AUTOMATIC EXTRACTION OF BUILDING FEATURES FROM TERRESTRIAL LASER SCANNING

Shi Pu and George Vosselman

International Institute for Geo-information Science and Earth Observation (ITC)
spu@itc.nl, vosselman@itc.nl

Commission VI

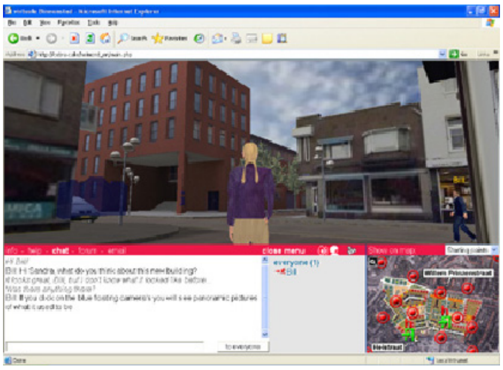
KEY WORDS: 3D modeling, terrestrial laser scanning, segmentation, feature recognition

ABSTRACT:

Realistic 3D city models are required for many purposes such as urban planning and safety analysis. The traditional modeling methods are mainly based on manual or semi-automatic reconstruction from close-range images, which are rather time-consuming. The lack of fully automatic reconstruction is mainly due to the difficulty to recover building structures from close-range images. The terrestrially scanned laser points usually contain useful information, and it can be a valuable data source for reconstructing 3D city models. This paper presents our approach to automatically extract building features from terrestrial laser scanned data. This is done by first processing the terrestrial laser points with various segmentation algorithms, then retrieving several important properties (size, position, direction, topology, etc.) from the segments, and finally recognizing potential building features (walls, windows and doors, etc.) with feature constraints, which are based on the properties of segments. The recognized features will form the basis for an automatic 3D building model reconstruction framework.

1 INTRODUCTION

Realistic 3D city models are required for many purposes such as urban planning and safety analysis. Originally the involvement of citizens in urban planning is generally limited to 2D design plans, which may be difficult to interpret. The availability of 3D models of the current urban environment, as well as new urban objects and their alternatives, would increase this involvement remarkably. 3D city models can also play an important role at security analysis as well. A realistic environment is essential for making good security analysis and training, particularly where physical security (security of infrastructure) and social security (livability, security feeling) are concerned. Figure 1 gives an example of the virtual city model of Helmond, the Netherlands.



Bill: Hi Sandra, what do you think about this new building?
It looks great, Bill, but I don't know what it looked like before.
Was there anything there?
Bill: If you click on the blue floating camera's you will see panoramic pictures of what it used to be

Figure 1: VR Helmond

However, nowadays the automatic construction of realistic city model is still not feasible. The traditional modeling methods are mainly based on manual reconstruction from 2D GIS data and close-range images, or semi-automatic reconstruction from close-range images. The manual approach normally starts with building outlines generated from 2D GIS data. Then the 2D outline is simply elevated with certain height to a 3D rough model. Detailed 3D structures can be created on the rough model with commercial 3D modeling packages such as 3DStudio Max. Finally the 3D model is textured by manually selecting certain parts from close-range images. Due to the huge number of urban objects in a city and variety of shapes, manual creation of a city model is a rather time-consuming and expensive procedure. There are also a few researches on semi-automatic city model reconstruction from close-range images (Dick et al. 2000, Schindler and Bauer 2003), but the results are not satisfactory. This is mainly due to the difficulty to recover 3D building structures from 2D images. Several studies (Brenner 2000, Maas 2001, Vosselman et al. 2004) show that laser scanning data can be a valuable data source for the automatic city model reconstructing. Comparing to close-range imagery, terrestrial laser scanning gives explicit 3D information, which enables the rapid and accurate capture of the geometry of a complex building facade; terrestrial laser scanning also provides high density point clouds, which gives enough raw data from which accurate and detailed 3D models can be obtained (Rabbani 2006). The reconstruction process based on terrestrial laser scanning can be generalized as three steps: feature recognition, where important building features (walls, windows, doors, etc.) are extracted; model fitting, where recognized features are fitted to simple geometric shapes such as polyhedron; model reconstruction, where models are combined from fitted geometric shapes and other data sources.

This paper presents our approach to extract building features from terrestrial laser scanned data. Section 2 gives an overview of our recognition method. Section 3 introduces the planar surface growing algorithm for segmentation. Section 4 describes the recognition procedure, by first defining the feature constraint categories and then giving specific feature constraints for different features. Sections 5 analyzes the recognition quantity. Some conclusions and future work are given in the last section.

2 METHOD OVERVIEW

A laser scanning point cloud contains information about building facades with x, y, z coordinates. However, the facade structures are not directly understandable by machines. Humans can easily find building features by comparing a set of characteristics such as position, color, topology, etc. For example, we know the feature ”wall” is usually the biggest plane in building facade, and it is usually vertical; we know the feature ”window” is on the wall, and it has a certain area range; we know the feature ”roof” is above the wall, and it is never vertical. This human knowledge about buildings can be ”modeled” and ”taught” to machines, so that features can be recognized from point clouds automatically.

The feature recognition procedure starts with segmentation, where a point cloud is categorized into different groups so that the points belonging to the same surface or region are in the same group. Each group (segment) is considered a potential building feature and will be analyzed further in the later stage. Next, several important properties (size, position, direction, topology, etc.) are retrieved from the segments. A couple of feature constraints are defined for each building feature, based on human knowledge about buildings. In the final step, building features are recognized out of segments by checking each segment’s properties through the feature constraints.

3 SEGMENTATION

Segmentation is the process of labeling each measurement in a point cloud, so that the points belonging to the same surface or region are given the same label. Building features can be roughly extracted from point cloud after segmentation, because different features usually belong to different surfaces or regions.

Several segmentation algorithms based on laser point cloud are available, and we adopted the planar surface growing algorithm by (Vosselman et al. 2004) because it is more suitable for segmenting planar surfaces. We only include a short explanation about the planar surface growing algorithm here because of its strong relevance to our feature recognition method.

The planar surface growing algorithm consists of the following steps:

1. Determine a seed surface. A seed surface consists of a group of nearby points that fit well to a plane. The algorithm selects an arbitrary unclassified point and tests if a minimum number of nearby points can be fitted to a plane. If this is the case, these points constitute the seed surface. Otherwise, another arbitrary point is tested.
2. Grow the seed surfaces. The growing can be based on one or more the following criteria:
   - **Proximity of points**. Only points within certain distance to a seed surface can be added to this seed surface.
   - **Globally planar**. For this criterion a plane equation is determined by fitting a plane through all surface points in this seed surface. Points can only be added if the perpendicular distance to the plane is below some threshold.

Figure 2 shows the original laser scanned point cloud of a building facade, and Figure 3 shows the segmentation results of this point cloud. It is clear that most windows, doors, roofs and extrusions are segmented successfully, although there are still some significant errors, for example, the wall results in 2 segments, and some parts of windows are not segmented.

In the ideal situation, each segment represents a building feature, and each building feature is represented in a segment. However, this can be hardly achieved because:

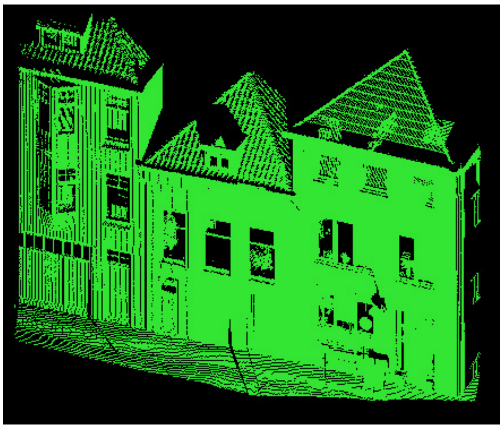


Figure 2: A terrestrial laser scanned building facade

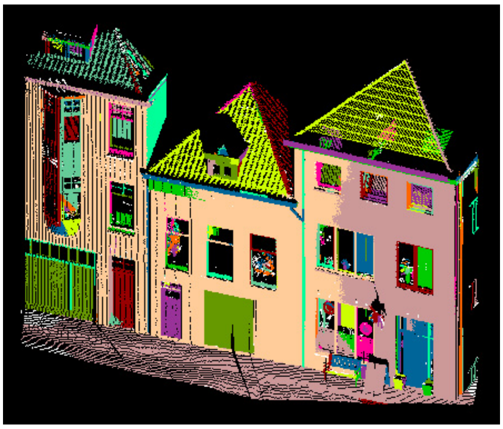


Figure 3: Segmentation result of a building facade

- A laser point cloud always contains irrelevant data, such as the objects behind windows, cars, benches besides buildings, and so on. The segmentation results normally contain these irrelevant segments too.
- Segmentation results are not 100% correct. Bad segmentation results in over-segmentation (one feature segmented to several segments), or under-segmentation (several features segmented to one segment), or miss segmentation (feature is not segmented).

Several parameters need to be specified for the planar surfaced growing algorithm, such as the number of seeds, the surface growing radius, the maximum distance between surfaces, etc. With the same number of seeds, larger surface growing radius or larger maximum distance between surface lead to fewer segments, which brings under-segmentation. In the other hand, smaller surface growing radius or smaller maximum distance between surface lead to more segment/over-segmentation. Our experience is that it is usually better to over segment laser point than under segment, because over-segmented parts have some similar properties, for example, they are attached to each other, they belong to the same plane, etc. According to feature constraints and these similar properties, over-segmented parts can always be combined to a complete feature in later phase. But under-segmented parts can hardly be split again.

4 FEATURE RECOGNITION

The result from segmentation gives potential building features, but it is still unknown whether a segment represents a feature, or which kind of feature a segment represents. Each building feature has its own characteristics, which can be formulated as feature constraints that are understandable by machines, so automatic feature recognition becomes possible.

4.1 Feature constraints

Considering the human knowledge about building features, and supposing all the building features are planar, we summarized a set of feature constraint categories as below:

- Size constraint. Walls, windows and doors can be easily distinguished from other features or noise segments by their sizes, as stated before.
- Position constraint. Certain features appear only in certain positions. For example, windows and doors are always on the walls; roofs are always on the top of walls.
- Direction constraint. Walls and roofs can be distinguished by their directions, as walls are usually vertical and roofs are not.
- Topology constraint. Building features have certain topology relation with other features or grounds. For example, ground always intersects some walls; roofs always intersect walls.
- Miscellaneous constraint. Some other information can be also helpful to feature recognition. For example, sometimes segments for windows usually have much lower point density, because glass reflects fewer laser pulses than other parts of a building. But this is just an optional constraint category, as sometimes windows are covered with curtains and reflect more laser pulses.

4.2 Recognition

We list the feature constraints for 7 important features (ground, wall, window, roof, door, extrusion, intrusion) in Table 1. Considering ground is very helpful to recognize building features, we also include ground as one of the features to be recognized, although ground itself is not a building feature.

Properties of a segment, such as area, direction and topology, can be hardly determined directly, as a segment is just a group of points. However, we could approximate each segment with a convex hull of all the points in this segment, and use the properties of this convex hull instead. In other words, the size of a segment is approximated with its convex hull's area; position is approximated with its convex hull's geometry center; direction is simply the normal of its convex hull; topology of segments can be approximated with topology of their convex hulls; and so on. Figure 5 gives an example of the convex hulls of the roof segments in Figure 4.

The implementation of door feature recognition is given below as an example. Recognition of other features have the similar structure and their implementations will not be repeated.

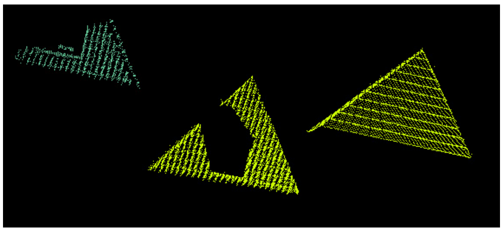


Figure 4: 3 roof segments

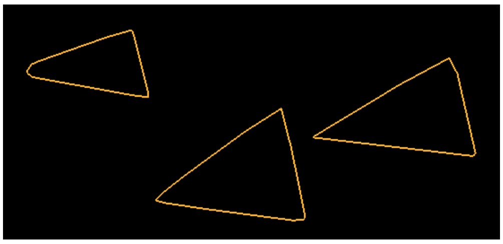


Figure 5: Convex hulls of 3 roof segments

```
Inputs: An array of convex hulls of all segments = hulls; estimated minimum door area = min; estimated maximum door area = max; convex hull of ground = ground; convex hull of wall = wall; distance threshold = d
Initialize counter $i = 0$
while $i < hulls.size$ do
    if $hulls[i].area3D \le min$ or $hulls[i].area3D \ge max$ then
        Not a door feature. Continue with the next convex hull.
    end if
    if $distance(hulls[i].center, wall) > d$ then
        Not a door feature. Continue with the next convex hull.
    end if
    if not $hulls[i].isVertical$ then
        Not a door feature. Continue with the next convex hull.
    end if
    if $hulls[i].Intesect(ground) = false$ then
        Not a door feature. Continue with the next convex hull.
    end if
    This is a door feature. Push $i$ to door feature list.
end while
```

For some features, the recognition is based on other already recognized features, or, in other words, certain features have higher priority during the recognition process. According to Table 1, recognition of window, roof, door, extrusion and intrusion require comparing with wall or ground, hence wall features and ground features should be recognized before processing the other 5 features. Recognition of walls requires again checking intersection with ground, therefore ground feature has even higher priority than wall feature. Recognitions of extrusion/intrusion are dependent on wall or roof, which yield another priority adjustment. So finally the recognition priority for the 7 features should be:

Ground>Wall>Roof=Window=Door>Extrusion>Intrusion

	Size	Position	Direction	Topology	Miscellaneous
Ground	Segment(s) with large area	Lowest			
Wall	Segment(s) with large area		Vertical	May intersect with the ground	
Window	Area from min < max	On the wall	Vertical		Low laser points clouds density
Roof	Segment(s) with large area	Above wall	Not vertical	Intersects with a wall	
Door	Area from min < max	On the wall	Vertical	Intersects with the ground	
Extrusion		A little bit outside the wall/roof		Intersects with a wall	
Intrusion		A little bit inside the wall/roof		Intersects with a wall	

Table 1: Constraints for 7 features

4.2.1 Ground Although ground is not a building feature, many building features can be easily recognized by comparing with ground. For example, segments of window and door have similar size, position and direction. However, doors always intersects with ground, while most windows usually don't.

A ground feature has a relatively large area, and it is the segment with the lowest position. Figure 6 shows the recognized ground from the segmented terrestrial laser scanning point cloud in Figure 3.

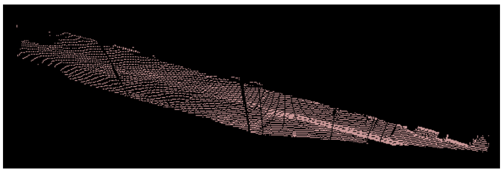


Figure 6: Recognized ground feature

4.2.2 Wall Wall feature is probably the most important building feature, and the determination of many features also depends on the detected wall features.

A wall feature has a relatively large area, it is vertical, and it intersects with the ground. Figure 7 left shows the recognized walls from the segmented terrestrial laser scanning point cloud in Figure 3.

Note that actually 2 segments are recognized as walls, which means the wall is over segmented. In this case we can automatically merge the 2 segments into 1 (Figure 7 right), according to that they face the same direction and are attached to each other.

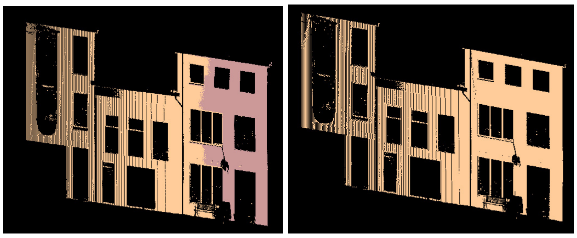


Figure 7: Recognized wall feature (left:before merge; right:after merge)

4.2.3 Roof A roof feature has a large area, it is above the wall, it is not vertical, and it has intersection with wall. Figure 4 shows the recognized roofs.

4.2.4 Door A door features has certain area range, it is on the wall, it is vertical, and it has intersections with ground level. Figure 8 shows the recognized doors.

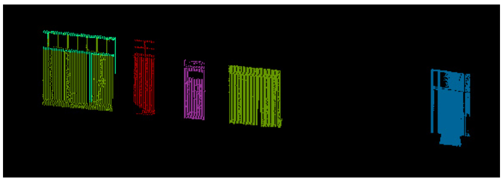


Figure 8: Recognized door feature

4.2.5 Window A window feature has certain area range, it is on the wall, it is vertical, and sometimes the point density is lower than average point density for all segments. Figure 9 shows the recognized windows after automatically merging over segmented segments.

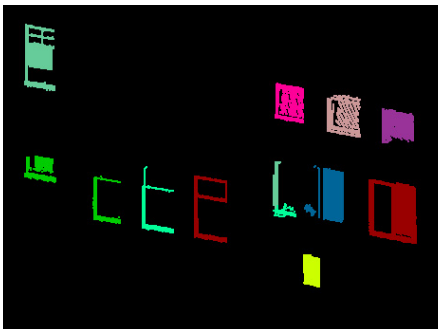


Figure 9: Recognized window feature

4.2.6 Protrusion A protrusion feature is a little bit outside the walls, and it has intersection with walls. Figure 10 shows the recognized protrusions.

4.2.7 Intrusion An intrusion is a little bit inside the walls, and it has intersection with walls. No intrusion is recognized from the building facade in Figure 3.

5 QUALITY ANALYSIS

The recognition method is implemented with C++ code, and experimented on a PC with Pentium 4 3.2G CPU, 1GB system memory and the NVidia Quadro FX540 video card. The tested

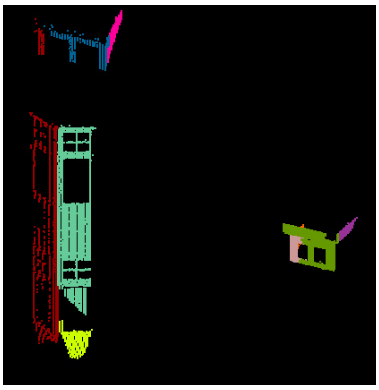


Figure 10: Recognized protrusion feature

point clouds contains 238034 points, and the overall running time is 42 seconds, including 18 seconds for segmentation, 7 seconds for convex hull creation and 17 seconds for feature recognition. We are satisfied with the processing speed.

	Total number	Recognized number
Ground	1	1
Wall	1	1
Window	11	7
Roof	3	3
Door	5	5
Protrusion	3	3
Intrusion	0	0

Table 2: Recognition quality

Table 2 gives the recognized numbers of the seven features. Ground, wall, roof, door and protrusion are all very well recognized, while only 7 window features are correctly recognized out of 11. This difference in recognition rate is mainly due to the different segmentation quality. The planar surface growing algorithm always starts with a seed surface, which is a group of nearby points that fit well to a plane. In the growing phase points are added to the seed surface, if the distance of a point to the plane is below some threshold. The plane parameters are updated after every added point, so the larger the plane the more reliable the plane parameters will be. That is why large features such as ground or wall will generate more reliable surfaces. However, windows have smaller size and windows without curtain reflect very few laser pulses. The points for windows are sometimes not enough to grow reliable surfaces, and hence windows are only partially segmented, or not segmented at all.

6 CONCLUSIONS AND FUTURE WORK

In this paper we described our approach to automatically extract building features from a terrestrial laser scanned point cloud. First the point cloud is segmented to several planar parts according to the point's location, direction, and the plane it belongs to; then we formulated a series of feature constraint categories, which represents the most significant characteristics that can distinguish one feature from another and the noise segments; finally each segment is checked through all feature constraints, to determine which kind of feature its is, or just noise segment.

The future work will be mainly on improvement of recognition rate, automatic determination of parameter values, and recognition of non planar features. All these detected features will then be used in an automatic process of modeling building facades.

ACKNOWLEDGEMENT

This research is partially funded by the Dutch BSIK research programme Space for Geo-Information, project Virtual reality for urban planning and safety. The authors would also like to thank the Oranjewoud and Cyclomedia for providing the data.

REFERENCES

Brenner, C., 2000. Towards Fully Automatic Generation of City Models, International Archives of Photogrammetry and Remote Sensing, vol. XXXIII-B3, Amsterdam, 2000, pp. 85-92.

Dick, A., Torr, P., and Cipolla, R., 2000. A model-based method for building reconstruction, Proceeding 11th British Machine Vision Conference, Bristol, UK, pp. 372-381.

Maas, H.-G., 2001. The suitability of Airborne Laser Scanner Data for Automatic 3D Object Reconstruction, Third International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, Ascona, Switzerland.

Rabbani, T., 2006. Automatic Reconstruction of Industrial Installations, Publication on Geodesy of the Netherlands Geodetic Commission, vol. 62, Phd thesis, TUDelft, Delft, the Netherlands

Schindler, K. and Bauer, J., 2003. A model-based method for building reconstruction, Proccedings ICCV Workshop on Higher Knowledge in 3D Modelling and Motion Analysis, Nice, France, pp. 74-82

Suveg, I. and Vosselman, G., 2004. Reconstruction of 3D building models from aerial images and maps. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 58(3-4), pp. 202-224.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognising Structure in Laser Scanner Point Clouds, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 46, part 8/W2, Freiburg, Germany, October 4-6, pp. 33-38.