Automatic Building Extraction From LiDAR data

Charalambos Poullis,
Suya You, Ulrich Neumann

Computer Graphics and Immersive Technologies Lab
Integrated Media Systems Center
University of Southern California
Motivation

Increasing need for 3D models of urban buildings:

- Simulation of different scenarios,
- Planning for natural or man-made events,
- Monitoring and surveillance
- Virtual Reality
- Games
- Feature films
Related work

- **Airborne LiDAR**
  - Semi-automatic methods rely on user interaction. More complete results.
  - Automatic methods often give incomplete results. Failure with very complex datasets.

- **Ground-based LiDAR**
  - Semi-automatic techniques require intense user interaction. Amount of data is large.
  - Some work in automating the process using 4D models.

- **Photogrammetry**
  Derive 3D models using a set of images. Too complicated for large-scale scenes. LOD is left to the user.

Time-consuming and still remains a difficult and complex problem.
Proposed Approach

- Uses airborne LiDAR data and satellite images.
- Runs in automatic and semi-automatic mode.
  - **Automatic mode:**
    - Unsupervised segmentation.
    - Model fitting and extraction.
  - **Semi-automatic mode:**
    - Supervised segmentation. The user selects a few vegetation points.
    - Building edge refinement using satellite images. *(optional)*
    - Model fitting and extraction
Pre-processing of the raw 3D point cloud data is performed in the following steps:
- Re-sampling
- Refinement
- Segmentation
Re-sampling

Goal: Use a regular grid to store the 3D data.

Step 1: Determine the size of the grid cell based on the:
(a) required resolution, and/or
(b) maximum error difference between samples

Step 2: Assign each sample of the 3D data to the correct grid cell.

Re-sampling introduces inconsistencies.
Re-sampling
Re-sampling
Goal: Refine the data while preserving important features such as discontinuities.

Step 1: Perform normal optimization using graph-cuts which results in smooth normals.

Step 2: Perform point optimization using the smooth normals using gradient-descent.
Refinement: Original point map
Refinement: Original normal map
Refinement: Optimized normal map
Refinement: Normal difference map
Refinement: Optimized point map
Refinement: Point difference map
Refinement: Comparison
Refinement: Summary

Initial point map
Refinement: Summary

Initial point map

Initial normal map
Refinement: Summary

- Initial point map
- Initial normal map
- Graph-cuts optimization
- Optimized normal map
Refinement: Summary

Initial point map → Initial normal map → Graph-cuts optimization → Optimized normal map → Optimized point map → Gradient descent optimization
Segmentation

Goal: Segment the 3D data into vegetation and buildings

- Supervised segmentation: Performs a region growing algorithm given a few user-selected vegetation points.
- Unsupervised segmentation: Performs skewness balancing.
Vegetation & Buildings
(Unsupervised)
Buildings (Supervised)
Parameterization of a Geometric Primitive

- Using a local coordinate system the vertices are computed:
  \[ v_0 = [-w/2,-h/2] \quad v_1 = [w/2,-h/2] \]
  \[ v_2 = [-w/2,h/2] \quad v_3 = [w/2,h/2] \]

- The internal points are parameterized as functions of the building's dimensions:
  \[ P_0 = [-aw/2,-bh/2] \quad P_1 = [aw/2,-bh/2] \]
  \[ P_2 = [aw/2,bh/2] \quad P_3 = [-aw/2,bh/2] \]

- This parameterization also allows for the symmetry constraints found in man-made structures, to be enforced.
  In order to ensure that the internal points are always inside the bounding box the condition \( 0 \leq a, b \leq 1 \) is enforced.
Parameterization of a Geometric Primitive

- The parameterization sub-divides the space in 5 areas.
- 3D plane fitting on the points inside each area.
- Error function is the sum of the euclidean distances of all the points from their plane.
- Use Non-Linear Bound-Constrained Minimization to derive the values for a, b which minimize the error function.

The process is fast because the number of unknowns is only 2 and they are also bound-constrained.
Parameterization of a Geometric Primitive

A single primitive handles buildings of different types.
Results
Results
Results
Results
Results
Results
Results
Results
Results: Misclassification
Conclusion

- We proposed a method for the detection and extraction of basic building types from airborne LiDAR data.
- Preprocessing:
  - Resampling
  - Refinement (normal optimization with a discontinuity preserving process)
  - Segmentation
- Parameterized geometric primitive for basic building structures
  - Very fast computation times.
  - Single primitive handles multiple roof types.
- Extend primitive to handle complex buildings.
Thank you!

http://www.poullis.org