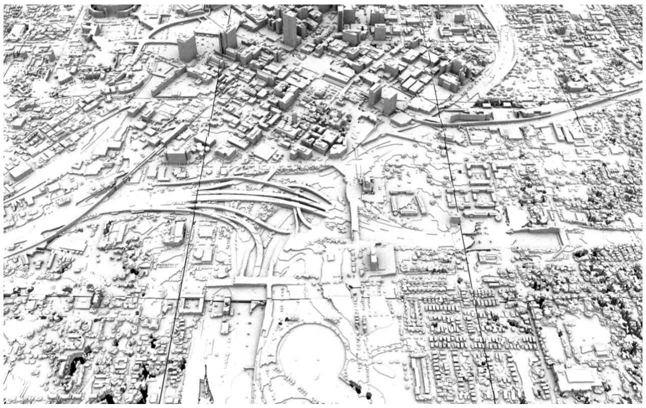
Automatic Creation of Massive Virtual Cities

Charalambos Poullis*
CGIT/IMSC
University of Southern California

Suya You†
CGIT/IMSC
University of Southern California



(a) North side.

(b) South side.

Figure 1: Automatic, complete reconstruction of the city of Atlanta.

ABSTRACT

This research effort focuses on the historically-difficult problem of creating large-scale (city size) scene models from sensor data, including rapid extraction and modeling of geometry models.

The solution to this problem is sought in the development of a novel modeling system with a fully automatic technique for the extraction of polygonal 3D models from LiDAR (Light Detection And Ranging) data. The result is an accurate 3D model representation of the real-world as shown in Figure 1. We present and evaluate experimental results of our approach for the automatic reconstruction of large U.S. cities.

Index Terms: I.3.5 [Computational Geometry and Object Modeling]: Boundary representations—Geometric algorithms, languages, and systems, Modeling packages; I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality—Visible line/surface algorithms

1 INTRODUCTION

Throughout the recent years, there has been an increasing demand and interest in the rapid generation of photorealistic virtual environments. A virtual environment provides a miniature representation of the real world which can then be employed in a wide range of applications ranging from computer graphics, virtual reality, games, feature films to Geographical Information Systems(GIS). A significant component to the success of these applications which employ a virtual representation of the real world, is visual realism. Achieving a high level of visual realism helps reduce the gap between the physical reality and virtual reality which in turn can improve the immersive experience of the users. However, current approaches and systems to produce the photorealistic 3D representations are still time-consuming, expensive and labor intensive. In fact, the creation of models is still widely viewed as a specialized art, requiring personnel with extensive training and experience to produce useful models.

This paper pursues a step change in the technique and system to create 3D models, in particular to rapidly produce realistic city models from remote sensor LiDAR data. The most significant contributions of the developed work is a complete modeling system and a fully automatic technique for the extraction of polygonal 3D models from LiDAR data. We have extensively tested the system with several city-size datasets including Baltimore downtown, Denver downtown, and city of Atlanta, etc. Our work represents a significant success to produce a consistent work flow that allows non-expert and non-artists to rapidly construct large-scale, high fidelity scene models.

The following sections will detail the developed techniques and system. Section 2 summarizes the state-of-knowledge in the areas of geometry modeling. Section 3 presents the overview structure of the system. Section 6 describes the techniques of automatic geometry modeling. Sections 7 presents the experimental results, and finally the Section 8 concludes the presented work.

2 RELATED WORK

A significant amount of research and a variety of different techniques have already been proposed and developed to address the problem of generating photorealistic virtual environments. Below we overview the state of the art in this area.

2.1 3D Model Reconstruction

In [9] the proposed system can deal with uncalibrated image sequences acquired with a hand-held camera. Based on tracked or matched features the relations between multiple views are computed. From this both the structure of the scene and the motion of the camera are retrieved. The ambiguity on the reconstruction is restricted from projective to metric through self-calibration.

*e-mail: charalambos@poullis.org
†e-mail:suyay@graphics.usc.edu

IEEE Virtual Reality 2009
14-18 March, Lafayette, Louisiana, USA
978-1-4244-3943-0/09/$25.00 ©2009 IEEE

In [8] Nevatia et al, propose a user-assisted system for the extraction of 3D polygonal models of buildings from aerial images. Low level image features are initially used to build high level descriptions of the objects. Using a hypothesize and verify paradigm they are able to extract impressive models from a small set of aerial images. The authors later extended their work in [6] to automatically estimate camera pose parameters from two or three vanishing points and three 3D to 2D correspondences.

In a different approach, [2] proposed an interactive system which can reconstruct buildings using ground imagery and a minimal set of geometric primitives. More recently [10] extended this system to incorporate pointcloud support as part of the reconstruction however the required user interaction increases considerably for large-scale areas. Moreover, the user interaction depends on the desired level of detail of the reconstructed models which may vary according to the application.

In [1] a ground-based LiDAR scanner is used to record a rather complex ancient structure of significant cultural heritage importance. Multiple scans were aligned and merged together using a semi-automatic process and a complete 3D model was created of the outdoor structure. The reconstructed model is shown to contain high-level of details however the complexity of the geometry limits this approach to the reconstruction of single buildings rather than large-scale.

Another ground-based approach is presented in [7] where a two-stage process is employed in order to quickly fuse multiple stereo depth maps. The results are impressive especially for a real-time system, however the resulting geometry is too complex and requires further processing in order to make it usable in another application.

A similar ground-based approach is presented in [14] where multiple range images are integrated by minimizing an energy functional consisting of a total variation regularization force and an L^1 data fidelity term. Similarly, the resulting geometry, although impressive, is too "heavy" for most applications, as is also the case with the proposed method in [12] which combines unregistered range and image sensing for reconstructing photorealistic 3D models.

A more recent primitive-based system [11] presented a method for the rapid reconstruction of photorealistic large-scale virtual environments using a minimal set of three primitives. In this case the authors sacrifice full automation to achieve high-fidelity and high-quality models.

In [5] the authors present a method for reconstructing large-scale 3D city models by merging ground-based and airborne-based LiDAR data. The elevation measurements are used to recover the geometry of the roofs. Facade details are then incorporated by the high resolution capture of a ground based system which has the advantage of also capturing texture information. The textures aid in the creation of a realistic appearance of the model. However, at the cost of having detailed facades they neglect to deal with the complexities and wide variations of the buildings' roof types.

3 System Overview

The processing pipeline consists of three main modules as shown in Figure 2. During the preprocessing the raw 3D poincloud data is resampled into a 2D XYZ map and missing data is recovered using local neighbourhood information. A novel automatic segmentation is then used to segment the resampled data into disjoint, contiguous regions which are used to extract the region boundaries. Finally, a simplification is performed on the extracted boundaries and surface fitting is performed on the segmented regions. The fitted surfaces are used in conjunction with the refined boundaries to extrude polygonal 3D models.

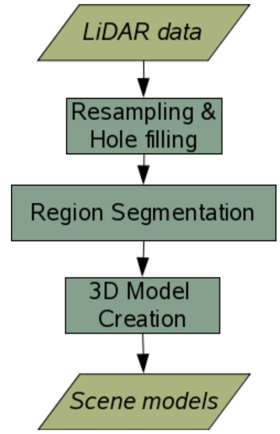


Figure 2: System Overview.

4 Preprocessing

The role of the Preprocessing module is to primarily divide the data into memory manageable parts and to convert those parts into the internal data representation used by our system. This module takes as input the raw 3D pointcloud data and converts it into a set of 2D XYZ maps, in two steps: Resampling and Hole filling. Traditionally, the size of the raw data is vast and unstructured and cannot be processed directly. Thus, the preprocessing step ensures that the data is subdivided into smaller, space/memory manageable parts, called 2D XYZ maps. The XYZ maps reduce the building extraction problem from a 3D to a 2D, therefore allowing all subsequent processing to be performed entirely in 2D and allowing the use of fast image processing techniques such as hole filling to be performed, which recovers any missing information from the local neighbourhood. This greatly reduces the computational complexity and improves the computational time of the system. Moreover, the XYZ maps are independent of each other which makes the process highly parallelizeable.

5 Region Segmentation

An automatic segmentation is performed on the resampled 2D XYZ maps to group neighbouring points of similar geometric properties into disjoint regions. Initially, a 1-D Gaussian distribution G_d and a 3-D Gaussian distribution $G_{\vec{N}}$ is created for each region R_i to describe the distributions of the depth and normals of all the points $P \in R_i$, respectively. The segmentation then begins by initializing a region R_0 with a starting point $P_{(x,y)} \in M$ in the XYZ map M. Candidate points in the 8-neighbourhood system are considered and are added iff the likelihood, of the candidate point's depth d_p and normal $\vec{n}_p$, belonging to the gaussian distributions G_d and $G_{\vec{N}}$ describing the region being processed, is above an adaptive threshold as indicated by equations 1 and 2,

$$Pr_d(d_{P_i}) \geq Pr_d(\mu_d + \kappa\sigma_d) \quad (1)$$

$$Pr_{\vec{n}}(\vec{n}_{P_i}) \geq Pr_{\vec{n}}(\vec{N}_\mu + \kappa\vec{V}_{\Sigma_{\vec{n}}}) \quad (2)$$

where μ_d and σ_d is the mean depth and variance of the distribution G_d respectively, $\vec{N}_\mu$ is the mean normal vector of the distribution $G_{\vec{n}}$, $\vec{V}_{\Sigma_{\vec{n}}}$ is the diagonal of the covariance matrix $\Sigma_{\vec{n}}$ of the distribution $G_{\vec{n}}$, and $\kappa = 1$ is a parameter which controls how similar the candidate point's properties have to be to be added to the region. Successfull candidate points are added to the region and the gaussian distributions describing that region are updated to reflect the change. This process is iteratively performed and a new region is initialized each time a region has considered all its neighbouring points and no change has occurred. The result is a set of disjoint regions shown with different colors in Figure 3(a) and Figure 3(b).

(a) Downtown Atlanta.

(b) Urban area in Denver.

Figure 3: Color-coded segmentation results.

Finally, we employ Suzuki's algorithm for the extraction of the region boundaries as described in [13]. The result is a two-dimensional closed contour B_{R_i} for each remaining region R_i. The boundary B_{R_i} is a dense set of sequentially, neighbouring points.

6 3D Model Creation

The 3D Modeling module takes the segmented regions and their boundaries and generates a set of lightweight, water-tight, polygonal 3D models. The process involves four steps. Firstly, linear surfaces are fitted to each of the regions. Secondly, the dense boundaries extracted in the previous section are simplified using a Douglas-Peucker approximation([3]) followed by an Iterative End-Point Fit([4]) for the refinement and the removal of irregularities resulting from the noise of the data and . Lastly, 3D models are created from the resulting boundaries and fitted surfaces.

7 Experimental Results

Figure 1 shows the result of the complete reconstruction of the city of Atlanta. The reconstruction was performed automatically in 13.5 hours on a single computer and contains urban and suburban areas of about $28km^2$. Densely built urban areas such as downtown districts generally resulted in very accurate and visually pleasing 3D models, which is mainly due to the fact that they are low-vegetation areas with buildings of high complexity and of relatively large areas. On the other hand, suburban and residential areas have performed the poorest primarily due to the high vegetation density, and in addition due to the minimal differences between the sizes and complexities of the buildings and the vegetation.

Figure 4 shows a top-view of the generated 3D models for downtown Denver. The area is $14km^2$ and the reconstruction was automatically performed in 5.2 hours.

In Figure 5 a close-up of the reconstruction of the entire downtown Baltimore is shown. The size of the area is $16km^2$ and the automatic reconstruction required 11 hours.

Figure 6 shows the complete reconstruction of the downtown Oakland. The size of the area is $17km^2$ and required 7 hours for the automatic reconstruction. Figure 7 shows a close-up of the same reconstruction.

8 Conclusion

We have presented a complete and novel approach for the automatic creation of massive virtual cities for virtual reality applications such as emergency personnel training and natural and man-made disaster simulations. Unlike existing techniques, we proposed a fast and automatic approach for reconstructing virtual worlds entirely

Figure 4: Complete reconstruction of downtown Denver.

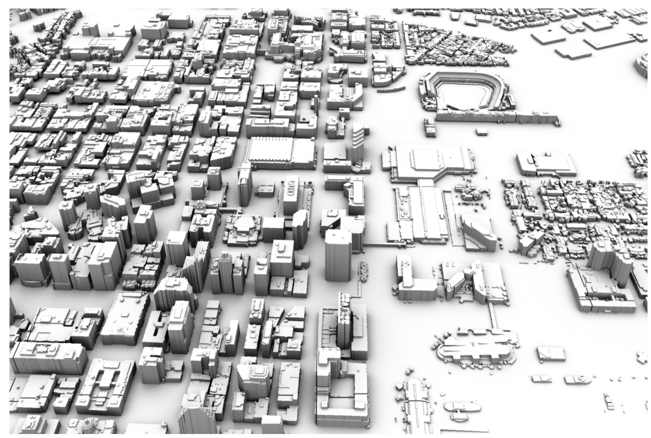


Figure 5: Close-up of the complete reconstruction of downtown Baltimore.

Figure 6: Complete reconstruction of downtown Oakland.

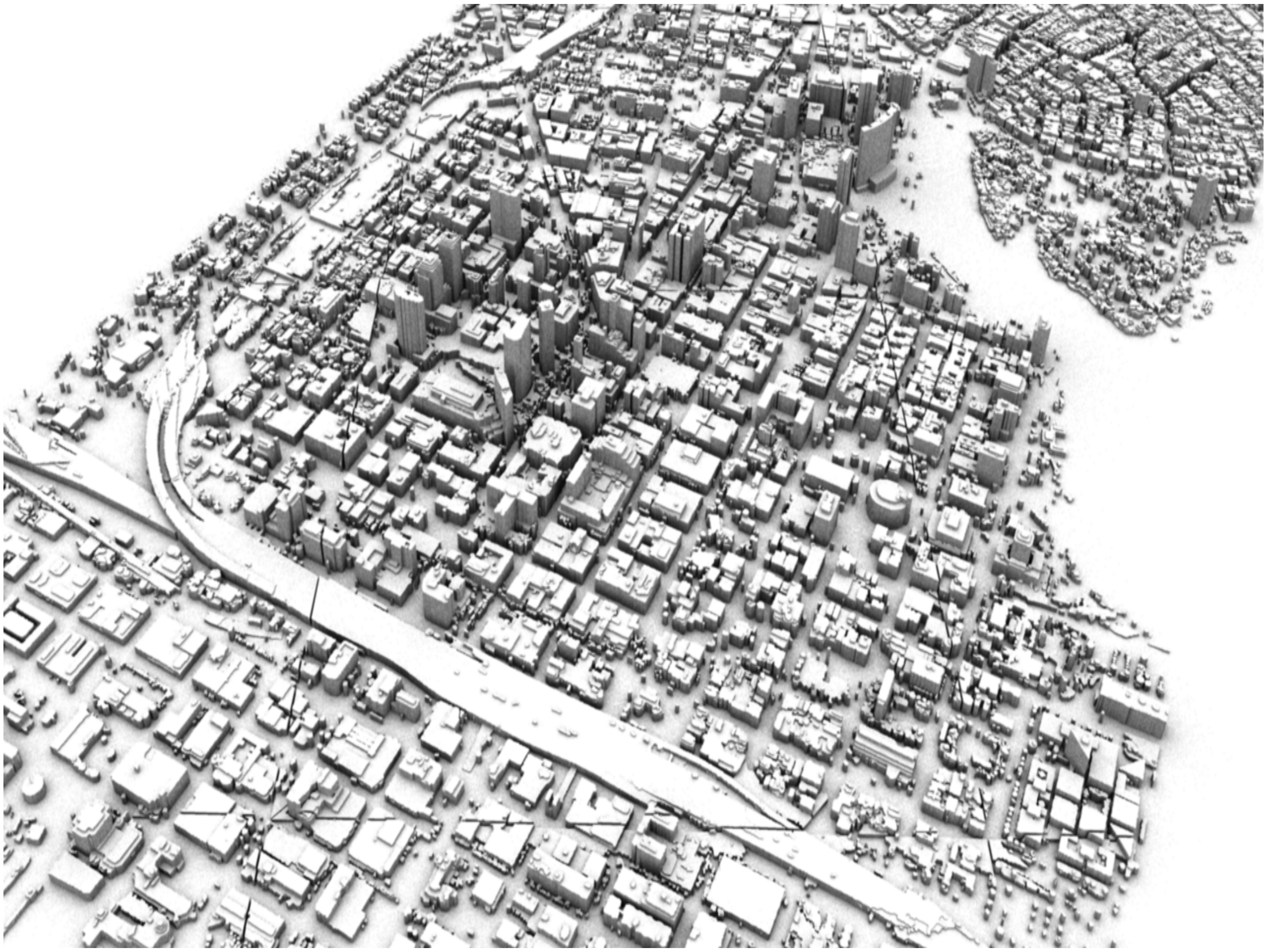


Figure 7: Close-up of downtown Oakland.

from remote sensory data captured by an airborne LiDAR scanner which does not rely on data dependencies and thresholds. To achieve this, we have introduced a robust and effective segmentation technique which segments the data based on a probabilistic analysis of their geometric properties. Experimental results were demonstrated, which verify the validity of our approach and its successfull application on a variety of different data sets. Finally, we have presented the effective qualitative and quantitative measures we have employed for the evaluation of the generated 3D models.

References

[1] P. Debevec, C. Tchou, A. Gardner, T. Hawkins, C. Poullis, J. Stumpfel, A. Jones, N. Yun, P. Einarsson, T. Lundgren, M. Fajardo, and P. Martinez. Estimating surface reflectance properties of a complex scene under captured natural illumination. Technical report, USC, ICT, 2004.

[2] P. E. Debevec. *Modeling and rendering architecture from photographs*. PhD thesis, University of California, Berkeley, 1996.

[3] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, 10:112–122, 1973.

[4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[5] C. Früh and A. Zakhor. Constructing 3D city models by merging ground-based and airborne views. In *CVPR*, pages 562–569. IEEE Computer Society, 2003.

[6] S. C. Lee, S. K. Jung, and R. Nevatia. Automatic pose estimation of complex 3D building models. In *WACV*, pages 148–152. IEEE Computer Society, 2002.

[7] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. M. Frahm, R. G. Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *International Conference on Computer Vision*, pages 1–8, 2007.

[8] R. Nevatia and K. E. Price. Automatic and interactive modeling of buildings in urban environments from aerial images. In *ICIP (3)*, pages 525–528, 2002.

[9] M. Pollefeys, L. J. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[10] C. Poullis, A. Gardner, and P. Debevec. Photogrammetric modeling and image-based rendering for rapid virtual environment creation. *Proceedings of ASC2004*, 2004.

[11] C. Poullis, S. You, and U. Neumann. Rapid creation of large-scale photorealistic virtual environments. In *VR*, pages 153–160. IEEE, 2008.

[12] I. Stamos and P. K. Allen. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, 88(2):94–118, 2002.

[13] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics and Image Processing*, 30:32–46, 1985.

[14] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *International Conference on Computer Vision*, pages 1–8, 2007.