Online Detection of Repeated Structures in Point Clouds of Urban Scenes for Compression and Registration

Sam Friedman & Ioannis Stamos

International Journal of Computer Vision

ISSN 0920-5691

Int J Comput Vis
DOI 10.1007/s11263-012-0575-y

Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your work, please use the accepted author's version for posting to your own website or your institution's repository. You may further deposit the accepted author's version on a funder's repository at a funder's request, provided it is not made publicly available until 12 months after publication.

Springer

Author's personal copy

Int J Comput Vis
DOI 10.1007/s11263-012-0575-y

Online Detection of Repeated Structures in Point Clouds of Urban Scenes for Compression and Registration

Sam Friedman · Ioannis Stamos

Received: 5 November 2011 / Accepted: 11 September 2012
© Springer Science+Business Media, LLC 2012

Abstract Laser range scans of urban areas have a distinctive geometry dominated by facade and ground planes and repetitive regular fenestration. Detection of these ubiquitous features provides profound insights into the scene. We present a novel method for detecting major planes and repetitive architectural features. Armed with this knowledge we illustrate its application in compression and registration of range scans. What is more our algorithm operates online, processing the scan as it is retrieved by the scanner. This realtime approach opens up new possibilities in range data segmentation, compression and registration.

Keywords Urban range scans · 3D scan registration · 3D scan compression · Regularity detection

1 Introduction

Understanding 3D images is increasingly important in fields as diverse as navigation, architecture, and biology. Each domain is replete with unique patterns that practitioners must uncover as they struggle to make sense of their data. In urban scenes, for example, the topology is characterized by the broad planes of facades, the slightly curving manifold of the street, and the fenestration of regular repeated architectural features like windows, balconies and cornices. Also notable is the fractal noise of vegetation, the Bezier curves of parked cars, and the high frequency spikes of objects in motion faster than the laser can catch them. In this paper we show how a focus on the repeated architectural structures yields significant scene understanding and can be used for compressing and registering range scans. We demonstrate a method to discover these regular structures online, as the scanner scans. We apply knowledge of these structures to facilitate compression and registration.

Processing point clouds quickly is crucial in many scenarios. Analyzing scans column by column allows seamless integration with the 3D camera because the camera is likewise scanning the scene column by column. This synchronizes the execution of the algorithm with gathering the data, accelerating our work to the point of just-in-time scene understanding. While focusing on individual columns gives excellent local knowledge on the scene we do not wish to lose sight of the big picture. After all, each column is just a sliver of a much larger tableau. However, by keeping an evolving data structure with macro features and greedily updating as the scan unfolds we are not forced to choose between speed and global knowledge, in fact our algorithm achieves both.

We can imagine this online processing as a line sweep algorithm. The scanline or column of measurements moves across the scene in discrete steps. At each step we search for periodicity and planes. We aggregate and update this data maintaining best estimates about higher level features in the scene. The processing occurs on the fly so that by the time the scanline reaches the edge of the scene our algorithm has done its work and requires no further manipulations of the data. Integration into the 3D camera hardware is a natural step forward for these types of algorithms.

This work has been supported in part by the following NSF grants: IIS-0915971, CCF-0916452 and MRI CNS-0821384. We would like to thank Tom Flynn for his labeling and registration software. We are also thankful to the anonymous reviewers for their helpful comments.

S. Friedman · I. Stamos ( )
Hunter College & Graduate Center of CUNY, CUNY, New York City, NY, USA
e-mail: istamos@hunter.cuny.edu

S. Friedman
e-mail: umpteee@yahoo.com

Published online: 28 September 2012

Author's personal copy

2 Related Work

Modeling 3D urban scenes from range scan data is a field of active research (Allen et al. 2003; Stamos et al. 2008; Zhao and Shibasaki 2003). There is a plethora of range image segmentation techniques from range data including edge detection (Bellon and Silva 2002; Wami and Batchelor 1994), region growing (Besl and Jain 1988; Pulli and Pietikinen 1993), and polynomial surface fitting (Besl and Jain 1988; Marshall et al. 2001). Most of these methods provide edge maps or regions expressed as polynomial functions. Yu et al. (2001) utilizes a graph-cut approach for segmentation. Our earlier work includes segmentation algorithms for the extraction of planar, smooth non-planar, and non-smooth connected components (Chao and Stamos 2007; Stamos and Allen 2002; Stamos et al. 2006).

Detecting symmetries and repetitive structures (such as windows and balconies) can be an important element of scene understanding. The concept of shape grammars for buildings introduced by Stiny (1982) as a formal approach to architectural design has been used for the image-based modeling of facades acquired through aerial photographs (Muller et al. 2007) and for the generation of synthetic cities (Muller et al. 2006). A shape-grammar approach that uses ground-based images was presented in Teboul et al. (2010, 2011). An earlier approach is the one of Lee and Nevatia (2004). The work of Mayer and Reznik (2007) uses a sequence of images to produce a sparse 3D point cloud as input to the detection algorithms. Work on the same topic is presented in Wu et al. (2010). Finally Park et al. (2009) detects regular patterns in images that are not constrained to be buildings.

Fewer approaches exist when the input is a 3D range scan. Pauly et al. (2008) derives regularities of substructures from a 3D model or range scan of a scene. This works by detecting symmetries (or similarity transformations) of basic structures in a regular grid. This general approach can be used for extracting regularities but it is sensitive in the calculation of curvatures and computationally intensive. Shen et al. (2011) presents a partitioning of urban facades which detects grids in an adaptive way. This extends the transformation voting technique of Pauly et al. (2008) to facades that are not globally rectilinear. In Stamos and Allen (2002) window-like rectangular features were extracted by using 3D edge detection on high-resolution 3D data. Line features are again utilized in Bokeloh et al. (2009) for symmetry detection. This symmetry detection of line features builds upon a graph based method for detecting symmetry presented in Berner et al. (2008). Nan et al. (2010) presents an interactive interface which exploits regular structures in urban scenes to improve sparse point clouds. Employing a Markov Network approach (Triebel et al. 2006) labels points as windows, but requires training. The work of Li et al. (2011) provides a fusion of 2D- and 3D-segmentation and symmetry detection for decomposition of urban facades. The authors of Martinet et al. (2006) use spherical harmonic decompositions to detect rotational and reflective symmetries in meshes. Our work also exploits harmonics, but we use them to detect translational symmetries per scanline. Our paper provides algorithms that work online and purely on 3D-range input without relying at all on 2D-images. We have presented work in the online detection of repeated patterns in urban scenes using 3D-information (Friedman and Stamos 2011). This paper extends the applicability of that work into the area of 3D-range registration. Table 1 compares and contrasts several of the algorithms discussed above.

There is a vast literature in the area of 3D-range registration Iterative Closest Point (ICP) is one of the most popular range registration algorithms (Besl and McKay 1992; Rusinkiewicz and Levoy 2001). ICP provides very accurate results but requires a good initialization of the registration transformation. Methods that do not require initialization in most cases detect and match features such as spin images (Huber and Hebert 2003) or lines (Stamos and Leordeanu 2003). A RANSAC-type automated ICP algorithm is presented in Aiger et al. (2008). In our paper we are describing an automated range registration algorithm that is based on online detection and matching of repeated patterns. Our algorithm is extremely fast and accurate without being confused by parts of the scene that may be different (for instance objects such as cars that can move and change positions during acquisition, or vegetation that may change between seasons).

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

(a) We extract repeated structures from large-scale scenes online, processing the data as it is acquired while also enabling extremely efficient offline processing.
(b) Using periodicity as the basis of our detection, we have robust results even in low resolution areas of the scene.
(c) We compress data-rich range scans while maintaining their architectural regularity.
(d) We present automatic registration opening the door to simultaneous acquisition and realtime registration from many range scanners at once.
(e) All of the above are achieved without requiring user interaction, or training data.

3 Data Acquisition

Our lab operates a Leica ScanStation2 laser scanner (Leica Geosystems 2012). The data retrieved is organized into a two dimensional array. The direction of the laser is controlled by delicately calibrated stepper motors. The laser rotates a small consistent angle vertically and records a measurement and then repeats. Once an entire column has been

Author's personal copy

Table 1 Comparison of regularity detecting algorithms. This table is adapted from Mitra et al. (2012)

Reference	Method	Input	Feature	Transforms
Berner et al. (2008)	feature graph matching	meshes points	slippage features	rigid
Bokeloh et al. (2009)	feature graph matching	meshes	lines	rigid
Martinet et al. (2006)	generalized moment functions	meshes	moments	rotation reflection
Pauly et al. (2008)	transformation voting	points	curvature	similarity transforms
Our Method	harmonics	points	eigenvalues curvature	translations

recorded another motor rotates the laser a small consistent angle horizontally and another column is recorded. In the pages that follow we will refer to single arrays within the 2D array as both columns and scanlines. In the coordinate system returned by the scanner the z-axis corresponds to the vertical vector.

This data is considered 2.5 dimensional because the 2D array of 3D points contains valuable information that distinguishes the scan from an unstructured point cloud. For example, investigating neighborhoods in the 2D array we are able to fit normals in constant time without resorting to an exhaustive search of nearest neighbors, or the additional overhead of an approximate nearest neighbors (ANN) library.

4 Extraction of Repeated Architectural Features Through Fourier Analysis

Windows and balconies are intriguing data points in urban laser scans. They are ubiquitous: virtually every commercial and residential structure is perforated by some kind of fenestration. In general, they are regular, occurring at a defined period within the building. Yet despite this regularity they are challenging to detect because of their high variability of appearance.

Glass windows without any shades often allow the laser into an interior from which only negligible light can return, yielding a missing data point. Some materials used to make window blinds appear extremely planar and may be mistaken for facades. Other materials like fabrics and plastic shades will present highly irregular geometry depending on the wind, the angle of incidence and other factors.

To compound the confusion many window sills are crowded with house plants, children's toys and other accouterments of apartment interiors. Balconies are marked by prominent self occlusions. Terraces which protrude from a building cast shadows on the facade which grow with the incident angle of the laser. This produces vastly different results for the same structure depending on its altitude.

All these facts make detection of regular architectural features difficult. Nevertheless the periodicity of the features is salient. While each individual feature may be unpredictable when analyzed together they are predictably unpredictable. This regularity can be extracted through Fourier analysis and this knowledge is fertile ground for downstream processing.

4.1 Finding the Major Planes

We begin by approximating the major planes for each column.[1] Most scanlines in the urban setting are dominated by points belonging to the ground and points from a facade. In general facades are extremely planar. The manifold of a city's pavement is usually curved, but its curvature is in general low and a planar approximation is often sufficient. To extract these major planes we fit minor planes in small neighborhoods around each point in the scanline. Using Principal Component Analysis (PCA) we determine the normal to these minor planes as the eigenvector corresponding to the smallest eigenvalue of the covariance matrix of the neighborhood. The eigenvalue itself is also stored as its magnitude is a measure of the goodness of fit of the local plane to the local data. Throughout the scan we maintain our best estimates of the facade and ground normals.

Buildings rise more or less perpendicularly from the more or less horizontal ground. So we may filter potential ground normals from potential facade normals by taking the dot product with the vertical vector. We assume the vertical vector is known as there a number of simple ways to determine it. We expect this product to be near 0 for ground normals and near 1 for facade normals. Now we have gathered two groups of small local planes from which we would like to deduce two global planes. The eigenvalues computed earlier help separate the noise from the signal by measuring the reliability of each local plane. However, simply selecting the normals corresponding to the smallest eigenvalues in the column would be a mistake. It is common, for example for a facade with a balcony to register two reliable vertical planes, or a scan that passes over a box truck or a ceiling from an interior may contain two or more horizontal planes with very low eigenvalues. To insure against these misdetections, the

[1] We will use the terms column and scanline interchangeably.

Author's personal copy

major planes are chosen as those with low eigenvalues and agreement with the majority of other potential vertical or horizontal planes.

As the scan progresses and more columns are recorded we greedily update our major plane estimates. While searching for the most reliable facade normal, it is important that we remain vigilant against facade shifts that occur when the scanner passes over a corner of a building or two adjacent facades. Before the running estimate of the facade is updated the angle between the old normal and the new normal is measured. If it is insignificant and the current normal is more reliable than the previous it is updated. If this angle is near $\pi/2$ we note a corner. If this angle is not near zero and not near $\pi/2$ this may be a noisy column or modern architecture. The angle between facade normals does not detect facade translation which are common when adjacent buildings are offset from the street by different amounts. To ensure that the facade has not been translated we must check that:

$$(f_1 - f_2) \cdot n_1 \approx (f_1 - f_2) \cdot n_2 \approx 0, \tag{1}$$

where f_1 is a point on the old facade, f_2 is a point on the new facade and n_1 and n_2 are normals of the two facades. If this is not the case then we must note a facade shift and going forward greedily improve the new facade. See Fig. 1. In settings where online processing is unnecessary, it is possible to improve previous global estimates after one pass has been made over the data.

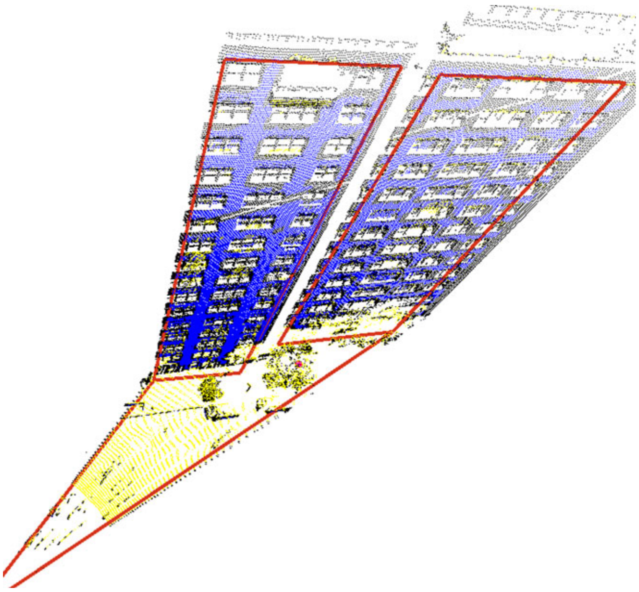


Fig. 1 The *red lines* indicate the edges of the major planes. The intersection of the ground plane with the facade plane can be computed even though this vector is often obscured in the raw data by vegetation, vehicles and pedestrians. The *darker blue points* are candidates for the facade plane and the *lighter yellow points* are ground plane candidates. Note that the facade shift has been correctly identified by the algorithm (Color figure online)

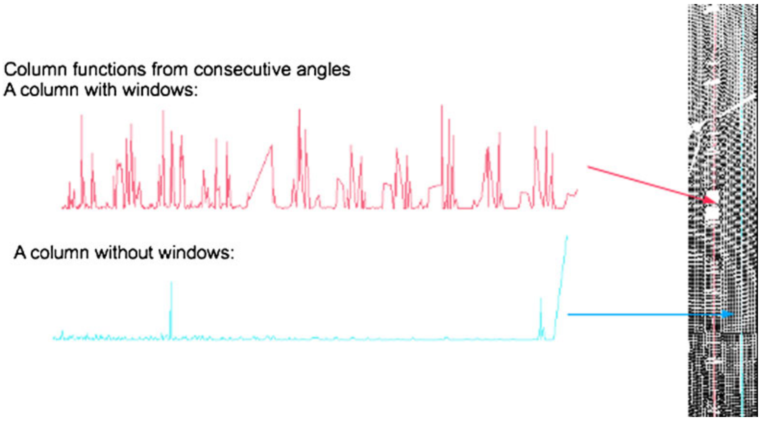
4.2 The Column Function

The global features of the major planes are complemented by the meso-level features of regular fenestration in windows and balconies. Fenestration tends to be regular occurring once for each floor. The first step in the detection of this regularity is construction of a column function. The estimates of the ground and facade planes allow us to distinguish all the points in the column on or near the facade. The rest of our analysis will focus on this subset.

The column function is derived from a local measure taken at each point in the column. Conceivably curvature, eigenvalues, or consecutive angles could be used. In our work we have chosen consecutive angles for the simplicity and speed of their computation. Two column functions are shown in Fig. 2. Despite being a noisy and simple metric the regularity of repetitive features shines through.

Before this regularity can be extracted the column function must be interpolated. As it stands the column function is distorted by an oversampling of close features and an under sampling of distant ones. Figure 3 illustrates the predica-

Fig. 2 Two column functions are displayed. The lighter one has covered a planar region wheres as the darker scanline and corresponding column function pass over a section of the building with windows. The regularity of the windows shines through the noisy column function. This regularity can be precisely extracted using Fourier analysis (Color figure online)



Author's personal copy

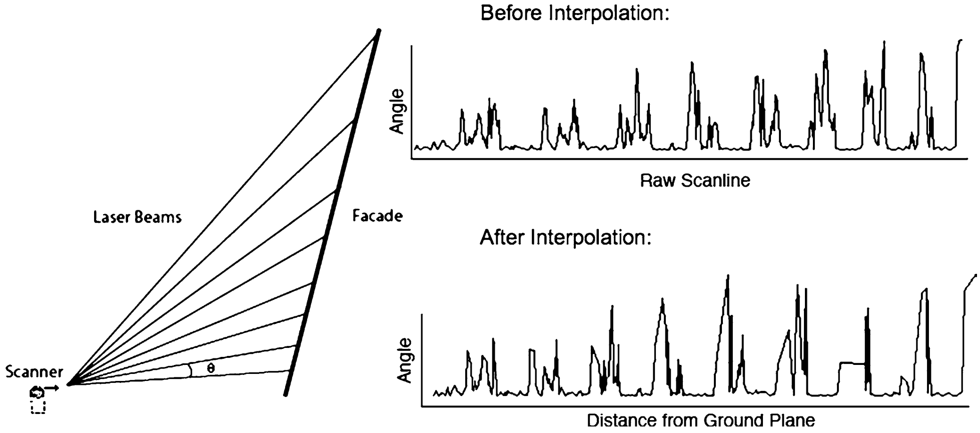


Fig. 3 The scanner over-samples near features and under-samples distant ones. As a result the beginning of the column function before interpolation displays a lower frequency period. It appears that the frequency increases throughout the column function when in fact the period is constant, as can be seen in the column function after interpolation. Interpolation is critical to precisely detect the period of the column function

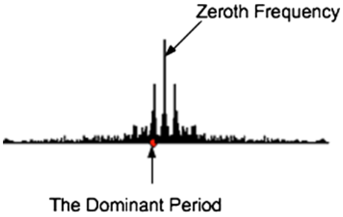


Fig. 4 The Fourier Transform of a column function containing windows has noticeable peaks corresponding to the frequency of the period. The column function is a real signal so the transform is symmetric which accounts for the double penultimate highest peaks. The highest peak in this transform is the zeroth frequency

ment. To rectify this, points are sorted by their height. The column function is scaled to reflect these heights. After the linear interpolation, the column function shows actual distances in the scene and not the concentration of measurements returned by the laser scanner.

4.3 Frequency Space

The signal after interpolation is truly periodic. Fourier analysis will indicate exactly which period is present in the function. The Discrete Fourier Transform maps our signal into the frequency domain. Figure 4 displays one such transform. It is symmetric because the column function input is real. In frequency space periodic functions have salient peaks at the frequencies of the period in the signal. The zeroth frequency is the average of the signal and is the largest peak in the transform, but tells us nothing about the period. Rather it is the secondary spikes which indicate the dominant period of a column function. These secondary spikes are detected in one pass over the data. At this point it is possible to return to the spatial domain and plot the periods found in the frequency space. The results are displayed in Fig. 5. The results show that analysis of a single column is enough to discover the period of repetitive architectural features.

4.4 Aggregation

The estimation of the period from a single column may be improved through aggregation. Since the algorithm is processing data online directly after uncovering the period latent in one column function another column function is

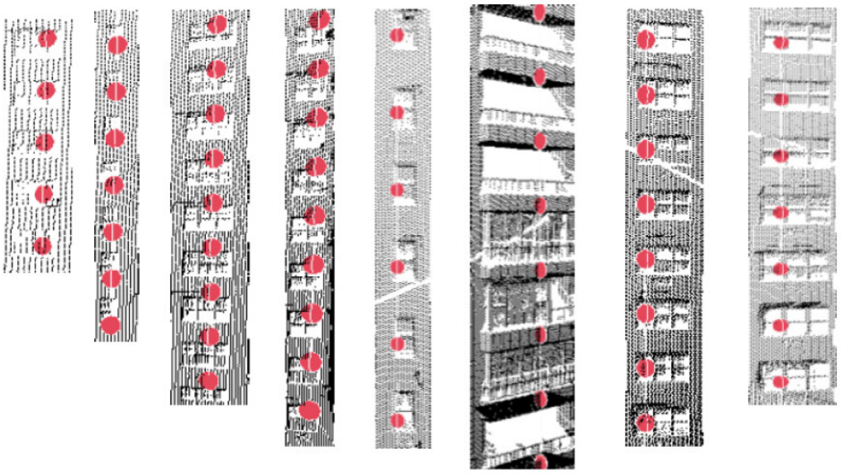


Fig. 5 The dominant period from a selection of column functions which covered areas of periodicity

Author's personal copy

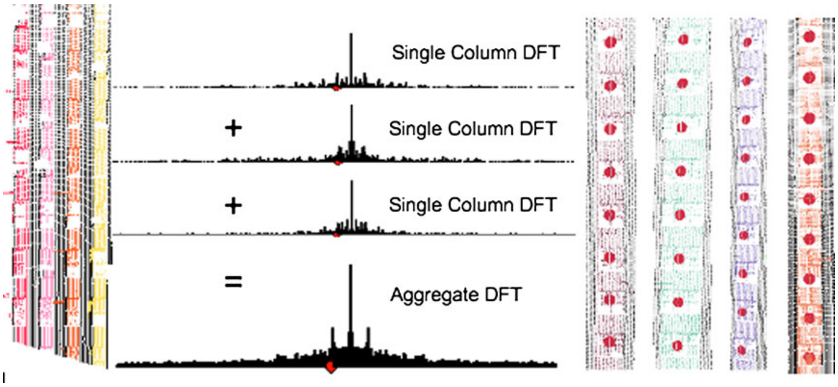


Fig. 6 Aggregating Fourier Transforms provides a more robust measure of the periodicity of a set of adjacent periodic columns. The second highest peak in the aggregated transform is more prominent than in the transform of any single column. The *black columns* are planar regions and the *colored columns* are repeated structures (Color figure online)

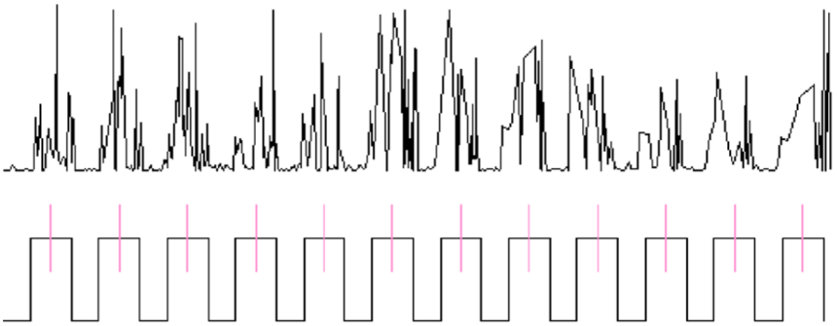


Fig. 7 A square wave can be fit to the column functions. This yields more information about the features and can be used to obtain robust translation vectors between different views

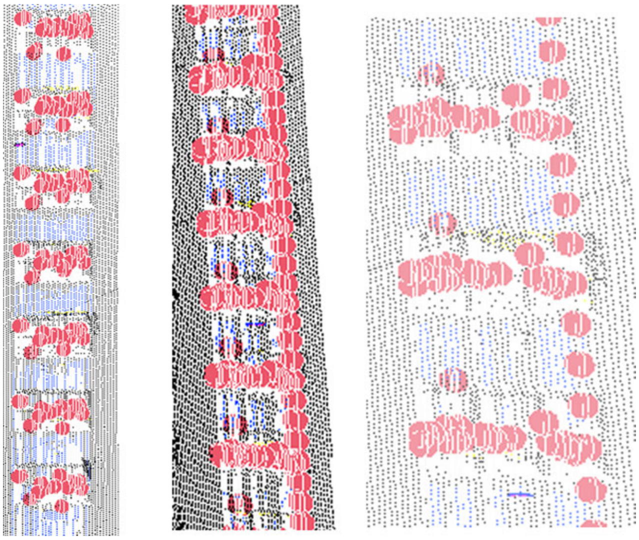


Fig. 8 The square wave gives a good estimation of the window centers. Note that the periodic points from the individual column do contain both high and low frequency outliers. However in the aggregate our window center estimation is robust. Furthermore high frequency "outliers" often contain additional information such as the frequency of panes within each window. See Fig. 9 for the aggregate centroids that are the medians of the many centers shown above

ready for processing. With high probability this column will record data from the same repeated architectural features that were uncovered previously. Adjacent columns with near identical period can then be grouped providing higher level insight into the features present in the facade. The period estimates can be improved by detecting the secondary spikes in the sum of all the transforms of the neighboring columns. Before summing the transforms must be centered about the zeroth frequency. This is because the columns are not necessarily the same size as occlusions may obstruct some scanlines and not others. Figure 6 illustrates this process. The resultant period is a more robust measure of the frequency of the repetitive features.

4.5 Fenestration Centroids from Square Waves

Deeper insight into the periodic feature can be obtained if we fit a square wave to the column function. This wave measures the actual height of the architectural feature and allows us to extract the exact middle of the windows and balconies, see Figs. 8 and 9. The previously gathered information about periodicity in the column allows us to uncover the best-fitting square wave quickly. It is practical to exhaustively check all possible square waves up to some finite resolution whose period coincides with the period the Fourier analysis uncovered in the column. The square wave that fits the column function will be our estimate of both the height and centers of the architectural features.

To fit a square wave the column function is turned into a binary function by taking all values above a threshold as 1 and all values below it as 0. We measure the difference between square waves and this binary thresholded version of

Author's personal copy

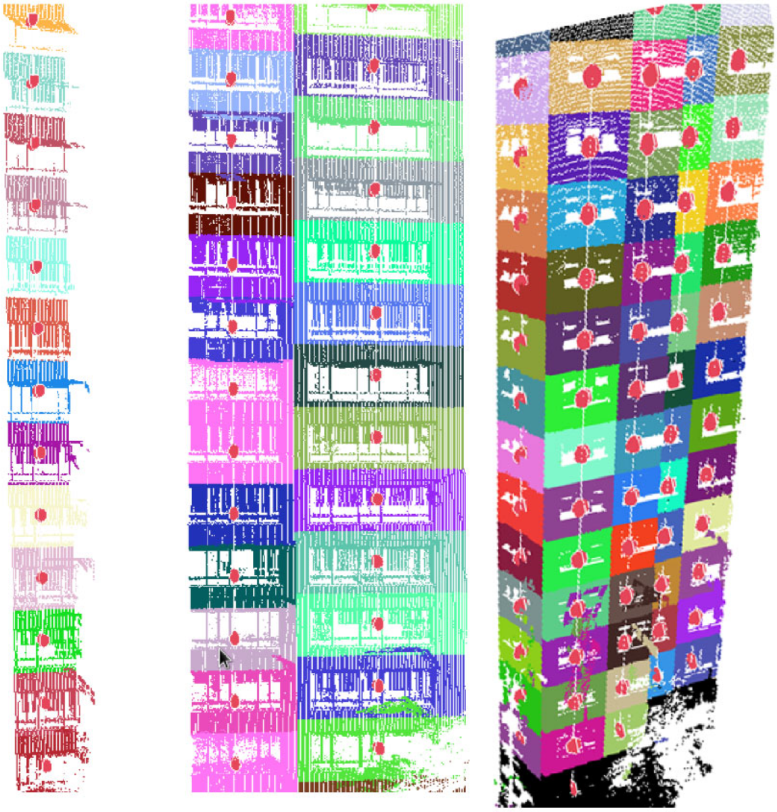


Fig. 9 The median feature middle from all the columns in the group is selected as the height of the groups centroid. The horizontal location is determined by the Chebyshev distance in x and y dimensions between the first and last scanlines in the group (Color figure online)

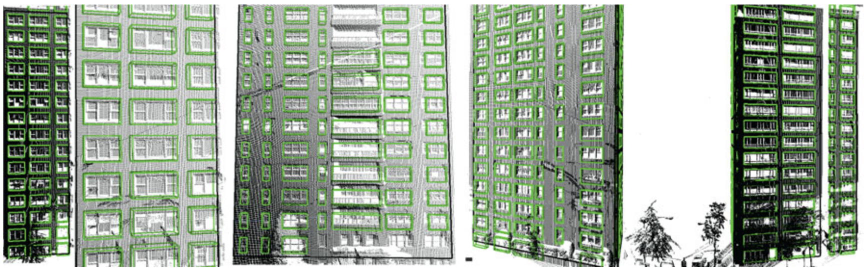


Fig. 10 The *green polygons* indicate the areas the algorithms detected as window regions. The size of each period is determined by the dominant frequency. The size and centering of the polygons is determined by the square wave fitting as described in Sect. 4.5. We compare these polygons against the ground truth from Fig. 11 to obtain Table 2 (Color figure online)

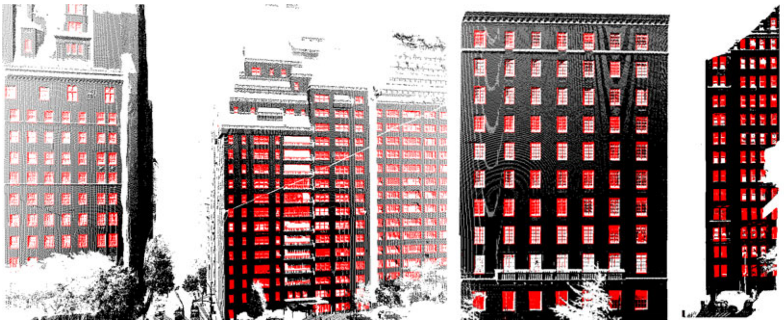


Fig. 11 Points belonging to window regions are colored *red*. These labeled scans are compared against our algorithms output to evaluate the regularity detection. The quantitative results are shown in Table 2 (Color figure online)

the column function. We repeat this process for every possible square wave within the period of the column, up to some discrete resolution. The wave with the smallest difference is chosen as the square wave approximation of the column function. A typical square wave and the column function it fits are shown in Fig. 7.

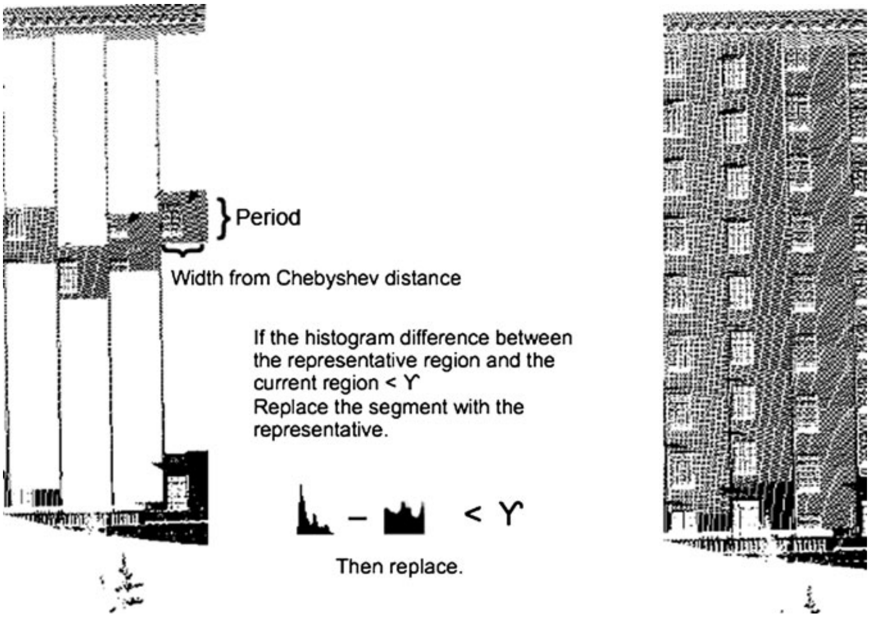
Author's personal copy

Table 2 The precision and recall of the regularity detection is presented here

Scan Name	True Positives	False Positives	False Negatives	Precision	Recall
Scan 1	95	1	7	.9896	.9314
Scan 2	124	6	9	.9538	.9323
Scan 3	53	3	4	.9492	.9333
Scan 4	56	1	3	.9824	.9491
Scan 5	58	1	4	.9830	.9355

Table 3 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

Scan	Points	Column Size	Periodic Columns	Run Time (in milliseconds)	Time Per Point
Scan 8	764154	7209	30	4859	.0064
Scan 2	546964	1364	232	6929	.0127
Scan 11	773376	2432	151	8390	.0012
Scan 14	1256904	4761	69	9415	.0075
Scan 9	1006740	1260	144	11419	.0113
Scan 6	1219106	3286	235	12705	.0104
Scan 3	1303359	2721	299	13951	.0030
Scan 1	805176	636	502	13316	.0165
Scan 4	1326117	2757	216	14943	.0112



If the histogram difference between the representative region and the current region < ϒ
Replace the segment with the representative.

Fig. 12 Our framework for scan compression. The compressed scan has reduced all the windows above while maintaining the ornamentation and irregularities

The middle of the high edge of the square wave corresponds with the middle of the repeated fenestration. Square waves are generated for each column in each aggregated window group computed in Sect. 4.4. The center of the feature can then be estimated as the median of the vertical centers of each column and the mean of the horizontal distance between the first and last column. The centers computed for each column can be seen in Fig. 8 and the aggregate centers are shown in Figs. 9 and 21.

4.6 Evaluating Regularity Detection

We employ several methods to evaluate the quality of the detected translational symmetries. First, polygons of the repeated architectural features are superimposed on the scan data. This gives a visual verification of the detection of both the period and the phase of the features. Figure 10 displays the superpositions in several different scans.

Author's personal copy

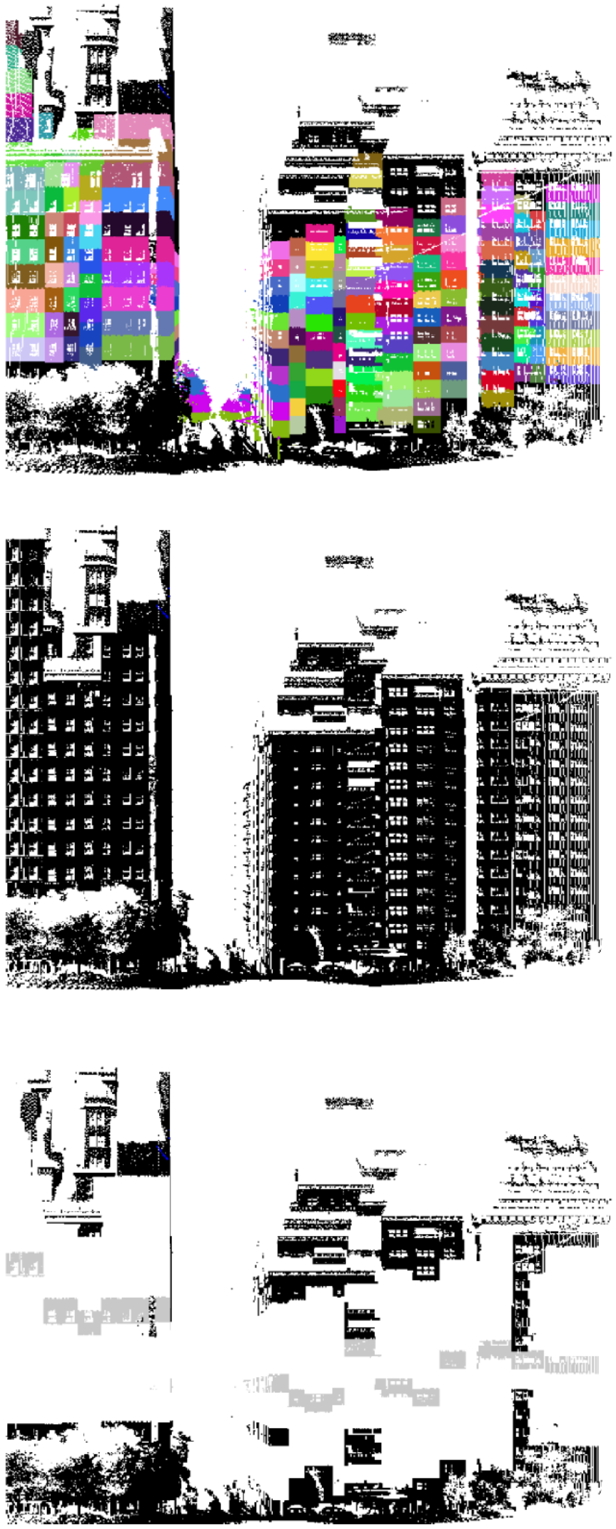


Fig. 13 The *top* image is a segmented scan. The *middle* image is a compressed version of the scan above it. The *bottom* image shows the amount of reduction. In the *bottom* image the *grey points* indicate the representative regions which are copied and translated to achieve the compression. See Sect. 5.3. The compression ratio is 48 % with 383735 points removed out of 805176. Even though some window groups contain multiple columns of windows compression is unaffected because the periodicity of all of them has been correctly identified (Color figure online)

For a quantitative assessment we have annotated the scans with labels for the regular features. For example, the red points in Fig. 11 are labeled as windows by a human visually selecting them. This ground truth is compared against the polygons displayed in Fig. 10. If 90 % or more of the labeled points fall within the area detected as a window it is considered a true positive. If there is a detected window and no labeled points fall within it, then it is a false positive, and lastly, if there are labeled points but no detected window it is a false negative. In this way, Table 2 is obtained.

4.6.1 Runtime Analysis

The algorithms presented here run in $O(n \log n)$ time where n is the size of a single column. The sorting of the column function and the subsequent Fast Fourier transform both come with a cost of $O(n \log n)$. The aggregation of adjacent columns has the effect of multiplying n by a constant but does not change our big-O analysis. This is because windows and other periodic features do not scale up with the size of the scan. A larger city does not have larger windows. Rather the windows have a more or less constant range of widths and heights regardless of the size of the scan in which they occur. By a similar argument our exhaustive search for a square wave approximation of the column function is limited by the size of the period.

We conducted time tests of our algorithm running a variety of scans. The results are presented in Table 3. 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 optimized.

4.6.2 Limitations

Many buildings in modern cities exhibit vertical periodicity. However many buildings contain aperiodic fenestration. Our algorithm can detect the major planes of such structures but will not detect regularity. Our algorithm is not appropriate for low-rise residential structures. These structures are unlikely to contain vertically repeated features.

In low resolution scans, the local planes fit to each point become unreliable. This obscures the alternation between planar facade and architectural features. If the resolution is very low even major plane estimation will become error-prone.

5 Compressing Range Scans

Range scans of urban scenes are dense. Data-rich point clouds of millions of points are not practical for many applications. Navigational robots may have limited computational resources onboard and processing millions of points

Author's personal copy

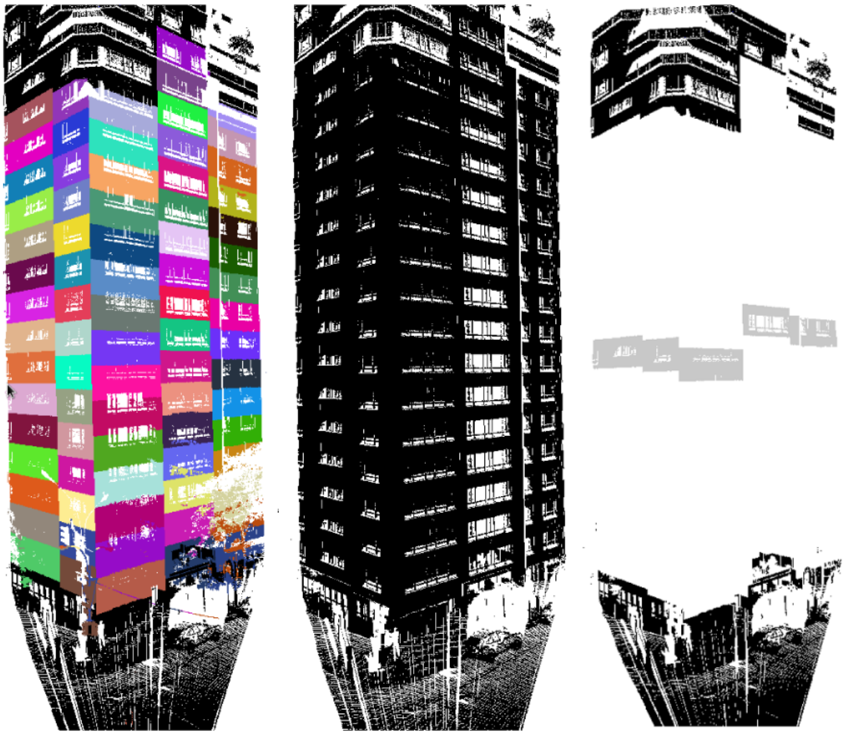


Fig. 14 The *left* image is a segmented scan. The *middle* image is a compressed version of the scan to the *left*. The *right* image shows the data reduction. The *grey points* are the representative regions. The compression ratio is 47 %, 513983 points have been removed out of a total of 1082958 points (Color figure online)

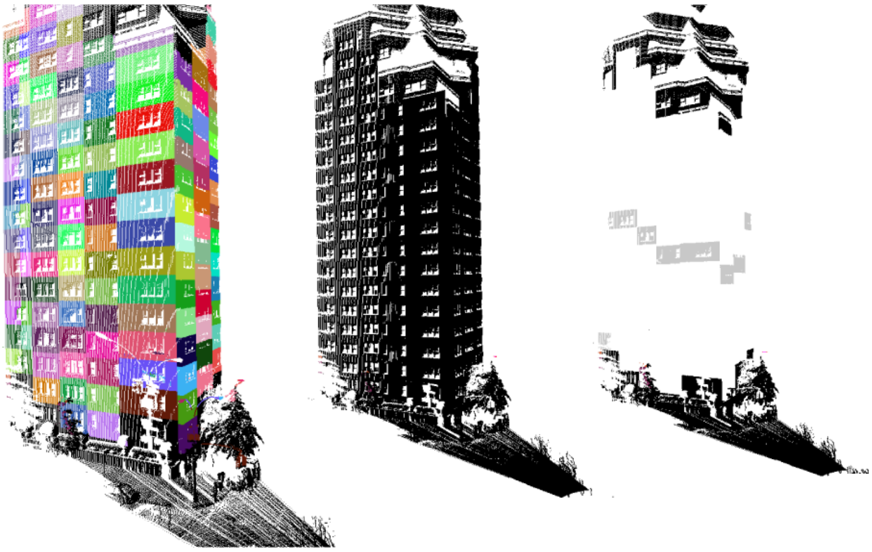


Fig. 15 The *left* image is a segmented scan. The *middle* image is a compressed version of the scan to the *left*. The *right* image shows the amount of reduction. The *grey points* are the representative regions. The compression ratio is 55 %, with 300628 points removed out of 546964 total points (Color figure online)

may exceed their capacity. Networked applications may not have the bandwidth necessary to transmit this huge volume of data. However, much of the urban scene is repetitive, especially fenestration. This redundancy can be reduced through compression.

5.1 Segmentation and the Representative Region

Compression begins by identifying a representative region which will be stored in full, and used as a replacement for the data we remove. The vertical period of the facade was discovered in Sect. 4. Moreover, in Sect. 4.4 horizontal sections of the facade were grouped and classified as planar or periodic. We now wish to segment the facade into like regions. One of these regions ought to be able to replace all the others vertically aligned with it, without compromising the character of the facade. Furthermore, the planar sections of the facade which are interspersed with the periodic sections of windows and balconies are also ripe for compression.

Segmentation proceeds by creating a rectangle around each periodic feature. The height of the rectangle is determined by the period detected in Sect. 4.3. The width of the

Author's personal copy

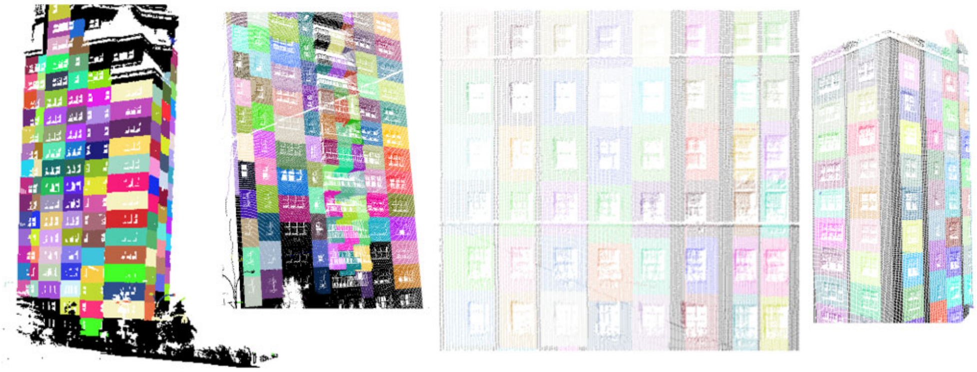


Fig. 16 These facades have been segmented by their periodicity. The Chebyshev metric identified the rectangles containing each periodic feature (Color figure online)

rectangle is determined by taking half the distance to the center of the neighboring periodic features, unless these features occur in the beginning or end of a facade in which case the rectangle is stretched to the facade's edge. We use the Chebyshev distance to extract this rectangle. If we used the Euclidean distance we would extract the circle around each window center.

Because windows typically have vertical and horizontal symmetry we wish to extract the feature's surrounding rectangle. The Chebyshev distance between two points α and β in $\mathbb{R}^3$ is defined as:

$$D_{Chebyshev}(\alpha,\beta):\max_{i\in(x,y,z)}\bigl(|\alpha_i-\beta_i|\bigr). \qquad (2)$$

We use this metric to determine the rectangular region with sides equal to the length of the dominant period. One of these regions is chosen as the representative, in our experiments we choose the median size feature from the sets aggregated in Sect. 4.4. At first glance it is tempting to choose the densest region as the representative. This strategy is often a mistake. Usually the region that is densest has a unique feature that distinguishes it from the rest of regions and disqualifies it as fair representative. Instead the region with median density is chosen.

To complete the compression we translate the chosen region vertically by the distance of the period. In this way the size of the column can be reduced by the number of windows found in the group of columns.

For a more ambitious reduction we assume that the building facade is planar. The vertical distance that came from the Fourier analysis can be coupled with the horizontal distance to the next adjacent group of periodic columns. The Chebyshev metric is altered to handle two different distances one vertical, the other horizontal. In the coordinate system given by the scanner the vertical axis corresponds to the z-axis of the points. However, the horizontal direction along the facade may be a combination of both the x and y coordinates. So, to find the horizontal distance to the next group of columns we take the Chebyshev distance in both the x and y directions.

$$D_{Vertical}(\alpha,\beta):\bigl(|\alpha_z-\beta_z|\bigr) \qquad (3)$$

$$D_{Horizontal}(\alpha,\beta):\max\bigl(|\alpha_x-\beta_x|,|\alpha_y-\beta_y|\bigr) \qquad (4)$$

Now much larger swaths of the building are included in the compression, see Fig. 16. The entire facade is compressed by a factor equal to the number of periodic elements it contains.

5.2 Comparison of Facade Segments

Some regions may contain extra information that should not be discarded when we compress with the median region. To avoid compressing regions with something that is not truly representative, the regions in question should be compared. Comparing 3D point clouds can be computationally intensive. To maintain the realtime flavor of the algorithm we choose a method based on Osada et al. (2002). Histograms are generated from a subset of points within each region. The histogram contains the Euclidean distance of randomly selected pairs of points from the subset. To prevent spurious matches, it is sensible to choose high curvature points (computed as in Cazals and Pouget 2005), or points whose eigenvalues computed during the local PCA were high, see Fig. 17. These points contain more of the unique character of each region rather than the homogeneous planarity. Comparing histograms is simple: one is subtracted from the other. Alternative histogram distance metrics exist as detailed in Pele and Werman (2010), however the $L1$ metric is simple and efficient for our purposes. A large absolute value in the difference between histograms indicates two distinct regions, both of which should be preserved in the compressed scan. Here we introduce a parameter γ as the threshold by which regions are declared similar. This parameter allows user's discretion over the degree to which their scans are compressed and the amount of data loss they will tolerate. Figure 18 shows the same scan with different levels of compression achieved by altering this threshold. The graphs in

Author's personal copy

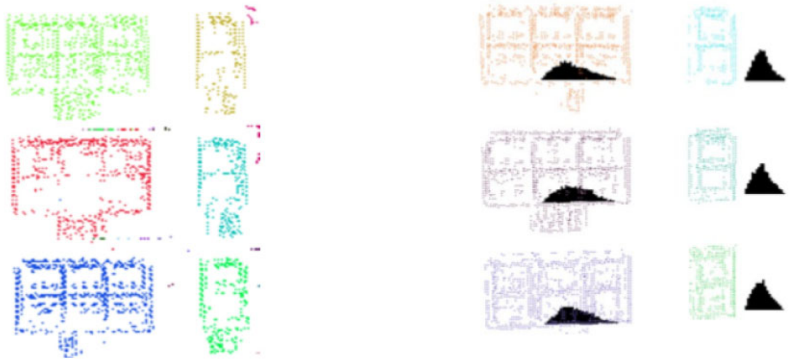


Fig. 17 Histograms are used as shape signatures to efficiently distinguish between distinct regions. *Left*: some regions of high curvature points. *Right*: same regions with their shape histograms. Similar regions have similar histograms

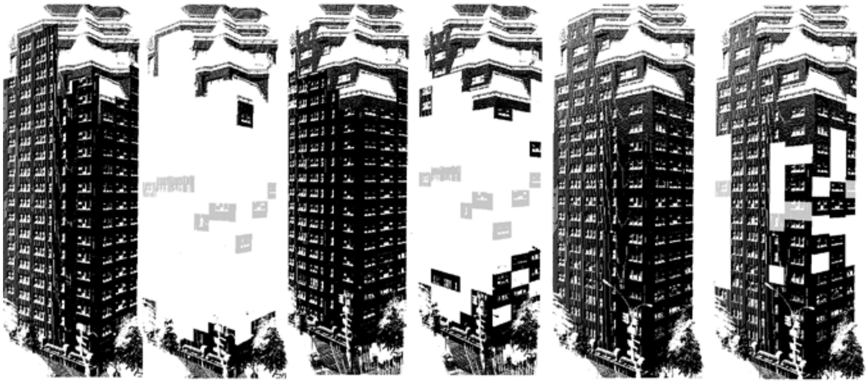


Fig. 18 The same scan with different levels of compression applied. The amount of compression is determined by changing the threshold which separates the histogram shape signatures, as described in Sect. 5.2. From *left* to *right* the compression rates are 57 %, 47 % and 13 %, and the histogram difference thresholds, γ are 2000, 700, and 500

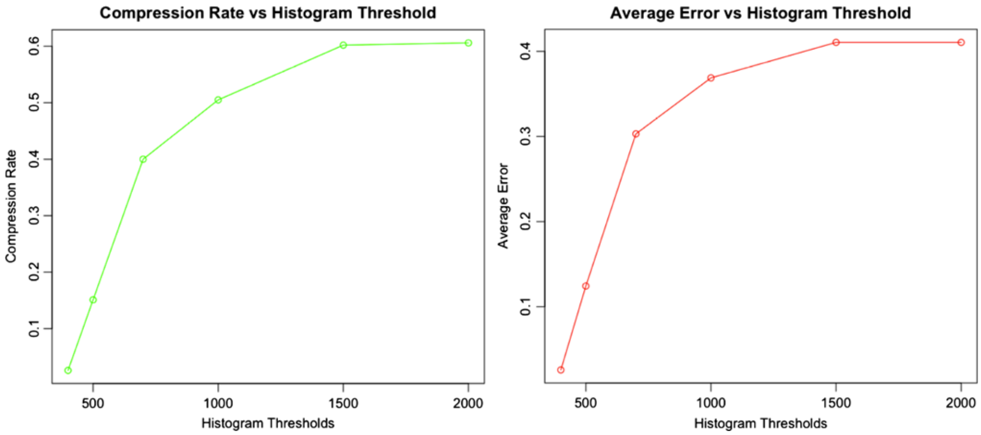


Fig. 19 At *left* the compression rates achieved at various histogram thresholds. At *right* the cumulative error at various histogram thresholds. The error is measured by averaging the Hausdorff distance (in meters) between the original regions and the representative that replaces them

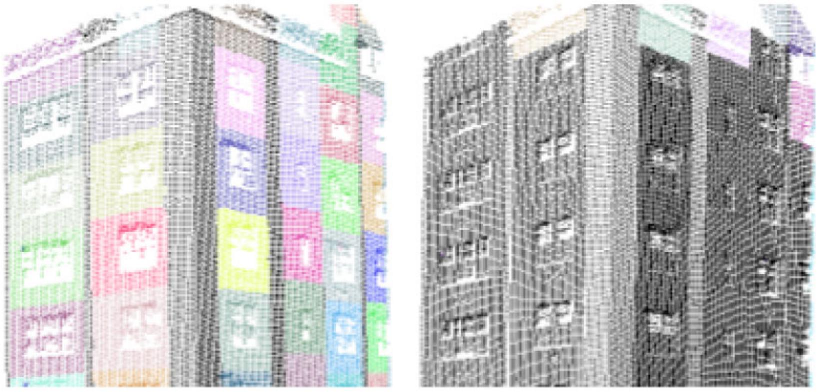


Fig. 20 Scans are compressed by repeating a representative region in place of the original data (Color figure online)

Author's personal copy

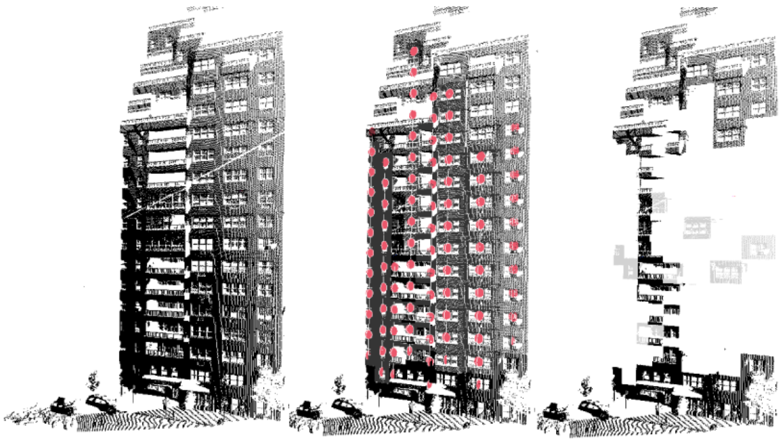


Fig. 21 The *left* image is the original scan. The *middle* images shows the feature centroids superimposed over the compressed scan. The *right* image shows all the points remaining after the compressed points are removed. The centroid of the architectural features discovered in Sect. 4.5 improve the compression results by aligning the groups by floor

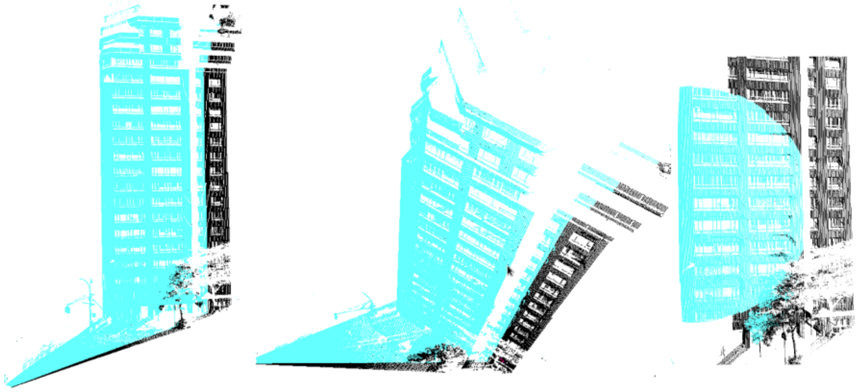


Fig. 22 Two views were synthesized from a single scan by applying a known transformation T. Our method then estimates a transformation T_*^{-1}, using the methods of Sect. 6

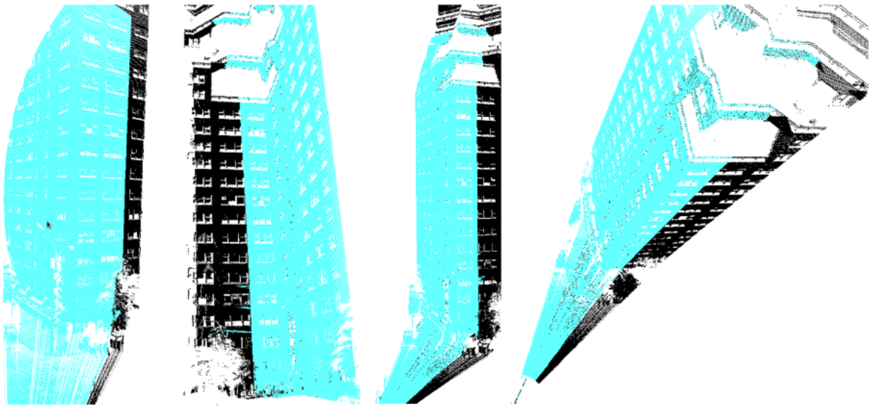


Fig. 23 A single group of regular features occurring in both scans is sufficient to achieve an accurate registration

Fig. 19 show the compression rates and error at different histogram thresholds. We measure the error by averaging the Hausdorff distance between the translated representative region and the region it replaces.

5.3 Replacing Regions with Translations of the Representative

Once the regions which are sufficiently similar to the representative are identified they can be removed. Scans with repetitive features removed are shown in Figs. 12, 13, 14, 15, 20 and 21. Working just with this reduced set it is still possible to recreate the look and feel of the initial scan. By translating the representative region vertically by multiples of the dominant period we are able to fill the missing data.

6 Registering in Real Time

Registration is a major topic of research in 3D computer vision community. Here we present a realtime reg-

Author's personal copy

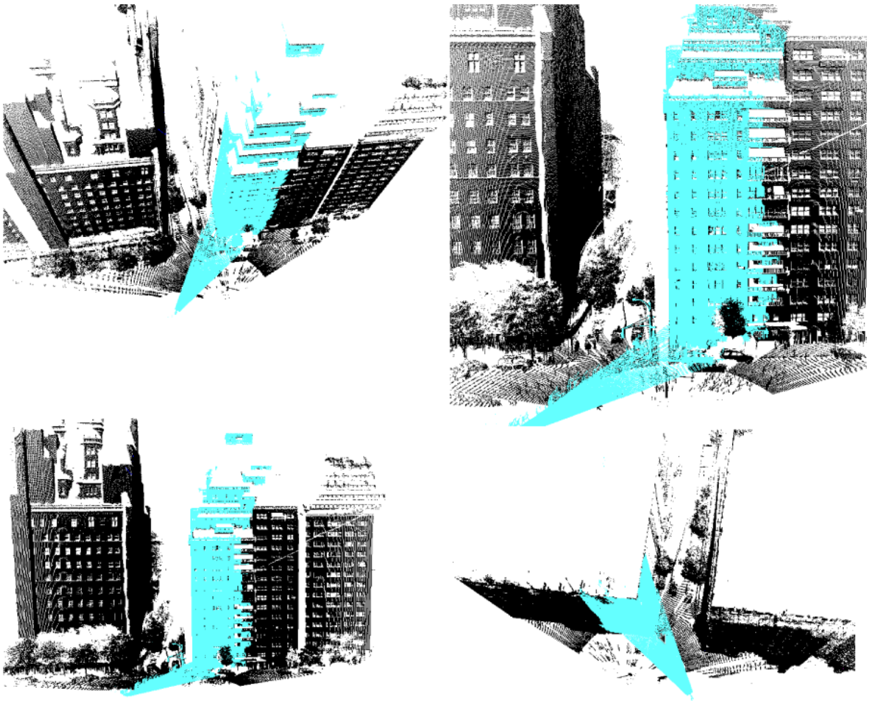


Fig. 24 Fours views of two distinct scans that have been registered following Sect. 6. The points from one scan are shown in blue and points from the other scan are shown in *black*

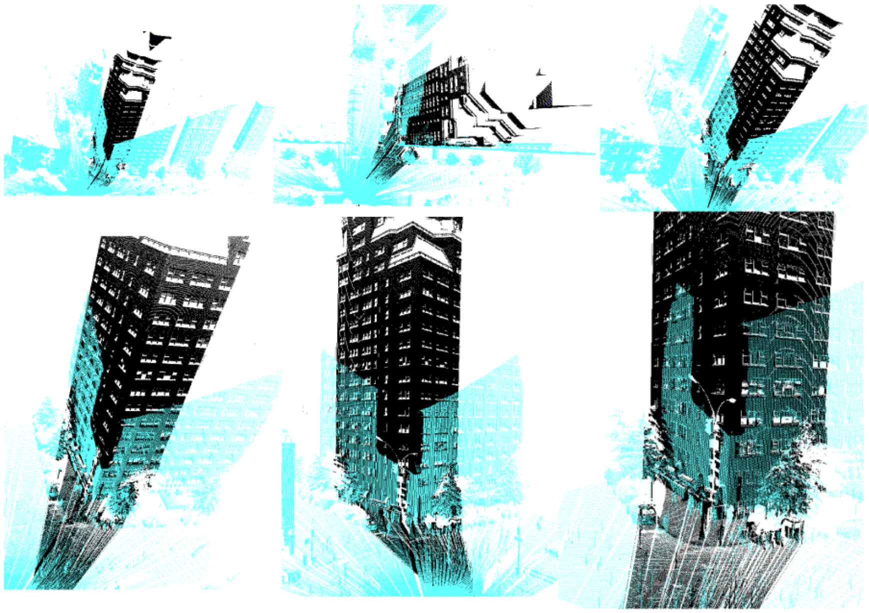


Fig. 25 Six views of two distinct scans one with *blue points* the other with *black points*. Despite a wide variance of resolution and incident angle of the laser, the transformation between the views can be computed online and automatically (Color figure online)

istration technique which builds upon the information acquired from the analysis of the periods of the windows and balconies. This strategy could be employed by two or more scanners, for simultaneous realtime multi-view scene acquisition. Our algorithm is extremely efficient and robust for scenes with repetitive patterns. Figures 22 and 23 show synthetic views, while Figs. 24 and 25 show two distinct real world scans registered using our algorithm.

6.1 Orthonormal Bases of Aggregated Columns

Solving for the rotation between two overlapping scans is essentially a problem of defining bases. We must find the orthonormal bases that describe the two sets to be registered. Let us call these two sets S_1 and S_2. For each point $x_1 \in S_1$ its corresponding point in S_2 called x_{rot} is given by:

$$x_{rot} = B_2 B_1^{-1} x_1, \tag{5}$$

Author's personal copy

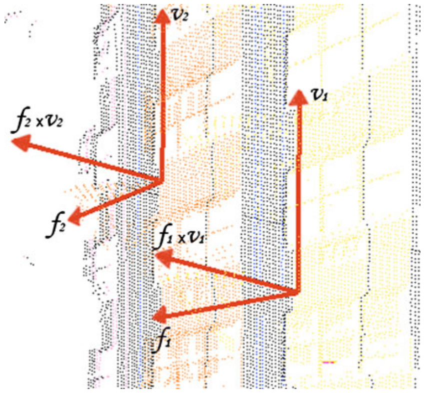


Fig. 26 Basis vectors for two window groups

where B_1 is the matrix composed of the orthonormal bases of S_1 and B_2 is a similar matrix for S_2. To find the orthonormal bases B_1 and B_2 we find the vertical vector v and the orthogonal facade vector f. The facade vector is established by taking the vector between any two points within the facade, called f_{start} and projecting this vector onto v. The facade vector is the vector between f_{start} and its projection onto v. By subtracting out the component of f_{start} in the direction of v we ensure that f is orthogonal to v.

$$f = f_{start} - \big((f_{start} \cdot v)v\big). \tag{6}$$

Both vectors f and v are normalized. Now their cross product yields the complete set of three basis vectors:

$$B_* = \begin{pmatrix} f \\ v \\ f \times v \end{pmatrix} \tag{7}$$

Figure 26 shows two sets of basis vectors and the repeated architectural features they were computed from.

6.2 Computing Corresponding Pairs

Once the rotation matrix $B_2 B_1^{-1}$ is known, solving for the translation between the scans is reduced to finding a pair of corresponding points. More precisely, we wish to find two locations in each scan of a single real world point. The translation between scans t is then:

$$t = x_2 - B_2 B_1^{-1} x_1, \tag{8}$$

where x_1 and x_2 are two data points that correspond to the same real world point.

There are several ways to compute x_1 and x_2. The probability that the laser will measure the exact same point in two different scans of the same scene is vanishingly small. Therefore choosing a pair of points and simply calculating the translation vector between them is bound to introduce some error. It is preferable to rely on higher level knowledge about the scene. To this end we propose two different methods for computing translations between different views. The methods are appropriate in different circumstances. The window width sequence method is best when the areas of overlap between the views are large and the resolution is high. The corner vector method is not as reliable as the window sequence but it requires less resolution and overlap.

6.2.1 *Window Width Sequences*

Registering two scans by a pair of architectural features should result in an alignment of all of the overlapping features. We compute the rotation matrix based on two sets of these features, the window groups from Sect. 4.4. If the mapping between feature groups is correct it should ensure that all the adjacent feature groups are also aligned. It is easy to check this by relying on their width. Expanding from the pair of feature groups we have aligned, we should find similar feature groups in all the remaining sets of adjacent features until the edge of the overlap.

6.2.2 *Corner Vector*

If the scan does not contain significant or high resolution overlap it may be difficult to distinguish this pattern of matching features because there might be only one set of features that we can align. In these circumstances we rely again on the macro data about major planes found in Sect. 4.1. Now we can use the ground plane estimate to compute the vertical coordinate of the translation vector.

For corresponding aggregated window column groups we try to match the first columns from each set of columns. This locks in two of the coordinates of the translation vector. To find the vertical translation we resort to our major plane detection from Sect. 4.1. The cross product of the facade and ground normals gives the direction of the line of intersection between the two planes (assuming they are not parallel):

$$i_d = n_g \times n_f. \tag{9}$$

Now this directional vector must be translated to align with the intersection of the major planes. For this translation vector we combine the z-coordinate of the ground plane and x and y coordinates of a point on the facade plane in the current scan line. Essentially dropping a line from the facade plane until it hits the ground.

6.3 Evaluation of Registration

Since the number of feature groups is relatively small it is not impractical to compute many possible transformations and then evaluate them all and select the best to register the scans. So, many transformations are generated and one will be selected as the best. We eliminate spurious transformations in a number of ways. First we ensure that the feature groups to be aligned have similar periods as detected in

Author's personal copy

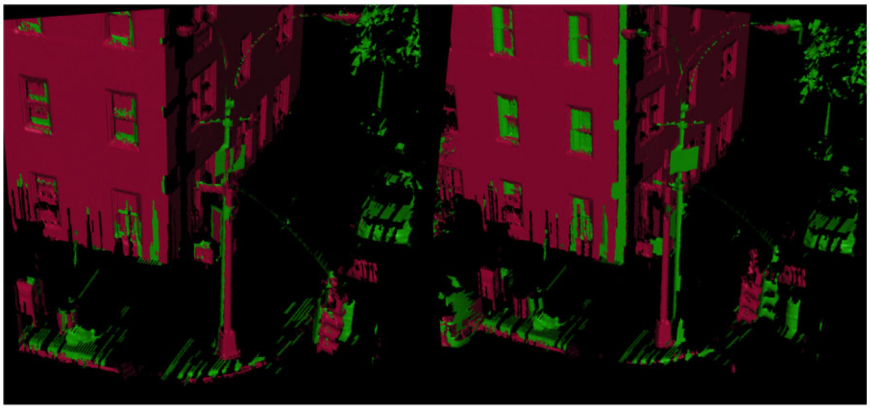


Fig. 27 The *left* image shows two views registered by our algorithm. The *right* images shows the registration after ICP is performed. Notice how the quality of the registration has been degraded by ICP. The lamppost, mailbox and trash can all exhibit significant error, which was not present in the real time registration on the *left*

Sect. 4.3. Furthermore we only compute transformations between window groups of similar widths. Then we evaluate the quality of the transformation by checking the distance between aligned columns once a transformation has been applied to one of them.

To further evaluate our registration algorithm we performed ICP on our results. Surprisingly, there was a notable degradation in the quality of the registration. After close analysis we concluded that the ICP algorithm tried to align regions like parked cars that did not actually correspond to the same real world structures. This experience illustrated another strength of our algorithm. Not only do we compute registration in realtime far faster than ICP, we base our registration on facade features which are lasting elements of the scene. ICP treats all close points equally, so scans performed on different days are susceptible to error when a new set of cars are parked on the street, for example. The average error returned by the ICP algorithm was 0.65 meters. ICP measures an error 0.914 meters before registration but much of this error comes from false correspondences and does not actually indicate a flaw in the registration. The scan before and after ICP is applied is shown in Fig. 27, clearly the registration before ICP is more accurate.

The transformation matrix output by our algorithm was compared against the transformation matrix computed from an interactive registration program where a user selects several corresponding points in the two scans. The interactive program then uses a least squares technique to find the transformation matrix between the scans. In our tests the per entry difference between the rotation matrices from our automatic approach and the interactive program never exceeded 0.0084 and averaged 0.0037. Alternative methods for comparing rotation matrices are presented in Huynh (2009). The translation vectors differed by an average of 0.12 meters.

7 Conclusion and Future Work

We have shown an online algorithm for detecting regular features and we have exploited this regularity to compress and register urban range scans. The meso-level information of the regular architectural fenestration provides tremendous insight into the scan. It will no doubt be shown to be applicable to a host of other applications in urban scene processing.

7.1 Smarter Scanners

One field ripe for research is integrating online algorithms of the kind described here into the hardware of the laser scanner. By processing the scan as it is acquired the offline computational resources are freed to pursue higher level recognition tasks.

Compression in particular could be helpful in an onboard application. A robot navigating a complex scene may simply not have the resources to store an entire point cloud of an urban scene. Integrating compression into the act of processing circumvents this need.

7.2 Parallel Range Scanning

Another area of future work will employ the registration algorithm presented here to allow simultaneous realtime registering of scans from multiple scanners. This application could prove critical in range finding sensor networks. If registration can occur as the scans are recorded each sensor could leverage all the global information acquired as it becomes available. In this way each sensor could enjoy the speed and robustness that parallelism affords without sacrificing the global knowledge of a serial scan.

References

Aiger, D., Mitra, N., & Cohen-Or, D. (2008). 4-points congruent sets for robust pairwise surface registration. In *SIGGRAPH* (Vol. 27).

Allen, P. K., Stamos, I., Troccoli, A., Smith, B., Leordeanu, M., & Murray, S. (2003). New methods for digital modeling of historic sites. *IEEE Computer Graphics and Applications*, *23*(6), 32–41.

Bellon, O. R. P., & Silva, L. (2002). New improvements to range image segmentation by edge detection. *IEEE Signal Processing Letters*, *9*(2), 43–45.

Author's personal copy

Berner, A., Bokeloh, M., Wand, M., Schilling, A., & Seidel, H. P. (2008). A graph-based approach to symmetry detection. In *Symposium on volume and point-based graphics* (pp. 1–8). Los Angeles: Eurographics Association.

Besl, P. J., & Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(2), 167–192.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2).

Bokeloh, M., Berner, A., Seidel, M. H. P., & Schilling, A. (2009). Symmetry detection using line features. *Computer Graphics Forum*, *28*, 607–706.

Cazals, F., & Pouget, M. (2005). Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, *22*, 121–146. doi:10.1006/cviu.2002.0963.

Chao, C., & Stamos, I. (2007). Range image segmentation for modeling and object detection in urban scenes. In *The 6th international conference on 3-D digital imaging and modeling*, Montreal, Canada.

Friedman, S., & Stamos, I. (2011). Real time detection of repeated structures in point clouds of urban scenes. In *The first conference of 3D imaging, modeling, processing, visualization and transmission* (pp. 220–227).

Huber, D. F., & Hebert, M. (2003). Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, *21*(7), 637–650.

Huynh, D. (2009). Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, *35*, 155–164.

Lee, S. C., & Nevatia, R. (2004). Extraction and integration of window in a 3D building model from ground view images. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 113–120). doi:10.1109/CVPR.2004.1315152.

Leica Geosystems (2012). http://hds.leica-geosystems.com/

Li, Y., Zheng, Q., Sharf, A., Cohen-Or, D., Chen, B., & Mitra, N. J. (2011). 2D–3D fusion for layer decomposition of urban facades. In *International conference of computer vision* (pp. 882–889).

Marshall, D., Lukacs, G., & Martin, R. (2001). Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(3), 304–314.

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

Mayer, H., & Reznik, S. (2007). Building facade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, *61*(6), 371–380.

Mitra, N. J., Pauly, M., Wand, M., & Ceylan, D. (2012). Symmetry in 3D geometry: extraction and applications. In *EUROGRAPHICS state-of-the-art report* (pp. 1–28).

Muller, P., Wonka, P., Haegler, S., Ulmer, A., & Gool, L. V. (2006). Procedural modeling of buildings. *Computer Graphics*, *25*(3), 614–623.

Muller, P., Zeng, G., Wonka, P., & Gool, L. V. (2007). Image-based procedural modeling of facades. *Computer Graphics*, *26*(3), 85.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., & Chen, B. (2010). Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics*, *29*(4), 93.

Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics*, *21*(4), 807–832.

Park, M., Brocklehurst, K., Collins, R., & Liu, Y. (2009). Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *31*(1). Special Issue on Probabilistic Graphical Models

Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., & Guibas, L. (2008). Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, *27*(3), 1–11.

Pele, O., & Werman, M. (2010). The quadratic-chi histogram distance family. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Lecture notes in computer science: Vol. 6312. Computer vision ECCV 2010* (pp. 749–762). Berlin: Springer.

Pulli, K., & Pietikinen, M. (1993). Range image segmentation based on decomposition of surface normals. In *Proceedings of the Scandinavian conference on image analysis* (Vol. 2, p. 893).

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *The 3rd international conference on 3D digital imaging and modeling* (pp. 145–152).

Shen, C. H., Huang, S. S., Fu, H., & Hu, S. M. (2011). Adaptive partitioning of urban facades. *ACM Transactions on Graphics*, *30*(6), 184.

Stamos, I., & Allen, P. K. (2002). Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, *88*(2), 94–118. doi:10.1006/cviu.2002.0963.

Stamos, I., & Leordeanu, M. (2003). Automated feature-based range registration of urban scenes of large scale. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 555–561).

Stamos, I., Yu, G., Wolberg, G., & Zokai, S. (2006). 3D modeling using planar segments and mesh elements. In *Intl. symposium on 3D data processing, visualization and transmission*, Chapel Hill.

Stamos, I., Liu, L., Chao, C., Wolberg, G., Yu, G., & Zokai, S. (2008). Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *International Journal of Computer Vision*, *78*(2–3), 237–260. Special Issue on Modeling and Representation of Large-Scale 3D Scenes

Stiny, G. (1982). Spatial relations and grammars. *Environment and Planning*, *9*, 313–314.

Teboul, O., Simon, L., Koutsourakis, P., & Paragios, N. (2010). Segmentation of building facades using procedural shape priors. In *IEEE conference on computer vision and pattern recognition* (pp. 3105–3112). doi:10.1109/CVPR.2010.5540068.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., & Paragios, N. (2011). Shape grammar parsing via reinforcement learning. In *IEEE conference on computer vision and pattern recognition* (pp. 2273–2280). doi:10.1109/CVPR.2011.5995319.

Triebel, R., Kersting, K., & Burgard, W. (2006). Robust 3D scan point classification using associative Markov networks. In *IEEE international conference on robotics and automation* (pp. 2603–2608).

Wami, M. A., & Batchelor, B. G. (1994). Edge-region-based segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(3), 314–319.

Wu, C., Frahm, J. M., & Pollefeys, M. (2010). Detecting large repetitive structures with salient boundaries. In *European conference on computer vision* (Vol. 6312, pp. 142–155).

Yu, Y., Ferencz, A., & Malik, J. Extracting objects from range and radiance images. *IEEE Transactions on Visualization and Computer Graphics*, 351–364 (2001).

Zhao, H., & Shibasaki, R. (2003). Reconstructing a textured CAD model of an urban environment using vehicle-borne laser range scanners and line cameras. *Machine Vision and Applications*, *14*(1), 35–41.