Non-local Scan Consolidation for 3D Urban Scenes

Qian Zheng[1] Andrei Sharf[1] Guowei Wan[1,2] Yangyan Li[1] Niloy J. Mitra[3] Daniel Cohen-Or[4] Baoquan Chen[1]

[1] SIAT, China [2] NUDT, China [3] IIT Delhi [4] Tel Aviv Univ.

Abstract

Recent advances in scanning technologies, in particular devices that extract depth through active sensing, allow fast scanning of urban scenes. Such rapid acquisition incurs imperfections: large regions remain missing, significant variation in sampling density is common, and the data is often corrupted with noise and outliers. However, buildings often exhibit large scale repetitions and self-similarities. Detecting, extracting, and utilizing such large scale repetitions provide powerful means to consolidate the imperfect data. Our key observation is that the same geometry, when scanned multiple times over reoccurrences of instances, allow application of a simple yet effective non-local filtering. The multiplicity of the geometry is fused together and projected to a *base-geometry* defined by clustering corresponding surfaces. Denoising is applied by separating the process into off-plane and in-plane phases. We show that the consolidation of the reoccurrences provides robust denoising and allow reliable completion of missing parts. We present evaluation results of the algorithm on several LiDAR scans of buildings of varying complexity and styles.

1 Introduction

Digital acquisition of objects has been an active research topic in recent years. This has been fostered by significant advances in scanning technologies, in particular devices that extract depth through active sensing. To faithfully capture a model, especially those large in extent, often an extensive acquisition process is required to guarantee a good coverage of the entire surface of the subject [Levoy et al. 2000]. However, due to time and accessibility limitations, such an elaborate acquisition setup is not always affordable for large scale urban landscapes and often the surface has to be recovered from rather imperfect scans, i.e., noisy, incomplete and corrupted with outliers.

Reconstruction of urban models is gaining increasing attention these days, motivated by ambitious applications that aim to build digital copies of real cities (e.g., Microsoft Virtual Earth 3D and Google Earth 3D). For such gigantic applications, rapid, robust and complete scanning is imperative. The dominant scanning technology for this purpose consists of LiDAR scanners mounted over airborne or street level vehicles, which scan buildings while the vehicles move at their normal driving speed. Although this yields coherent 3D points of scanned models, such scans are often noisy and incomplete (see Figure 1).

In this paper we focus on the enhancement and consolidation of imperfect scans of urban models. The main challenge is how to significantly improve the quality of the data starting from such noisy, non-uniform, and incomplete scans. Direct surface reconstruction from such poor quality of input data is inconceivable, and an unrealistic goal (see Figure 2). Luckily, models of urban landscapes exhibit a high degree of self-similarity and redundancy. We explicitly make use of this characteristic of urban scenes to enable plausible geometry recovery. The key observation is that the same geometry is scanned multiple times over reoccurrences of the repeated parts. The non-local multitude of geometry provides opportunities to denoise the data by applying a non-local filter and to complete missing parts using information from remote regions. The challenge lies in automatically determining which points to retain and which ones to prune out. Instead of making strong prior assumptions about the models and blindly recreating geometry using predefined procedural rules, we work directly with the scans and consolidate them, attempting to extract maximum information from the messy scans.

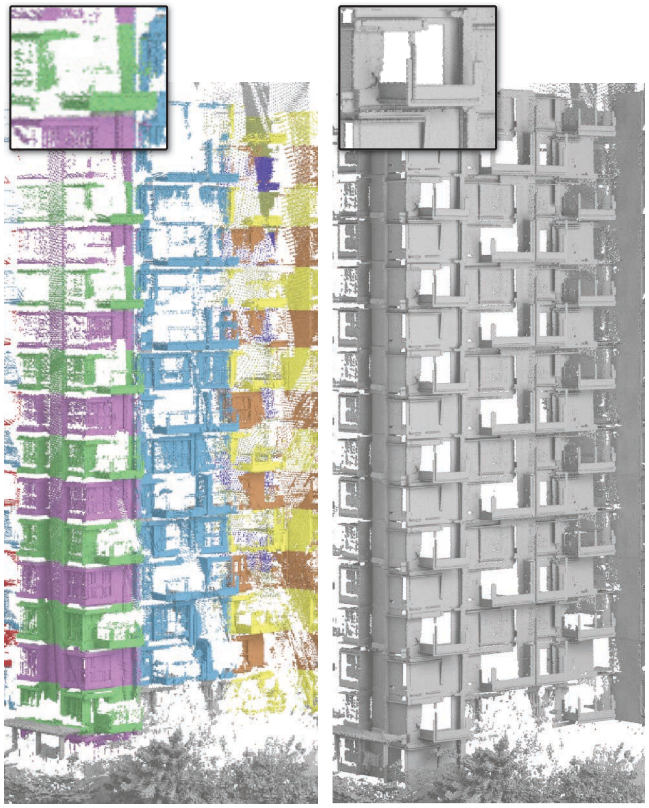

Figure 1: *Consolidating a LiDAR scan captured 3D building containing noise and missing regions. (Left) Repeated parts are detected and colored. (Right) Result of non-local filtering and consolidation of the repeated parts.*

Regularity and self-symmetry in urban buildings is not a chance occurrence, but is demonstrated universally across countries and cultures. Such large scale repetitions arise from manufacturing ease, build-ability, aesthetics, etc. Also because of functional requirements and constraints, buildings are mostly comprised of flat or near-planar faces. While in recent years many techniques have been developed to detect repeated parts in models [Debevec et al.

ACM Reference Format
Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N., Cohen-Or, D., Chen, B. 2010. Non-local Scan Consolidation for 3D Urban Scenes. *ACM Trans. Graph. 29*, 4, Article 94 (July 2010), 9 pages.
DOI = 10.1145/1778765.1778831 http://doi.acm.org/10.1145/1778765.1778831.

∈ }

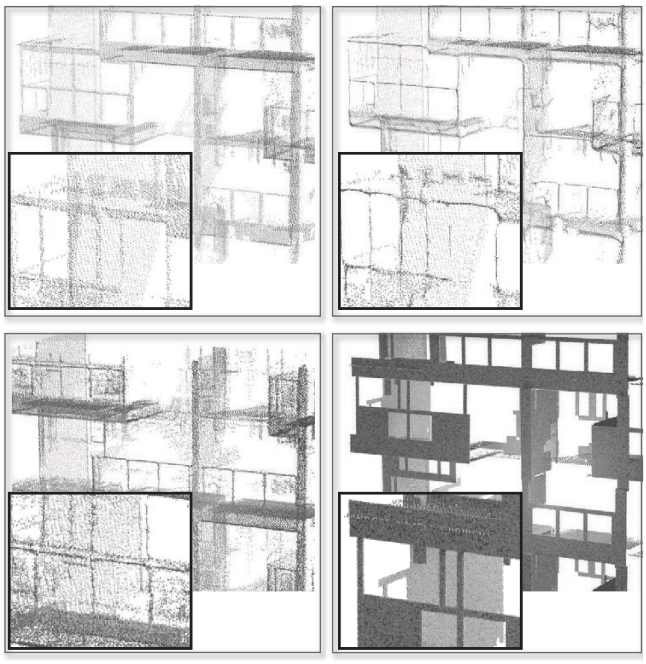

Figure 2: *Comparison with state-of-the-art point consolidation method. The input (top-left), result using WLOP [Huang et al. 2009] (top-right), result using WLOP on the union of detected repetitions aligned to one instance (bottom-left), and result using our consolidation method (bottom-right). Respective zooms for one balcony are shown.*

1996; Hays et al. 2006; Mitra et al. 2006; Korah and Rasmussen 2007; Pauly et al. 2008; Musialski et al. 2009], most of these works do not investigate how to best use the strong regularity present in 3D scans, specifically in urban buildings. Moreover, most of the techniques are applied in image space by analyzing photometric 2D images sampled over an underlying regular domain. Only few attempts have been made towards detection of regularity directly on 3D geometry (e.g., [Pauly et al. 2008]).

We investigate the central question of given a set of imperfectly scanned repeated parts, how to enhance the quality of each of the reoccurrences. We show that consolidation of the registered reoccurrences using non-local filtering provide superior robust denoising and allows reliable completion of missing parts. Most building are designed and generated in a procedural and modular fashion [Merritt and Ricketts 2001]. However, instead of learning parameters for a codebook of rules, we attempt to learn the repetitions pattern directly from the data, using high level user guidance when data quality is poor. We split the denoising into two steps: *off-plane* and *in-plane* denoising based on the registration of corresponding planes and lines, respectively. By partitioning the algorithm into two distinct phases, we significantly reduce the cross contamination of data across planes. Buildings are largely made of dominant planes or low complexity faces, e.g., cylinders, separated by large angles allowing effective off-plane denoising. While a naive reconstruction from scanned building data produces poor results, we demonstrate that working with higher order primitives like lines and planes, and explicitly recovering their mutual relations and regularity allow us to create superior results, which cannot be achieved by local methods. We applied our method to a large selection of urban models containing varying amount of repetitions. We demonstrate our scan consolidation results on data sets acquired rapidly by a moving device producing rather sparse and low quality data.

Contributions. We present a scan-consolidation framework that:

uses non-local filtering for scanned data filtering across multiple recordings of repeated geometry in urban buildings,

operates in the parametric space using high-order primitives for reliable processing, and

incorporates a statistical error metric minimization scheme to robustly handle noise, outliers, and deal with missing data.

2 Related Work

Non-local filtering. Data acquired by scanning devices always contains some degree of noise and outliers. Denoising has been extensively studied in image processing, e.g., [Perona and Malik 1990; Rudin et al. 1992; Lindenbaum et al. 1994; Tomasi and Manduchi 1998], and has also been applied for surface denoising, e.g., [Fleishman et al. 2003; Jones et al. 2003; Oztireli et al. 2009]. A different approach for removing noise is by non-local filtering recently proposed by Baudes and colleagues [2005; 2008]. The basic idea is to average similar neighborhoods irrespective of their spatial proximity. Dabov et al. [2007] further utilize non-local similarity groups and filter an image in 3D transformation domain. The idea of non-local filtering has also been investigated for surfaces [Yoshizawa et al. 2006]. Such an approach is more resilient to noise than local filters provided that the data exhibits sufficient self-similarity and redundancy. Our work takes the non-local idea a step further by explicitly using the pronounced self-similarity commonly found in most urban building models.

Our work is closely related to the work of Musialski et al. [2009] where the regular structures commonly present in (orthographic) images of building facade are exploited using a non-local filter. Prevalent symmetries, specifically dominant translational and reflective ones, are searched for and a lattice of repeated patterns detected. Subsequently, information is propagated by a diffusion process based on similar symmetric pixels. The propagation allows rejecting outliers like street signs, cables, lights, etc., and repairing the facade image. Unlike their work, we operate directly on 3D point cloud data and consolidate the raw scans using the extensive underlying repetitions.

Fitting-based reconstruction. The primary goals of our work are denoising, completion, and in general consolidation of the scanned data, as a pre-process for any surface reconstruction (see also [Huang et al. 2009]). Such a goal of recovering surface from poor, noisy, insufficiently sampled point clouds is ill-posed and unrealistic. Previous attempts regularized the problem by using priors in the form of primitive shapes and using the data for fitting and parameter finding. Such an approach is common in CAD, where the process can be regarded as reverse engineering since often models are defined by a combination of basic geometric primitives.

Gal et al. [2007] match local geometric priors to local neighborhoods of 3D scans at multiple scales using partial shape matching. As a consequence, the scan is augmented with noise-free data, high-quality normal information, and sharp feature markups. The augmented data can be considered as a consolidation of the scanned data allowing a reliable reconstruction. Recently, Schnabel et al. [2009] present a hole filling algorithm that is guided by primitive shapes that have been detected in the input point cloud. Surrounding primitive structures around the holes are extended to complete the holes and synthesize edges and corners from the primitives' intersections. The problem is formulated as a surface energy minimization solved using graphcut. However, both methods assume moderate quality of data and quickly degenerate on poor quality inputs, as in our case.

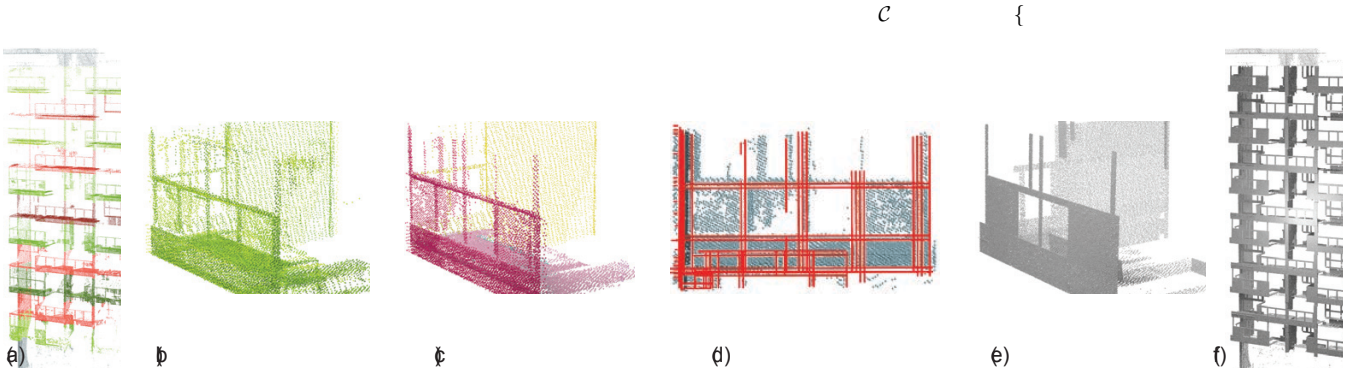

Figure 3: *Consolidation pipeline. (Left-to-right) (a) Repeated instances in an input scan are detected, and each instance is partitioned into groups of points using RANSAC. Here the repeated instances are partitioned into planes with associated confidences. (b) The instances are brought together by factoring out their repetition, followed by ICP registration. (c) Planes across multiple instances are clustered, and each group represented by its weighted median. Points are then projected onto the planes for off-plane denoising. (d) In each plane, edge lines are detected, weighted by their confidence, and a set of representative weighted median lines selected. (e) The line segments induce a partition on the plane, and regions with low confidence are removed. (f) Consolidated instances, with both off- and in-plane noise reduced, are projected back to obtain the final result.*

Repetitions detection. Our method is heavily based on the existence and detection of repeated parts in architectural models. Detection of regular or near-regular patterns has been widely studied and analyzed for images (e.g., [Liu et al. 2004a; Liu et al. 2004b; Park et al. 2009], with some work specifically focusing on analyzing facades. The early work of Schaffalitzky and Zisserman [1999] automatically detects imaged elements that repeat on a plane typically occurring in urban facades. They compute features and use RANSAC to detect repetitions under projective transformations. Yu et al. [2001] extract full objects from scanned data by an iterative clustering process similar to us followed by a coarse-to-fine segmentation. Wang et al. [2002] computes a facade texture map by removing occlusions and noise from multiple images associated with camera model and a coarse 3D geometric proxy for buildings. They compute a weighted-average consensus facade from the images aligned with the 3D model, then deblur and detect windows by edge detection and rectangular fitting. Missing data is recovered by assuming periodicity in horizontal and vertical directions on the facade.

Muller et al. [2007] perform autocorrelation analysis on 2D facade images to generate a 3D procedural model counterpart. To detect the facade structure, they subdivide and cluster the image into repetitive tiles under translational symmetry. Models are inferred by comparing against a predefined library of 3D ones, projecting them onto 2D, and matching in the image space. The algorithm performs extensive analysis on 2D image edges, with the user required to manually adjust the depth of the fitted models in the scene. Recently, Korah and Ransmussen [2008] address the problem of automatically detecting 2D grid structures such as windows on building facades images taken in urban settings. They work under the assumption that the background is strongly structured, which allows searching for near-regular textures in the image and the detection of rectangular structures in a grid-like pattern.

In the area of shape analysis, Pauly et al. [2008] present a framework for discovering repetitive structures in 3D geometry. Structure discovery is performed by analyzing the space of pairwise similarity transformations of local surface patches. They observe that in many cases the spatial coherence of repetitive structures leads to accumulative patterns in the corresponding transformation space. Appropriate functions are presented to map similarity transformations onto planar uniform grids. To detect regular structures, a grid is fitted to the clusters using a global optimization method. Bokeloh et al. [2009] observe that line features are more stable than point features, and propose an iterative closest line algorithm to match line features. Repeated elements are detected using a region growing starting from the constellations of the matched line features. These state-of-the-art techniques focus on detecting repeating elements in 3D models, but do not investigate means to use the detected structures for extensive data improvement or completion.

Recently, Xiao et al. [2009] proposed 3D facade reconstruction from street view images using a multi-view semantic segmentation along with inverse patch-based orthographic composition and structure analysis to create compelling results using strong priors of building regularity. In contrast, we make use of the self-similarity present in typical buildings and street-view scanned data, to perform non-local filtering and scan repair, resulting in significantly improved reconstructions (see Figure 1). Unlike previous work in surface completion and reconstruction [Pauly et al. 2005], the input to our data is large volumes of poor non-uniformly scanned data, on which most existing methods cannot be applied. By gathering and consolidating from many repeated instances, we show that significantly better, though not perfect, results are achievable. Note that comparison with ground truth is based on visual validation in absence of a better reference data for comparison.

3 Overview

In this work, we investigate means to explore, detect, and use large scale repetitions, regular or unstructured, for consolidating noisy and incomplete scans acquired using state-of-the-art LiDAR scanners. The acquired data comes in the form of point clouds, and lack any segmentation or high-level structural information. Such data quality makes it challenging to detect repetitions, where we simultaneously look for repeated elements and also infer how they are repeated. While recent research efforts [Müller et al. 2007; Pauly et al. 2008] have demonstrated success when working with images or scans as inputs, they either make strong assumptions about the pattern of repetitions and data quality, or may fail when the number of repeated instance are not large.

Our consolidation framework works in the following key stages:

1. Using user guidance repeated facade elements are extracted as *instances*.
2. Each instance is segmented into a set of planes.
3. The instances are registered to a consistent coordinate system, and corresponding planes across instances are identified using parameter space clustering.

4. Representative planes are selected for groups of clustered ones, and *off-plane* denoising performed.
5. Using a similar scheme in 2D, occupied polygonal regions are identified in respective planes in the *in-plane* denoising phase.

Finally, the consolidated instances are propagated back to the original positions to get the consolidated building scans. Except for the repeated instance extraction that requires user interaction, the other steps in the pipeline are fully automatic.

Since steps 2 – 5 for data consolidation strongly depends on the successful repetition detection, later we present a simple and general strategy for detecting and extracting repeated instances in raw scans. Assume that we detect $\mathcal{I} = \{I_1, \ldots, I_n\}$ as repeated instances of some point set geometry, along with the transformations that relate them. Say T_{ij} denotes the transformation when applied to instance I_i aligns it to I_j, i.e., $T_{ij}(I_i) \rightarrow I_j$. As multiple such repeated instance sets are typical in our data sets, our consolidation procedure is applied simultaneously to all of them. For each group, we bring their elements into a consistent coordinate frame, i.e., $\mathcal{I}' = \{I'_1, I'_2, \ldots, I'_n\} = \{I_1, T_{21}(I_2) \ldots, T_{n1}(I_n)\}$.

Incompleteness and sparsity of data due to scanning can be assumed to be random and appear at different places across instances. Once aligned, missing parts in instances can be consolidated using better data from across instances. However, we demonstrate that a naïve application of this philosophy leads to only marginal improvement. Our scans, originating from urban buildings, largely consist of flat faces. We demonstrate that making explicit use of this information as a prior, and working on primitives such as planes, and their subsequently in-planes, lead to significantly better results. The key idea is to denoise by clustering corresponding parts and projecting the points of corresponding parts to their cluster representative. We make the method robust to outliers and varying sampling density by incorporating locally adaptive confidence weights.

Building facades are typically made of flat faces, and lack in enough geometric details to aid in registration or alignment [Aiger et al. 2008]. Further, details when present, are often missed by LiDAR like scanners due to their low scanning resolution. We start by working with such planar faces that can be reliably estimated. First, each instance I_i is segmented into planes $\mathcal{P}_i = \{P_1^i, \ldots\}$, and each point of the instance associated with one of the planes (see Figure 3a). The segmentation is performed using a RANSAC approach, and planes and their coupled points are assigned weights based on the confidence in the plane. Note that $|\mathcal{P}_i|$ may vary across instances due to large chunks of missing data.

To establish correspondence between the planes of $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ we take an indirect approach. First, the instances in $\mathcal{I}'$ are locally aligned and registered using iterative closest point (ICP) algorithm [Besl and McKay 1992], while using I_1 as the reference. To make this alignment step robust we weight each point with respect to the confidence of its plane. Once the instances are aligned (see Figure 3b), the constituent planes are simply grouped based on proximity in the (plane) parameter space. Finally, for each plane cluster we choose a representative. The choice for representative directly affects the final result. We show that the obvious choices are ill-suited for our purpose, and a careful weighted median in the parameter space gives good performance (see Figure 4). Then all the points are projected onto their representative plane to remove *off-plane* noise (see Figure 3c). Next, we apply a similar procedure to the *in-plane* points, where line segments take the role of planes. In this in-plane phase, we also remove points that reside in sparse regions of the union point set, which typically arise due to presence of transparent elements such as glass pane windows (see Figure 3d). Finally, the point sets are consolidated and the data propagated across instances, to get a consolidated point set (see Figure 3e). The consolidated data can then be used for reconstruction or prior-driven procedural synthesis. Next we provide further details about the individual steps of our algorithm.

4 Algorithm

Plane estimation. In the first step of the algorithm we use RANSAC to segment data corresponding to each instance I_i into planes, and associate each point with its plane normal. We reject planes that are below a threshold in RANSAC voting (50 votes in our experiments). To robustly handle noise and variable sampling density, we compute a confidence value for each plane. For an estimated plane P_i let the projected points be denoted by $\{\mathbf{p}_1, \mathbf{p}_2, \ldots\}$. The local density d_j around each point $\mathbf{p}_j$ is approximated by measuring the radius of its k-nearest neighbor (k=10 in our implementation). Then the confidence weight $w(P_i)$ in the plane is a combination of three factors: (i) its area $\sigma(P_i)$, (ii) its homogeneity, i.e., the variance of the point density across the plane $\eta(P_i)$, and (iii) its anisotropy $\phi(P_i)$, i.e., the aspect ratio between the two eigenvalues of the covariance matrix of the corresponding projected point set. The combined expression for confidence of plane P_i is given as,

$$w(P_i) = \sigma(P_i) \cdot \eta(P_i) \cdot \phi(P_i). \quad (1)$$

ICP registration. Next, we apply a weighted ICP based alignment of the planes, where higher weights are given to planes with higher confidence. In this step we align multiple instances using pairwise alignments, keeping one selected plane fixed as the base. Since such a series of alignments is order-dependent, we select the instance with the largest sum of confidence as base. In each ICP iteration, we subsample the point cloud and associate each point with its nearest point in the base based on the L_2 deviation of their normal vectors. Figure 3b shows the overlaid results after the ICP step.

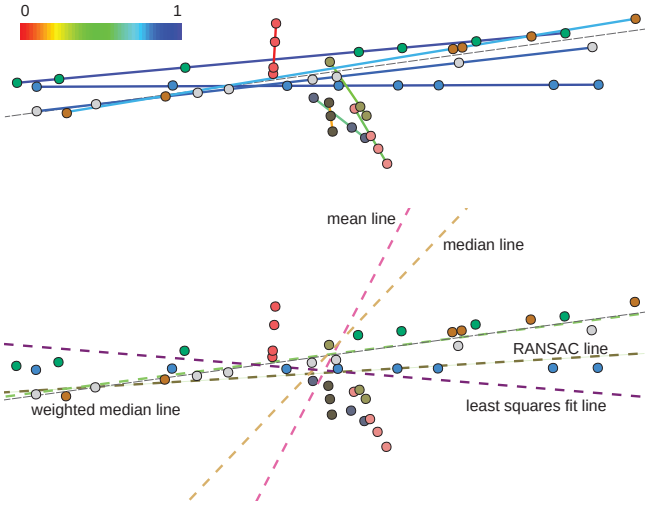

Figure 4: *Illustration of cluster representative. (Top) A cluster of lines and their associated points, generated by sampling from a base line (in dotted gray) with noise added in the parameter space. The lines are assigned colors based on computed confidence computed based on respective point distributions. (Bottom) The least squares fit line to the complete set of points is shown in purple. The RANSAC line is shown in olive green. The mean, median and weight-median lines, computed in the parameter space, are illustrated in pink, yellow, green, respectively. The weighted median line takes into account the length of the respective line segments, and are hence robust to outlier segments.*

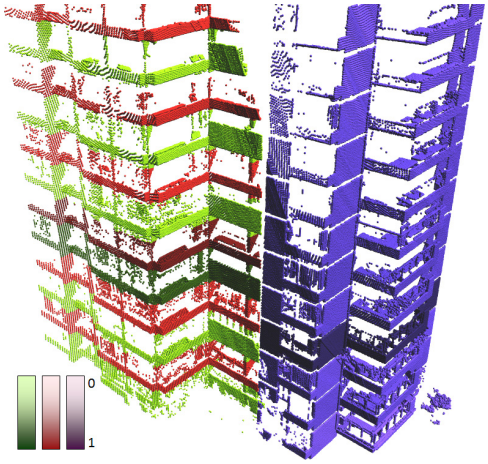

Figure 5: *Within each repetition, we weight planes by their confidence values. In this figure, we indicate grouping using shading (green, red, purple), and confidence in each group using luminosity. Darker shades denote higher confidence. Note due to oblique angles, the data is noisier and sparser at higher levels.*

Instead of greedily fixing a base, one can employ simultaneous registration techniques [Biber and Strasser 2006]. However, we found such a procedure slow and unsuitable for handling very large number of planes, as in our case. Although the results are not perfectly aligned, they are close enough to perform a reliable clustering of the planes, which we describe next.

Clustering. Planes that are similar are clustered together, with similarity measured using a L_2 norm between their parametric coefficients. We represent each plane P presented in the form $\mathbf{n} \cdot \mathbf{p} + d = 0$, with $\|\mathbf{n}\| = 1$ and $\mathbf{p} \in P$, using a three tuple $(\mathbf{n}_x, \mathbf{n}_y, d)$, i.e., as a point in 3D. Distance between planes P_i, P_j is measured using $\|P_i - P_j\|_2$. While conceptually this is simply clustering in a 3D space, we give more weight to the dominant planes obtained from multiple repetitions. This is an important detail that allows to robustly handle point sets with missing data, or those arising from thin unstable planes.

For reliability, we deliberately order the clustering process, and apply it incrementally. We insert the planes from all the instances into a priority queue ordered by their respective confidence weights. We then create one cluster at a time, guided by the priority order. A plane that is inserted into a cluster is removed from the priority queue. The clustering radius is a user defined parameter and depends on the data quality. In this procedure, the dominant planes get priority and are created early on. We discard small clusters below a threshold size of $\max(|\mathcal{I}|/4, 2)$. Thus, for a planar component to appear in the final consolidation, it has to be captured well at least in two instances.

Cluster representative. For each cluster of planes $\mathcal{C} = \{P^1, \ldots\}$ we need to select a representative P_c. Subsequently, we project all the points associated with the cluster onto the representative plane. The resulting point set is now free of off-surface (see Figure 3c).

The many points in a cluster come from different instances and form a *thick* point cloud. A natural choice for representative plane seem to be the least squares fit to the thick point set. However, our key observation here is that it is more effective not to ignore the point source, but rather to compute the representative plane of the cluster in a parametric plane space.

We represent each plane by its coefficients in the parametric form using $(\mathbf{n}_x, \mathbf{n}_y, d)$. Thus, we operate in a 3D space, seeking for a proper representative plane P_c in 3D. A simple approach is to take the average of of all point in that 3D plane space, or the L_1-median similar to the proposal by Lipman et al. [2007]. However, in the presence of strong outliers, the mean and even the L_1-median can yield erroneous planes. This is demonstrated in a 2D example in Figure 4. On the top, the lines and their associated points are depicted. On the bottom, we see the position of the median and L_1-median lines. As we can see their orientation does not agree with the major lines in the cluster, since they are affected by the outliers.

For a given cluster $\mathcal{C}$, we formulate the weighted L_1-median minimization as:

$$P_c^{L_1} = \frac{\sum_{i \in \mathcal{C}} P^i \cdot w(P^i) \cdot \theta(\|P^i - P_c^0\|)}{\sum_{i \in \mathcal{C}} w(P^i) \cdot \theta(\|P^i - P_c^0\|)}. \qquad (2)$$

This defines a reweighted iterative solution using a fast-decaying weight function $\theta(r) = \exp(-r^2/(h/4)^2)$. The support radius h is set in our algorithm to the cluster size. Outlier planes are characterized by being rather sparse or of rather small area, i.e., of low confidence. Thus, by taking the weighted L_1-median as the representative plane, where the weights are defined by our plane density measure (Equation 1), we down-weigh the outliers to get a good representative.

Note the location of the weighted median plane in Figure 4. It should be emphasized that by construction our clusters contain only planes with similar orientations and outlier planes are unlikely to have such a strong affect as illustrated in Figure 4. Nevertheless, the weighted median yields a reliable representative, and given that the confidence value are readily available, the computation is simple, efficient, and effective.

In-plane denoising. To consolidate the in-plane lines in each of the representative planes, we perform a similar clustering process, but in 2D (see Figure 6). Using the detected prominent in-plane lines we partition each plane into regions. The density of the points in each region is used to classify the regions into inliers and outliers. Inliers regions are then upsampled, while points belonging to outlier regions are discarded. Note that analyzing the in-plane data and detecting lines is naturally supported by much less data, posing an additional difficulty in their correct consolidation and outlier removal. In the following, we provide some details.

For each plane P_i in each instance, we separately perform the following: We first detect all possible boundary points by examining the local neighborhood distribution of each point (we use a neighborhood of 20 points). Let the eigen-values of the covariance matrix of neighborhood of point $\mathbf{p}_i$ be λ_0^i, λ_1^i in increasing order. Then we mark point $\mathbf{p}_i$ as a boundary point, if $\lambda_0^i/\lambda_1^i < \eta$, with $\eta = 0.2$ in our examples.

Next we perform a RANSAC line detection over all marked boundary points and remove those with few vote (≤ 3). Each line, being defined by a subset of the points, are further partitioned into segments by removing large unsampled sections along the line. For each line segment l_i we define the line confidence $w(l_i)$ based on its local point density and length.

We perform a similar incremental clustering algorithm as in the case of planes, starting with the dominant, high confidence lines and clustering based on metric distance in a 3D line parameter space $\{\mathbf{m}_i^x, \mathbf{m}_i^y, \mathbf{n}_i^x\}$ denoting line segment centers $\mathbf{m}_i$ and orientations $\mathbf{n}_i$, respectively. We always cluster lines from corresponding planes. For each cluster, the weighted L_1-median is used to as the the representative line. We remove the clusters with small number of elements, and regularize the remaining representative

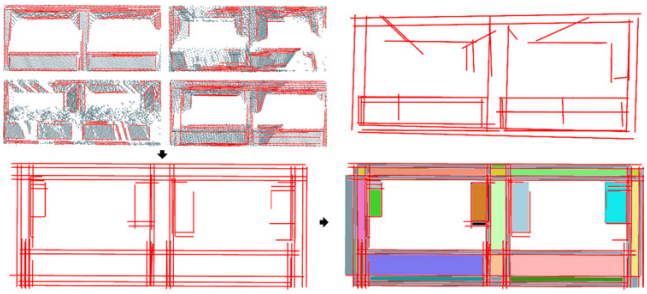

Figure 6: *(Top-left) Four repeated instances with corresponding in-plane detected lines segments. Due to sparsity of data and high amount of outlier, simple consolidation of line segments lead to mediocre results (top-right). However, consolidation with weighted median filter coupled with an orthogonality regularization produces significantly improved results (bottom-left), which is used to partition the plane into inlier-polygons (bottom-right). Note that the lines are* not *constrained to be axis aligned.*

line segments using a filter that gives preference to orthogonal arrangement of lines. The resulting segments induce a partition of the plane, and based on the number of points (per unit area) in each polygon we classify them as inliers or outliers (see Figure 6).

Extracting repeated parts. Detecting, extracting, and exploiting such large scale repetitions provide a mean to consolidate the data exploiting the underlying redundancy. Repetitions are detected using local descriptors. Two parts are considered to be similar if they contain similar configurations of local descriptors. The problem is challenging as we have to detect similarity across parts with only partial matching, with small repeated subparts (see Section 3). The detection is an integral part of our problem since we assume that large missing parts require consolidation exploiting underlying repetitions. After an offline pre-processing phase, we allow fast user generated query based search for identifying similar sections of the scans. False positives appear at this stage, which we prune out in a validation stage. Remaining outliers marginally affect the final result since later we employ robust statistical tools for data consolidation.

We start by computing a large set of local descriptor D_i across the point cloud that is first embedded inside a 3D grid with each voxel being 0.1m along each dimension. Each descriptor is a cube consisting of $4 \times 4 \times 4$ binary voxels, which are concatenated and represented as a binary vector. A voxel is set to '1' if its density of points is large enough (more than 50 points per cell in our examples), and to '0' otherwise. Two descriptors D_i and D_j are said to be similar if their bitwise AND exceeds a prescribed value ς, i.e., D_i AND $D_j > \varsigma$. Similar descriptors are clustered into types, and those with significant cluster size retained.

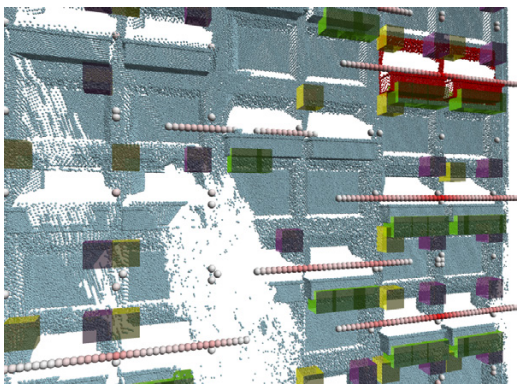

Figure 7: *The user selection (marked in red), and the resulting clusters and votes. Here, three clusters, marked by boxes of three different colors, were detected. The red spheres illustrate the accumulation of votes per cluster repetitions, with the saturation of the red reflecting the amount of votes.*

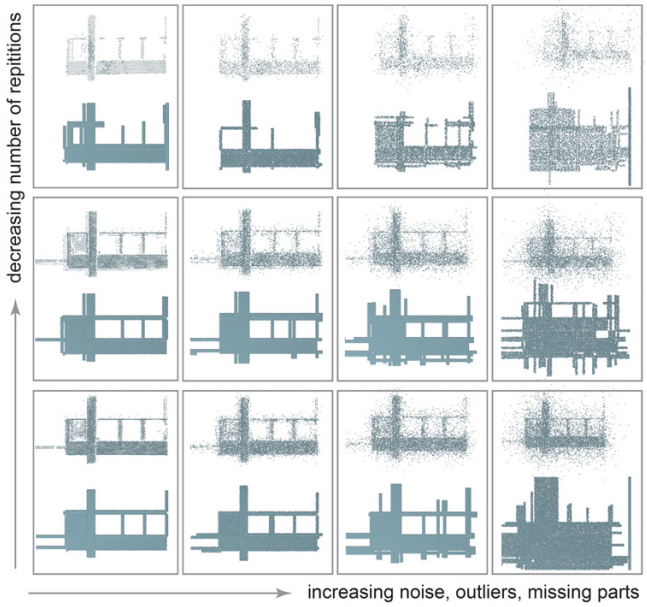

Figure 8: *Performance of in-plane denoising with varying number of repetitions (5, 4, 3 along the vertical direction) and increasing amount of noise and outliers (uniform random noise* 1.25%*,* 2.5%*, and* 5% *with respect bounding box diagonal length). Repeated instances are independently generated, but of similar quality.*

After this pre-processing phase, the user loosely selects a region as a query (see Figure 7). The selection results in a set of significant descriptors of various types. We encode the configuration formed by such descriptors using (rigid invariant) relative coordinates at some pivot point of configuration. Detection of repeated sections now amounts to finding similar configuration of descriptor types (see also sub-graph matching for line features employed by Bokeloh et al. [2009]). Based on the relative coordinates (possibly more than one) associated with each descriptor type, all the descriptors vote for absolute coordinates. Each descriptor type is associated with some offsets or a relative coordinate defined by their location in the query configuration. Note that it is possible that in the query there are a number of descriptors in different locations and each adds one offset. Then, absolute coordinates with many votes are candidates to be a pivot of a similar configuration.

The generated candidates often have false-positives. We prune them out in a validation step, where a candidate region with similar configuration is considered similar to the query region after a validation using a cube of voxels, however, this time at a higher resolution. Again, two cubes are said to be similar if there is a significant partial match between them, which is determined by counting the number of voxels that contains about the same number of points.

5 Results

Since real data rarely comes with ground truth to validate consolidation results, we tested our algorithm on synthetic 2D data with progressive amount of noise added, and with increasing number of

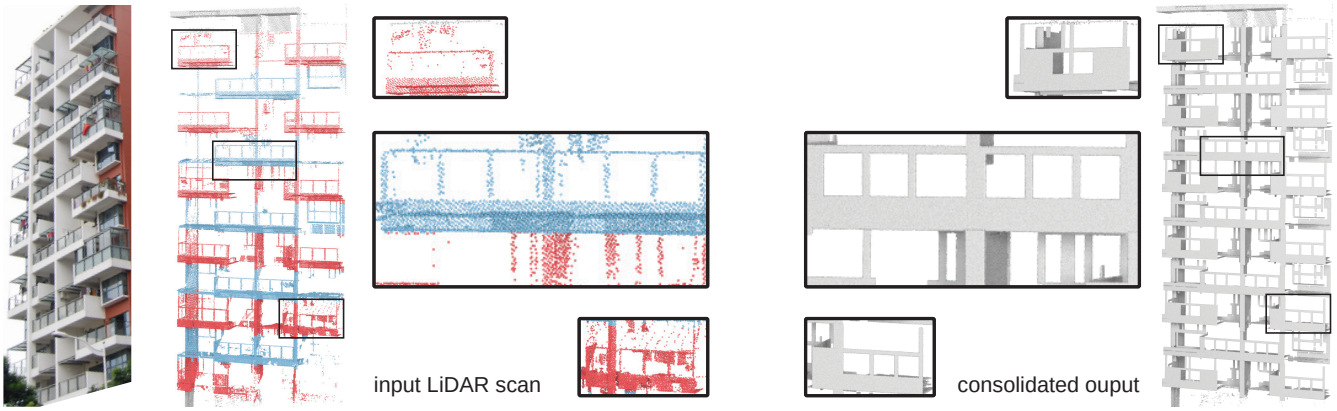

Figure 9: *Input and consolidation results on a building with only six floors. The consolidation result can be judged by comparing with the photograph of the building.*

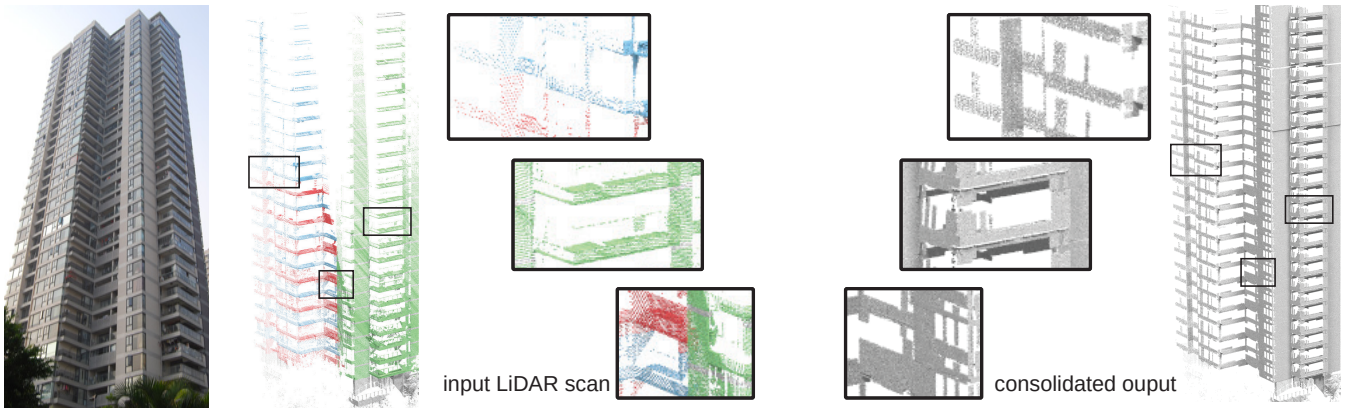

Figure 10: *Input and consolidation results on a very tall building with progressively poor data quality with height. Even though the data quality looks worse in comparison with Figure 9, the consolidation output is superior due to the high amount of repetitions.*

repetitions (see Figure 8).

We tested our algorithm on a variety of buildings scans obtained using a Optec Lynx LiDAR scanner mounted to a jeep driving typically at 20 mph. We list below the default set of parameters used in the various stages of our system. Further information can be found on our project webpage.

- Out-Plane Denoising: (i) cluster radius: 3.0 $\times$ median distance to nearest plane, (ii) cluster size: max($|\mathcal{I}|$/4, 2), where $|\mathcal{I}|$ is the number of repeated instances, and (iii) cluster weight: min(|cluster size|/10, 0.3).
- In-Plane Denoising: (i) $\lambda_0/\lambda_1 < 0.2$, (ii) scan interval: 0.005, and (iii) density threshold for validating a polygon: 0.7 $\times$ average density of the corresponding plane.
- Repetition Detection: If half of two voxelized regions are same, we consider them to be repeated instances.

Table 1 shows the performance of our system on various models presented in this paper.

In all the examples, the raw scans were directly processed and consolidated by our system. In each case, the user bootstrapped the process by marking a few query regions (see Figure 7). Compared to 2D, the consolidation effects in 3D are more pronounced, due to the additional off-plane denoising. Figures 9 and 10 shows consolidation results on two buildings with varying repetition pattern. Note that although visually the input data quality of the tall building looks worse, due to a higher number of repetitions the consolidation results are in fact better as compared to the complex building example.

For Figure 11, a building with fine details and sharp features, the consolidation results are satisfactory for the areas with large features. However, in regions where we have delicate structures in surrounding regions, we end up learning false 'details'. This is not surprising, since in this case noise is high, large parts of the data are missing, and the number of detected repetitions is only three, which is probably too few for effective consolidation.

Buildings, specially high rise ones, sometimes have characteristic cylindrical parts, e.g., capsule elevators. We extended our consolidation framework to detect and handle such patterns. In our algorithm, we supplement the plane fitting to RANSAC based cylinder

model	# pts.	# queries	# repet.	prep. time	cons. time
complex	128,558	2	5—6	60s	160s
walls	327,230	3	4—12—12	150s	164s
tall	433,325	3	7—17—25	55s	135s
cylinder	1,354,305	15	[3-32]	320s	215s
teaser	737,723	12	[6-8]	300s	99s

Table 1: *Performance statistics on a 2.67 GHz Intel Core i7-920 with 6GB RAM.*

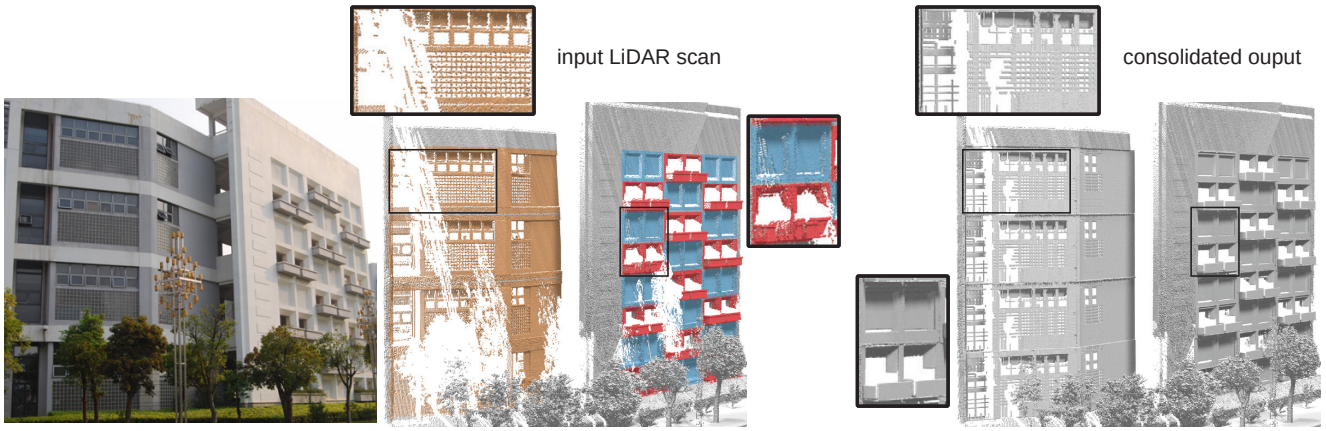

Figure 11: *Captured walls with high detail and large noise (left) are consolidated (right), while our weighted median in-plane consolidation preserves the fine detail (bottom zooms). In presence of high noise and large missing parts, we falsely detect additional lines (compare with simulation results in Figure 8).*

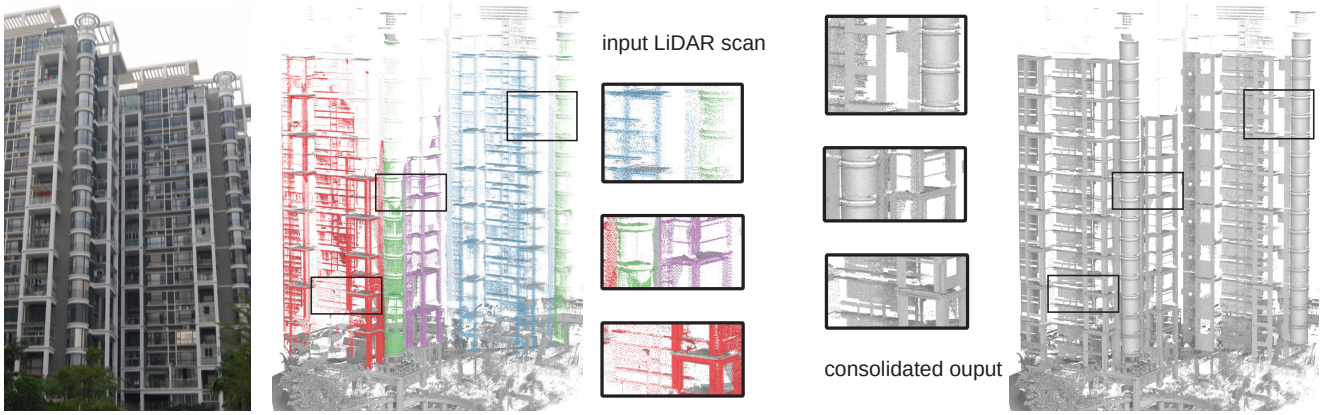

Figure 12: *We consolidate a large urban scene, containing cylinders as a repetitive component.*

fitting (see Figure 12). Although the system can be expanded to handle other low parameter count primitives, such inclusions are only justified if such patterns are common on urban facades.

Limitations. Our scan consolidation framework is designed for urban buildings with large scale repetitions. When the data assumptions are not met, we get incorrect results (see Figure 11). Our system needs some user queries to start looking for repetitions. For the type of data we consider, we believe that user assistance is probably unavoidable. While having the user in the loop prevents the system to be used in an unsupervised mode, loosely speaking as we work close to noise margin some crude guidance is needed to differentiate between signal from noise. Although repetition detection is not the focus of this work, its accuracy does dictate the quality of the consolidation results.

6 Conclusions

We presented a pipeline for consolidating imperfectly scanned data of urban buildings. We exploit the large scale repetitions commonly found in building scans and use it to denoise the input in two phases: off-plane and in-plane. Using robust statistical weighted medians for planes and for lines, we demonstrate that the original input can be significantly denoised, rectified, and consolidated. With the growing popularity and availability of LiDAR scanners, we expect our method to help produce consolidated scans sufficient for many of the online navigation and virtual city generation applications. The output of our system can possibly be used for reconstruction and procedural synthesis.

In the future we plan to focus on developing an algorithm to facilitate automatic detection of repetitions. The problem is hard because of the data quality and noise margin. Nevertheless, using high level data descriptors as well as limiting the search space to only few symmetry classes could alleviate the problem. The ultimate goal is to reconstruct good quality models using the consolidated scans possibly in conjunction with other easily accessible data sources.

Acknowledgements

We thank Shachar Fleishman for his thoughtful comments and the anonymous reviewers for their valuable suggestions. This work was supported in part by National Natural Science Foundation of China (60902104), National High-tech R&D Program of China (2009AA01Z302), CAS Visiting Professorship for Senior International Scientists, CAS Fellowship for Young International Scientists, Shenzhen Science and Technology Foundation (GJ200807210013A). Niloy was partially supported by a Microsoft outstanding young faculty fellowship.

References

AIGER, D., MITRA, N. J., AND COHEN-OR, D. 2008. 4-points congruent sets for robust surface registration. *Proc. of ACM SIGGRAPH 27*, 3, #85, 1–10.

BESL, P., AND MCKAY, N. 1992. A method for registration of 3D. In *IEEE PAMI*.

BIBER, P., AND STRASSER, W. 2006. nscan-matching: Simultaneous matching of multiple scans and application to slam. In *In Robotics and Automation*, 2270–2276.

BOKELOH, M., BERNER, A., WAND, M., SEIDEL, H.-P., AND SCHILLING, A. 2009. Symmetry detection using line features. *Computer Graphics Forum (Proceedings of Eurographics)*.

BUADES, A., COLL, B., AND MOREL, J.-M. 2005. A non-local algorithm for image denoising. In *Proc. of IEEE Conf. on Comp. Vis. and Pat. Rec.*, 60–65.

BUADES, A., COLL, B., AND MOREL, J.-M. 2008. Nonlocal image and movie denoising. *Int. J. Comp. Vis. 76*, 2, 123–139.

DABOV, FOI, KATKOVNIK, AND EGIAZARIAN. 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on 16*, 8, 2080–2095.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering arch. from photographs: A hybrid geometry- and image-based approach. *Proc. SIGGRAPH 30*, 11–20.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. *Proc. of ACM SIGGRAPH 22*, 3, 950–953.

GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *Proc. of Eurographics Symp. on Geometry Processing*, 253–262.

HAYS, J. H., LEORDEANU, M., EFROS, A. A., AND LIU, Y. 2006. Discovering texture regularity as a higher-order correspondence problem. In *Proc. Euro. Conf. on Comp. Vis.*

HUANG, H., LI, D., ZHANG, H., ASCHER, U., AND COHEN-OR, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. 28*, 5, Article 176.

JONES, T. R., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. In *Proc. of ACM SIGGRAPH*, 943–949.

KORAH, T., AND RASMUSSEN, C. 2007. 2d lattice extraction from structured environments. In *ICIP*, 61–64.

KORAH, T., AND RASMUSSEN, C. 2008. Analysis of building textures for reconstructing partially occluded facades. In *Proc. Euro. Conf. on Comp. Vis.*, 359–372.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project: 3d scanning of large statues. In *Proc. of ACM SIGGRAPH*, 131–144.

LINDENBAUM, M., FISCHER, M., AND BRUCKSTEIN, A. M. 1994. On gabor's contribution to image enhancement. *Pattern Recognition 27*, 1, 1–8.

LIPMAN, Y., COHEN-OR, D., LEVIN, D., AND TAL-EZER, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. 26*, 3, 22.

LIU, Y., COLLINS, R. T., AND TSIN, Y. 2004. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE PAMI 26*, 3, 354–371.

LIU, Y., LIN, W.-C., AND HAYS, J. H. 2004. Near regular texture analysis and manipulation. 368 – 376.

MERRITT, F., AND RICKETTS, J. 2001. *Building Design and Construction Handbook*, 6th ed. McGraw-Hill.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. In *Proc. of ACM SIGGRAPH*, vol. 25, 560–568.

MÜLLER, P., ZENG, G., WONKA, P., AND GOOL, L. J. V. 2007. Image-based procedural modeling of facades. *ACM Trans. on Graphics 26*, 3, 85.

MUSIALSKI, P., WONKA, P., RECHEIS, M., MAIERHOFER, S., AND PURGATHOFER, W. 2009. Symmetry-based facade repair. In *Vision, Modeling, and Visualization Workshop 2009 in Braunschweig, Germany (VMV09)*.

OZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Proc. Eurographics) 28*, 2, 493–501.

PARK, M., BROCKLEHURST, K., COLLINS, R. T., AND LIU, Y. 2009. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE PAMI 31*.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3d scan completion. In *Proc. of Symp. of Geometry Processing*, 23–32.

PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. 2008. Discovering structural regularity in 3D geometry. *ACM Trans. on Graphics 27*, 3.

PERONA, P., AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 7, 629–639.

RUDIN, L. I., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D 60*, 1-4, 259–268.

SCHAFFALITZKY, F., AND ZISSERMAN, A. 1999. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, 165–181.

SCHNABEL, R., DEGENER, P., AND KLEIN, R. 2009. Completion and reconstruction with primitive shapes. *Computer Graphics Forum (Proc. of Eurographics) 28*, 2, 503–512.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proc. of Int. Conf. on Comp. Vis.*, 839.

WANG, X., TOTARO, S., TAILL, F., HANSON, A. R., AND TELLER, S. 2002. Recovering facade texture and microstructure from real-world images. In *Texture Analysis and Synth.*, 381–386.

XIAO, J., FANG, T., ZHAO, P., LHUILLIER, M., AND QUAN, L. 2009. Image-based street-side city modeling. In *ACM SIGGRAPH Asia 2009 papers*, 1–12.

YOSHIZAWA, S., BELYAEV, A., AND SEIDEL, H.-P. 2006. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *SMI*, 38–44.

YU, Y., FERENCZ, A., AND MALIK, J. 2001. Extracting objects from range and radiance images. *IEEE Transactions on Visualization and Computer Graphics 7*, 4, 351–364.