

Consolidation of Low-quality Point Clouds from Outdoor Scenes

Jun Wang¹, Kai Xu^{2,3}, Ligang Liu⁴, Junjie Cao⁵, Shengjun Liu⁶, Zeyun Yu⁷ and Xianfeng David Gu⁸

¹Nanjing University of Aeronautics and Astronautics, China

²National University of Defense Technology, China

³Shenzhen Institutes of Advanced Technology, China

⁴University of Science and Technology of China, China

⁵Dalian University of Technology, China

⁶Central South University, China

⁷University of Wisconsin-Milwaukee, USA

⁸State University of New York at Stony Brook, USA

Abstract

The emergence of laser/LiDAR sensors, reliable multi-view stereo techniques and more recently consumer depth cameras have brought point clouds to the forefront as a data format useful for a number of applications. Unfortunately, the point data from those channels often incur imperfection, frequently contaminated with severe outliers and noise. This paper presents a robust consolidation algorithm for low-quality point data from outdoor scenes, which essentially consists of two steps: 1) outliers filtering and 2) noise smoothing. We first design a connectivity-based scheme to evaluate outlierness and thereby detect sparse outliers. Meanwhile, a clustering method is used to further remove small dense outliers. Both outlier removal methods are insensitive to the choice of the neighborhood size and the levels of outliers. Subsequently, we propose a novel approach to estimate normals for noisy points based on robust partial rankings, which is the basis of noise smoothing. Accordingly, a fast approach is exploited to smooth noise, while preserving sharp features. We evaluate the effectiveness of the proposed method on the point clouds from a variety of outdoor scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: 3D point data—Outlier detection; Normal estimation; Noise smoothing; Feature preserving

1. Introduction

Recent advances in scanning technologies promote the growing popularity of fast, easy-to-use, affordable scanning devices and cameras, which makes it possible for rapid acquisition from large-scale scenes. More recently, considerably increasing attention has been attracted to data acquisition of outdoor scenes, such as urban facades, heritage sites, which supports an abundance of ambitious applications in digital city modeling, GIS mapping and navigation, and commerce [KMYG12]. For such massive applications, the scanning devices are either put statically far away from the objects of interest, or even mounted over airborne or vehicles, where the carriers move at their normal driving speed during scanning. Due to the imperfect outdoor scanning conditions, like the quantization or motion of objects and mul-

tle reflections, object occlusion, the acquired point data are often corrupted by severe outliers and noise. These low-quality point data present dramatic challenges in the downstream applications (e.g. model reconstruction). Therefore, an elaborate consolidation is imperative to remove outliers and smooth noise from such point data.

Outliers detection is a non-trivial task for point clouds for several reasons: lack of prior knowledge of the statistical distribution of outliers, variances in local point densities, and geometrical discontinuities [Sot07]. When scanning a building facade, attached with static posters and flying flags, the scan data of the posters and flags are outliers. Their statistical distribution and densities are obviously unknown, which leads to serious troubles in detecting outliers. Another principal challenge is how to remove noise

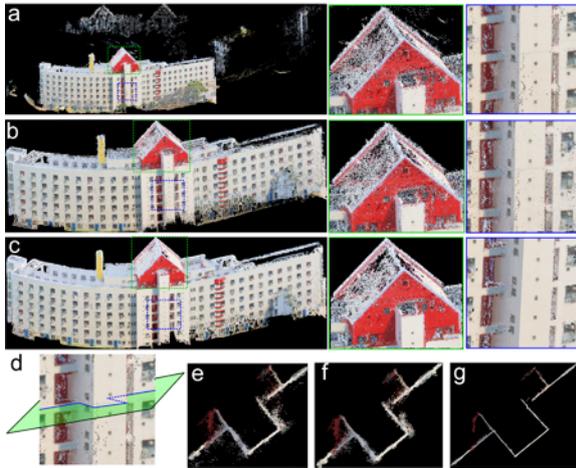


Figure 1: Outlier filtering and noise smoothing of 3D scene point data. (a) The original point data of the building with color mapping, containing a high level of outliers and noise; (b) The outlier filtering result; (c) The noise smoothing result. The middle and right columns of (a), (b) and (c) show the zoom-in details of "roof" and "wall" parts of the building. The section views of the walls, as shown in (d), are given in (e), (f) and (g), corresponding to the results in (a), (b) and (c) respectively. From (c) and (g), the outliers are significantly removed, the noise is nicely smoothed, and the edges of the wall are well preserved.

around sharp features. The geometric structures of the objects (e.g., buildings) in outdoor scenes are designed with many sharp edges and corners with the consideration of stability and aesthetics. For the geometry of the objects containing sharp features, differentiating between sharp features and noise is fairly demanding as both are inherently ambiguous, and oftentimes the limitations of the measuring devices and techniques may prevent proper generation of such high-frequency features [FCOS05].

In this paper, we present an effective consolidation solution to automatically remove outliers and smooth noise from low-quality point clouds of outdoor scenes, as demonstrated in Figure 1. Considering the characteristics of outliers, we design an effective outlier factor based on the relative density deviation of the local neighborhood and the average local neighborhood, providing a scoring strategy that includes a normalization to become *independent* from the specific data distribution. Furthermore, a clustering-based method is used to further remove small dense outliers, which exist ubiquitously in the scene data.

Next, we present a fast point cloud denoising method which can remove noise effectively, while preserving sharp features. Noisy normals are filtered iteratively by adaptively

weighted averaging of neighboring normals; and point positions are updated to agree with the filtered normals. The weight function for normal filtering is fairly simple, which makes the denoising algorithm fast and thus suitable for the large-scale scene data. Moreover, the filtering is anisotropic so that the normals filtered are sensitive to sharp features, based on which the points are updated. Thus, sharp features can be well preserved.

Our denoising method substantially relies on the normals of points. We observe the geometry of objects (e.g., architectures) in outdoor scenes is mainly composed of planar patches. Based on this observation, we propose a robust approach for estimating normals, which essentially seeks the best associated planes of points by clustering consistent points from the same underlying substructures based on partial rankings. As a result, the estimation is still accurate for sharp features that usually locate around multiple substructures.

The main contributions are summarized as follows:

1. An effective connectivity-based outlier detection method is proposed, which is insensitive to the size of chosen neighborhood. Additionally, a clustering-based outlier detection algorithm is designed to properly filter out small dense outliers.
2. A novel approach is designed to estimate normals of points guided by the information derived from residual sorting, which yields reliable results even in the presence of highly noisy point data.
3. A fast denoising method presented is capable of smoothing noise effectively, while recovering sharp features of the original point data.

1.1. Related Work

Point Cloud Processing of Indoor Objects. In the past decade, a variety of approaches have been proposed to process 3D point clouds of traditional indoor objects, such as mechanical parts, small sculpture models [WPK*04, SB-S08, KNRS12]. When scanning the point data from those kinds of objects, the devices are usually fixed near the objects, so the level of noise and outliers is relatively *low*. Schall et al. [SBS05] used a kernel density estimation technique for noise filtering of a noisy point set sampled from a smooth surface. Liu et al. [LCW12] designed a framework to consolidate unorganized point data by an iterative procedure of interlaced down-sampling and up-sampling steps. Those methods are able to handle noise to a certain degree; however, they have difficulties recovering sharp features. Lange et al. [LP05] presented a method for anisotropic fairing of point clouds using mean curvature flows, which is capable of recovering sharp edges from a low level of noise. Overall, the current methods may achieve satisfactory results on point clouds with a low level of noise, but they usually fail in the presence of a high level of outliers and noise. Moreover, the scale of outdoor scene data is usually so large that most of those methods cannot handle efficiently.

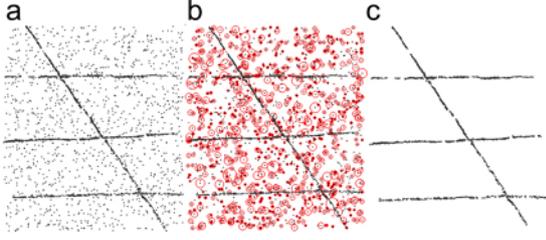


Figure 2: 2D illustration of outlier detection. (a) 2D point data with outliers; (b) Outlier detection with DDF, where the outliers are associated with circles, the radii of which correspond to their DDF values; (c) Outlier removal result. Note that the outliers are accurately detected.

Point Cloud Consolidation of Outdoor Scenes. As mentioned earlier, the quality of data acquisition from outdoor scenes is poor, always suffering from severe missing data, noise and outliers. Recently, several researchers have started to put efforts in processing 3D point clouds from real scenes [SHFH11, LZS*11, FS13]. Zheng et al. [ZSW*10] proposed a pipeline for consolidating deficient scanned data of urban buildings, particularly, for data completion. In their denoising step, outlier removal and sharp feature preservation are not considered. DeVore et al. [DPH*13] presented a solution for terrain point cloud processing, where a metric is defined for determining the outlierness of a point. As stated by themselves, the metric performs well at identifying outliers along smooth areas; however, it treats sharp edges, corners, and boundaries as outliers frequently. Our method is capable of preserving sharp features even in the presence of severe outliers and noise.

2. Outliers Filtering

2.1. Filtering Sparse Outliers

Outlier is considered as "a datum which differentiates so much from the other data as to arouse suspiciousness that is generated by a different mechanism" [Haw80]. The goal of outlier detection is to find those unusual data in a given data set. Recently, outlier detection has become an important and well-studied problem in data mining, machine learning and statistics communities [BKNS00, KKSZ09]. For example, Papadimitriou et al. [PKGf03] proposed an effective outlier detection method based on the correlation integral of local density for data mining applications. Inspired from this technique, we design a new connectivity-based approach to detect sparse outliers from 3D point clouds.

Given a point set $P = \{p_1, p_2, \dots, p_n\}$ (n is the number of points), let $d(p_i, p_j)$ be the distance between p_i and p_j , we have the following definitions:

- *k*-distance of a point p . For any positive integer k , the *k*-distance of a point p , represented by $k\text{-dist}(p)$, is defined

as the distance $d(p, q)$ between two points $p, q \in P$ such that: 1) for at least k points $q' \in P$, $d(p, q') \leq d(p, q)$, and 2) for at most $(k - 1)$ points $q' \in P$, $d(p, q') < d(p, q)$.

- *k*-distance neighborhood of a point p . Given the *k*-distance of p , the *k*-distance neighborhood of p , denoted by $N_{kd}(p)$, contains every point whose distance from p is not greater than the *k*-distance of p .

Suppose $N_{kd}(q)$ is the *k*-distance neighborhood of q ($q \in N_{kd}(p)$), the average distance from q to all points in $N_{kd}(q)$ is defined as:

$$\bar{d}_k(q) = \frac{1}{|N_{kd}(q)|} \sum_{q' \in N_{kd}(q)} d(q, q') \quad (1)$$

where $|\cdot|$ is the cardinality of (\cdot) . Then, the average of $\bar{d}_k(q)$ over all the points in the *k*-distance neighborhood of p is $\bar{D}_k(p) = \frac{1}{|N_{kd}(p)|} \sum_{q \in N_{kd}(p)} \bar{d}_k(q)$. Given a point p and a positive integer k , we define a distance-based deviation factor (*DDF*) as:

$$DDF_k(p) = \left| \frac{\bar{D}_k(p) - \bar{d}_k(p)}{\bar{D}_k(p)} \right| = \left| 1 - \frac{\bar{d}_k(p)}{\bar{D}_k(p)} \right| \quad (2)$$

The *DDF* at $N_{kd}(p)$ is the relative deviation of the densities of its local neighborhood and the average local neighborhood. Intuitively, a point whose neighborhood density is close to the average local neighborhood density will have a *DDF* of 0, while an outlier will have a *DDF* much greater than 0. Accordingly, we can pick a cut-off to determine whether a point is outlier or not. However, it is non-trivial to set up an appropriate cut-off without any hints from the given data set. Here, we utilize the probabilistic logic theory [Nil86] to exploit the property of *DDF*.

The standard deviation of $\bar{d}_k(q)$ over $N_{kd}(p)$ is:

$$\sigma_k(p) = \sqrt{\frac{\sum_{q \in N_{kd}(p)} (\bar{d}_k(q) - \bar{D}_k(p))^2}{|N_{kd}(p)|}} \quad (3)$$

and the normalized standard deviation $\sigma_k(p)$ of the average of $\bar{d}_k(q)$ over $N_{kd}(p)$ is:

$$\sigma_{DDF_k(p)} = \frac{\sigma_k(p)}{\bar{D}_k(p)} \quad (4)$$

Based on the reasonable assumption that pairwise distances at a sufficiently small scale are drawn from a single distribution, we have the following criterion to determine the outlierness of a point p :

$$p = \begin{cases} \text{an outlier} : & \omega(p) = \frac{DDF_k(p)}{\sigma_{DDF_k(p)}} > \omega; \\ \text{an inlier} : & \text{otherwise.} \end{cases} \quad (5)$$

where ω is a positive coefficient. $\omega = 1.25$ generally produces satisfactory results in our experiments. Accordingly, given a positive value k , we compute *DDF* and σ_{DDF} for each point of the input point data, and then label an outlier whenever *DDF* is greater than $\omega \cdot \sigma_{DDF}$. Figure 2 gives the 2D illustration of outlier detection.

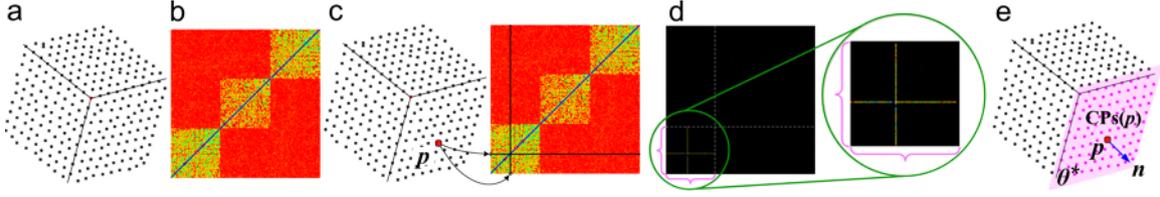


Figure 3: Illustration of normal estimation. *a)* A neighboring point set with three plane substructures. For illustration, we arrange the point set substructure-by-substructure. *b)* The similarity matrix from all point pairs, where the similarity value is colored from red (lowest) to blue (highest). Note that two points from the same substructure have a relatively high similarity value, while those from distinct substructures have a small value. *c)* One random point p and its corresponding row and column in the similarity matrix; *d)* The points that have relatively high similarity values to p in the matrix; and *e)* The consistent point set $\text{CPs}(p)$, the fitted plane θ^* and the normal $n = n(\theta^*)$.

This outlieriness indicator based on the standard deviation is the main feature of the outlier detection method. Note here we do not need to choose many magic cut-offs, while using the probabilistic reasoning based on $\sigma_{DDF_k(p)}$. From the definitions, $\bar{d}_k(q)$ and $\bar{D}_k(p)$ are pairwise distances and hence we consider their distribution and also compare each object of those within its neighborhood. Even if the point data possess a mixture of different distributions, the local deviation characteristic enables our method to detect the outliers successfully. In reality, for real data sets, the distribution of pairwise distances generally abides by a certain distribution over all or most scales.

2.2. Filtering Small Cluster Outliers

With this method above, sparse outliers can be filtered out and locally dense points are considered as inliers. As a result, the small dense subsets may still survive, which, however, could be outliers. To eliminate those outliers, we exploit a clustering-based scheme to remove the small cluster outliers.

The k -nearest neighbor graph (k -NNG) is constructed for the point data (e.g. $k = 3$). We observe that the points on the objects of interest generally are denser than those on the objects of no interest. Accordingly, we search the point with the smallest value ϖ_{min} of ϖ in (5) among the un-clustered point data, and regard this point as a seed point, from which the region grows by adding its k -NN into the region. Each added point q satisfy two compatible conditions within the region: 1) $\|p - q\| < 0.75\hat{d}$; 2) $|\varpi(p) - \varpi(q)| < 2.5\varpi_{min}$, where \hat{d} is the mean distance of k -NNG over all point data. The growing process repeats iteratively until no more points can be added into the region, referred to as a cluster. Finally, the point set is partitioned into distinct clusters, where the small clusters are regarded as outliers.

3. Partial Ranking based Normal Estimation

After removing outliers, we perform noise smoothing, which is based on the normal information. Considering the geometry characteristics of the objects in outdoor scenes, we

present a robust method for estimating normals on noisy point data that preserves sharp features.

3.1. Partial Ranking from Residual Sorting

Given a point p on a piecewise surface and its neighborhood $N(p)$, there are two categories w.r.t. its location: 1) far from any sharp features; 2) near sharp features (e.g. edge and corner). For the former case, we can randomly choose three non-collinear points from $N(p)$ to determine a plane and thereby obtain the normal; for the latter case, there may be several piecewise surfaces surrounding p , that is, $N(p)$ consists of several piecewise sub-neighborhoods. Thus, choosing three non-collinear random points cannot result in a correct normal. Therefore, we need to find the best plane determined by one of sub-neighborhoods. Toward this end, we design a clustering algorithm to seek the best underlying substructures of points based on partial rankings and thereby estimate the normals for both cases.

For any two points, if they are inliers from the same substructure, they share similar preferences to the hypotheses as measured by residuals, while cross-structure inliers and noise do not exhibit this characteristic. Building on this observation, we construct a top- k permutation list to indicate the preference, and thereby design a robust point similarity metric capable of differentiating inliers of the same underlying structure from other point data.

Given a point p , let $Q = \{q_i | i = 1, 2, \dots, n\}$ be the neighborhood of p , and $\Theta = \{\theta_i | i = 1, 2, \dots, m\}$ be a set of m hypothetical planes, where each hypothetical plane θ_i is fitted by three non-collinear points. For each point q_i of Q , we compute its absolute residuals as measured to the m putative planes to obtain a residual list:

$$\mathbf{r}^{(i)} := [r_1^{(i)}, r_2^{(i)}, \dots, r_m^{(i)}] \quad (6)$$

Then we sort the elements in $\mathbf{r}^{(i)}$ in non-descending order as:

$$\tilde{\mathbf{r}}^{(i)} := \left[r_{\pi_1}^{(i)}, r_{\pi_2}^{(i)}, \dots, r_{\pi_m}^{(i)} \right] \text{ s.t. } r_{\pi_1}^{(i)} \leq r_{\pi_2}^{(i)} \leq \dots \leq r_{\pi_m}^{(i)} \quad (7)$$

Then the top- k list of q_i is defined as the first k elements of the permutation $\Pi^{(i)} := \{\pi_1^{(i)}, \pi_2^{(i)}, \dots, \pi_m^{(i)}\}$, i.e.,

$$\pi^{(i)} := \{\pi_1^{(i)}, \pi_2^{(i)}, \dots, \pi_k^{(i)}\} \quad (8)$$

Essentially, the sorting $\pi^{(i)}$ ranks the k hypothetical planes according to the preference of q_i ; the higher a hypothetical plane is ranked, the more likely q_i is an inlier to it.

3.2. Kendall's tau based Similarity

Intuitively, two points share many common hypothetical planes at their top- k preference list $\pi^{(1)}$ and $\pi^{(2)}$, if they come from the same underlying structure. Theoretically, we measure the similarity of two top- k lists using Kendall's tau [Ken38]. Let π be a top- k list, D_π be a set of elements contained in π , $\pi(e)$ the position of the element $e \in D_\pi$. Then the Kendall's tau distance between $\pi^{(1)}$ and $\pi^{(2)}$ is given by:

$$K(\pi^{(1)}, \pi^{(2)}) = \sum_{\{e_i, e_j\} \in \mathcal{P}} \bar{K}_{e_i, e_j}(\pi^{(1)}, \pi^{(2)}) \quad (9)$$

where $\mathcal{P} \in \{(i, j) | i \neq j, i, j \in D_{\pi^{(1)}} \cup D_{\pi^{(2)}}\}$ and $\bar{K}_{e_i, e_j}(\pi^{(1)}, \pi^{(2)})$ is defined as:

$$\bar{K}_{e_i, e_j} = \begin{cases} 0 & : \prod_{k=1,2} (\pi^{(k)}(e_i) - \pi^{(k)}(e_j)) > 0 \\ 1 & : \text{otherwise.} \end{cases} \quad (10)$$

Accordingly, we measure the similarity of two points using the Kendall's tau distance of their top- k preference lists. The definition of similarity of two points p_1 and p_2 is represented by:

$$\mathcal{S}(p_1, p_2) = 1 - \frac{1}{k(k+1)} K(\pi^{(1)}, \pi^{(2)}) \quad (11)$$

From the definition, if the two points are from the same underlying structure, the similarity value is large; otherwise, it is small.

3.3. Consistent Structure Clustering

As aforementioned, the planar structures basically constitute the geometry of the objects in outdoor scenes. Therefore, we design a guided clustering scheme based on the similarity function to seek the best underlying planar structures of points. For each point, it clusters as many points as possible from its neighborhood which come from the same substructure as the point. Given a point p , its neighboring point set Q and the m hypothetical plane set Θ (e.g. $m = 100$), our goal is to gather a group of consistent points $Q' \subseteq Q$ such that all points in Q' are from the same underlying substructure as p . First, we define a weighting function between two points p and q :

$$w(p, q) := \begin{cases} \mathcal{S}(p, q) & : \text{if } p \neq q; \\ 0 & : \text{otherwise.} \end{cases} \quad (12)$$

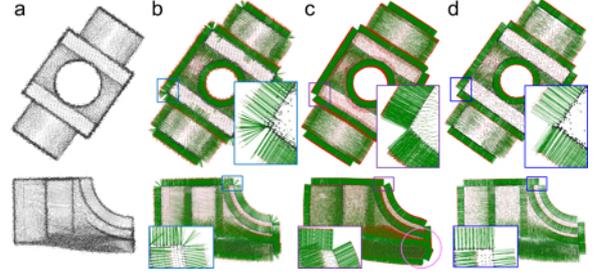


Figure 4: Normal estimation comparison on the block and fandisk point data (both with 3% Gaussian noise). (a) The noisy block data; (b) The normals estimated from Li et al.'s [LSK*10]; (c) Huang et al.'s [HWG*13] and d) Ours. The second row gives the corresponding results of the noisy fandisk data from three methods. The results from Huang et al.'s [HWG*13] and ours are relatively better than that of Li et al. [LSK*10]. We notice that some parts in Huang et al.'s [HWG*13] results are slightly deformed (see the circular region in the second row), which may lead to inaccurate normals.

where \mathcal{S} is the similarity function (11). Given the point p , the conditional probability $P(q|p)$ of choosing $q \in Q$ as the consistent point is thus determined by the following monotonic relation:

$$w_p(q_u) \geq w_p(q_v) \Rightarrow P(q_u|p) \geq P(q_v|p) \quad (13)$$

where $w_p(\cdot)$ is a weighting function conditioned on p , i.e. $w_p(\cdot) := w(\cdot, p)$. The monotonic relation function (13) suggests that points which are consistent to p are more likely to be chosen. More specifically, the points that are more consistent to p are more likely to be from the same underlying surface. Based on this idea, the clustering algorithm is proposed to gather a group of points from Q of p such that all of them are consistent to p , and then we apply plane fitting on the cluster to obtain the normal of the plane, i.e., the normal of p . The approach is implemented in Algorithm 1, where $\bar{\mathcal{S}}(p)$ is the mean similarity value of all points in Q to p . Figure 3 illustrates the idea of normal estimation.

From the algorithm, the fundamental step is to generate m hypothetical planes from the neighboring point set Q of p . Each hypothetical plane is determined by 3 non-collinear points. It is observed that the normals of those 3 points should be close to the plane normal, if all 3 points are potentially from the same underlying surface, based on which we manage to obtain more accurate hypothetical planes. Specifically, when generating hypothesis, if the normal of the determined plane is not close to the normals, if available, of all three points, this hypothetical plane should be discarded and choose another points. As a result, more reliable hypothetical planes are obtained. Moreover, this pre-filtering also speeds up the hypothesis generation.

Algorithm 1 Partial Ranking based Normal Estimation

```

1: procedure ESTIMATENORMAL( $p$ )
2:    $\mathbf{CPs}(p) \leftarrow \emptyset$   $\triangleright$  the consistent point set
3:    $Q \leftarrow$  the neighboring point set of  $p$ 
4:    $\Theta \leftarrow$  generate  $m$  hypothetical planes from  $Q$ 
5:   compute the top- $k$  list  $\pi$  of  $p$ 
6:   for each  $q_i \in Q$  do
7:     compute the top- $k$  list  $\pi^{(i)}$  of  $q_i$ 
8:     compute the similarity  $\mathcal{S}(p, q_i)$ 
9:     compute the weighting function  $w(p, q_i)$ 
10:    if  $w(p, q_i) \geq 0.25\bar{\mathcal{S}}(p)$  then
11:      append  $\mathbf{CPs}(p) := \mathbf{CPs}(p) \cup \{q_i\}$ 
12:    end if
13:  end for
14:   $\theta^* \leftarrow$  apply plane fitting on  $\mathbf{CPs}(p)$ 
15:  return  $n(\theta^*)$   $\triangleright \theta^*$ : the best tangent plane
16: end procedure

```

The described normal estimation algorithm is able to estimate the normals of point data properly even in the presence of noise. Furthermore, this method is capable of preserving them, unlike the regression-based techniques that tend to smooth sharp features and fail to estimate normals near sharp features. Note that the normals estimated here are not oriented so that the orientation method [LW10] is used. Figure 4 shows the normal estimation results of the *block* and *fandisk* models from Li et al.'s [LSK*10], Huang et al.'s [HWG*13] methods and ours. From the comparisons, Huang et al.'s [HWG*13] method and ours obtain better results than Li et al.'s [LSK*10]. Huang et al.'s [HWG*13] method essentially propagates point resampling from smooth regions to sharp edge regions so as to smooth noise and preserve sharp edges. Therefore, the points from their results are denser than the original data, as shown in Figure 4 (c). Due to noise, the shape of the model is slightly deformed after resampling. Consequently, the normals on the deformed regions could be inaccurate.

4. Noise Smoothing

In this section, we introduce a robust method to smooth noise from point clouds. *Mesh* denoising has been extensively studied in the last decade. Belyaev and Ohtake [BO03] presented an interesting comparison of *mesh* smoothing methods based on the two-step technique: 1) face normal averaging, and 2) mesh fitting. This type of methods show good characteristics in denoising mesh with sharp features with a low level of noise. However, in real-scanning data, the level of noise could be fairly high. To address this problem, we propose a more robust two-step iterative approach to smooth noise based on adaptively anisotropic normal filtering.

4.1. Normal Filtering

In the noisy point data, the normals could be corrupted more or less by the noise, even though the estimation above is well performed. Therefore, we take into account the neighboring normals of a point to alleviate the effect of noise. However, inaccurate estimation of normals in the neighborhood may affect the current point, especially near sharp edges and corners. To address this issue, we apply normal filtering on anisotropic neighborhoods. As is known, a surface with sharp features is generally piecewise smooth and a feature lies on the intersection of multiple surrounding patches. When noise filtering is applied to a feature point, it is expected that the point is only influenced by the neighboring points from the same patch. From Algorithm 1, each point has an associated best tangent plane θ^* , so we consider this point and its Consistent Point set (\mathbf{CPs}). In the following context, we refer to the consistent point set of each point as its supporting neighborhood, based on which, an effective filtering method is proposed to filter normals.

For each point, we cluster all the points from its supporting neighborhood which are similar to this point in terms of certain attributes. Given a point p , let $\mathbf{CPs}(p)$ be the supporting neighborhood of p , the weighted normal of p is represented by:

$$n(p) = \frac{\sum_{q \in \mathbf{CPs}(p)} \mathfrak{S}(p, q) \cdot n(q)}{\sum_{q \in \mathbf{CPs}(p)} \mathfrak{S}(p, q)} \quad (14)$$

where $n(q)$ is the normal of q , and $\mathfrak{S}(p, q)$ is a function measuring the normal similarity of p and q , which is defined as:

$$\mathfrak{S}(p, q) = e^{-\frac{\left[\frac{(n(p) - n(q)) \cdot n(p)}{\zeta}\right]^2}{2\sigma^2}} \quad (15)$$

where σ^2 is the variance and ζ is a positive number. Based on these definitions, we have the following approach to update $n(p)$:

1. Initiate the result with the original normal: $n_0(p) \leftarrow n(p)$;
2. Calculate the $(j+1)$ -iteration result from the j -iteration result by:

$$n_{j+1}(p) = \frac{\sum_{q \in \mathbf{Q}(p)} \mathfrak{S}_j(p, q) \cdot n(q)}{\sum_{q \in \mathbf{Q}(p)} \mathfrak{S}_j(p, q)}, \quad j = 0, 1, \dots \quad (16)$$

until $\|n_{j+1}(p) - n_j(p)\| < \Psi = 0.999$, or the max iteration reaches a given number τ (e.g. 50), where $\mathbf{Q}(p) = \{q | q \in \mathbf{CPs}(p), |(n_j(p) - n(q)) \cdot n_j(p)| < \zeta\}$.

From (16), we do not take into account for calculation all the points from $\mathbf{CPs}(p)$, but from $\mathbf{Q}(p)$ where the normals of all points are close to $n(p)$. Essentially, the process ignores heavy noise and thus the method is resistant to noise. The final normal corresponds to the local maximum of the probability density of the normals, estimated with a certain kernel.

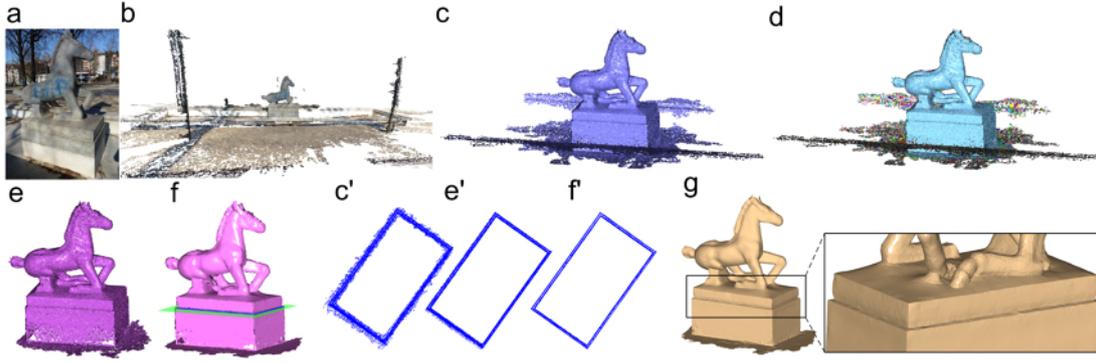


Figure 5: Processing result on 3D scene point data ($k = 50$, $\zeta = 0.5$, $n_p = 12$). (a) The image of 3D scene with the ground, tree and horse; (b) The point data of the scene with color mapping; (c) The outlier filtering result; (d) The clustering result (12,827 clusters); (e) The clustering-based outlier removal result (the biggest cluster); (f) The noise smoothing result; (g) The triangular mesh of (f), generated from [OBS05]. The section views of the "base" part, as shown in (f), are given in (c'), (e') and (f'), corresponding to the results in (c), (e) and (f), respectively. From the section views, the outliers and noise are removed satisfactorily, while sharp corners are well preserved. Consequently, it results in a nice surface mesh, as shown in the zoom-in view of (g).

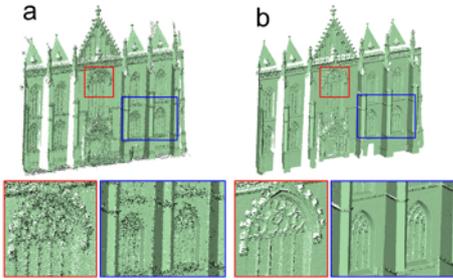


Figure 6: Consolidation result of the point cloud of a catholic church ($k = 50$, $\zeta = 0.55$, $n_p = 10$). (a) The church point cloud and (b) The consolidation result. We notice that sharp edges of windows are well preserved, and small decorating details are also recovered.

4.2. Point Updating

Having the accurate normals, we evolve the point data to match the new normal field based on an iterative point updating strategy. Technically, we introduce the point similarity function (11) to design a fast and effective vertex updating scheme.

Given a point p , the k -nearest neighbor is searched, denoted by $Q_k(p)$ (e.g. $k = 6$). Then, the point p is updated to p' by the following formula:

$$p' = p + \frac{\sum_{q \in Q_k(p)} \mathcal{S}(p, q) [n(q) \cdot (q - p)] n(q)}{k \sum_{q \in Q_k(p)} w(p, q)} \quad (17)$$

where q is one of the neighboring points in $Q_k(p)$, $n(q)$ is the normal of q . From the definition, the points which are more

similar to p will have bigger contribution to the updated position of p . By performing this algorithm over the point data, all point positions are updated. As a result, noise is significantly smoothed, while sharp features are well preserved.

5. Results and Discussion

This section provides experimental results to validate our proposed algorithm. We implemented the algorithm in C++ and ran it on an Intel 2.40GHz machine with 4GB of memory. We have tested it on a variety of point clouds from outdoor scenes, as well as indoor objects.

5.1. Parameters

Our proposed method requires three parameters: 1) the size of neighborhood k ; 2) the normal difference threshold ζ ; and 3) the point updating iterations n_p . A plenty of experiments with different levels of density show the results are affected inconsiderably by k . Hence, we set $k = 50$ in all cases. The choice of ζ depends on the sharpness of features. It is set relatively small for sharp features (e.g. [0.1, 0.3]), while big for smooth features (e.g. [0.3, 0.65]). The setting of n_p relates to the level of noise. If the noise is heavy, n_p should be comparatively big. A relatively high value of n_p usually guarantees the success of smoothing. Typically, it is set within the range of [5, 20].

We have conducted a series of experiments on the same point data by choosing different parameter settings. The consolidation results suggest to be insensitive to the parameter settings. Overall, $k = 50$, $\zeta = 0.5$, $n_p = 15$ generally yield satisfactory results according to a large number of testing

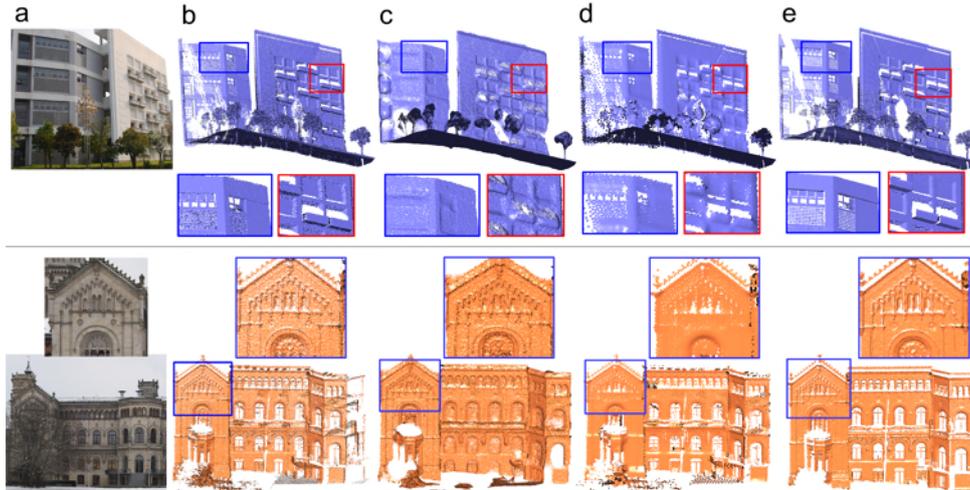


Figure 7: Consolidation results of the scan data of two building facades. (a) The image of a building facade; (b) The original scan data; and the consolidation results from (c) Liu et al. [LCW12], (d) Huang et al.'s [HLZ*09] methods and (e) Ours. The facade contains many planar substructures (i.e., walls and windows). From the results, the sharp edges of windows and walls are comparatively better preserved with our method. The second row presents the consolidation results of another facade data from three methods. From the comparison, our method obtains more satisfactory results in which the small details are well recovered and the sharp edges are preserved successfully.

cases. Moreover, it usually does not require multiple iterations.

5.2. Outdoor Scenes

To extract the objects of interest from the scene point clouds, a lot of tedious user interactions are required in previous work so that users need to determine where the outliers and noise are, and then delete them manually. Comparatively, our method is able to detect and remove outliers, and smooth noise with very limited interactions. To verify that, we first run our algorithm on point clouds from outdoor scenes, which include high levels of noise and outliers. Figure 5 shows the consolidation result of the point cloud including the ground, trees and horse, where the horse is the object of interest. By running the outlier removal algorithm, the horse is successfully extracted in Figure 5 (c). The original points of horse are bumpy, which are smoothed favorably by our noise smoothing method (as shown in Figure 5 (f)), resulting in a nice triangular mesh in Figure 5 (g). The whole consolidation process only requires the input of three parameters (i.e., $k = 50$, $\zeta = 0.5$, $n_p = 12$). From the section views, the outliers and noise are removed, while sharp corners are well preserved. Figure 6 presents the consolidation result of the point clouds of a catholic church. The facade contains a number of decorating details and sharp edges. The original point cloud is severely corrupted with outliers and noise. From the zoom-in view, sharp features are well preserved, and small details are also recovered.

5.3. Comparison to Other Methods

We compare two point cloud consolidation methods [LCW12, HLZ*09] with ours. Figure 7 shows the processing results of the scan data of two building facades with Liu et al.'s [LCW12], Huang et al.'s [HLZ*09] methods and ours. There are many sharp edges among walls and windows in the point cloud of the first row. The noise is well smoothed by Huang et al.'s method [HLZ*09]; however, sharp edges are smoothed as well. Liu et al.'s method [LCW12] can handle noise to some extent, while sharp features are also blurred. Our method achieves the best result in terms of removing outliers, smoothing noise and preserving sharp features, as shown in the zoom-in views. The second row presents the consolidation results of another facade data with three methods. There are some trees in the scene; in addition, on the building facade there are a number of small decorating details and sharp edges (see Figure 7 (a)). The trees lead to heavy outliers in the point cloud, as shown in Figure 7 (b) of the second row. The outliers in Huang et al.'s [HLZ*09] and Liu et al.'s [LCW12] results are removed completely. Huang et al.'s method [HLZ*09] smooths out small details as well as sharp edges in the point cloud. Liu et al.'s method [LCW12] retains some of small features, which, however, are still noise-ridden. Comparatively, our result is better, in which outliers are removed thoroughly and noise is smoothed properly, whilst small details and sharp feature are also retained. Table 1 shows the computational timings of our method on those scene point clouds.

Table 1: Timings of our method on scene point clouds.

Figure	Points	Consolidation Time (seconds)			
		Outlier	Normal	Noise	Total
1	1,400k	90	272	87	449
5	1,814k	116	301	114	531
6	1,447k	92	281	91	464
7[1]	632k	42	120	39	201
7[2]	421k	26	82	26	134

5.4. Indoor Objects

Our method also works well on the point data from indoor objects. To evaluate the performance, the *fandisk* model is tested in Figure 8, where the synthetic noise is added by a zero-mean Gaussian function with standard deviation proportional to the diagonal length of the bounding box of the groundtruth point cloud. We run Fleishman et al.'s [FCOS05], Sun et al.'s [SRML07], Zheng et al.'s [ZFAT11] methods and ours on the noisy point data. The first row shows the noisy point data and the corresponding results from four methods. Then, we adopt the surface reconstruction method [OBS05] with the same parameter setting to generate surface meshes from the processed point data, respectively. From the comparisons, Fleishman et al.'s [FCOS05] and Zheng et al.'s [ZFAT11] results are not as good as those from Sun et al. [SRML07] and ours. We also notice that Sun et al.'s [SRML07] method produces a number of bumpy meshes. Since the same reconstruction parameters are used, it suggests the superiority of our algorithm to other methods. Meanwhile, we provide some quantitative comparisons. To demonstrate the fidelity of the processed data to the groundtruth data, we compute the Hausdorff distance from the processed data to the groundtruth data. Figure 9 shows detailed comparisons of the Hausdorff distance results, where the horizontal axis is the distance value between the smoothed and the groundtruth data, and the vertical axis is the corresponding histogram (in percentage) with respect to each distance value. From this figure, our method yields smaller Hausdorff distances, suggesting that our method produces faithful point data to the groundtruth data.

To evaluate the capability of dealing with heavy outliers and noise, our algorithm is ran on extremely high levels of outliers and noise in Figure 10. As outliers increases, the processing quality gradually gets worse. Even though 75% outliers are added, the sharp features are still captured.

5.5. Limitation

In scene scanning data, the points could be fairly sparse around the boundaries of the object of interest due to the scanning process. Consequently, some of those points are probably detected as outliers. In addition, the outdoor scenes (e.g. buildings and architectures) contain a high percentage of regular structures, while some scenes may consist of more complex structures. For example, the forrest terrain scenes

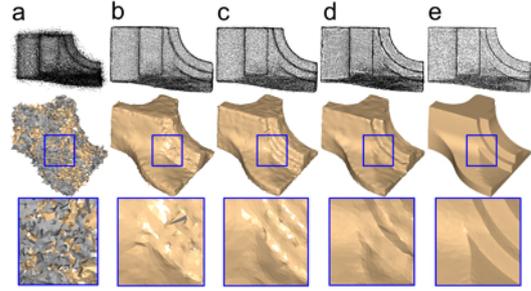


Figure 8: Comparison on the *fandisk* model. (a) The noisy point data (3% Gaussian noise); and the results from (b) Fleishman et al.'s [FDCO03] ($k = 1, n = 5$); (c) Zheng et al.'s [ZFAT11] ($n_1 = 5, \sigma_s = 0.35, n_2 = 10$); (d) Sun et al.'s [SRML07] ($n_1 = 30, n_2 = 20, T = 0.5$) methods and (e) Ours ($k = 50, \zeta = 0.16, n_p = 20$). The second row shows the reconstructed meshes from the corresponding point data in the first row. From the zoom-in views in the third row, sharp edges are better preserved from our method.

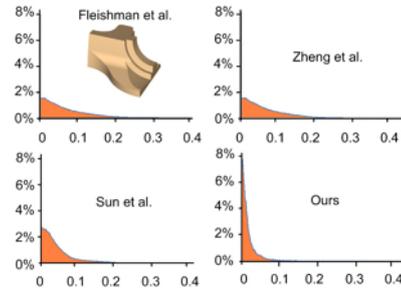


Figure 9: The histograms show the Hausdorff distances between the smoothed and groundtruth data from different methods. The horizontal axis is the error (absolute distance value), and the vertical axis is the corresponding percentage to each error value.

include trees, grasses, bushes and so on. In this situation, the consolidation method may not work well in terms of preserving features.

6. Conclusion

In this paper, we propose an effective and efficient consolidation method for point clouds from outdoor scenes, which are always contaminated with severe outliers and noise. The combination of connectivity-based and clustering-based outliers detection is capable of removing all types of outliers without any user interactions. A robust normal estimation method is proposed for noisy point clouds that preserves sharp features. Furthermore, we describe a robust, two-step denoising approach to process noisy point data, which is capable of removing noise effectively, while preserving sharp features. On a variety of experiments, our method achieves

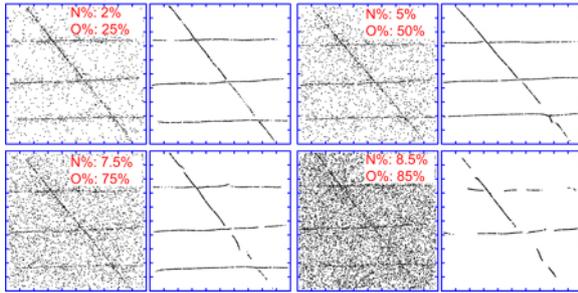


Figure 10: Processing results of 2D point data with different levels of outliers and noise. *N%* and *O%* are the percentages of noise and outliers, respectively. Note that we replace plane with line during processing. From the results, our method can tolerate 7.5% noise and 75% outliers. When the level of outliers reaches to 85%, the method tends to be unstable.

satisfactory results in the presence of a high level of noise and outliers.

Acknowledgement

We thank Shi-Min Hu for his thoughtful comments and the anonymous reviewers for their valuable suggestions. We are grateful to Hui Huang, Tai-Pang Wu, Qian Zheng, Chaohui Shen and the LGG at EPFL for providing data.

References

- [BKNS00] BREUNIG M. M., KRIEGEL H.-P., NG R. T., SANDER J.: Lof: identifying density-based local outliers. *SIGMOD Rec.* 29, 2 (2000), 93–104. 3
- [BO03] BELYAEV A., OHTAKE Y.: A comparison of mesh smoothing methods. In *In Proceedings of the Israel-Korea BiNational Conference on Geometric Modeling and Computer Graphics* (2003), pp. 83–87. 6
- [DPH*13] DEVORE R., PETROVA G., HIELSBERG M., OWENS L., CLACK B., SOOD A.: Processing terrain point cloud data. *SIAM Journal on Imaging Sciences* 6, 1 (2013), 1–31. 3
- [FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *SIGGRAPH '05*, pp. 544–552. 2, 9
- [FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *SIGGRAPH '03*, pp. 950–953. 9
- [FS13] FRIEDMAN S., STAMOS I.: Online detection of repeated structures in point clouds of urban scenes for compression and registration. *International Journal of Computer Vision* 102, 1-3 (2013), 112–128. 3
- [Haw80] HAWKINS D.: *Identification of Outliers*. Chapman and Hall, London, 1980. 3
- [HLZ*09] HUANG H., LI D., ZHANG H., ASCHER U., COHEN-OR D.: Consolidation of unorganized point clouds for surface reconstruction. *SIGGRAPH Asia '09*, pp. 176:1–176:7. 8
- [HWG*13] HUANG H., WU S., GONG M., COHEN-OR D., ASCHER U., ZHANG H. R.: Edge-aware point set resampling. *ACM Trans. Graph.* 32, 1 (2013), 9:1–9:12. 5, 6
- [Ken38] KENDALL M. G.: A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (June 1938), 81–93. 5
- [KKSZ09] KRIEGEL H.-P., KRÖGER P., SCHUBERT E., ZIMEK A.: Loop: local outlier probabilities. *CIKM '09*, pp. 1649–1652. 3
- [KMYG12] KIM Y. M., MITRA N. J., YAN D.-M., GUIBAS L.: Acquiring 3d indoor environments with variability and repetition. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 138:1–138:11. 1
- [KNRS12] KÄÜHLER J., NÄÜLL T., REIS G., STRICKER D.: Robust outlier removal from point clouds acquired with structured light. In *Eurographics* (2012). 2
- [LCW12] LIU S., CHAN K.-C., WANG C. C. L.: Iterative consolidation of unorganized point clouds. *IEEE Comput. Graph. Appl.* 32, 3 (May 2012), 70–83. 2, 8
- [LP05] LANGE C., POLTHIER K.: Anisotropic smoothing of point sets. *Comput. Aided Geom. Des.* 22, 7 (Oct. 2005), 680–692. 2
- [LSK*10] LI B., SCHNABEL R., KLEIN R., CHENG Z., DANG G., JIN S.: Robust normal estimation for point clouds with sharp features. *Comput. Graph.* 34, 2 (2010), 94–106. 5, 6
- [LW10] LIU S., WANG C. C. L.: Orienting unorganized points for surface reconstruction. *Comput. Graph.* 34, 3 (June 2010), 209–218. 6
- [LZS*11] LI Y., ZHENG Q., SHARF A., COHEN-OR D., CHEN B., MITRA N. J.: 2d-3d fusion for layer decomposition of urban facades. *ICCV '11*, pp. 882–889. 3
- [Nil86] NILSSON N.: Probabilistic logic. *Artificial Intelligence* 28 (1986), 71–87. 3
- [OBS05] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: An integrating approach to meshing scattered point data. *SPM '05*, pp. 61–69. 7, 9
- [PKGf03] PAPADIMITRIOU S., KITAGAWA H., GIBBONS P., FALOUTSOS C.: Loci: fast outlier detection using the local correlation integral. In *ICDE' 03* (2003), pp. 315–326. 3
- [SBS05] SCHALL O., BELYAEV A., SEIDEL H.-P.: Robust filtering of noisy scattered point data. *SPBG'05*, pp. 71–77. 2
- [SBS08] SCHALL O., BELYAEV A., SEIDEL H.-P.: Adaptive feature-preserving non-local denoising of static and time-varying range data. *Comput. Aided Des.* 40, 6 (June 2008), 701–707. 2
- [SHFH11] SHEN C.-H., HUANG S.-S., FU H., HU S.-M.: Adaptive partitioning of urban facades. *SA '11*, pp. 184:1–184:10. 3
- [Sot07] SOTOODEH S.: Hierarchical clustered outlier detection in laser scanner point clouds. In *Laser07* (2007), p. 383. 1
- [SRML07] SUN X., ROSIN P. L., MARTIN R. R., LANGBEIN F. C.: Fast and effective feature-preserving mesh denoising. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (Sept. 2007), 925–938. 9
- [WPK*04] WEYRICH T., PAULY M., KEISER R., HEINZLE S., SCANDELLA S., GROSS M.: Post-processing of scanned 3d surface data. *SPBG'04*, pp. 85–94. 2
- [ZFAT11] ZHENG Y., FU H., AU O. K.-C., TAI C.-L.: Bilateral normal filtering for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics* 17, 10 (Oct. 2011), 1521–1530. 9
- [ZSW*10] ZHENG Q., SHARF A., WAN G., LI Y., MITRA N. J., COHEN-OR D., CHEN B.: Non-local scan consolidation for 3d urban scenes. *SIGGRAPH '10*, pp. 94:1–94:9. 3