Robust Context Free Segmentation of Unordered 3D Point Cloud Models

Andrey DIMITROV[1] and Mani GOLPARVAR-FARD[2]

[1] Ph.D. Student, Department of Civil Engineering and Engineering Mechanics, Columbia University, 10025; PH (713) 854-7744; email: ad2895@columbia.edu

[2] Assistant Professor, Department of Civil and Environmental Engineering, and Department of Computer Science, University of Illinois at Urbana-Champaign, 61801; PH (217) 300-5226; email: mgolpar@illinois.edu

ABSTRACT

Accurate and rapidly produced 3D models of the as-built environment can be significant assets for a variety of Civil Engineering scenarios. Starting with a point cloud of a scene – generated using laser scanners or image-based reconstruction method– the user must first identify collections of points that belong to individual surfaces, and then, fit surfaces and solid geometry objects appropriate for the analysis. When performed manually, this task is often prohibitively time consuming and, in response, several research groups have recently focused on developing methods for automating the modeling process. Due to the limitations of the data collection processes as well as the complexity of as-built scenes, automated 3D modeling still presents many challenges. To overcome existing limitations, in this paper, we propose a new region growing method for robust context-free segmentation of unordered point clouds based on geometrical continuities. In our method, only one parameter is required to be set by the user to account for the desired level of abstraction. Preliminary experimental results from two challenging scenes of the built environment demonstrate that our method can account for variability in point cloud density, surface roughness, curvature, and clutter within a single scene.

INTRODUCTION

3D modeling of the as-built environment is used by the AEC industry in a variety of engineering analysis scenarios. Significant applications include progress monitoring of construction sites, quality control of fabrication and on-site assembly, energy performance assessment, and structural integrity evaluation. In recent years, point clouds have become the predominant data type collected on site and used as a basis for modeling. This data can originate from image-based 3D reconstruction methods using images or videos, as well as structured light methods, mainly laser scanners. The process of generating 3D models from point cloud data involves two steps: 1) identifying collections of points that belong to each surface, and 2) fitting geometry (meshes, primitives, NURBS, subdivision) to them. This process is manual and very time-consuming. In response, several research groups (e.g., Xiong et al. 2013, Zhang et al. 2013) have recently focused on developing methods for automated modeling.

Although, the type of geometry to be fitted is heavily dependent on a specific analysis, the common task in Civil Engineering scenarios is the segmentation of point clouds into identifiable surfaces. Figure 1 shows examples of the use of automated segmentation to identify specific elements that need to be modeled. The complexity of as-built scenes produces significant challenges for automated segmentation:

1) *Density*: Point cloud models exhibit locally variable densities based on orientation and distance from the capture device. Furthermore, occlusions from surface irregularities and adjacent objects produce regions with missing data.

2) *Surface Roughness*: the physical texture of common surfaces can range from smooth (steel, marble) to very irregular (grass, crushed stone). Because a given scene can contain a wide range of surface *roughnesses*, no priors about noise levels can be reliably used.

3) *Curvature*: surfaces can be flat, single curved, double curved, or have undulations at multiple scales, making boundaries hard to define.

4) *Clutter*: a scene can be made up of multiple objects in close proximity, making feature detection difficult.

5) *Abstraction*: 3D modeling is inherently a process of abstraction. This required that some decisions have to be made by the user. In consequence, automation needs to balance flexibility with the ease of use.

In this paper, we present a segmentation method that addresses these challenges. We test the performance of the proposed method on two challenging point cloud models that are generated using multiple laser scans of the built environments.

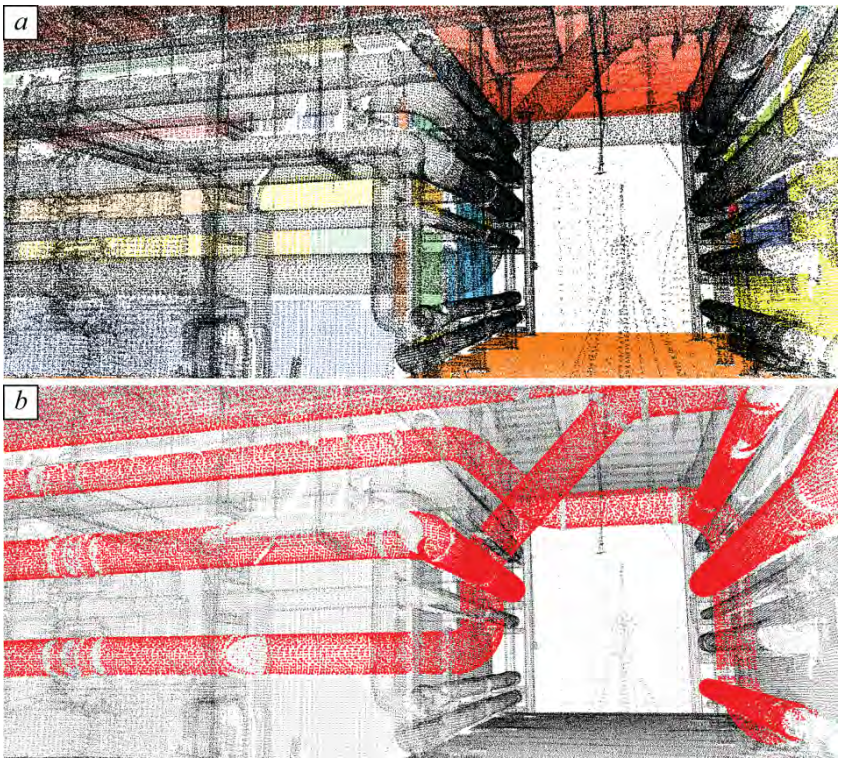

Figure 1. Example result of our automated segmentation method in a challenging point cloud including various structural and MEP/FP elements: a) flat segments for the purposes of architectural modeling; b) pipe segments of similar diameter for the purposes of MEP modeling.

RELATED WORK

Recent automated segmentation strategies for the purposes of geometric modeling (Tang et al. 2010) fall into two families:

1) Surface based methods (e.g., Zhang et al. 2013, Reisner-Kollmann et al. 2012, Li et al. 2011, Borrmann et al. 2011, Deschaud et al. 2010) attempt to find sets of points in the scene that fit planes and primitives. These methods are generally able to identify surfaces in cluttered

scenes. However, they leverage expectations of the types of surfaces to be modeled and are thus context specific. They also work best when prior knowledge of the surface roughness is available to define the inlier/outlier thresholds. In Zhang et al. (2013), these thresholds take the form of global parameters for a Maximum Likelihood Estimator Sample Concensus (a probabilistic approach to the RANSAC paradigm).

2) Region growing methods (Son et al. 2013, Rabbani et al. 2006) are graph based and to not constrain the user to modeling with primitives. They rely on some formulation of connectivity between points and a measure of similarity. Similarity is generally derived from local features extracted from a neighborhood around each point. Surface normal (Klasing et al. 2009, Dey et al. 2005), and curvature are the most commonly used features. Curvature estimation can depend on normal estimation and orientation (Kalogerakis et al. 2009) to produce detailed features (principal curvature vectors) or use points alone to provide an indirect measure (Mérigot et al. 2011, Pauly et al. 2003). Alternatively, local NURBS fitting can be used (Son et al. 2013), with the assumption that curvatures are relatively low, and a surface does not fold on to itself (e.g., small diameter pipe). Similarity measures based on local features used in this family of research works assume a locally manifold (non-self-intersecting) surface. However, when multiple objects are in close proximity, the point clouds describe effectively non-manifold surfaces and thus these features become unreliable. Figure 1 and Figure 4a show cluttered scenes can render this approach ineffective.

METHOD

We propose a context-free method for segmentation of unordered point clouds in $\mathbb{R}^3$. No prior assumptions are made about the intent of the segmentation process as it applies to a larger context of engineering analysis. In particular, there are no priors about 1) possible types of shapes in the scene or the surface models used to describe them (e.g., primitives, NURBS, subdivision surfaces, meshes); 2) materials, surface roughness, object proximity and clutter, or noise levels; and 3) the completeness of the scene, the point cloud density, or other constraints stemming from the sensing process. Our method requires only one parameter to be set by the user, the radius of interest r. This parameter provides a "soft" upper bound on the amount of detail that is to be considered, and allows segmentation to account for local context. For example, a single wood plank on a concrete floor should be segmented as one element, while collections of planks (e.g., formwork, decking, fences) should be considered as singular textured materials and would be grouped together. Thus, we consider r as a locally adaptive threshold, allowing the capture of the representative scales of various surfaces in the scene. Furthermore, the process is able to dynamically reevaluate local features during segmentation, thus mitigating unreliability in regions where multiple objects are near one another. The method starts with a pre-processing stage that subsamples the point cloud. A multi-scale feature detection stage follows, describing surface roughness and curvature around each 3D point. Finally a seed finding and region growing stage segments the points. Figure 2 shows an overview of the entire segmentation process.

Pre-processing

A point cloud P is subsampled to one point per voxel of size $0.1r$, generating point cloud S, such that $S \subseteq P$. An upsample data structure is kept to map each point $p_i \in S$ to the set of removed points from its representative voxel.

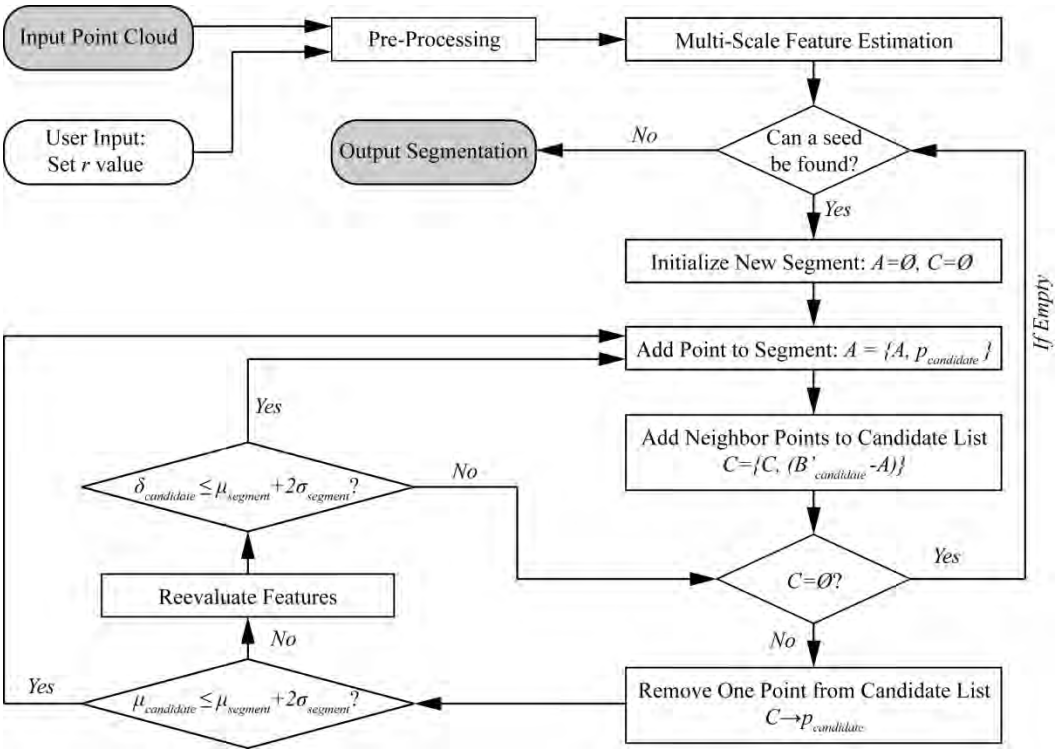

Figure 2. The automated segmentation process.

Multi-Scale Feature Detection

A point $p_i \in S$ has a full neighborhood $B_i = \{b_0, \ldots, b_n\}$ where all $b_u \in P$ and are within distance r of p_i. A partial neighborhood $B_i(d_j, q) \subseteq B_i$ is defined as the set of at most q points sampled uniformly from all points in B_i with a distance to p_i smaller than $d_j \in \{r, 0.72r, 0.52r, 0.37r, 0.27r, 0.19r\}$.

For each point p_i, scale-dependent features $F_{ij} = \{R_{ij}, k^1_{ij}, k^2_{ij}, \epsilon_{ij}\}$ are computed based on a neighborhood of points $B_i(d_j, q)$, where $R_{ij} = \langle \overrightarrow{n_{ij}}, \overrightarrow{K^1_{ij}}, \overrightarrow{K^2_{ij}} \rangle$ is the local orthonormal basis represented by the unit surface normal $\overrightarrow{n_{ij}}$, and unit directions of *max* and *min* curvatures $\overrightarrow{K^1_{ij}}$, $\overrightarrow{K^2_{ij}}$; k^1_{ij} and k^2_{ij} are the absolute valued *max* and *min* curvatures, and $\epsilon_{ij} = \{\mu_{ij}, \sigma_{ij}\}$ is the surface roughness.

First, a rough estimate of the surface normal is computed based on a Principal Component Analysis (PCA) on all points in $B_i(r, 50)$. Next, to compute surface roughness and curvature, a set of planes $L_i = \{l^0_i, \ldots, l^m_i\}$ passing through p_i and rotating uniformly around the surface normal are generated.

For each neighborhood $B_i(d_j, q)$ and each plane l^α_i, all points with offset smaller that $0.1r$ from plane form the plane neighborhood Φ^α_{ij}. These points are represented in the $\mathbb{R}^2$ subspace formed by the plane. We fit a circle to these points using the least squares method, and the curvature k_α, center o_α, and fit error $e_\alpha = \{\mu_\alpha, \sigma_\alpha\}$ (average and standard deviation distance of the points to the circle) are recorded. For each scale, k^1_{ij} is derived from the plane with

maximum curvature, and k_{ij}^2 derived from the plane perpendicular to the one with maximum curvature. The centers recorded from these two planes are projected into the world $\mathbb{R}^3$ basis and the final surface normal becomes $\overrightarrow{n_{ij}} = unit(p_i - (o_{max} + o_{min})/2)$. The local basis is completed by finding $\overrightarrow{K_{ij}^1}$ and $\overrightarrow{K_{ij}^2}$, two unit vectors perpendicular to the surface normal and lying on the two principal planes respectively. Finally, ϵ_{ij} is computed as the mean fit error from the circles in the two principal planes. If no features can be computed due to insufficient neighbors, $F_{ij} = \{\emptyset, 0, 0, \emptyset\}$. The representative features for each point p_i is derived by finding $F_i \leftarrow F_{ij}$ such that $\mathbf{argmin}_j(\mu_{ij})$. Figure 3 illustrates the estimated surface roughness at each scale and the derived representative roughness and radii for the floor in scene in Figure 1. The circular red zones that appear at smaller scales are zones where features cannot be found due to low point density. In Figure 3b, red represents zones where the feature estimation is unreliable.

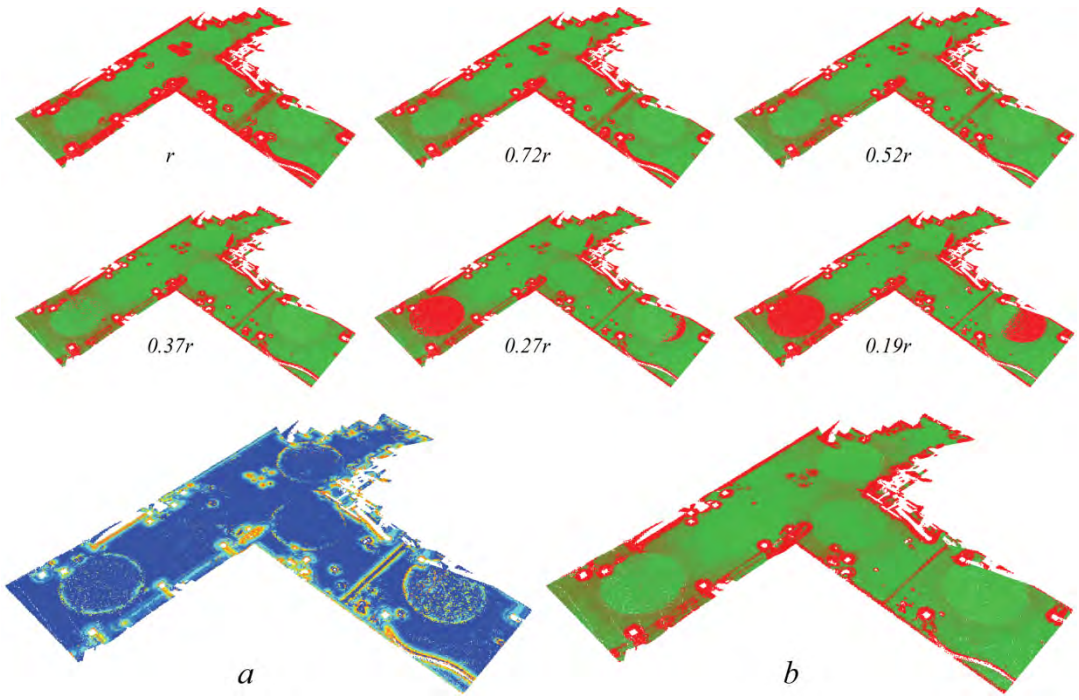

Figure 3. Showing estimated surface roughness for a floor surface at different scales. a) The representative radii per point (blue is largest, red is smallest); b) The estimated surface roughness from the representative features.

The representative neighborhood B'_i for each point is formed by fitting a local double curved surface (torus) derived from the principal curvatures and directions. From all points in B_i, only those whose distance from this surface satisfies $\delta \leq \mu_i + 2\sigma_i$ are kept.

Finding Seeds

Each segment starts by finding one seed point that is then grown by incrementally adding new points to it. When growth is complete, a new seed point is needed for the growth of the next segment. Seed finding proceeds in successive steps, starting from the smoothest and most

uniform patches in the scene. Thresholds for surface roughness, $\tau = 0.01r$, and radius, $\gamma = 0$, are initialized. A valid seed is found by looking for an non-segmented point with minimum surface roughness μ_i that satisfies $\mu_i \leq \tau$ and $r_i = d_\gamma$. If no such point exists, the radius is incremented, $\gamma \leftarrow \gamma + 1$. When all possible radii have been exhausted, the roughness threshold is incremented, $\tau \leftarrow 2 \times \tau$ and $\gamma \leftarrow 0$. If the roughness threshold τ reaches r and no valid seed can be found, the segmentation terminates.

Segment Growth

At the start of each segment growth, empty sets of candidates $C = \emptyset$ and current segment points $R_c = \emptyset$ are initialized. The surface roughness of the segment is initialized as $\mu_{segment} = \mu_{seed}$ and $\sigma_{segment} = \sigma_{seed}$. Once a seed is chosen, it is added to A and all points from its representative neighborhood B^r_{seed} are added to C. Then, a point from C is removed and examined. If its local roughness satisfies $\mu_{candidate} \leq \mu_{segment} + 2\sigma_{segment}$, it is added to R_c, the segment roughness is updated, and all points in its representative neighborhood $B^r_{candidate}$ are added to C. If not, this point's features are considered unreliable, and are recomputed using only points from $B_{candidate}(r, q) \cap R_c$. This avoids non-manifold local neighborhoods from affecting segmentation as all points in R_c are guaranteed to belong to a manifold surface. We reject those points that do not meet this criteria. Finally, segmentation ends when there are no more points left in C. Figure 4 shows how reevaluated points correspond to regions where multiple objects are in close proximity.

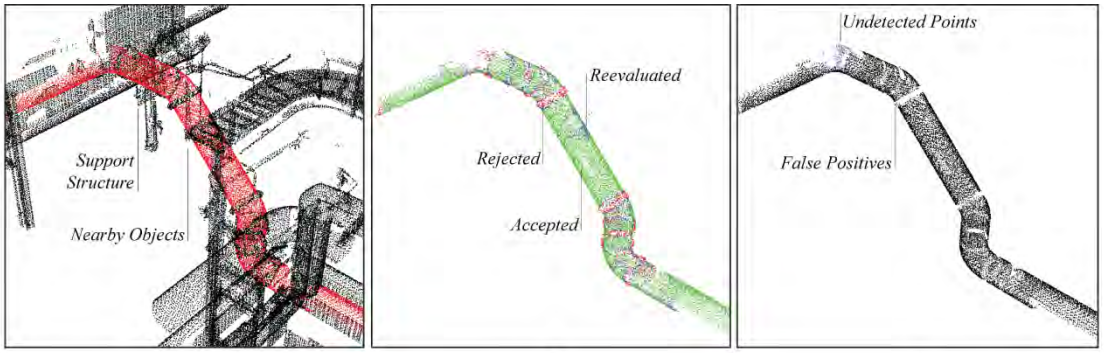

Figure 4. Example segmentation of a composite pipe in a cluttered scene. a) Shows the red ground truth segmentation, highlighting regions where other objects are touching the pipe. b) During region growing, green points are accepted, blue points are reevaluated, and red points are rejected. c) Comparison with ground truth: red are false positives, blue are false negatives.

EXPERIMENTAL RESULTS

Ground truth

The ground truth in our experiments was manually generated from point cloud scenes. Segments were defined based on a judgment of surface continuity and not object completeness; i.e., segments ignore small local features that are deemed part of an object but deviate from the

continuity of the overall surface. Figure 4a illustrates the ground truth of a composite pipe. The connection details between pipe sections– although part of the overall composite pipe object– are deemed to belong to surfaces that are discontinuous from the main pipe surface and are segmented separately. Object definition –what points belong to the same object– can be very subjective. By focusing on surfaces, without bias towards objects, we aim to reduce the potential variability of ground truth judgment by the user.

The ground truth segmentation of P is a set $G = \{G_0, \ldots, G_n\}$, where each $G_i \subseteq P$ is a ground truth segment. Each p_i can belong to at most one segment. If it does not belong to any segment, it was dismissed as noise or there was inadequate sensing and, as a result, the user could not perceive a surface from that region.

Validation Metrics

The performance measure of a segmentation algorithm is derived from its ability to segment sets of points that properly describe the underlying shapes in the scene. In practice, describing the underlying shapes includes fitting surfaces to segmented sets of points. In this paper, no assumptions about fitting strategies are made, thus we define the proper description of the underlying shapes as the similarity between of the segmented point sets and the ground truth point sets.

Given a test segmentation $T = \{T_0, \ldots, T_n\}$ and a ground truth segment G_i, we define a testing segment T' as the union of all T_i with at least k points in G_i. In our experiments, we set $k = 5\%$ of the number of points in G_i. We define: true positives (TP) as the number of points in G_i and T'; true negatives (TN) as the number of points not in G_i and not in T'; false positives (FP) as the number of points not in G_i and T'; false negatives (FN) ad the number of points in G_i and not T'. The performance of the segmentation per ground truth segment G_i is measured by:

1- The extent at which T' provides a false description of the underlying shape, using the *false positive rate* (α), or type I error:

$$\alpha = FP/(FP + TN)$$

2- The extent at which T' provides an incomplete description of the underlying shape, using the *false negative rate* (β), or type II error:

$$\beta = FN/(TP + FN)$$

3- The extent of over-segmentation of G_i, which makes the complete underlying shape harder to assemble, using η as the number of T_i used to form T'.

Experiments

Experiments were run on two challenging point clouds representing two types of built environments. The first, the corridor data set, is a 20 million point interior scene with a lot of very densely packed MEP equipment. The second, the street data set, is a 2 million point outdoor scene with a variety of surface roughnesses (from smooth walls to rough grass and bushes), as well as many invalid points produced from people and vehicles in movement during scanning. The corridor scene was processed with $r = 0.3$, yielding a subsampled cloud of 950,000 points. Figure 5 shows the relationship between each metric and the mean maximum curvature of the segment. It indicates that all three metrics are not affected by the curvature of the segments, thus the algorithm performs consistently over these various surfaces.

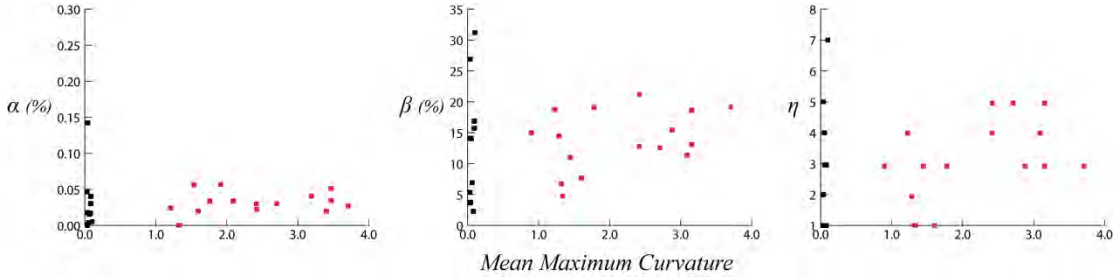

Figure 5. Results from the corridor data set. Each data point represents a segment in the scene (red for pipe segments, black for flat segments).

Overall our method results in a mean α of 0.03%, with standard deviation 0.02%. This means that detected segments have very little points that should not belong to them. Because false positives contribute to erroneous surface fitting, a very low rate is crucial in proper automated modeling. The mean β for all segments is 12% with standard deviation of 8%, meaning that most detected segments are more than 80% complete. The scene also contains over-segmentation of an average of 3.5 detected segments per ground truth segment. Figure 6 shows an example from our data set where incomplete detection and over-segmentation in this scene is caused by insufficient sensing due to occlusions.

Figure 6. a) Ground truth segmentation of two composite pipes. The red circles show areas with missing data due to occlusions. b) The over segmentation resulting from missing data.

The use of a robust curvature feature can facilitate the classification of segments. Figure 5 shows a clear distinction between curved and flat surfaces. This information is used in Figure 7 to separate walls, floors and ceilings, from pipes of various diameters. The street scene was processed with $r = 0.75$, yielding a subsampled cloud of 930,000 points. Figure 8 shows how the roughness feature can be used to distinguish between smooth surfaces (street, sidewalk, building walls), and rough surfaces (vegetation, roof shingles, detailed molding). The data sets, videos, and additional results from these experiments can all be found at http://raamac.cee.illinois.edu/segmentation.

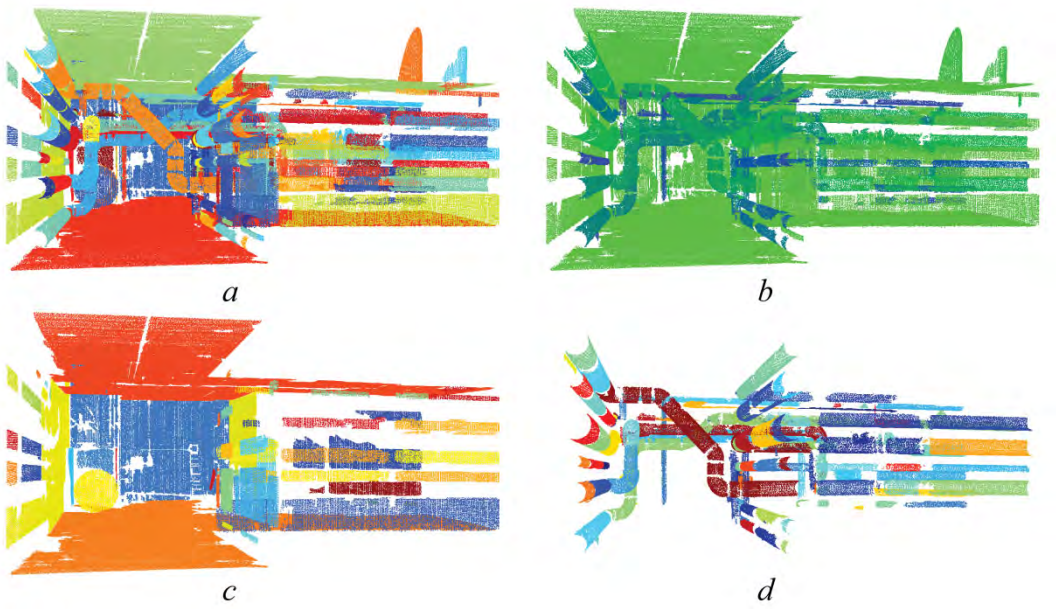

Figure 7. a) Segmentation of the corridor data set. b) Color map of mean maximum curvature of each segment. c) Isolation of low curvature surfaces. d) Isolation of high curvature surfaces (pipes).

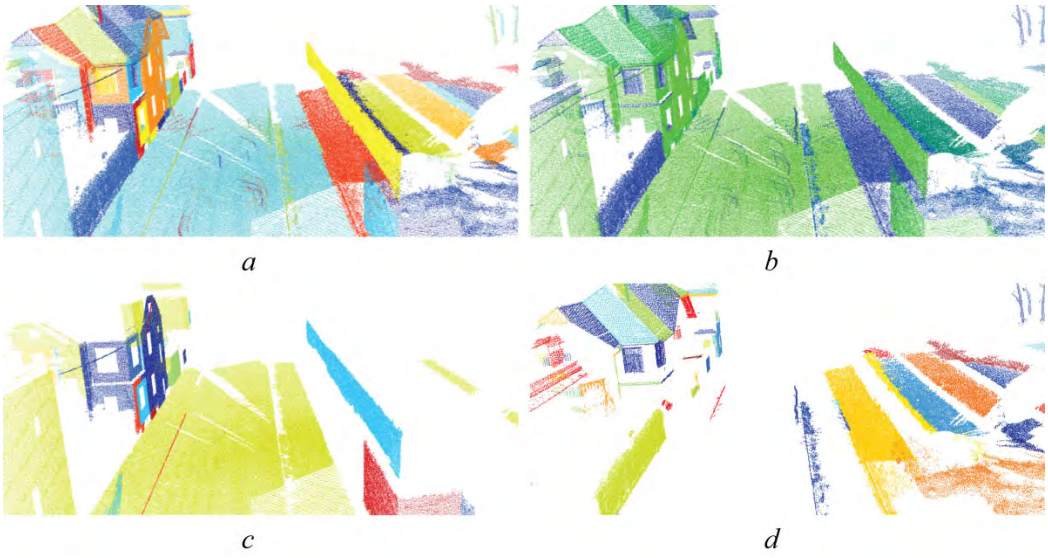

Figure 8. a) Segmentation of the street data set. b) Color map of mean surface roughness of each segment. c) and d) Isolation of smooth and rough surfaces.

CONCLUSION

A novel method for robust context-free segmentation of unordered point clouds was presented. The process can account for variability in point cloud density, surface roughness, curvature, and clutter within a single scene. Only one parameter is required to be set by the user to account for the desired level of abstraction. Preliminary experiments result in a mean α of

0.03% and a mean β of 12% in challenging scenes of the built environment and demonstrate the potential of this method to be implemented in a variety of civil engineering analysis scenarios. Future work will focus on further validation and refinement of performance metrics as well as a reduction of computation time through implementations leveraging GPU.

AKNOWLEDEMENT

We would like to thank our partner construction companies for their technical support with point cloud data. We also like to thank the members of the RAAMAC lab for their help and support with forming a large database of ground truth models.

REFERENCES

Borrmann, D., Elseberg, J., Lingemann, K., and Nüchter, A. (2011). "The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design." *3D Research*, *2*(2), 1-13.

Deschaud, J. E. and Goulette, F. (2010). "A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing." In *Symp. on 3D Data Processing, Visualization and Transmission*, Paris, France.

Dey T. K., Li G., and Sun J. (2005). "Normal estimation for point clouds: A comparison study for a Voronoi based method." In *Symp on Point-Based Graphics*, 39-46.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). "Comparison of Surface Normal Estimation Methods for Range Sensing Applications." *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (ICRA)*, .3206-3211.

Kalogerakis, E., et al. (2009). "Extracting lines of curvature from noisy point clouds." *Computer-Aided Design*, 41(4), 282-292.

Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J. (2011). "GlobFit: consistently fitting primitives by discovering global relations." In *ACM Transactions on Graphics (TOG)*, 30(4).

Mérigot, Q., Ovsjanikov, M., and Guibas, L. (2011). "Robust voronoi-based curvature and feature estimation." *IEEE Transactions on Visualization and Computer Graphics*, 17(6), 743-756.

Rabbani, T., van Den Heuvel, F., and Vosselmann, G. (2006). "Segmentation of point clouds using smoothness constraint." *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(5), 248-253.

Reisner-Kollmann, I. and Maierhofer, S. (2012). "Segmenting multiple range images with primitive shapes." Proc., *Conf. on Systems, Signals & Image Proc*, 320-323.

Son, H., and Kim, C. (2013) "Fully Automated As-Built 3D Pipeline Segmentation Based on Curvature Computation from Laser-Scanned Data." *Computing in Civil Engineering*, 764-772.

Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. (2010). "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques." *Automation in Constr.*, *19*(7), 829-843.

Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). "Automatic creation of semantically rich 3D building models from laser scanner data." *Automation in Constr.*, *31*, 325-337.

Zhang, G., Vela, P. A., and Brilakis, I. (2013). "Detecting, Fitting, and Classifying Surface Primitives for Infrastructure Point Cloud Data." *Computing in Civil Engineering*, 589-596.