use image data to improve the accuracy and robustness of the detection as images normally have much higher resolution and more visual information than LiDAR. 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. 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 being 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.
References