

Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction

Katja Wolff^{1,2} Changil Kim² Henning Zimmer¹ Christopher Schroers¹
Mario Botsch³ Olga Sorkine-Hornung² Alexander Sorkine-Hornung¹

¹Disney Research ²Department of Computer Science, ETH Zurich ³Bielefeld University

katja.wolff, sorkine, kimc@inf.ethz.ch
henning.zimmer, christopher.schroers, alex@disneyresearch.com botsch@techfak.uni-bielefeld.de

Abstract

Point sets generated by image-based 3D reconstruction techniques are often much noisier than those obtained using active techniques like laser scanning. Therefore, they pose greater challenges to the subsequent surface reconstruction (meshing) stage. We present a simple and effective method for removing noise and outliers from such point sets. Our algorithm uses the input images and corresponding depth maps to remove pixels which are geometrically or photometrically inconsistent with the colored surface implied by the input. This allows standard surface reconstruction methods (such as Poisson surface reconstruction) to perform less smoothing and thus achieve higher quality surfaces with more features. Our algorithm is efficient, easy to implement, and robust to varying amounts of noise. We demonstrate the benefits of our algorithm in combination with a variety of state-of-the-art depth and surface reconstruction methods.

1. Introduction

Acquiring the 3D geometry of real-world objects is a long standing topic in computer vision and graphics research, with many practical applications, ranging from scanning small objects up to modeling complete cities. Consequently, there is an abundance of 3D reconstruction techniques, which can be roughly classified into active techniques [3] relying on illuminating the scene (e.g. by lasers or structured light), and passive techniques that analyze a multitude of images of the scene and are thus referred to as multi-view stereo or photogrammetry methods [33]. The latter, image-based methods have a number of benefits compared to active techniques. One main advantage is that the capture process is simple and cheap, only requiring standard imaging hardware like consumer digital cameras. Additionally, image-based methods provide color information of the scene and offer high resolution scanning thanks to the advances in image sensors. A popular approach to image-based 3D reconstruction is to

first compute camera poses and then estimate per-view depth maps by finding corresponding pixels between views and triangulating depth [14]. All pixels are then projected into 3D space to obtain a point cloud, from which a surface mesh is extracted using point cloud meshing techniques [2].

As illustrated in Figure 1 (a)–(c), a downside of image-based methods is that they are prone to producing outliers and noise in the depth maps due to matching ambiguities or image imperfections (lens distortion, sensor noise, etc.). The resulting point clouds are thus often noisy, and even state-of-the-art meshing methods often fail to produce reasonable results. Typically, the meshes computed from such noisy point clouds either miss many details (when a lot of regularization is applied) or reconstruct wrong geometry such as disturbing blobs. A common remedy to reduce outliers in image-based methods is, similarly to the surface reconstruction, to use strong smoothing or regularization in the depth computation, but this inevitably destroys fine details and is also costly to compute as it typically comes down to solving large global optimization problems.

We take a different approach in this paper. Starting from many, high resolution input images of the scene, we compute per-view depth maps using a depth estimation method of choice, but preferably one with little to no regularization (such as [21]) to reconstruct as much detail as possible. Our main idea is then to detect and remove noisy points and outliers from each per-view point cloud by checking if points are consistent with the surface implied by the other input views. Not only do we evaluate geometric consistency, but also consider photometric consistency between the input views, which improves the robustness of the method and is typically not possible for active techniques such as laser scanning. As shown in Figure 1 (d)–(e), merging the denoised point clouds from all views retains full coverage of the captured scene while being more compact and less noisy. This renders the subsequent surface reconstruction less demanding, allowing common techniques to produce favorable surface meshes with a high degree of detail.

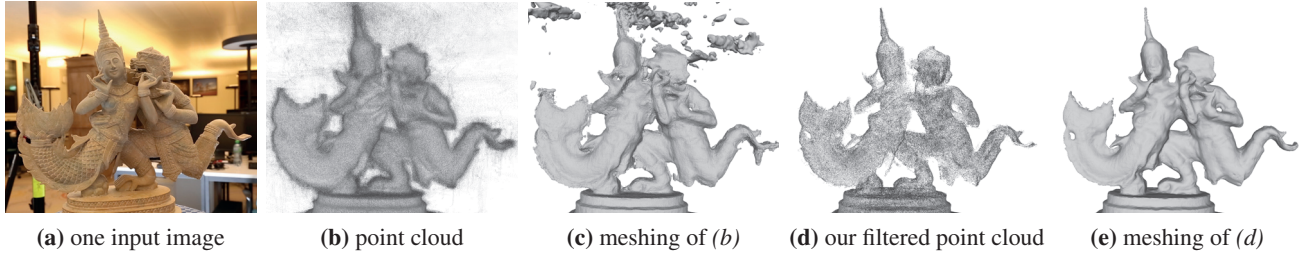


Figure 1. From a set of images of a scene (a), multi-view stereo methods such as [9] can reconstruct a dense 3D point cloud (b), which however often suffers from noise and outliers. This leads to disturbing artifacts when used in subsequent surface reconstruction (meshing) methods such as [20] (c). We propose a simple and efficient filtering method that produces clean point clouds (d) that allow for a favorable surface reconstruction (e).

Our method is simple to implement and easy to parallelize, while effective in removing noise and outliers in point clouds. As shown in Section 4, it can handle varying amounts and types of noise produced by several multi-view stereo methods [6, 9, 21, 45] and tangibly improves the results of various surface reconstruction techniques [5, 8, 20, 39] that are subsequently applied. Our method works with virtually any image-based technique that reconstructs scene geometry in the form of depth maps and any surface reconstruction method based on point sets. We thus believe that our method is a versatile tool bridging the two steps of image-based 3D reconstruction and facilitating the standard workflow. We demonstrate the benefits of our method on a variety of dense and high-resolution multi-view datasets.

2. Related work

Active 3D acquisition techniques, such as laser scanning and structured light approaches, have been predominantly used in professional domains, as they provide high accuracy albeit requiring specialized and expensive equipment. Due to their limitations (e.g. the size of scannable objects) and the restricted environment and illumination conditions, passive image-based techniques have also been developed and deployed widely. However, such image-based multi-view stereo methods are much more susceptible to produce noisy depth estimates due to image imperfections, triangulation inaccuracy, depth quantization, as well as outliers due to matching ambiguities and non-diffuse surfaces. For these reasons, image-based 3D reconstruction pipelines perform denoising and outlier removal at virtually every step of the pipeline, as outlined below.

Most multi-view stereo methods refine the reconstructed depth maps, and often this is integrated into the depth estimation stage and formulated as a (global) optimization problem [12, 45]. Furukawa et al. [10] use a filter based on quality and visibility measures for merging points while handling errors and variations in reconstruction quality. Tola et al. [38] use a robust descriptor for large-scale multi-view stereo matching in order to reduce the amount of outliers in

the computed point cloud. However, as shown in Section 4, these approaches still often leave a significant amount of noise and outliers in the final reconstructions, necessitating additional outlier removal steps to render the point sets suitable for later surface reconstruction. Among a few such attempts, Shan et al. [34] reconstruct dense depth maps from sparse point clouds and use them to remove points that are in significant visibility conflict and to augment the input point cloud. However, they treat each view separately when densifying the sparse depth maps and they need to modify the standard Poisson surface reconstruction method. Similarly, a free space constraint was used to clean up depth maps in [29] and [27].

While the above techniques are presented as part of depth reconstruction methods, there exist more dedicated point cloud denoising and outlier removal techniques. Sun et al. [37] propose a point cloud denoising method imposing sparsity of the solution via L_0 minimization. The method optimizes both point normals and positions with the piecewise smoothness assumption, thereby preserving sharp features. Rusu et al. [31] present a point cloud refinement method with the application of indoor environment mapping. They propose an outlier removal technique based on statistical analysis of input points. Both methods consider the point positions only and do not consider further information like color or scanner positions. Rusu et al.’s method explicitly assumes laser scanning as the point input source. Yücer et al. [43] use accurate foreground segmentation of a dense image set to refine the bounding volume of the object, resulting in a detailed visual hull that is subsequently used to filter outliers from the point cloud. However, the visual hull does not filter points in concavities and may not be tight enough.

Since geometry acquisition inevitably includes measurement noise at varying degrees, many surface reconstruction methods provide some form of smoothing mechanisms to deal with the acquisition noise and to adapt to the varying quality of the acquired point clouds. A family of methods uses moving least-squares (MLS) to resample the input point cloud to a potentially smoother and more uniform point set by projecting points onto a locally fitted smooth surface

represented by a low-degree polynomial [1, 13]. Instead of computing local projections, implicit moving least-squares (IMLS) methods [35] allow to reconstruct an implicit representation of the surface. Although IMLS becomes more robust to noise and also preserves sharp features when using robust statistics [30], it still does not handle outliers very well. Similarly, the parameterization-free projection operator [25] results in a resampled point cloud by means of point projections, but onto a multivariate median, being more robust to noise and able to detect outliers. By taking into account the point density, the method was extended to deal with sharp features [18] and a high level of non-uniformity [17]. This last work led to a class of methods very relevant to our method, called *point consolidation*. These methods include multiple stages of point cloud processing, from merging points to denoising, decimating, and redistributing them such that they become more suitable for later surface reconstruction [17]. The recent work of Wu et al. [42] further completes the missing parts of a scanned object by utilizing point skeleton estimation. Our method also proposes to facilitate the surface reconstruction, but exploits the information available exclusively to the image-based reconstruction workflows, namely, color information and 3D camera poses, which purely geometry-based methods usually do not have access to.

Streaming surface reconstruction using wavelets [26] allows for fast processing of large point clouds but is only resilient to a low amount of noise. The popular Poisson surface reconstruction technique [19] estimates a smoothed indicator function of a surface by minimizing the distance between the smoothed gradient of the unknown indicator function and the smoothed surface normal vector field implied by the oriented points. This renders the method resilient to noise, but at the cost of overly smooth reconstructions. In a recent extension, the energy functional includes a screening term, such that the influence of the original point positions can be adapted [20]. Still, noisy point clouds require low screening, resulting in smooth reconstructions and losing detailed features. Similar restrictions apply to other methods, that explicitly model a smoothness assumption [5, 32, 41].

Our method implicitly uses a surface represented by the input depth maps when examining each point, similarly to range image integration methods such as [7, 15] and the more recent KinectFusion [28]. While most existing methods use a volumetric representation to cache the implicit function in 3D space, our algorithm operates directly in image space, avoiding premature explicit discretization and large memory usage. We use a photo-consistency criterion in our filter, which was first proposed in the space-carving literature [23]. Despite that, color information has rarely been used for surface reconstruction or outlier removal, except for semantic analysis; see [2] for details. We use color information in conjunction with the input point geometry.

In general, there have been many filtering approaches for image-based reconstruction pipelines [4, 22, 24, 36, 40, 44], but the combination of ideas proposed in this paper has not been considered before.

3. Denoising and outlier removal

Our denoising algorithm removes inconsistent points from a set of input depth maps $\{D_i \mid i = 1, \dots, N\}$ by analyzing their geometric and photometric consistency with other views.

3.1. Geometric consistency

To determine the geometric consistency, each 3D point $\mathbf{p}$ originating from a depth map has to be examined against all other depth maps. For this purpose, we measure how far $\mathbf{p}$ is from the true surface by estimating and examining the signed distance of $\mathbf{p}$ to the surface entailed by the input depth maps. Since the actual surface is yet to be known and the estimation of signed distances at all $\mathbf{p}$ would be expensive, we utilize several steps of efficient approximation, which were inspired by range image integration methods.

The depth maps are first trivially tessellated and back-projected to represent triangulated range surfaces, as illustrated in Figure 2(a) and (b). Ill-shaped triangles having an angle less than a threshold (1° in our implementation) are removed to permit opening concavities over depth discontinuities. We intend to compute the average distance of each point $\mathbf{p}$ to the range surfaces. However, computing the distance from a 3D point to a number of meshes may potentially require building spatial acceleration structures and multiple point-to-triangle projections, which is computationally expensive. Further, as we deal with noisy surfaces, we need to assure that the distance estimation is robust enough.

Instead of computing the exact point-mesh distance, we calculate the distance along the viewing ray from the camera center $\mathbf{v}_i$ to point $\mathbf{p}$. This still requires to intersect the ray with the triangulated range surface, but since the range surface is simply the back-projection of the depth map D_i , the intersection can be efficiently calculated by projecting $\mathbf{p}$ to the image space of D_i . Then the vertices of the 2D triangle in the tessellated depth map into which $\mathbf{p}$ was projected correspond to the vertices of the intersected triangle in 3D. The depth at the intersection point is interpolated barycentrically from the three vertices. We approximate the signed distance $d_i(\mathbf{p})$ between $\mathbf{p}$ and the range surface of depth map D_i by the z -distance between $\mathbf{p}$ and the intersection point in camera space, i.e.,

$$d_i(\mathbf{p}) = z_i(\mathbf{p}) - z, \quad (1)$$

where z is the depth (z -coordinate) of $\mathbf{p}$ and $z_i(\mathbf{p})$ is the interpolated depth at the intersection.

When considering the distance of $\mathbf{p}$ to a range surface D_i , a negative distance d_i implies that $\mathbf{p}$ lies behind the range surface and could not have been observed from this view.

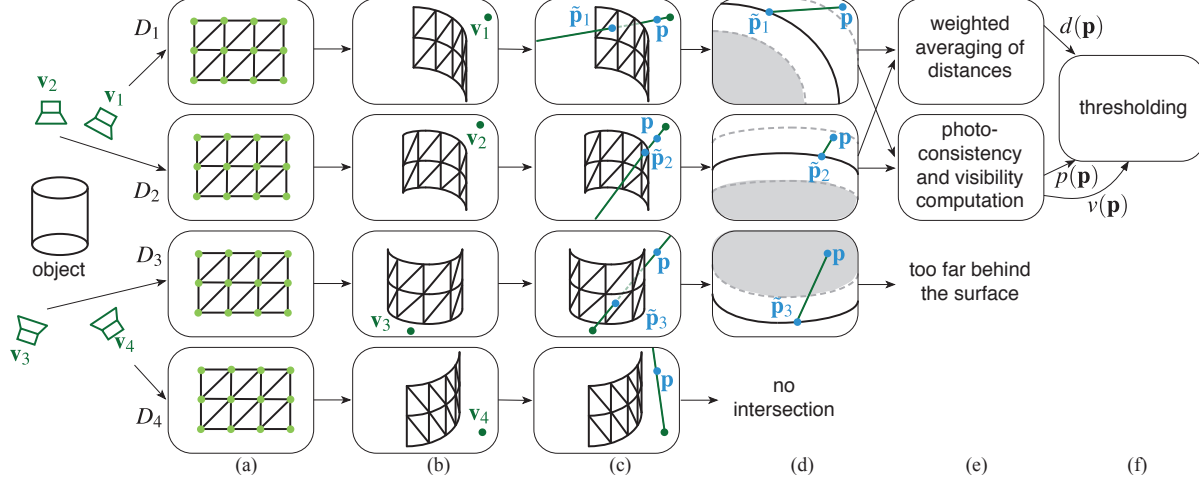


Figure 2. Our point denoising pipeline: An object is captured from different views, with camera positions $\mathbf{v}_i$, resulting in several depth maps D_i . The depth maps are triangulated (a) and represent range surfaces in 3D space (b). For each point $\mathbf{p}$ from each depth map, intersection points $\tilde{\mathbf{p}}_i$ with all other depth maps are calculated (c). (We do not display the fifth depth map here from which $\mathbf{p}$ originates.) Color, depth and weight values are available at the triangle vertices and can be interpolated for the intersection point. The signed distances between $\mathbf{p}$ and the intersection points $\tilde{\mathbf{p}}_i$ (green lines in (d)) are approximated. Only range images for which $\mathbf{p}$ does not lie too far behind the surface (gray area in (d)) are considered further. A weighted average of the signed distances $d(\mathbf{p})$ is calculated together with a photo-consistency measure $p(\mathbf{p})$ and visibility measure $v(\mathbf{p})$ (e). All three values are used to decide whether $\mathbf{p}$ should be discarded (f).

Therefore, we discard such d_i in computing the weighted average. Allowing for a certain error margin, we define an indicator function that specifies whether a point lies no farther than a certain distance σ behind the surface:

$$\mathbb{I}_{\sigma}^G(d_i) = \begin{cases} 1 & \text{if } -\sigma < d_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

A large positive distance d_i implies that $\mathbf{p}$ could have been seen from this view but is far away from the actually observed range surface. To limit the influence of these outliers, we truncate the signed distance d_i to σ if $d_i > \sigma$, but *still* include it in the computation of the weighted average since it has been seen from this view and makes the average computation more robust against cases where $\mathbf{p}$ is an outlier (instead of a depth value $d_i(\mathbf{p})$). In cases where no intersection exists, e.g., D_4 in Figure 2, the range surface is not further considered for the distance calculation for $\mathbf{p}$.

Additionally, to reflect greater uncertainty when a point $\mathbf{p}$ from the range image D_i has been observed at a grazing angle to the surface, we introduce the weight

$$w_i(\mathbf{p}) = \mathbf{n}(\mathbf{p})^T \frac{\mathbf{p} - \mathbf{v}_i}{\|\mathbf{p} - \mathbf{v}_i\|}, \quad (3)$$

where $\mathbf{n}(\mathbf{p})$ is the point normal at $\mathbf{p}$. The weight w_i measures the similarity between the viewing direction $\mathbf{p} - \mathbf{v}_i$ and the normal direction $\mathbf{n}$ at $\mathbf{p}$ and thus becomes small in absolute value at a grazing angle. Point normals are calculated using principal component analysis of their image-space neighbors [16] and are oriented towards the camera center, hence

$w_i(\mathbf{p}) > 0$. Although more sophisticated normal estimation could be used, we found this method to be fast and sufficient for our purposes.

Depth maps from opposite sides of the object do only overlap in small regions, usually at grazing angles, which makes these observations unreliable without contributing much to the overall distance estimate. To significantly decrease computation time, we do not consider depth maps whose viewing direction $\mathbf{v}_j$ differs too much from the viewing direction $\mathbf{v}_i$ under which $\mathbf{p}$ was observed, by limiting the angle between both viewing directions. Keeping only depth maps for an angle smaller than 90° , i.e., $\mathbf{v}_j^T \mathbf{v}_i > 0$, yields good results in our implementation.

We finally compute the signed distance d to the surface as a weighted average over all range surfaces:

$$d(\mathbf{p}) = \frac{1}{w(\mathbf{p})} \sum_i \mathbb{I}_{\sigma}^G(d_i(\mathbf{p})) w_i(\mathbf{p}) \min\{d_i(\mathbf{p}), \sigma\}. \quad (4)$$

In practice, the weight w_i is calculated only at vertices of the range image and interpolated in the same manner as for the signed distance in Equation 1. The normalization factor $w(\mathbf{p})$ is the summation of all weights: $w(\mathbf{p}) = \sum_i \mathbb{I}_{\sigma}^G(d_i(\mathbf{p})) w_i(\mathbf{p})$. Note that $\mathbf{p}$ itself and its weight are included in the average, with the distance of 0 to the range surface it originates from, since we want to compute the distance to an averaged surface from *all* depth maps.

3.2. Photometric consistency

In addition to the geometric consistency, the consistency of colors of intersection points, as well as the visibility of

$\mathbf{p}$ are calculated. In contrast to the averaging of distances, where outliers are truncated, we only want to consider range surfaces that lie close to the point $\mathbf{p}$, as only they provide reliable color estimates. To this end, we define a second indicator function that is similar to the first, but now encodes whether a point is closer to the range surface than the distance σ for both positive and negative directions:

$$\mathbb{I}_{\sigma}^P(d_i) = \begin{cases} 1 & \text{if } -\sigma < d_i < \sigma \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

We use the same σ for both indicator functions.

The visibility is obtained by simply counting the depth maps that fall into this margin and thus contribute to the color consistency calculation:

$$v(\mathbf{p}) = \sum_i \mathbb{I}_{\sigma}^P(d_i(\mathbf{p})), \quad (6)$$

which gives us an estimate of the number of depth maps in which $\mathbf{p}$ is visible.

The photometric consistency is measured by the standard deviation of the color distribution:

$$p(\mathbf{p}) = \left(\frac{1}{v(\mathbf{p})} \sum_i \mathbb{I}_{\sigma}^P(d_i(\mathbf{p})) \|\mathbf{c}_i(\mathbf{p})\|^2 - \frac{1}{v(\mathbf{p})^2} \left\| \sum_i \mathbb{I}_{\sigma}^P(d_i(\mathbf{p})) \mathbf{c}_i(\mathbf{p}) \right\|^2 \right)^{1/2}, \quad (7)$$

where $\mathbf{c}_i$ denotes the (interpolated) color value at the intersection of $\mathbf{p}$ and the range surface of D_i .

3.3. Point filtering

The last step is to decide whether $\mathbf{p}$ should be kept based on its geometric and photometric consistency. We retain a point if it satisfies all of the following three conditions:

$$-t_d < d(\mathbf{p}) < 0, \quad p(\mathbf{p}) < t_p, \quad v(\mathbf{p}) > t_v, \quad (8)$$

where $t_d < \sigma$, t_p , and t_v are thresholds for distance, photometric consistency, and visibility, respectively.

While σ influences the possible thickness of reconstructed features of the object, t_d decides how much deviation from the surface we allow and thus controls the level of removed noise. A small value of t_d reduces the number of retained points significantly and results in smoother mesh reconstructions from the filtered point clouds. If the input depth maps are already sparse, a higher value should be chosen. In practice, choosing t_d as a fixed ratio of σ (e.g. $t_d = 0.1 \cdot \sigma$ in all our examples) and only adjusting σ to the object scale works well.

The choice of keeping only points with a negative signed distance to the surface (first condition of Equation 8) is based on the observation that most of the noise appears on the “outside” of the surface, which can be attributed to the image-based capturing process. The simple trick of retaining points only on the inside removes most of such noise. Figure 3 shows the effects of doing so.

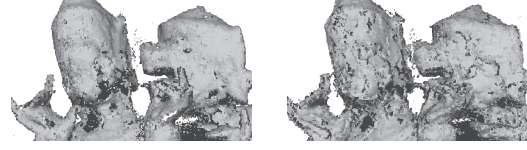


Figure 3. On the left we show a close-up of a denoised point cloud with $t_d = \sigma$ and $-t_d < d < 0$. On the right we use $-t_d < d < t_d$ and set $t_d = 0.5 \cdot \sigma$ to keep the interval size the same and the comparison fair. The amount of noise is visibly reduced in the first approach.

4. Results

In this section, we validate our denoising and outlier removal algorithm on several multi-view image datasets. Since our method is designed to work with existing multi-view depth and surface reconstruction methods, we provide results with a selection of such methods. For all depth and surface reconstructions presented in our paper, we hand-picked the parameters so as to achieve the best possible results. For our own method, we used fixed parameters for all results. The value of σ should be chosen according to the scale of the scene, so we set it to 1% of the depth range (e.g., the length of the bounding box along the z -axis); we set $t_{\sigma} = 0.1 \sigma$. The visibility parameter t_v is set to be 7.5% of the number of input depth maps. For the photo-consistency threshold, we always set $t_p = 0.2$. To ease reproducibility, the supplementary material accompanying this paper includes noisy input point clouds, our denoised point clouds, as well as the parameters used for the meshing.

Results for different depth reconstruction algorithms.

Figure 4 shows the reconstructed surfaces from several datasets recently released by Yücer et al. [43]. These datasets feature a very dense sampling of the scene in terms of views per baseline and also offer a high spatial resolution, potentially allowing to reconstruct a high degree of detail, but also challenging the computational efficiency of reconstruction methods. We used four different dense multi-view depth reconstruction algorithms with different algorithmic principles, levels of regularization, and noise and outlier characteristics. While Fuhrmann et al. (MVE) [9] and Zhang et al. (ACTS) [45] use sophisticated global regularization, Kim et al. (LFD) [21] use local regularization only, and our implementation of the plane-sweep algorithm (PS) [6] uses no regularization at all. We used screened Poisson surface reconstruction (PSR) [20] for surface reconstruction as it is very resilient to input noise and also widely used. Each pair of images in Figure 4 shows the results without and with our denoising algorithm.

We used about 200 input views for all depth reconstruction methods. For MVE we used the level-2 depth maps ($4 \times$ downsampling) as advised in the paper and the software documentation. The LFD method proposes a simplistic outlier

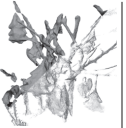
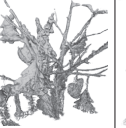
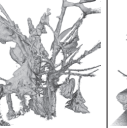
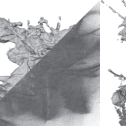
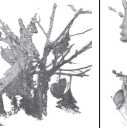
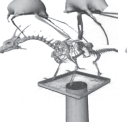
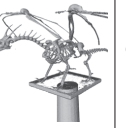
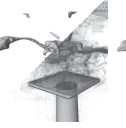
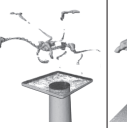
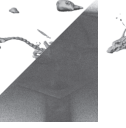
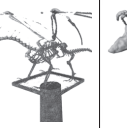
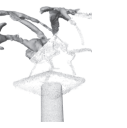
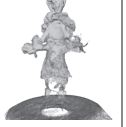
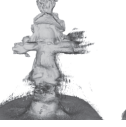
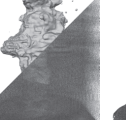
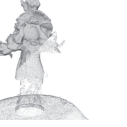
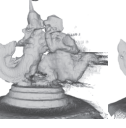
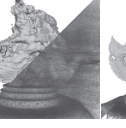
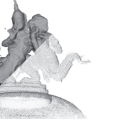
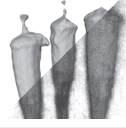
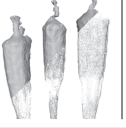
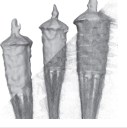
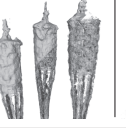
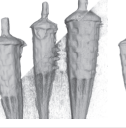
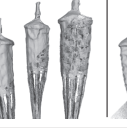
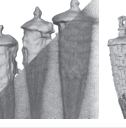
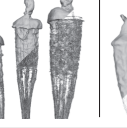
Datasets	MVE		LFD		ACTS		PS		PMVS unfiltered
	unfiltered	with our filter	unfiltered	with our filter	unfiltered	with our filter	unfiltered	with our filter	
									
									
									
									
									

Figure 4. Meshes and point clouds (shown in *upper* and *lower* triangles, respectively) obtained on dense multi-view datasets [43] using various depth estimation methods: Multi-view environment (MVE) [9], light field depth reconstruction (LFD) [21], dense depth reconstruction from video (ACTS) [45], and the plane-sweep algorithm (PS) [6]. The point clouds were meshed using the screened Poisson surface reconstruction (PSR) [20] without our denoising method (*left* in each pair) and after our denoising (*right* in each pair). We hand-tuned all parameters of the depth estimation methods and PSR to achieve the best possible results. We also show the result of PMVS [11] in the last column as reference. Please see the supplementary material for a more extensive presentation of these results.

filtering step which we disabled when applying our filter (to not filter twice), but we keep it enabled for the baseline results. ACTS required the input images to be at the resolution of about 720p HD. Since the resulting point clouds often contained multi-million points, we had to decimate some input point clouds so that PSR can run with the available memory (64 GB on our machine). In such cases, which only happened with unfiltered point clouds, we downsampled the input images using bicubic interpolation until PSR could process them while keeping the number of views the same. Note that we did not have to downsample the images for our denoised results, allowing us to use the full input resolution.

As can be seen in Figure 4, the results of MVE exhibit outliers that are more structured and consistent across views, hence forming areas with densely clustered points. Thus, it is generally more difficult for a denoising algorithm to detect them as outliers or noise, often leaving them as features. However, our method measures the photometric consistency as well, rendering it easier to detect such outliers than solely with geometric consistency. Without removal of these outliers, we had to use a higher amount of smoothing to remove the clutter, which was often impossible without

removing features. To demonstrate our method’s robustness to noise, we reconstructed scene depth using the LFD method that only performs local regularization, and also a simple plane-sweeping (PS) algorithm with *no* regularization. The resulting point clouds show an extreme amount of noise, but also capture a lot of details. As can be seen, our method is able to remove most of the noise and allows the subsequent surface reconstruction to yield favorable meshes with many details preserved. A similar observation can be made for ACTS which results in less noise due to the global regularization but still produces too many outliers to make meshing feasible from unfiltered point clouds.

As a comparison we also show results of PMVS which does not produce dense depth maps, but densifies sparse points. While yielding reasonable results in general, dense depth methods in combination with our denoising often produce more favorable meshes with more details revealed.

Results for different meshing algorithms. As shown in Figure 4, PSR handles noisy input very well, but at the price of increased smoothing and less accurate feature localization, which sometimes results in missing features. Without

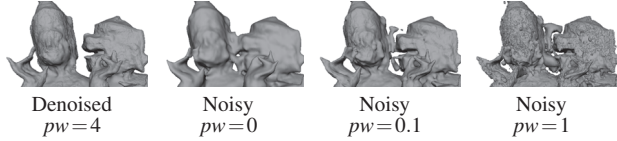


Figure 5. Comparison between our algorithm and the smoothing of PSR with varying amount of screening. Leftmost: mesh reconstructed from our denoised point cloud with a large screening weight $pw=4$ and 5 samples per node (spn). To the right: meshes reconstructed from the noisy point cloud for different screening weights. We had to use very low screening due to the noisy input and use 20 spn to achieve smoother results. Without our denoising, the resulting meshes are either too noisy or lack detail.

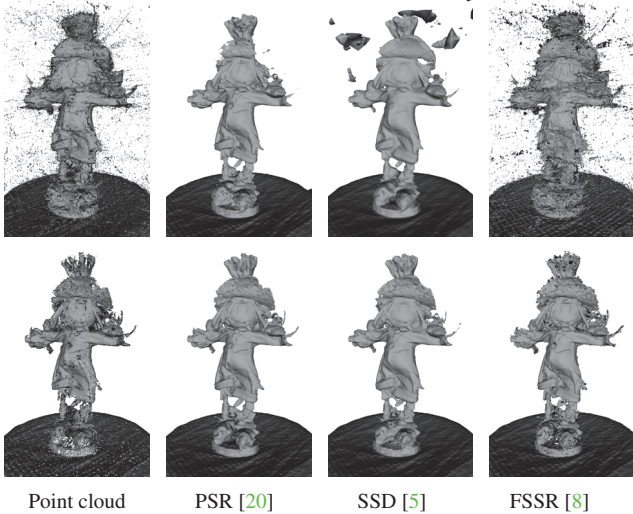
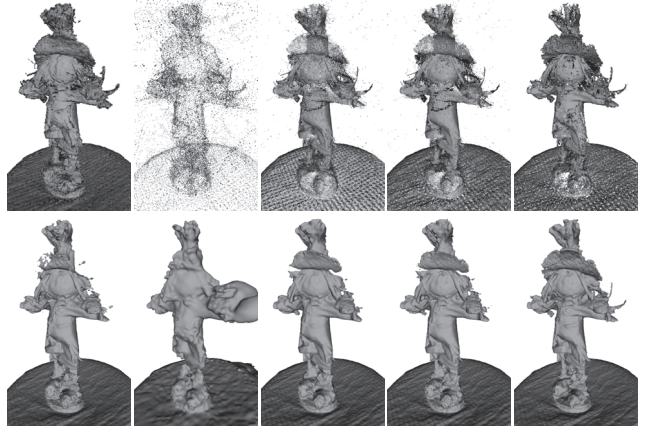


Figure 6. The first column shows an input point cloud (top) and our denoising result (bottom). The remaining columns compare meshes reconstructed using different surface reconstruction techniques. The meshes computed from the denoised points always exhibit more details and less artifacts.

strong smoothing, the reconstructed surface is rough and includes substantial amount of clutter; see Figure 5 for the results of varying PSR parameters. We typically used a higher screening term for our denoised results and a very low to no screening term for noisy input point clouds.

While PSR is very resilient to noise, other surface reconstruction methods tend to respect input points more. In such cases, our method can be even more valuable. Figure 6 shows the meshing results of noisy input and our denoised point clouds using a range of widely used surface reconstruction techniques. The effect of our method is consistent across different meshing techniques.

Comparison against other denoisers. In Figure 7, we compare our method with other point cloud denoising, smoothing, or resampling methods. The point cloud consolidation method (WLOP) of Huang et al. [17] results in



Rusu et al. [31] RIMLS [30] WLOP [17] EAR [18] Ours

Figure 7. Comparison of our denoising method with other outlier removal, resampling, or smoothing methods. Top: Filtered point clouds, bottom: corresponding meshes computed using PSR. Our result exhibits the most detail and least amount of artifacts.

very smooth, clean point clouds, however, lacking detailed features. Edge-aware resampling of Huang et al. [18] also presents very smooth results and while succeeding at removing noise, both methods left a significant amount of outliers. We also tried the more recent work of Wu et al. [42], but since our input point clouds do not include missing parts, the effect was negligible. Robust IMLS of Öztireli et al. [30] produces a relatively sparse point cloud and suffers from many outliers. Rusu et al.’s [31] outlier removal method successfully removes outliers, but did not handle noisy points. Also none of these methods uses color information to remove the noise or outliers, whereas our method handles such noisy point clouds using the information that is available for image-based techniques, but that is ignored by methods that only process oriented point clouds.

Comparison to ground truth data. To assess the results more quantitatively, we measured the bias of the reconstructed meshes from ground truth results. Figure 8 shows the errors of reconstructed DRAGON meshes taken from Figure 4. We evaluate the accuracy and completeness of each mesh according to the metrics used in the Middlebury multi-view stereo benchmark [33]. The meshes are color-coded with green indicating no error and where blue and red denoting negative (surfaces placed inside the ground truth) and positive errors, respectively. We observe that meshes resulting from our denoising algorithm consistently mark higher scores for all depth reconstruction methods.

Performance analysis. Figure 9 summarizes the performance of our algorithm, where accuracy and completeness errors as well as the runtime were measured with varying number of input depth maps. It takes about 30 seconds to

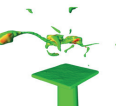
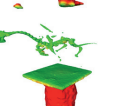
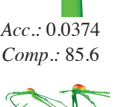
	MVE	LFD	ACTS	PS
PSR				
Ground truth				
Ours + PSR				
	Acc.: 0.0953 Comp.: 64.4	Acc.: 0.0374 Comp.: 85.6	Acc.: 0.0373 Comp.: 38.4	Acc.: 0.2107 Comp.: 50.5
	Acc.: 0.0647 Comp.: 74.0	Acc.: 0.0218 Comp.: 91.6	Acc.: 0.0244 Comp.: 47.3	Acc.: 0.0385 Comp.: 86.0

Figure 8. Quantitative evaluation of the DRAGON meshes using different depth reconstruction methods and PSR meshing. The top and bottom rows show the results without and with our denoising algorithm, respectively. We measured errors in terms of accuracy (in world units; the lower, the better) and completeness (in percent; the higher, the better), using an accuracy threshold of 90%, and a completeness threshold of 0.1 world units.

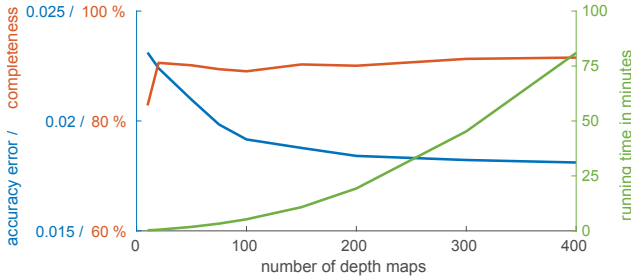


Figure 9. The run-time and output quality of our method with varying numbers of input depth maps calculated on the DRAGON dataset for the LFD method.

process 20 depth maps at 1920×1080 resolution, about 5 minutes for 100, and 20 minutes for 200 depth maps, using our simple, OPENMP-based parallel implementation on a 3.2 GHz 12-core Intel CPU. As can be seen in the graph, with more than 200 depth maps as input, the output quality does not change much while the runtime increases further. When comparing different datasets using the PS method, the running time for 200 depth maps is 10 minutes for DECORATION, 13 for DRAGON, 16 for SCARECROW and STATUE, and 7 minutes for TORCH.

The time complexity of our algorithm is $\mathcal{O}(MN) = \mathcal{O}(KN^2)$, where M is the number of input points, i.e., all pixels from all N depth maps, and K is the depth map resolution (thus $M = KN$). With small N and a parallel implementation, the complexity becomes close to $\mathcal{O}(M)$, but the complexity increases quadratically with the number of depth maps N . Still, the algorithm can process large datasets since it does not perform costly optimizations nor requires much additional memory except for the point set itself. Often the worst

case of $\mathcal{O}(NM)$ will not be reached, as we do not compare depth maps from opposite sides of an object.

Limitations. As we rely on the redundancy of points and need to calculate intersections with a range surface formed by the depth maps, our method might fail for very sparse input, e.g., very sparse depth maps (such as those reprojected from a sparse point cloud), or for a low number of depth maps. Also when the input images are taken under vastly different lighting situations, the photo-consistency calculation might be inaccurate. To mitigate this problem we can choose a higher photo-consistency threshold, which however reduces the efficiency of the filtering.

5. Conclusions

We presented an efficient, simple, and robust algorithm for noise and outlier removal from the often extremely noisy point sets generated by image-based 3D reconstruction techniques. Our method reduces the amount of erroneous and extraneous points in the input, which significantly improves the reconstruction quality while reducing the computational and storage overhead. We demonstrated the benefits of our method in conjunction with a variety of existing depth estimation and surface reconstruction techniques and thus believe that we presented a practical and useful tool for virtually any image-based 3D reconstruction workflow.

Classic multi-view reconstruction methods often perform costly optimizations for smoothing and regularizing the results, which removes a significant amount of detail present in the scanned scene. With our method, simple reconstruction techniques without much (or any) smoothing, e.g. [6, 21], can be used to create over-redundant points. As shown in the experiments, our method is able to reduce these large and noisy point clouds so that meshing becomes feasible and often even produces more accurate surface reconstructions that preserve many details. We hence hope that our method opens up the door to fundamentally novel basic approaches for image-based 3D reconstruction.

Acknowledgements. This work was supported by the NCCR Digital Fabrication, funded by the Swiss National Science Foundation, NCCR Digital Fabrication Agreement #51NF40-141853. Mario Botsch was supported by the Cluster of Excellence Cognitive Interaction Technology CITEC (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

- [1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Trans. Vis. and Comput. Graph.*, 9(1):3–15, 2003. 3

- [2] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva. State of the art in surface reconstruction from point clouds. In *Eurographics 2014 - State of the Art Reports*, pages 161–185, 2014. 1, 3
- [3] F. Blais. Review of 20 years of range sensor development. *J. Electronic Imaging*, 13(1):231–243, 2004. 1
- [4] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proc. CVPR*, pages 1–8, 2008. 3
- [5] F. Calakli and G. Taubin. SSD: smooth signed distance surface reconstruction. *Comput. Graph. Forum*, 30(7):1993–2002, 2011. 2, 3, 7
- [6] R. T. Collins. A space-sweep approach to true multi-image matching. In *Proc. CVPR*, pages 358–363, 1996. 2, 5, 6, 8
- [7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. ACM SIGGRAPH*, pages 303–312, 1996. 3
- [8] S. Fuhrmann and M. Goesele. Floating scale surface reconstruction. *ACM Trans. Graph.*, 33(4):46:1–46:11, 2014. 2, 7
- [9] S. Fuhrmann, F. Langguth, and M. Goesele. MVE – a multi-view reconstruction environment. In *Eurographics Workshop on Graphics and Cultural Heritage*, pages 11–18, 2014. 2, 5, 6
- [10] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Proc. CVPR*, pages 1434–1441. IEEE, 2010. 2
- [11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 6
- [12] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *Proc. ICCV*, pages 1–8, 2007. 2
- [13] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26(3), 2007. 3
- [14] A. Hartley and A. Zisserman. *Multiple view geometry in computer vision (2nd ed.)*. Cambridge University Press, 2006. 1
- [15] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Winder. Reliable surface reconstruction from multiple range images. In *European conference on computer vision*, pages 117–126, 1996. 3
- [16] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. ACM SIGGRAPH*, pages 71–78, 1992. 4
- [17] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.*, 28(5):176:1–176:7, 2009. 3, 7
- [18] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang. Edge-aware point set resampling. *ACM Trans. Graph.*, 32(1):9:1–9:12, 2013. 3, 7
- [19] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. SGP*, pages 61–70, 2006. 3
- [20] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, 2013. 2, 3, 5, 6, 7
- [21] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73:1–73:12, 2013. 1, 2, 5, 6, 8
- [22] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. 3
- [23] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 3
- [24] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *Proc. CVPR*, pages 1–8. IEEE, 2007. 3
- [25] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, 26(3), 2007. 3
- [26] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. *Comput. Graph. Forum*, 27(5):1411–1420, 2008. 3
- [27] P. Merrell, A. Akbarzadeh, L. Wang, J.-M. Frahm, and R. Y. D. Nistr. Real-time visibility-based fusion of depth maps. In *Proc. CVPR*, 2007. 2
- [28] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality, ISMAR*, pages 127–136, 2011. 3
- [29] D. Nister. *Automatic Dense Reconstruction from Uncalibrated Video Sequences*. Numerisk analys och datalogi, Stockholm, 2001. 2
- [30] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *Comput. Graph. Forum*, 28(2):493–501, 2009. 3, 7
- [31] R. B. Rusu, Z. C. Marton, N. Blodow, M. E. Dolha, and M. Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008. 2, 7
- [32] C. Schroers, S. Setzer, and J. Weickert. A variational taxonomy for surface reconstruction from oriented points. *Comput. Graph. Forum*, 33(5):195–204, 2014. 3
- [33] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, pages 519–528, 2006. 1, 7
- [34] Q. Shan, B. Curless, Y. Furukawa, C. Hernández, and S. M. Seitz. Occluding contours for multi-view stereo. In *Proc. CVPR*, pages 4002–4009, 2014. 2
- [35] C. Shen, J. F. O’Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph.*, 23(3):896–904, 2004. 3
- [36] S. N. Sinha, P. Mordohai, and M. Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *Proc. ICCV*, pages 1–8. IEEE, 2007. 3
- [37] Y. Sun, S. Schaefer, and W. Wang. Denoising point sets via L0 minimization. *CAGD*, 35(C):2–15, 2015. 2
- [38] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012. 2

- [39] B. Ummenhofer and T. Brox. Global, dense multiscale reconstruction for a billion points. In *Proc. ICCV*, pages 1341–1349, 2015. [2](#)
- [40] G. Vogiatzis, P. H. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proc. CVPR*, volume 2, pages 391–398. IEEE, 2005. [3](#)
- [41] C. Walder, B. Schölkopf, and O. Chapelle. Implicit surface modelling with a globally regularised basis of compact support. *Comput. Graph. Forum*, 25(3):635–644, 2006. [3](#)
- [42] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or. Deep points consolidation. *ACM Trans. Graph.*, 34(6):176:1–176:13, 2015. [3](#), [7](#)
- [43] K. Yücer, A. Sorkine-Hornung, O. Wang, and O. Sorkine-Hornung. Efficient 3D object segmentation from densely sampled light fields with applications to 3D reconstruction. *ACM Trans. Graph.*, 35(3):22:1–22:15, 2016. [2](#), [5](#), [6](#)
- [44] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. ICCV*, pages 1–8. IEEE, 2007. [3](#)
- [45] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009. [2](#), [5](#), [6](#)