Online Facade Reconstruction from Dominant Frequencies in Structured Point Clouds [*]

Sam Friedman
Hunter College and CUNY Graduate Center
695 Park Avenue
umpteee@yahoo.com

Ioannis Stamos
Hunter College and CUNY Graduate Center
695 Park Avenue
istamos@hunter.cuny.edu

Point Cloud Processing Workshop, CVPR 2012

Abstract

We present an online method for filling holes in point clouds by exploiting the regularity of urban areas. Sweeping a plane across the scene we compute periodicity, major planes, and occlusions. Extending rays from the laser that have been occluded gives a planar approximation for holes in facades. The periodicity of the architecture is used to vastly improve this approximation yielding facades that seem complete and natural. Both abstract and high resolution mesh data is constructed from the improved point clouds. All this processing is performed online allowing for seamless integration with scanner hardware.

1. Introduction

The urban landscape is replete with repetition. Windows, balconies, bricks, and streets all repeat with remarkably regular periods in cities around the globe. Discovering this regularity in 3D urban images enables an expanse of applications for scene understanding. Discovering it in real-time offers even more possibilities. With precise knowledge of scene regularity we may compress data or conversely generate synthetic data to fill holes. Overlapping regularities in different views can be used for registration. Meshes can be created from point clouds on-the-fly. Scans can be classified into a number of helpful categories, distinguishing ground, facade, moving, vegetation, and architectural feature points.

Occlusions are a major problem in processing Lidar data. Even small objects located near the scanner can cast tremendous shadows on the scene. These shadows interfere with many algorithms by leaving large discontinuities in the input data. A prototypical example is illustrated in Figure 1.

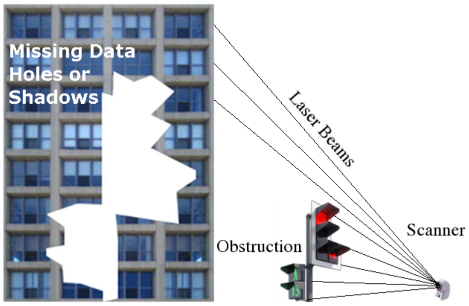

Figure 1. Objects near the scanner can cast tremendous shadows. If some of the occluded facade is periodic it is possible to infer the missing data using methods described in this paper.

Occlusions are not the only cause of holes in laser scans. Highly specular surfaces like car exteriors, and glass can deflect the laser preventing the scanner from measuring the time of flight. Additionally if the laser is pointed at the sky or objects beyond its range no data can be gathered. Our algorithm recognizes all of these types of holes and fills them by leveraging the planarity and periodicity of the facades in the scene. The synthetic data generated for the reconstruction conforms to architectural patterns of windows and balconies and the 2D structure of the point cloud acquired by the scanner.

Point clouds of urban scenes can be dense, containing millions of points. Processing these large data sets is computationally expensive and generally beyond the capacities of embedded processors included in laser scanning hardware. For this reason data acquisition and processing are commonly performed separately in sequence. We present a novel algorithm that allows nearly simultaneous acquisition and processing.

Our approach uncovers regularity in real-time by performing Fourier analysis on each scanline. This information is combined with the planarity in the scene allowing us to fill holes, generate meshes, plan views and classify points all in lock step with scene acquisition.

The laser scanner performs a plane sweep across the scene recording data points along all the nearest intersec-

[*] This work has been supported in part by the following NSF grants: IIS-0915971, CCF-0916452 and MRI CNS-0821384.

tions of the plane with scene elements. Similarly our algorithm sweeps a plane across the scene processing each set of returned points for planarity and periodicity. By designing an algorithm that follows the same paradigm as our sensor we enable seamless integration with scanner hardware.

2. Related Work

In recent years, reconstructing images with missing data has received much attention. For the input of a 2D image [1] presents a method which treats the input image as a training set from which the missing data is inferred. Leveraging databases of millions of images [4] completes scenes by finding semantically similar pictures. Fusing both 2D and 3D images [6] computes a layer decomposition from a registration between the 2D and 3D data sets which allows for outlier removal and geometry propagation. [14] introduces a context sensitive surface completion. Their algorithm iteratively aligns surface patches along the edge of holes until the surface is complete. A comprehensive survey of urban reconstruction techniques is presented in [11].

Regularity detection in 3D data is a field of active research. By learning global properties of orthogonality, orientation, and shape [5] reconstructs engineered shapes from noisy point clouds. In [8], the authors present a method for extracting approximate symmetries by voting in transformation space. [9] presents a technique for symmetrization based on a similar method as [8]. By detecting lattices in the transformation space [13] extends the transformation voting method to find smaller repeating symmetries akin to the architectural features detected in this paper. [15] presents a method that similarly exploits the regularity of urban environments to reconstruct facades by using non-local filtering. Another approach which uses urban regularity for reconstruction is found in [12]. The authors of that work use user interaction to highlight repeating sections and automatically reconstruct high floors where the laser data is sparse. In contrast our approach requires no user interaction and operates online. Shape grammars are another approach to model facade regularity exploited in [10]. The use of harmonics for symmetry detection can be traced to [7]. Using spherical harmonic decompositions they develop a method that recognizes precise symmetries in 3D mesh data.

Online algorithms for processing Lidar data are gaining attention as 3D cameras increase in popularity. In [3] the authors present an online method that uses Markov chains and detection of abrupt changes to classify vegetation, horizontal and vertical points. The authors of [2] introduce the method of online detection of repeated features by using Fourier analysis of column functions. We build upon those results and extend the applications to hole-filling and mesh creation.

Our *contributions* with respect to earlier work can be summarized as follows:

(a) Exploiting the periodicity of urban architecture we fill large holes in facades caused by occlusions, missing data, and noise from building interiors.

(b) Holes are filled in a manner that maintains the 2D structure of the point cloud facilitating downstream processing.

(c) We generate high-resolution triangular meshes as well as more abstract polygonal meshes of the scene.

(d) Processing occurs online, in a plane sweep, allowing for seamless integration with scanner hardware.

3. Regularity Detection

Data retrieved from the laser scanner is organized into a 2D grid of 3D points. Each column is a vertical sampling of the scene. Our algorithm performs a plane sweep across the 3D scene processing each column for its regularity while aggregating and updating global features at each step. The major planes in each scanline are identified thus separating the ground points from the facade. We search for periodicity in the facade to identify architectural features like windows and balconies. Points that cast shadows such as signs, cars and vegetation are classified and the obscured data is synthesized using the regularity of the architecture.

3.1. Major Planes

Scanlines in an urban area typically pass over two planar surfaces: a ground plane and a facade plane. We detect these macro features online by fitting small planes to the neighborhood around each point as the scene is recorded. The local plane is determined by using standard Principal Components Analysis (PCA). Each of these local planes will be a candidate major plane. We distinguish candidate ground planes from candidate facade planes by taking the dot product with the vertical axis. We store the eigenvalues from the eigen decomposition of the covariance matrix, because their magnitude measures the goodness of fit of the local plane to the local data (in the least squares sense). However, the local plane with the smallest eigenvalue may not correspond to the major plane. Frequently, scans contain planar areas like ceilings from interiors or box trucks which are extremely planar but are neither ground nor facade. To protect against these mis-detections we find the largest cluster of agreement in the local planes and then select the plane with the smallest eigenvalue within that cluster. The major plane estimates are greedily updated as the scan proceeds unless a shift or a corner is detected. Facade corners can be detected by measuring an orthogonal relationship between the current scanline and the major plane's normal. Shifts are detected when the vector between the current scanline and a point on the major plane is not perpendicular to the major plane's normal.

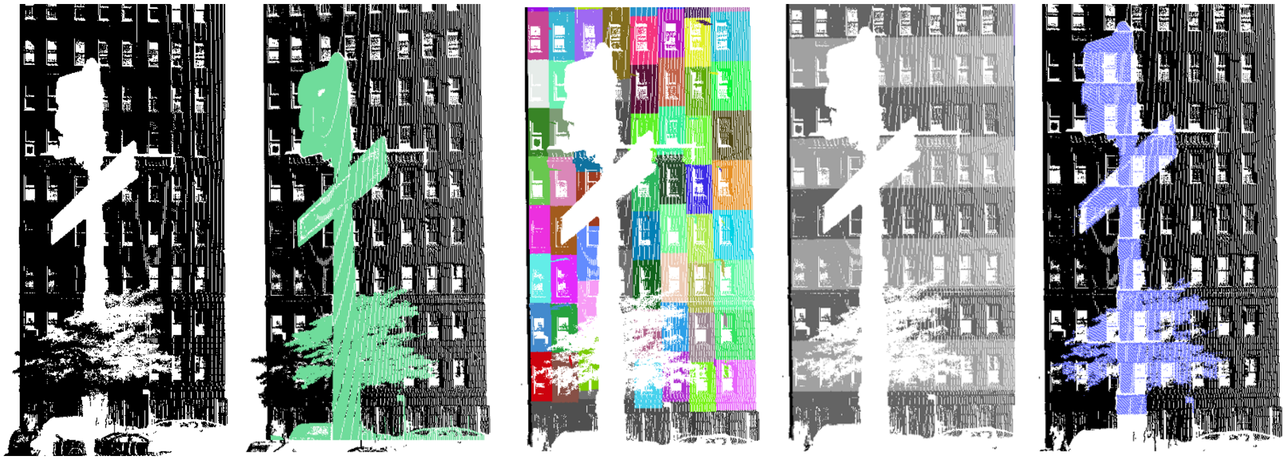

Figure 2. An overview of our method. From left to right is the input data with a large occlusion caused by a street sign and vegetation, then planar hole-filling is shown in green, periodicity segmentation by architectural features yields the colored grid, and per facade periodicity gives the gray stripes. Finally in blue is a synthetic hole-filling which conforms to the regularity of the facade and the 2.5D structure of the point cloud.

3.2. Periodicity Detection

We narrow our focus to the facade points to search for repeated architectural features such as windows and balconies. Operating online we construct a column function for each scanline as it is retrieved, following the approach described in [2]. The column function is derived by taking a local measure at each point. Many measures will suffice to expose the regularity in the scanline. Angles between the vectors connecting consecutive data points is a quick but noisy metric. We primarily use the eigenvalues from the local PCA, which were computed in 3.1. The component in the direction of the facade plane's normal also reveals the regularity in the column. Figure 3 displays three distinct column functions taken over the same periodic column.

To find the period in the column we take the Discrete Fourier Transform (DFT) of the column function. The frequency domain of a periodic column will register two secondary peaks corresponding to the prevalent frequency in the column. Plotting this frequency in the spatial domain shows the repetition of the regular features. Our Fast Fourier Transform implementation requires the size of the input function to be a power of 2. To accommodate this, for a column function with s values we zero pad by finding the k such that $2^{k-1} < s < 2^k$. If the index difference between the zeroth frequency and the dominant frequency is d_i, and the vertical height of the entire column is h, then the period is given by:

$$period = \left(\frac{h}{d_i \left(\frac{s}{2^k} \right)} \right) = \left(\frac{2^k h}{d_i s} \right) \quad (1)$$

Typically the dominant period of a facade corresponds to the height of each floor. While the DFT gives us the frequency of the period we still must compute the phase. We fit square waves to the column function. The phase is shifted to align with middle of the high values of the square wave. This corresponds with the center of the architectural features.

Processing the scan column by column is a natural and efficient way to look at laser scans but at the end of the day we are interested in the image itself not just its constituent columns. To give a higher level interpretation of the scene we aggregate adjacent columns with similar frequencies together. We can sum their frequency spaces by centering about the zeroth frequency. This gives a more robust measure of the period of the feature. We may also use this aggregate to find a more robust choice of feature middle by selecting the median middle from all the grouped columns.

4. Applications

In this section, we present several applications of online periodicity detection and plane estimation. Each approach adheres to the online nature of the regularity detection. Holes caused by occlusions, interiors and missing data are filled. Both triangular and abstract polygonal meshes are constructed from the point clouds. Lastly we introduce methods for sensor planning and point classification.

4.1. Hole Filling

Armed with the current estimates of the major planes we are ready to approximate the missing and occluded data. If the distance between a point and the scanner is less than the distance between the scanner and the plane we label the point as casting a shadow. To approximate the data inside the hole we extend the ray from the scanner until it inter-

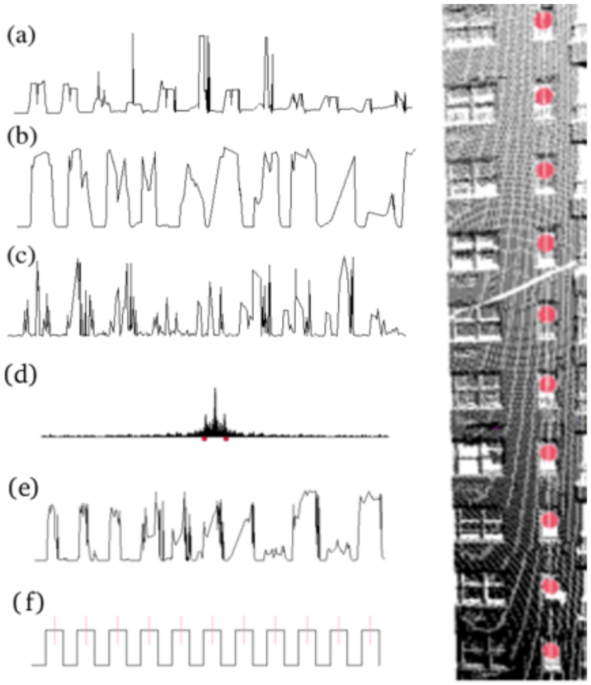

Figure 3. (a) column function derived from orthogonal projection to the major plane, (b) column function of eigenvalues from local PCA, (c) column function from consecutive angle, (d) Discrete Fourier Transform (DFT) of a periodic column function, (e) the eigenvalue column corresponding to the column with red dots in the scan at right, (f) square wave fit to the column to detect the phase of the dominant frequency so that we can center the segmentation around the windows.

sects with one of the major planes. This is a reasonable approximation because the laser travels in a straight line. The point $\mathbf{p_i}$ of intersection is given by scaling the ray to the obstructing object $\mathbf{p_{occlusion}}$ by d:

$$d = \frac{(\mathbf{p_0} - \mathbf{l_0}) \cdot \mathbf{n}}{\mathbf{p_{occlusion}} \cdot \mathbf{n}} \tag{2}$$

where $\mathbf{p_0}$ is a point on the plane, $\mathbf{n}$ is the normal to the plane, and $\mathbf{l_0}$ is a point on the ray from the laser to $\mathbf{p_{occlusion}}$. The origin of the coordinate system is the location of the scanner. Therefore all rays from the scanner pass through the origin, and $\mathbf{l_0}$ can be set to the zero vector. The planar approximation is:

$$\mathbf{p_i} = (d)(\mathbf{p_{occlusion}}) \tag{3}$$

This planar approximation of the missing data may be satisfactory for some applications. It maintains the 2.5D structure of the scan since each shadow point can be replaced with its ray's intersection with the major plane. This structure simplifies many algorithms so maintaining it is beneficial for downstream processing. However we are replacing what may be an ornate piece of architecture with a blank flat wall.

To do better justice to the occluded data we may consider the periodicity of the column in question. Using the period extracted in Equation 1 we may fill this hole by extending a periodic feature into the shadow. To find the appropriate filling, the facade is segmented by its period. This separates the different floors in the building, as can be seen in the alternating colors in Figure 4. Of the segments without any occlusions we select the highest density floor as the representative. For each point in the planar filling of the facade we associate the closest point of the representative floor translated vertically to align with the floor of the planar point. This method ensures that this periodic filling maintains the 2.5D structure of the scan. Facades filled in this manner can be seen in Figures 4, 5, 6 and 7. The floor on which a point p_i lives is given by:

$$floor = \left(\frac{\mathbf{p_i} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}\right) \mod period \tag{4}$$

where $\mathbf{v}$ is the vertical vector. For each planar point $\mathbf{p_i}$ we must associate a periodic point $\mathbf{p_{periodic}}$ extracted from the representative region, $\mathcal{R}$. If f_r is the floor of the representative region and f_i is the floor of the planar point the associated periodic point is given by:

$$\mathbf{p_{periodic}} = \min_{\mathbf{p_k} \in \mathcal{R}} ||\mathbf{p_i} - (\mathbf{p_k} - (f_r - f_i)(period)\mathbf{v})|| \tag{5}$$

This definition suggests an exhaustive search of the entire representative region. However since the column of $\mathbf{p_i}$ is known we can safely limit our search to this column and a few of the neighboring columns for efficient computation.

Once the holes caused by occlusions have been filled we turn our attention to facade holes caused by missing data. This missing data is likely the result of specular or translucent surfaces like windows. On those surfaces we may be unable to obtain a measurement or we may record a point from the interior. Either way these points are inconsistent and disrupt the continuity of the facade. We reconstruct the continuous facade so that the synthesized data conforms to the periodicity of the building and the structured 2D array of the point cloud. Specifically, for each scanline the increment angle θ between readings and the axis of rotation $\mathbf{a}$ are determined. θ is found by taking a sample of points in the scanline computing the angle between them and dividing this angle by the absolute value of index difference between the points. For example if we have points $\mathbf{p_i}$ and $\mathbf{p_j}$ at indices i and j respectively then:

$$\theta = \frac{cos^{-1}\left(\frac{\mathbf{p_i} \cdot \mathbf{p_j}}{||\mathbf{p_i}||||\mathbf{p_j}||}\right)}{|i-j|} \tag{6}$$

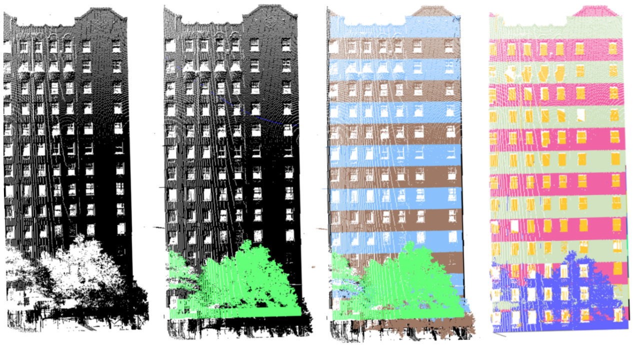

Figure 4. A scan with almost the entire bottom occluded can still be reconstructed. Left to right is the input, the planar filling, segmentation by the dominant period, and finally the periodic filling is shown in blue and the points in orange fill discontinuities that were not caused by occlusion, such as missing data and interior points.

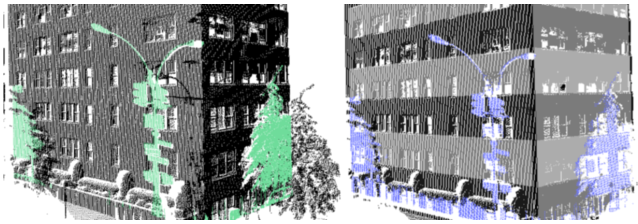

Figure 5. A corner with street lamp and vegetation. At left is the planar filling and at right in blue is the period aware synthesis of the occlusions.

Averaging over a small sample of point pairs from the scanline is sufficient to get a robust estimation of the vertical angle the scanner rotates between subsequent readings. Similarly the normalized axis of rotation for a scanline is given by:

$$\mathbf{a} = \frac{\mathbf{p_i} \times \mathbf{p_j}}{||\mathbf{p_i} \times \mathbf{p_j}||} \tag{7}$$

For numerical stability it is wise to choose i and j that are not close neighbors in the scanline. Now if an interior point or missing data is detected in a scanline we may construct an appropriate replacement that will maintain the continuity of the facade. If our last reliable reading was at point $\mathbf{p_k}$ and point $\mathbf{p_l}$ is determined to be a discontinuity we can replace $\mathbf{p_l}$ by rotating $\mathbf{p_k}$ by $(k - l)\theta$ about the axis $\mathbf{a}$ to obtain $\mathbf{p_c}$.

$$\mathbf{p_c} = \mathbf{R_a}((k - l)\theta)\mathbf{p_k} \tag{8}$$

Finally this vector is intersected with the facade plane using Equations 2 and 3 with $\mathbf{p_c}$ replacing $\mathbf{p_{occlusion}}$. Points obtained this way are colored orange in Figures 4 and 6.

4.2. Online Mesh Creation

Some algorithms require a mesh as input, and cannot directly process point clouds. Using the regularity extracted in Section 3 we can create a hole-free polygonal mesh in realtime with minimal processing requirements. Two such meshes are displayed in Figure 9. The major planes are perforated with intrusions or extrusions occurring at the frequency detected by the Fourier analysis. Each feature is classified as an intrusion or an extrusion depending on where the majority of its points lie in relation to the facade plane. The amount of extrusion or intrusion is determined as the mean perpendicular projection to the facade plane of the intruding or extruding points. We may enforce global properties of orthogonality, adjacency and parallelism between facades in the scene when appropriate as described in [5].

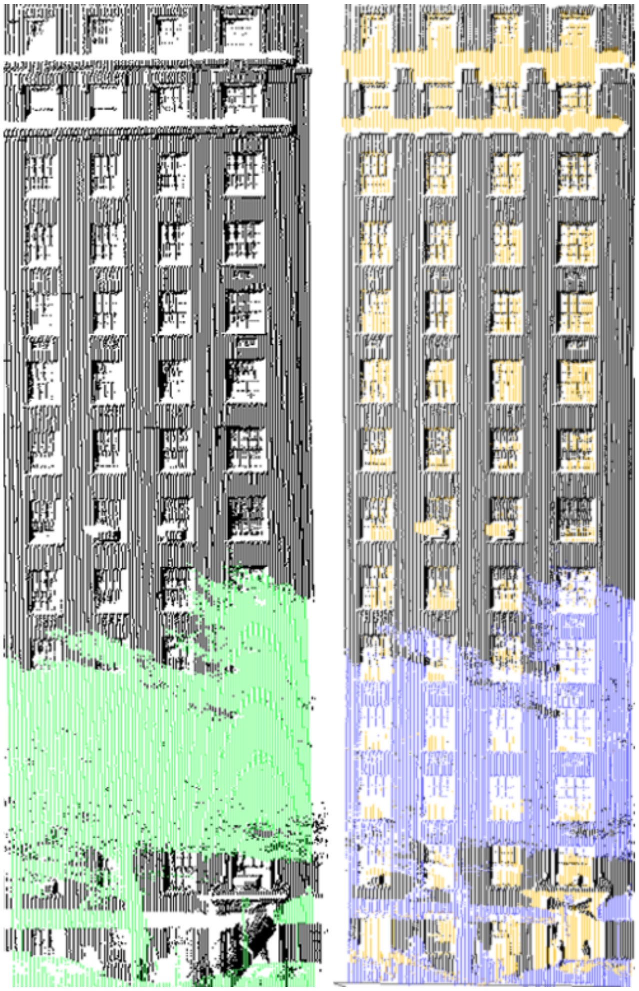

Figure 6. A facade with an occlusion that cuts across its the entire width reconstructed using our algorithm. The green points are the planar intersections of the laser rays with the facade plane, and the blue points are synthesized from the representative floor. The orange points fill holes that were not caused by cast shadows, as described in 4.1.

A triangular mesh can be generated online from a structured point cloud, see Figure 10. Since the data is orga-

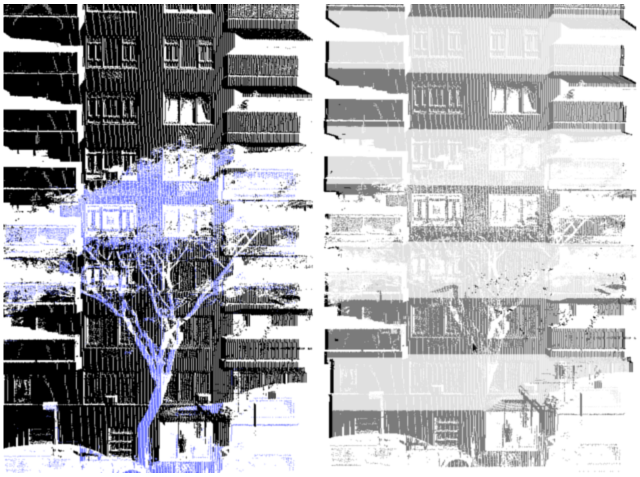

Figure 7. At left is a facade reconstruction with the synthesized points colored blue. At right the same points are colored as the rest of the facade according to the dominant period. Notice how it is not easy to distinguish between the points that were inferred and the starting data.

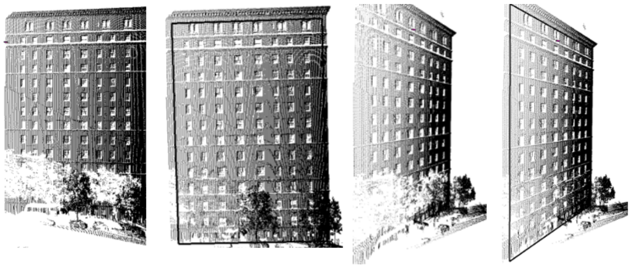

Figure 8. Two views of the same building before and after our hole filling is applied. Notice the seamless integration of the synthetic points and the input data.

nized into a 2D array we can construct triangles by connecting each 3D point to their nearest neighbors in the 2D array. Normally this naïve approach will generate deplorable meshes because the connectivity in the array only roughly corresponds to proximity in 3D space. However since we have removed all occlusions and discontinuities in 4.1 this simple algorithm can be employed to generate acceptable triangular meshes in a single pass over the data, as shown in Figure 10.

4.3. Sensor Planning

The 3D data retrieved by laser scanners is far more informative than conventional 2D images. However, like 2D images a single range scan provides only one viewpoint onto the scene and important information may be missing. For this reason many scans are often taken of the same scene and registered together. We define a "mystery plane" as the planar polygon connecting two detected facades. We compute the normals and area of all the "mystery planes" in the scene. The normal is calculated by taking the cross product of two spanning vectors. The spanning vectors are the vectors between the two known planes between which we have detected a jump. Now we extend this normal by the average length of all the vectors in current scene minus their projections in the vertical direction. This extension of the normal indicates a sensible place for locating the scanner as it will have an excellent view of the missing data (see Figure 11). If we have many "mystery" planes we may sort by their area and return an ordered list of subsequent scanner locations.

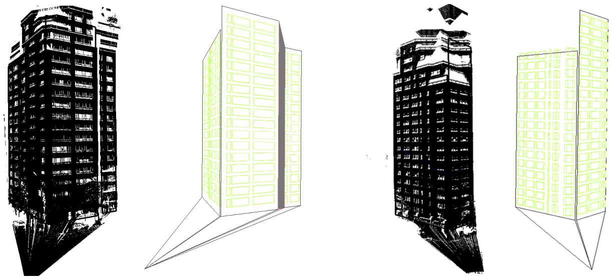

Figure 9. Two point clouds and the polygonal meshes extracted using our algorithm.

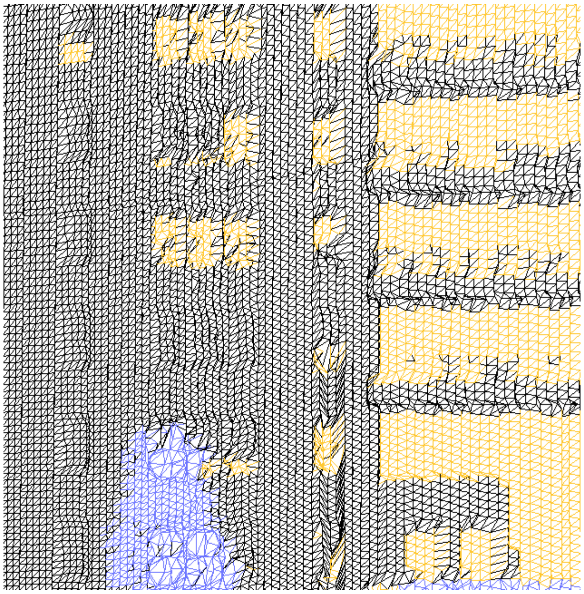

Figure 10. After filling holes from occlusions and from interiors and missing data we can generate a triangular mesh online. Blue triangles contain points that were synthesized from the periodicity of the building. Orange triangles contain points that were missing or from an interior that were intersected with the facade, as described in 4.1.

4.4. Evaluation

Evaluating a reconstruction of the type presented here is not straightforward. The algorithm synthesizes data we were unable to acquire. We must therefore evaluate our accuracy in hitting an unknown target. To begin we visually evaluate our periodicity and phase detection by overlaying

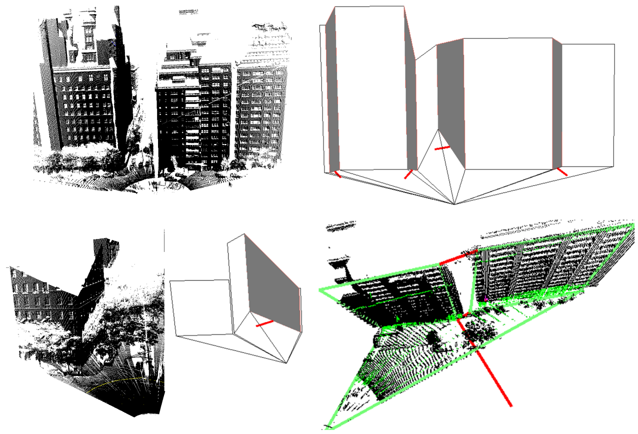

Figure 11. Here the "mystery planes" are colored in gray. The red vectors protruding orthogonally indicate wise placements for subsequent scans.

the polygonal windows on input data. Figure 12 shows the windows in green superimposed upon the scan.

To further illustrate the accuracy of our method we create artificial occlusions, fill them, and then compare our reconstructions with the real data. Several of those results are displayed in Figure 13. To ensure the boundaries of the synthetic holes do not effect the results we create holes using Euclidean, Manhattan, and Chebyshev distance metrics. We then measure the Hausdorff distance from the hole-filling to the real data. The distance averaged about a quarter of a meter and was never greater than a half of a meter.

For a structured point cloud of n rows and m columns our algorithm runs in $O(mn\log_2 n)$ time. For each column we sort and perform the fast Fourier transform both with run time $O(n\log_2 n)$. The computational cost of running our algorithm on several large-scale scans is presented in Table 4.4. The per point processing time varies from 3 to 16 microseconds. This is several orders of magnitude faster than the rate at which the scanner acquires a single point. Additionally, our code is implemented in Java and not fully optimized.

5. Conclusion

We have presented an online algorithm which exploits the regularity of the urban scene to fill holes in point clouds and generate abstract meshes. Our algorithm works online allowing for near simultaneous processing and acquisition of 3D scenes. We foresee a slew of algorithms of this ilk, creating intelligent 3D cameras which generate excellent data out of the box.

Scan	Points	Run Time (in milliseconds)	Time Per Point
Scan 8	764154	4859	.0064
Scan 2	546964	6929	.0127
Scan 11	773376	8390	.0012
Scan 14	1256904	9415	.0075
Scan 9	1006740	11419	.0113
Scan 6	1219106	12705	.0104
Scan 3	1303359	13951	.0030
Scan 1	805176	13316	.0165
Scan 4	1326117	14943	.0112

Table 1. This table shows the algorithm's running time on range scans of various sizes. Run time is not completely proportional to scan size. This is because the individual column size and the ratio of periodic columns also factor into the algorithm's performance.

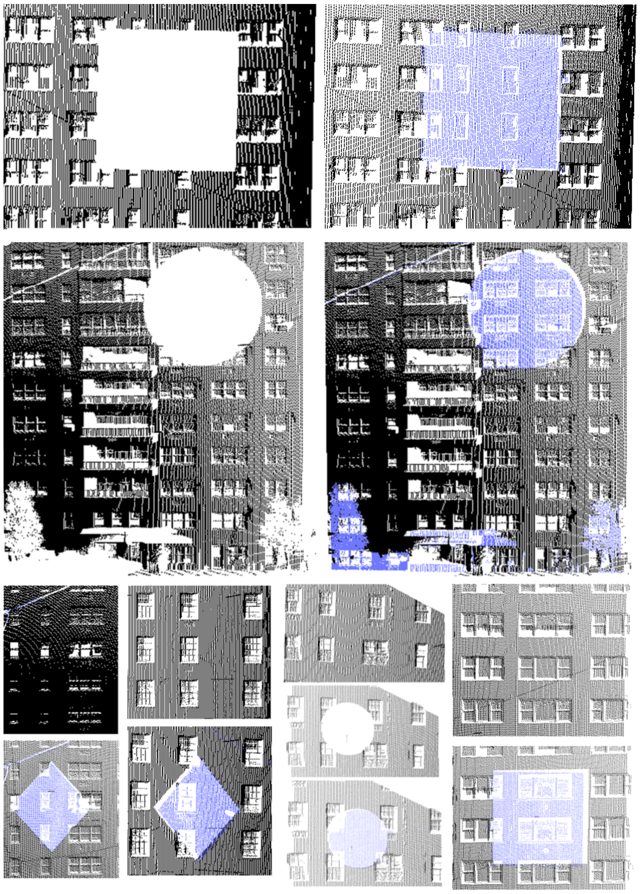

Figure 13. We generate synthetic shadows and fill them. We measure the Hausdorff distance from the filling to the real data to estimate the accuracy of our approach. In the top image the error distance was 0.28 meters. From left to right on the second row the error distances were: 0.32, 0.47, 0.07, and 0.44 meters.

5.1. Limitations

Our algorithm requires at the very least planarity of the obstructed data and hopefully periodicity as well. While most urban facades fit this description many do not. Our method does not require globally rectilinear facades that

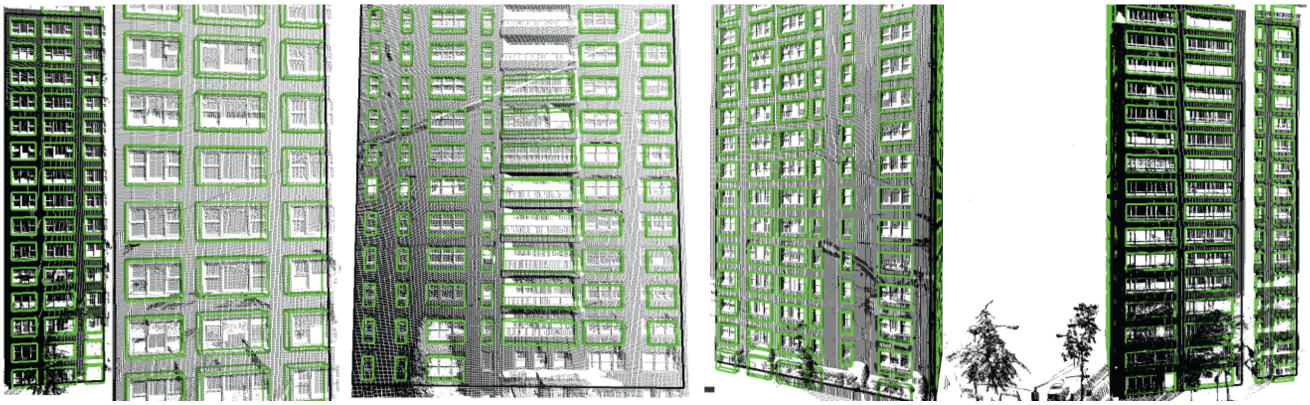

Figure 12. Superimposing the extracted mesh on the point cloud we can verify the accuracy of our results.

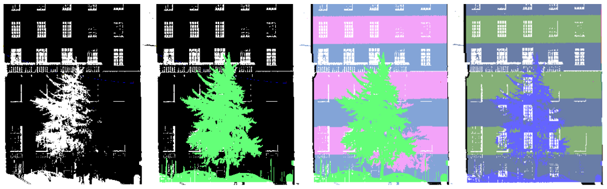

Figure 14. Here the first two floors of the building have a different dominant frequency from the rest of the facade. Our algorithm incorrectly propagates the features from the upper floors into the occlusion caused by the tree.

conform to a regular lattice, because we can update our period estimates on-the-fly when new dominant frequencies are detected. However if the period changes in the middle of the scanline, as in Figure 14, our algorithm may propagate one period into an area with a different dominant frequency.

In scans of low resolution the periodicity of the signal is often obscured by noise and we must settle for planar approximations. If resolution is extremely low the local planes become unreliable and even a planar approximation becomes error-prone.

5.2. Future Work

Our early forays into classifying the shadow points merit further research. Other semantic groupings of these points are reasonable such as vehicles, signs, vegetation, and humans, to name a few. Classifying the points online would be highly advantageous for autonomous scanners to navigate safely. Many of these subsets exhibit their own regularity such as the branching of trees or the Bezier curves of cars. We leave the detection and classification of such structures to future work.

References

[1] I. Drori, D. Cohen-Or, and H. Yeshuron. Fragment-based image completion. *ACM SIGGRAPH*, 2003.

[2] S. Friedman and ioannis Stamos. Real time detection of repeated structures in point clouds of urban scenes. In *3DIMPVT*, 2011.

[3] O. Hadjiliadis and I. Stamos. Sequential classification in point clouds of urban scenes. In *Fifth International Symposium on 3D Data Processing, Visualization and Transmission*, May 2010.

[4] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM SIGGRAPH*, 2007.

[5] Y. Li, X. Wu, Y. Chrysanthou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4), 2011.

[6] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. J. Mitra. 2d-3d fusion for layer decomposition of urban facades. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[7] A. Martinet, C. Soler, N. Holzschuch, and F. X. Sillion. Accurate detection of symmetries in 3D shapes. *ACM Transactions on Graphics*, 25(2):439–464, 2006.

[8] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3D geometry. In *ACM SIGGRAPH*, pages 560–568, 2006.

[9] N. J. Mitra, L. J. Guibas, and M. Pauly. Symmetrization. In *ACM SIGGRAPH*, pages 63–71, 2007.

[10] P. Muller, , G. Zeng, P. Wonka, and L. V. Gool. Image-based procedural modeling of facades. In *ACM SIGGRAPH*, 2007.

[11] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer. A survey of urban reconstruction, 2012.

[12] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen. Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics*, 29(4), 2010.

[13] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 27(3):#43, 1–11, 2008.

[14] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM SIGGRAPH*, 2004.

[15] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3d urban scenes. *ACM Transactions on Graphics*, 29(3), 2010.