Vision, Modeling, and Visualization (2017)
M. Hullin, R. Klein, T. Schultz, A. Yao (Eds.)

Dense and Scalable Reconstruction from Unstructured Videos with Occlusions

Jian Wei[1,2] and Benjamin Resch[1] and Hendrik P. A. Lensch[1]

[1]Computer Graphics, Tübingen University, 72076 Tübingen, Germany
[2]Communication Engineering, Jilin University, 130012 Changchun, China

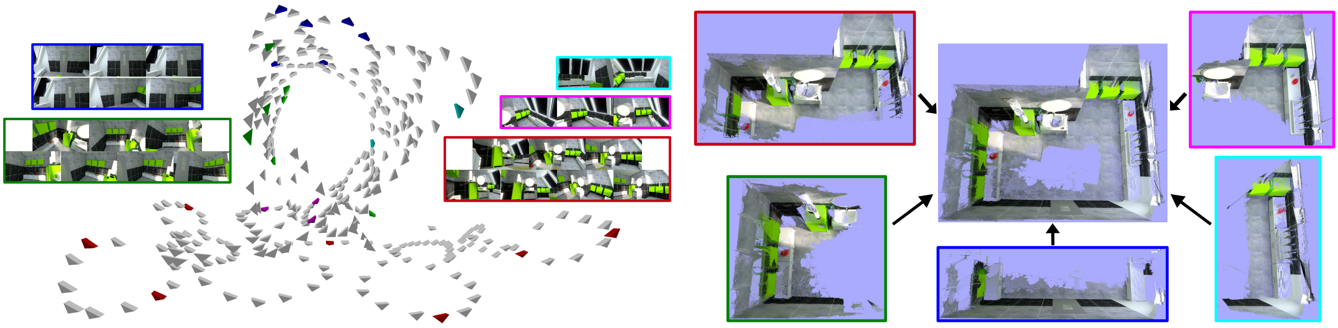


Figure 1: *A complete indoor scene recovered by our scalable method. Left: Sampled camera trajectory and clusters of views our approach selects. Right: Our partial and final reconstructions. The cameras, images, and point cloud for each view cluster are marked with the same color. Redundancy and visibility conflicts are removed at both per- and multi-cluster levels.*

Abstract

Depth-map-based multi-view stereo algorithms typically recover textureless surfaces by assuming smoothness per view, so they require processing different views to solve occlusions. Moreover, the highly redundant viewpoints of videos make exhaustive calculation of depth maps unfeasible for large scenes. This paper achieves dense and scalable reconstruction from videos by adaptively selecting a minimum subset of views from the unstructured camera paths, that are most beneficial for incremental occlusion handling and coverage improvement. Furthermore, we simplify and optimize each set of locally consistent points as the points accumulated from a cluster of previously processed views. By combining content-aware view selection and clustering, as well as cluster-wise point merging, our approach can reduce both computational and memory costs while producing accurate, concise, and dense 3D points, even for homogeneous areas. The superior efficiency and point-level fashion of our operations facilitate 3D modeling at large scales.

Categories and Subject Descriptors (according to ACM CCS): I.4.5 [Image Processing and Computer Vision]: Reconstruction—I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, Shading, Shadowing and Texture

1. Introduction

As acquisition of high-frame-rate videos got easier, scene reconstruction with an arbitrarily moved video camera has become a hot research topic. Many Multi-View Stereo (MVS) methods have been proposed for fast, dense, or accurate modeling from videos. However, when more frames are needed for larger scenes, the system's *scalability* has to be considered as well. For this purpose, most methods target modeling of small scenes, process sparse views for partial recovery, or decrease one or both of computational efficiency and surface quality. Especially in the presence of textureless areas, the depth-map-based methods typically rely on the smoothness assumption. The dense per-view estimates, *e.g.*, plane segments [GFP10] or depth interpolants [WRL16], probably lead to wrong surfaces that *occlude* the estimates of other views, producing *visibility conflicts*. These inconsistent errors are usually solved via cross-view propagation. But the reconstruction would scale poorly on very long videos, since redundant computations have to be done on a lot of densely sampled views and large memory space is inevitably occupied to maintain the near-duplicate depth maps.

This paper proposes a more scalable approach to obtain a concise point cloud from far fewer views, while retaining the superior visibility consistency and completeness (see Fig. 1). Its most parts

DOI: 10.2312/vmv.20171259

are point-wise, and thus can be easily parallelized on GPUs and multithreaded on CPUs. Our main contributions are three-fold:

1) *Content-adaptive view selection*, for incremental improvement of visibility consistency and surface completeness. In this way, the Next-Best-View (NBV) determination and model updating can benefit from each other.

2) *Automatic view clustering*, each cluster generating a concise, optimal, and locally consistent point set. This enables us to perform the detection and removal of visibility conflicts only on the current cluster while storing merged points in the previous clusters.

3) *Ray-wise point merging*, that not only respects points' scale and uncertainty but also preserves high resolutions. So the surfaces are simplified and optimized but never coarsened.

2. Related Work

Incremental/Online MVS. Some methods incrementally get 3D models represented by points [TS17], voxels [NZIS13, PKMR15, OKI15], patches [KM15a, LPVG16], triangles [ND10], tetrahedra [STO13], or surfels [KTSP14]. Bayesian estimation and convex optimization are combined in [PFS14]. Online subvolume registration is used in [FTF*15]. The surface quality of these methods depends largely on the simultaneously computed camera poses, and most of them concentrate on modeling RGB-D input or small scenes. In [PNF*08], a real-time system aided by INS/GPS data is designed. The large, sparse point cloud is progressively decomposed in [KM15b] for parallel recovery, and a total variation prior is used in [KHSM16]. The prioritized region growing method [LPVG16] improves surface quality by adaptively expending and branching 3D patches. Schöps *et al.* [SSHP17] propose a pipeline on a mobile device using motion stereo. Unlike these methods, we focus on recovering homogeneous surfaces and our occlusion handling can be parallelized more easily.

View Selection. For less computational and memory overhead, the views can be pre-selected from all available [GSC*07, GFMP08] or only the clustered [FCSS10, MRS*14a] views, but the scene's geometric properties are neglected. This has been solved by the content-aware NBV strategy. In [HH12], covariance propagation is used and the views decreasing uncertainties are preferred. The triangle-based work [DF09] also incorporates image resolution and visual saliency. The distance to object and viewing angle with already selected views are considered in [MRS*14b]. Coverage and visibility are first maximized in [HZK08] within a voxel grid and then unreliable results are solved based on the photo consistency. The optical-flow-guided method [ND10] exploits a base mesh, and ray casting is used in [MHPB16]. Some work [ND10, MRS*14b, MRS*14a] operate on the sparse points from Structure-from-Motion (SfM). Differently from them, our view selection handles potential occlusions progressively and works on dense points.

View Clustering. Independent and parallel processing of view clusters can achieve scalability. View clustering can be based on an initial geometry [ZCI*08], graph partitioning and spectral clustering considering visibility and camera information [LIN09], or iterative cluster division and view addition to enforce the size and coverage constraints [FCSS10]. Mauro *et al.* utilize graph-based dominant set clustering [MRVG*13] and leveraged affinity propagation [MRS*14a]. In contrast, our clustering is after each partial recovery by assessing the visibility consistency.

Point Merging. Many methods [MAW*07, BFL12, KM15a] create photo-consistent depth maps using specific criteria, but often get noisy results that are then removed by checking geometric consistency. The estimates can be merged via moving least-squares [SOS05], bundle optimization [LLCX10], least commitment [HM12], or height maps [ZDRF12]. In [BBH08], the averages of point positions in each leaf node of a tree are kept but the final points' density relies on the node size. Poisson Surface Reconstruction [KH13] struggles with high-frequency noise by defining the point scale as the density of accumulated points. In [MKG11], graph cuts are performed on a global confidence map, causing difficulty to find the exact maximum from the unsigned confidence values (typically between 0 and 1). Some volumetric methods construct the signed distance function with the point scale neglected [CL96] or respected for globally optimal surfaces [CT11]. In [SZV*12], the closest signed distances to the surface are proved more accurate than the signed distances measured along the lines of sight. The point scale has been respected in local solutions by utilizing discretized [FG11] and floating-scale [FG14] implicit functions. Our point merging is based on the latter work because the continuous function can evaluate surface hypotheses anywhere.

Dense Depth Maps from Videos. This paper is based on our previous work [WRL16] that creates depth maps from videos focusing on homogeneous areas. After executing SfM [RLW*15] for all camera extrinsics, they pre-select a dense subset of views for sufficient triangulation angles, and produce a depth map for every four views to further alleviate the redundancy. For each depth map, the edges' depths are first computed via a robust, ray-wise, and subpixel-level sweeping strategy, and then diffused into textureless areas by assuming smooth surface. As shown in Fig. 2, invisibility of edges leads to wrong surfaces occluding the geometry from different views. They solve this problem by back-projecting all interpolants to the edge depth maps of other views. The depth errors are computed for the definitely wrong interpolants and then spread in 2D space. Finally, the depths with large errors are filtered out.

3. Approach Overview

Our method uses the diffusion-based depth map creation and edge-based conflict invalidation of [WRL16], but extends this work for higher scalability and more concise output. We perform the same pre-processing and use the pre-selected views as our candidate reference views. The scene is divided into multiple parts, each recovered from a view cluster (see Fig. 1). So our reconstruction is separated into two levels: per-cluster recovery and multi-cluster integration. It begins by picking any candidate view as the first reference view of the first cluster.

At the per-cluster level, we incrementally improve the visibility consistency and coverage of the current part of the scene using the newly created points and do not get to another part, *i.e.*, generate a new view cluster, until all conflicting and redundant estimates are solved. The reference view selection and clustering are both based on approximating the local consistency achieved so far. More specifically, this level works as follows:

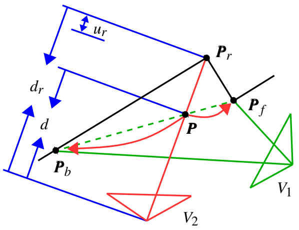


Figure 2: *Cause and removal of visibility conflicts. Diffusing the depths of two edges $\boldsymbol{P}_f$ and $\boldsymbol{P}_b$ in the view V_1 where the edge $\boldsymbol{P}_r$ is invisible creates a wrong surface (dashed line). This is solved by invalidating the interpolant $\boldsymbol{P}$ using another view V_2 where $\boldsymbol{P}_r$ is recovered with $d_r - d > 3u_r$ (u_r is its uncertainty), and afterwards growing it (red arrows).*

1) For the current reference view, a set of new points are first created from its dense depth map estimated using the method of [WRL16] (after diffusion of the edges' depths).

2) The detectable visibility conflicts are invalidated from *both* the new and existing points produced from the current view cluster. If using the invalidation method of [WRL16] (see Fig. 2), the 2D-space region growing has to be performed for all views in this cluster. So we enhance it for growing over the accumulated points. This step keeps the scene model extended by the new points visibility-consistent with respect to the currently selected views. See Fig. 3 and the supplementary material for details.

3) Then we select the NBV as the next reference view to see as many wrongly occluded surfaces as possible. Once the consistency improvement converges, it creates a new cluster with a view that captures novel scene content. See Section 4 for this step.

4) If the current view cluster finishes the partial reconstruction, the points accumulated from this cluster are merged, *i.e.*., simplified and optimized, to obtain a locally consistent and concise point cloud. See Section 5 for this step.

5) The above steps are iterated until all parts of the scene are traversed. See Step 3 of our view clustering scheme.

At the multi-cluster level, the steps of conflict invalidation and point merging are performed in order over all view clusters for global consistency and conciseness.

We represent each point $\boldsymbol{P}$ from the v^{th} view of the c^{th} cluster with $\{v,c,b,u,s,\boldsymbol{x},\boldsymbol{n},\boldsymbol{c},\boldsymbol{N}\}$. $b \in \{0,1\}$ indicates whether $\boldsymbol{P}$ is created from an edge pixel. The point uncertainty u measures the depth imprecision of $\boldsymbol{P}$ and we use the depth's standard deviation defined in [WRL16] but measured along the visual ray. The point scale s conveys the surface area $\boldsymbol{P}$ covers and we use the footprint [FG11] of its depth map pixel. $\boldsymbol{x}$, $\boldsymbol{n}$, and $\boldsymbol{c}$ are the position, surface normal, and color. $\boldsymbol{n}$ is computed using the central difference of point positions. $\boldsymbol{N}$ is the index set of n neighboring points around $\boldsymbol{P}$ (n might change during the reconstruction). It is initialized as the 4-connected neighborhood and renewed once the points are updated.

4. Dynamic View Selection and Clustering

We introduce a content-aware method for determining which view to process next, the so-called NBV, and where to encapsulate the views into one cluster. These tasks depend on the visibility of currently recovered surfaces and the views' abilities to handle potential occlusions. Completeness of the entire scene is increased by constructing more view clusters. This section assumes a point set $\mathcal{P}$ created from a view set $\boldsymbol{V}$. Our view evaluation to select the NBV for one view cluster is first described and then extended for adaptive view clustering.

NBV Determination. As depicted in Fig. 2, the conflict invalidation needs to reconstruct the edges occluded by wrong surface interpolants. Therefore, we select the NBV by preferring the views seeing the recovered surfaces in a *novel* and *front* perspective.

To solve the occlusion errors of $\mathcal{P}$, we define a view score $s(\boldsymbol{P}_i, V_j) = (\max(0, \boldsymbol{n} \cdot (-\boldsymbol{v})))^2$ for each NBV candidate $V_j \notin \boldsymbol{V}$ at a point $\boldsymbol{P}_i \in \mathcal{P}$. Here $\boldsymbol{n}$ denotes the point normal and $\boldsymbol{v}$ is the viewing direction in which V_j observes $\boldsymbol{P}_i$. The squaring operation is used for more distinguishable score peak over all view candidates. This definition assigns V_j a higher score at $\boldsymbol{P}_i$ if it is closer to the fronto-parallel view of the point's surface patch, and a zero score if seeing the surface from the opposite side. Let m be the number of points possibly visible in V_j, *i.e.*, $\boldsymbol{n} \cdot (-\boldsymbol{v}) > 0$. For enough overlap of the captured content, we incorporate a parameter α[†] to only consider the views with $m > \alpha|\mathcal{P}|$.

The NBV can be simply determined by computing all point-wise scores for each candidate and picking the view with the maximum score average. However, this would suggest the view next to the previously selected view due to their similar viewing perspectives. Since our scalable method processes *sparsely* sampled views, the NBV should have a clear perspective distinction from all already processed views. Towards this end, we favor the views causing the most significant increase of the view scores.

Specifically, we incorporate the maximum score $s_{\max}(\boldsymbol{P}_i, \boldsymbol{V}) = \max s(\boldsymbol{P}_i, V_k)$, $V_k \in \boldsymbol{V}$, at $\boldsymbol{P}_i$ achieved by all previously selected views $\boldsymbol{V}$. Then the change of $s_{\max}$ when selecting V_j as the NBV is $s^{\Delta}_{\max}(\boldsymbol{P}_i, \boldsymbol{V}, V_j) = s_{\max}(\boldsymbol{P}_i, \boldsymbol{V} \cup \{V_j\}) - s_{\max}(\boldsymbol{P}_i, \boldsymbol{V})$. We determine the NBV by finding the view maximizing the average of $s^{\Delta}_{\max}$:

$$s^{\Delta}_{\text{mean}}(\mathcal{P}, \boldsymbol{V}, V_j) = \frac{1}{n} \sum_{\boldsymbol{P}_i \in \mathcal{P}} s^{\Delta}_{\max}(\boldsymbol{P}_i, \boldsymbol{V}, V_j). \tag{1}$$

If s^{Δ}_{mean} of the NBV is smaller than a threshold τ, we say that all valuable views for improving the visibility consistency of $\mathcal{P}$ have been selected and thus terminate the reconstruction.

As visualized in Figs. 1 and 3, our method can dynamically select a small subset of views that are still sufficient to recover all scene structures and remove all occlusion errors.

Adaptive View Clustering. The criterion of Eq. 1 merely applies to small scenes due to the visibility constraint ($m > \alpha|\mathcal{P}|$). For large-scale MVS, we design an adaptive view clustering scheme: On the premise of sufficient visibilities of the points from the *current* view cluster, we select the NBV to deal with the occlusion errors from the *existing* (current and previous) clusters; Once the surfaces are locally consistent, we lower the request in visibilities to create a new cluster with the NBV. This method enables us to improve the consistency of other parts of the model and gradually extend the surface coverage. It also results in a great saving in computational

[†] See Section 6 for our parameter settings.

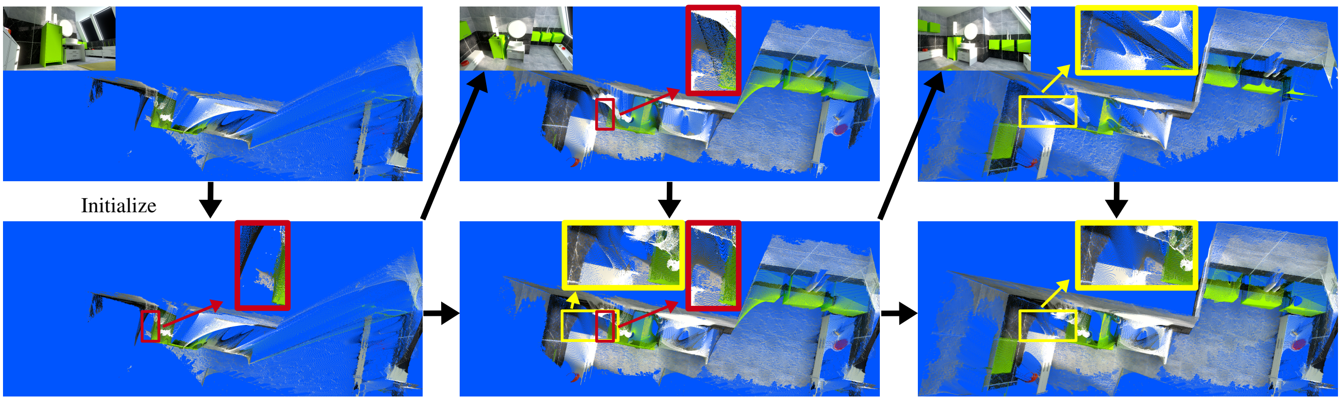


Figure 3: *Intermediate results of our method. Top: automatically selected views (top left) and created new points; Bottom: accumulated points (after invalidation). Due to space limitation, only the first three of all the selected views in Fig. 1 are presented. The marked regions show how occlusion errors in the accumulated points are gradually addressed using the new points (red) and how the errors generated from the new views are left out of the model (yellow).*

and memory efficiency, because we can perform conflict invalidation only for the current view cluster while merging the points from each previous cluster.

Concretely, let $\boldsymbol{V}$ contain multiple view clusters and $\mathcal{P}_c \subset \mathcal{P}$ be the points accumulated from the current cluster $\boldsymbol{V}_c \subset \boldsymbol{V}$. By using two thresholds τ_l and τ_g[†] as the criterions for local consistency of $\mathcal{P}_c$ and global consistency of $\mathcal{P}$, our method works as follows:

1) We first select the NBV V_n by maximizing $s^{\Delta}_{\text{mean}}(\mathcal{P},\boldsymbol{V},V_j)$, V_j being a view candidate. We use $\mathcal{P}$ and $\boldsymbol{V}$ here to avoid picking the NBV which is similar to a view in the previous clusters. If $s^{\Delta}_{\text{mean}}(\mathcal{P}_c,\boldsymbol{V}_c,V_n) > \tau_l$, it means that $\mathcal{P}_c$ is probably *inconsistent*. Hence our reconstruction continues by refining $\mathcal{P}_c$ with the new points from V_n and adding V_n into $\boldsymbol{V}_c$.

2) Otherwise, we say that $\mathcal{P}_c$ is *locally consistent* under the visibility constraint of α. Then we reselect V_n by halving this limit, *i.e.*, $\alpha \leftarrow 0.5\alpha$. The view candidates satisfying the visibility constraint of Step 1 are not considered here. Again, if $s^{\Delta}_{\text{mean}}(\mathcal{P}_c,\boldsymbol{V}_c,V_n) > \tau_l$, we guess that $\mathcal{P}_c$ is still *inconsistent* since V_n probably sees the occluded geometry behind some small surfaces of $\mathcal{P}_c$. As more content visible in V_n is novel, we initialize a new cluster with V_n and leave the invalidation of potentially remaining conflicts to the multi-cluster level.

3) Otherwise, V_n is reselected by neglecting the visibility constraint. If $s^{\Delta}_{\text{mean}}(\mathcal{P},\boldsymbol{V},V_n) < \tau_g$, it means that the content of V_n has already been recovered from $\boldsymbol{V}$, *i.e.*, $\mathcal{P}$ is *globally consistent*. Thus the whole reconstruction finishes. Otherwise, a novel part of the scene is to be recovered by generating a new cluster with V_n.

Figure 1 shows that our method can efficiently reconstruct large scenes by clustering the selected views. The overlap of scene content between neighboring clusters allows conflict removal among clusters and the redundant points are merged finally (see Section 5).

5. Ray-Wise Point Merging

The accumulated points are redundant and potentially noisy leading to expensive memory consuming and poor surfaces, so we merge them to get a *concise* and *optimal* point cloud. The merging is done at both per- and multi-cluster levels. At the per-cluster level, the points are refined gradually but merged only after all conflicts are removed. We use the continuous, signed implicit function of Fuhrmann and Goesele [FG14] but with the following differences:

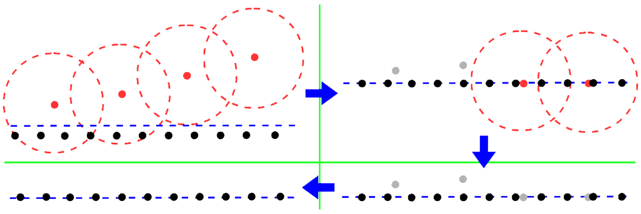


Figure 4: *Merging two point sets at different resolutions. The low-resolution (LR) points (left two red dots) whose neighborhood (dashed circles) covers high-resolution (HR) points (black dots) are not optimized. Some farther LR points (right two red dots) optimized onto the isosurface (blue dashed line) of HR points are post-detected via a second neighborhood checking. All LR points (gray dots) are removed subsequently.*

1) They *discretely* sample the implicit function via a scale-aware octree and extract an isosurface *mesh* corresponding to the zero-level set. For fast 3D-space search, we also insert all input points into such an octree (See [FG14] for the octree building). But we obtain optimized *points* by seeking the *exact* intersections between the isosurface and visual rays of input points.

2) Although their method respects the point scale, the points with large scales might still slightly decrease the resolution of merged surfaces. Instead, we aim at preserving the surface resolution while removing redundancy at the same time. To this end, the large-scale points are not optimized but used for optimizing other points and then eliminated. This reduces our computational effort and also avoids degrading the high-resolution surfaces.

3) Our approach also incorporates the point uncertainty to limit the search range.

Particularly, our point merging consists of pre-labeling, optimization, and simplification:

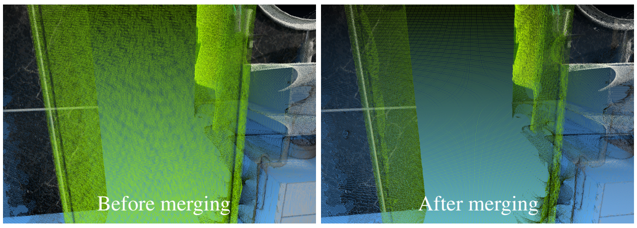


Figure 5: *Points are merged from 32,434 to 10,251 but the surface resolution is preserved.*

Pre-Labeling. Because only the highest-resolution points survive in our method, it makes no sense to optimize the points that will be removed later anyway. Thus, before optimization we first roughly judge whether each point $\boldsymbol{P}$ has a relatively low resolution and ignore it if yes. For this, we seek a set of smaller-scale points $\mathcal{P}_r$ within a spherical neighborhood centered on $\boldsymbol{P}$ such that $\boldsymbol{P}_j \in \mathcal{P}_r(\boldsymbol{x}) : ||\boldsymbol{x}-\boldsymbol{x}_j|| < \frac{1.5s}{2}$, $s > s_j$, $v \neq v_j$. Here $v \neq v_j$ lets $\mathcal{P}_r$ exclude the points created from the same view of $\boldsymbol{P}$. If $\mathcal{P}_r \neq \emptyset$, it means that $\boldsymbol{P}$ has a low resolution, and then we label it but do not remove it for the moment (see Fig. 4).

Optimization. Afterwards, for each *unlabeled* point we optimize it using *all* input information. This is performed by testing position hypotheses along its visual ray and calculating the implicit function for each hypothesis until the zero crossing is found.

The implicit function has positive values in front of the surface and negative behind. Its value at 3D position $\boldsymbol{x}$ is calculated using a weighted average of basis functions:

$$F(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{P}_i} w(\boldsymbol{x}_i) f(\boldsymbol{x}_i)}{\sum_{\boldsymbol{P}_i} w(\boldsymbol{x}_i)}, \quad \boldsymbol{P}_i \in \mathcal{P}_f(\boldsymbol{x}) : ||\boldsymbol{x}-\boldsymbol{x}_i|| < 3s_i. \tag{2}$$

$\mathcal{P}_f$ is the point subset influencing $\boldsymbol{x}$. For each point in $\mathcal{P}_f$, the basis function f is rotation invariant around its normal, and contributes to F with the same volume but more distribution if its scale is smaller. The polynomial weighting function w assigns the regions before surface more weight and falls smoothly off to 0 otherwise. See [FG14] for how to formulate f and w.

Let $\boldsymbol{v}$ be the visual ray direction of a point $\boldsymbol{P}$. Its position $\boldsymbol{x}$ is optimized as follows:

1) We search for the zero crossing of the implicit function F along the line segment $\boldsymbol{l}$ connecting $(\boldsymbol{x}-\boldsymbol{v}u)$ and $(\boldsymbol{x}+\boldsymbol{v}u)$ by considering the point uncertainty u.

2) Instead of redundantly finding $\mathcal{P}_f$ (see Eq. 2), we first find a larger point subset $\mathcal{P}_f(\boldsymbol{l})$, within which each tested position $\boldsymbol{x}'$ can yield its $\mathcal{P}_f(\boldsymbol{x}')$ more easily.

3) We do a progressive search starting from $\boldsymbol{x}$ in the direction based on the sign of F after each testing. The shift along $\boldsymbol{v}$ is in inverse depth to allow for distance to the camera. Its absolute value is initialized as 0.001 and halved if F values of the two previous hypotheses cross zero. The optimal position $\hat{\boldsymbol{x}}$ is determined until the change of F is below a threshold.

Simplification. Relatively distant low-resolution points might survive the neighborhood checking in the pre-labeling step and then optimized onto the isosurface of high-resolution points (see Fig. 4). We remove them by checking the neighborhoods again. To avoid rebuilding the octree and redoing radius search using new point positions, we obtained almost the same results by finding the point subset $\mathcal{P}_r(\hat{\boldsymbol{x}})$ within $\mathcal{P}_f(\boldsymbol{l})$. We eliminate the newly detected low-resolution points together with the pre-labeled points from the merged point cloud.

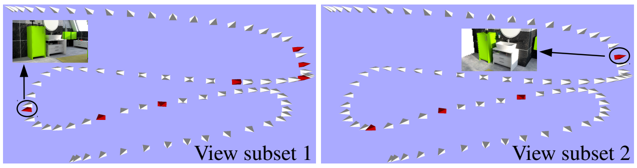


Figure 6: *Our selected views for Bathroom by beginning from different views (see the images). For clear visualization, the trajectories of unselected views are sampled (every 10).*

After each point gets a new position, other attributes are updated accordingly (see the supplementary material). As shown in Fig. 5, our merging method removes redundant points while maintaining high surface resolution. See Section 6 for quantitative comparison.

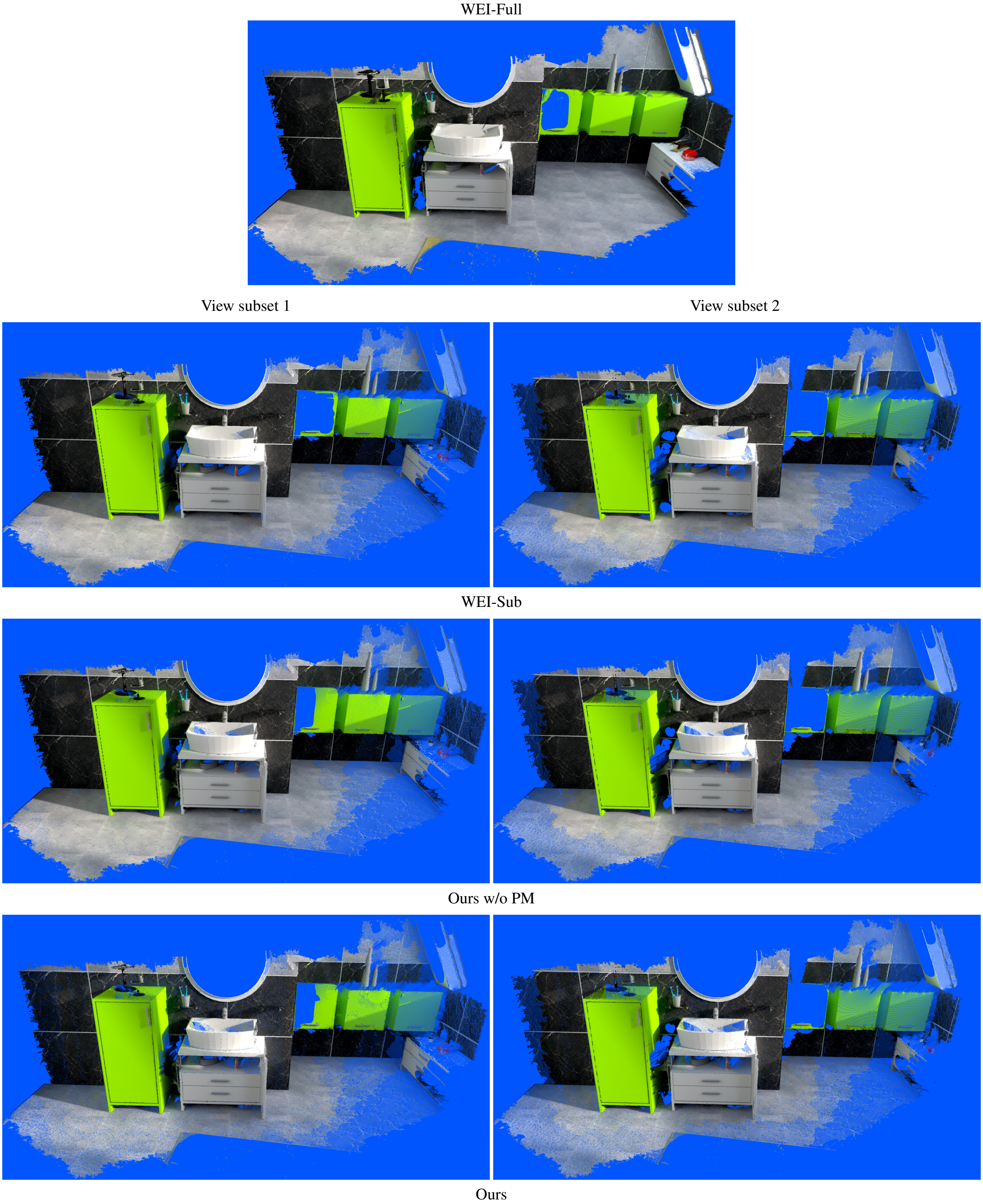
6. Results

Four videos are tested to evaluate our approach. Two synthetic datasets named Bathroom and BathroomL separately capture one part and the entire geometry of an indoor scene. Two real-world datasets named BuildingL and ChurchL are both in outdoor, large-scale environment. As in [WRL16], our method relies on accurate recovery of object edges for surface interpolation and conflict removal, so full HD resolution (1920×1080) are used for all images.

We compare our algorithm with [WRL16] (WEI) and another cluster-based MVS scheme [FCSS10] (CMVS). The superiority of [WRL16] over [BFL12], [ESC14], [KZP*13], and [WRL14] is proved by the authors, particularly on recovering homogeneous areas. CMVS achieves scalability by independently processing the view clusters, unlike our incremental manner. Since it works based on the SfM features and associated visibilities while we used the computer-generated camera extrinsics for synthetic scenes, we only implemented CMVS on real-world scenes. All tests were run on NVIDIA GeForce Titan GPUs and multithreaded CPUs (OpenMP).

Parameter Settings. We used $\alpha = 0.8$ in Section 4 to let more than 80% of the currently generated points be visible in the NBV. A smaller value would produce fewer and larger view clusters. As the convergence criteria for NBV selection, we used $\tau_l = 0.007$ for the points of the current cluster and $\tau_g = 0.009$ for the whole point cloud. Letting $\tau_g > \tau_l$ avoids small, unnecessary clusters at the end of the reconstruction which are built by the views capturing duplicate scene content but still leading to a slight increase of the s^{Δ}_{mean} value. Increasing τ_l and τ_g probably makes some occlusion errors survive the conflict invalidation procedure, and decreasing them would make us process redundant views.

Evaluation. Ground-truth depth maps are available for synthetic scenes. We project each output 3D point into a set of sampled views, and measure its inaccuracy using the smallest error between its position and the ground-truth positions of per-view projections to avoid potential occlusions. The completeness is computed as the


WEI-Full

View subset 1

View subset 2

WEI-Sub

Ours w/o PM

Ours

Figure 7: *Point clouds of Bathroom reconstructed using our method without (w/o) and with the point merging (PM) step, as well as WEI processing all candidate (WEI-Full) and only our selected views (WEI-Sub). The two columns compare the results produced from different view subsets (see Fig. 6).*

Measurement	Bathroom							BathroomL		
	WEI-Full	View subset 1			View subset 2					
		Ours w/o PM	Ours	WEI-Sub	Ours w/o PM	Ours	WEI-Sub	Ours w/o PM	Ours	WEI-Sub
Selected views (clusters) ↓	100	7 (1)	7 (1)	7	4 (1)	4 (1)	4	25 (5)	25 (5)	25
Point number ($\times 10^3$)	151,210	10,921	3,920	10,922	6,313	3,426	6,276	31,436	11,648	31,477
Completeness (%) ↑	86.774	85.452	84.769	84.767	82.261	81.596	82.183	84.831	81.798	84.577
Mean error ($\times 10^{-3}$) ↓	3.670	3.857	6.760	3.990	4.498	6.237	4.373	2.949	5.094	3.020
Runtime (sec.) ↓	3,267	219	1,152	140	115	583	78	746	5,691	481

Table 1: *Statistics for reconstructed synthetic scenes (See Fig. 7 for the corresponding visualizations). Bathroom have 100 candidate reference views and BathroomL have 800. We failed to implement WEI-Full on BathroomL due to the huge memory requirements. The arrows indicate preferred directions.*

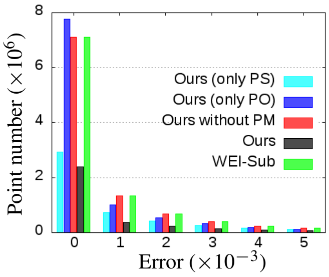


Figure 8: *Error distributions for Bathroom obtained by the methods and view subset 1 in Table 1. PS and PO denote the steps of point simplification and optimization when merging points.*

average percentage (related to the image size) of the projections whose inaccuracies are smaller than 0.1. We fill up the projection gaps using one-pass morphological closing operation.

Figures 7, 8, and Table 1 compare different results by turning on/off our point merging (PM) step and implementing WEI on all (WEI-Full) or only our selected views (WEI-Sub). We test the flexibility of our view selection by reconstructing the Bathroom scene separately beginning from two distinct perspectives (see Fig. 6).

Our method without PM obtains comparable surface quality, *i.e.*, point numbers, completeness, and mean errors, to WEI-Sub. The increase of runtime is induced by iterative conflict invalidations. When performing PM, the point numbers are significantly reduced but this does not lead to a large decrease of surface completeness. Figure 8 explains the increase of mean error. Although point optimization improves surface accuracy, the redundant estimates pruned away by the simplification step have high precision while most of wrong points are left due to their low density. Our runtime is much longer since we need to do radius search in the octree for finding $\mathcal{P}_f(\boldsymbol{l})$ when optimizing points. A more effective solution to this problem would be helpful. Note that WEI-Sub requires manual view selection. Even so, our method still produces more concise points in substantially shorter time compared with WEI-Full while maintaining high surface coverage. Moreover, comparing the reconstructions from the two selected view subsets, the setting of the first reference view affects the process of our incremental reconstruction but we still get similar results.

Figure 9 shows the results of real-world scenes. CMVS fails to recover the homogeneous geometry and produces noisy points. It processes more views because the intermediate results are not considered in its view selection and clustering. Conversely, our geometry-aware method generates dense points and processes much less views. Please see the supplementary material for the clusters of views our method selects and videos.

7. Conclusion

We designed a scalable, video-oriented MVS method for generating a dense and concise point cloud rather than a redundant depth map sequence. Content-aware view selection enables us to merely process the most valuable views while maintaining the surface completeness and occlusion robustness, and also adaptively cluster the selected views. As the reconstruction continues, we merge the locally consistent points of each view cluster obtaining optimized and simultaneously simplified points with the highest surface resolution preserved. By testing on both small- and large-scale datasets, the high surface coverage even without sufficient textures and superior efficiency of our system are proved.

References

[BBH08] BRADLEY D., BOUBEKEUR T., HEIDRICH W.: Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *CVPR* (2008). 2

[BFL12] BAILER C., FINCKH M., LENSCH H. P. A.: Scale robust multi view stereo. In *ECCV* (2012). 2, 5

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. *ACM Trans. Graph. 30* (1996), 303–312. 2

[CT11] CALAKLI F., TAUBIN G.: Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum 30*, 7 (2011), 1993–2002. 2

[DF09] DUNN E., FRAHM J. M.: Next best view planning for active model improvement. In *BMVC* (2009). 2

[ESC14] ENGEL J., SCHOPS T., CREMERS D.: Lsd-slam: Large-scale direct monocular slam. In *ECCV* (2014). 5

[FCSS10] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Towards internet-scale multi-view stereo. In *CVPR* (2010). 2, 5

[FG11] FUHRMANN S., GOESELE M.: Fusion of depth maps with multiple scales. In *Proc. SIGGRAPH Asia* (2011). 2, 3

[FG14] FUHRMANN S., GOESELE M.: Floating scale surface reconstruction. *ACM Trans. Graph. 33*, 4 (2014), 46. 2, 4, 5

[FTF*15] FIORAIO N., TAYLOR J., FITZGIBBON A., DI STEFANO L., IZADI S.: Large-scale and drift-free surface reconstruction using online subvolume registration. In *CVPR* (2015). 2

[GFMP08] GALLUP D., FRAHM J. M., MORDOHAI P., POLLEFEYS M.: Variable baseline/resolution stereo. In *CVPR* (2008). 2

[GFP10] GALLUP D., FRAHM J. M., POLLEFEYS M.: Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR* (2010). 1

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-view stereo for community photo collections. In *ICCV* (2007). 2

[HH12] HANER S., HEYDEN A.: Covariance propagation and next best view planning for 3d reconstruction. In *ECCV* (2012). 2

[HM12] HU X., MORDOHAI P.: Least commitment, viewpointbased, multi-view stereo. In *3DIMPVT* (2012). 2

[HZK08] HORNUNG A., ZENG B., KOBBELT L.: Image selection for improved multi-view stereo. In *CVPR* (2008). 2

[KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Trans. Graph. 32*, 3 (2013), 29. 2

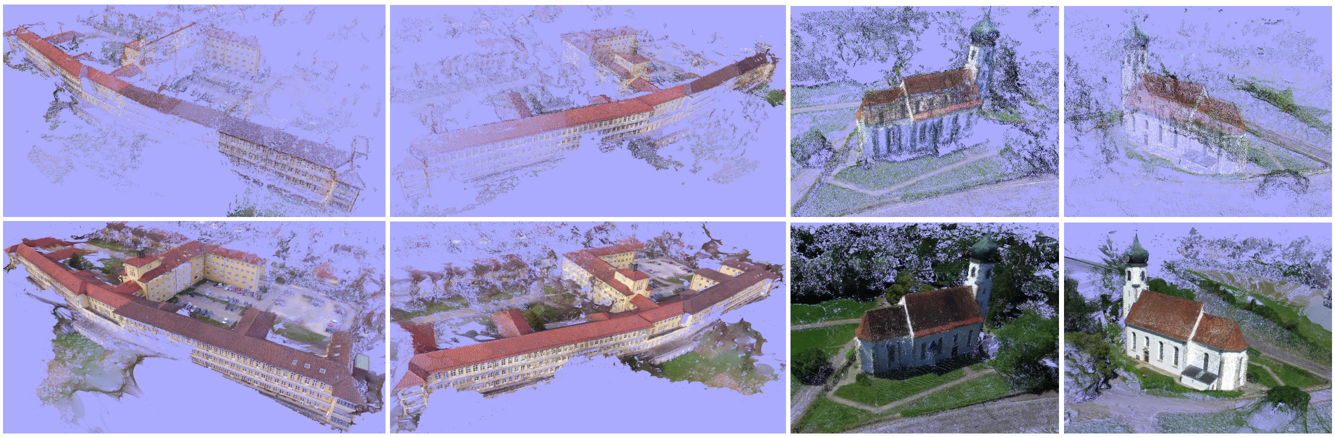


Figure 9: *Produced points for real-world scenes. CMVS (top) reconstructs BuildingL (first two columns) using 4 clusters of 49 views (from 1000) and ChurchL (last two columns) using 4 clusters of 47 views (from 1500). Our method (bottom) constructs 3 clusters with only 13 views for each dataset. However, we can even recover the poorly textured surfaces.*

[KHSM16] KUHN A., HIRSCHMÜLLER H., SCHARSTEIN D., MAYER H.: A tv prior for high-quality scalable multi-view stereo reconstruction. *Int. J. Comput. Vision* (2016). 2

[KM15a] KANG Z., MEDIONI G.: Progressive 3d model acquisition with a commodity hand-held camera. In *WACV* (2015). 2

[KM15b] KUHN A., MAYER H.: Incremental division of very large point clouds for scalable 3d surface reconstruction. In *ICCV Workshops* (2015). 2

[KTSP14] KOLEV K., TANSKANEN P., SPECIALE P., POLLEFEYS M.: Turning mobile phones into 3d scanners. In *CVPR* (2014). 2

[KZP*13] KIM C., ZIMMER H., PRITCH Y., SORKINE-HORNUNG A., GROSS M. H.: Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph. 32*, 4 (2013), 73:1–73:12. 5

[LIN09] LADIKOS A., ILIC S., NAVAB N.: Spectral camera clustering. In *ICCV Workshops* (2009). 2

[LLCX10] LI J., LI E., CHEN Y., XU L.: Bundled depth-map merging for multi-view stereo. In *CVPR* (2010). 2

[LPVG16] LOCHER A., PERDOCH M., VAN GOOL L.: Progressive prioritized multi-view stereo. In *CVPR* (2016). 2

[MAW*07] MERRELL P., AKBARZADEH A., WANG L., MORDOHAI P., FRAHM J. M., YANG R., NISTÉR D., POLLEFEYS M.: Real-time visibility-based fusion of depth maps. In *ICCV* (2007). 2

[MHPB16] MENDEZ O., HADFIELD S., PUGEAULT N., BOWDEN R.: Next-best stereo: Extending next-best view optimisation for collaborative sensors. In *BMVC* (2016). 2

[MKG11] MÚCKE P., KLOWSKY R., GOESELE M.: Surface reconstruction from multi-resolution sample points. In *VMV* (2011). 2

[MRS*14a] MAURO M., RIEMENSCHNEIDER H., SIGNORONI A., LEONARDI R., VAN GOOL L.: An integer linear programming model for view selection on overlapping camera clusters. In *3DV* (2014). 2

[MRS*14b] MAURO M., RIEMENSCHNEIDER H., SIGNORONI A., LEONARDI R., VAN GOOL L., BRESCIA I.: A unified framework for content-aware view selection and planning through view importance. In *BMVC* (2014). 2

[MRVG*13] MAURO M., RIEMENSCHNEIDER H., VAN GOOL L., LEONARDI R., BRESCIA I.: Overlapping camera clustering through dominant sets for scalable 3d reconstruction. In *BMVC* (2013). 2

[ND10] NEWCOMBE R. A., DAVISON A. J.: Live dense reconstruction with a single moving camera. In *CVPR* (2010). 2

[NZIS13] NIESSNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph. 32*, 6 (2013), 169. 2

[OKI15] ONDRUSKA P., KOHLI P., IZADI S.: Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. *IEEE Trans. Vis. Comput. Graph. 21*, 11 (2015), 1–1. 2

[PFS14] PIZZOLI M., FORSTER C., SCARAMUZZA D.: Remode: Probabilistic, monocular dense reconstruction in real time. In *ICRA* (2014). 2

[PKMR15] PRISACARIU V. A., KÄHLER O., MURRAY D. W., REID I. D.: Real-time 3d tracking and reconstruction on mobile phones. *IEEE Trans. Vis. Comput. Graph. 21*, 5 (2015), 557–570. 2

[PNF*08] POLLEFEYS M., NISTÉR D., FRAHM J.-M., AKBARZADEH A., MORDOHAI P., CLIPP B., ENGELS C., GALLUP D., KIM S.-J., MERRELL P., SALMI C., SINHA S., TALTON B., WANG L., YANG Q., STEWÉNIUS H., YANG R., WELCH G., TOWLES H.: Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision 78*, 2 (2008), 143–167. 2

[RLW*15] RESCH B., LENSCH H. P. A., WANG O., POLLEFEYS M., SOLKINE-HORNUNG A.: Scalable structure from motion for densely sampled videos. In *CVPR* (2015). 2

[SOS05] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph. (Proc. ACM SIGGRAPH)* (2005), 896–904. 2

[SSHP17] SCHÖPS T., SATTLER T., HÄNE C., POLLEFEYS M.: Large-scale outdoor 3d reconstruction on a mobile device. *Comput. Vis. Image Unders. 157* (2017), 151–166. 2

[STO13] SUGIURA T., TORII A., OKUTOMI M.: 3d surface extraction using incremental tetrahedra carving. In *ICCV Workshops* (2013). 2

[SZV*12] SCHROERS C., ZIMMER H., VALGAERTS L., BRUHN A., DEMETZ O., WEICKERT J.: Anisotropic range image integration. In *DAGM* (2012). 2

[TS17] THOMAS D., SUGIMOTO A.: Modeling large-scale indoor scenes with rigid fragments using rgb-d cameras. *Comput. Vis. Image Unders. 157* (2017), 103–116. 2

[WRL14] WEI J., RESCH B., LENSCH H. P. A.: Multi-view depth map estimation with cross-view consistency. In *BMVC* (2014). 5

[WRL16] WEI J., RESCH B., LENSCH H. P. A.: Dense and occlusion-robust multi-view stereo for unstructured videos. In *CRV* (2016). 1, 2, 3, 5

[ZCI*08] ZAHARESCU A., CAGNIART C., ILIC S., BOYER E., HORAUD R.: Camera-clustering for multi-resolution 3-d surface reconstruction. In *M2SFA2* (2008). 2

[ZDRF12] ZHENG E., DUNN E., RAGURAM R., FRAHM J. M.: Efficient and scalable depthmap fusion. In *BMVC* (2012). 2