Least Commitment, Viewpoint-based, Multi-view Stereo

Xiaoyan Hu
Stevens Institute of Technology
Hoboken, New Jersey, USA
xhu2@stevens.edu

Philippos Mordohai
Stevens Institute of Technology
Hoboken, New Jersey, USA
mordohai@cs.stevens.edu

Abstract

We address the problem of large-scale 3D reconstruction from calibrated images relying on a viewpoint-based approach. The representation is in the form of a collection of depth maps, which are fused to blend consistent depth estimates and minimize violations of visibility constraints. We adopt a least commitment strategy by allowing multiple candidate depth values per pixel in the fusion process and deferring hard decisions as much as possible. To address the inevitable noise in the depth maps, we explicitly model its sources, namely mismatches and inaccurate 3D coordinate estimation via triangulation, by measuring two types of uncertainty and using the uncertainty estimates to guide the fusion process. To the best of our knowledge, this is the first attempt to model both geometric and correspondence uncertainty in the context of dense 3D reconstruction. We show quantitative results on datasets with ground truth that are competitive with the state of the art.

1. Introduction

Undeniably, there has been significant progress in multi-view 3D reconstruction in terms of accuracy, scalability and more rigorous benchmarking [27, 28]. One question that has not been resolved yet, however, since the answer depends on the specific variant of the problem one is faced with, is which is the "best" approach for multi-view reconstruction. There are methods that achieve outstanding results on single objects surrounded by cameras [13, 3, 26, 37, 20, 38] by taking advantage of silhouettes and the fact that the final surface is a watertight manifold. These assumptions, however, require almost the entire surface of the object to be visible and often relatively easy foreground/background segmentation. In this paper we focus on a different flavor of multi-view stereo where the inputs are images of large-scale scenes that do not provide 360^o coverage of the surfaces. In the last few years, several algorithms [12, 23, 30, 9, 11] with moderate processing and storage requirements that do not require any global computations have been instrumental to the success of large-scale, dense 3D modeling [25, 8, 7]. Our approach can also be a component of such systems.

A common characteristic of many problems in computer vision, including 3D reconstruction, is that more regularization is needed when the data are barely sufficient to solve the problem, as for example in binocular stereo which has been addressed via sophisticated global optimization techniques [36, 1]. As more data become available, the need for regularization is reduced since outliers are easier to detect and remove [23, 9] and *the solution emerges from the consensus of inliers*. To achieve this, it is important that the outliers are random and uncorrelated with the inliers so that systematic errors do not reinforce each other. As long as there is even a

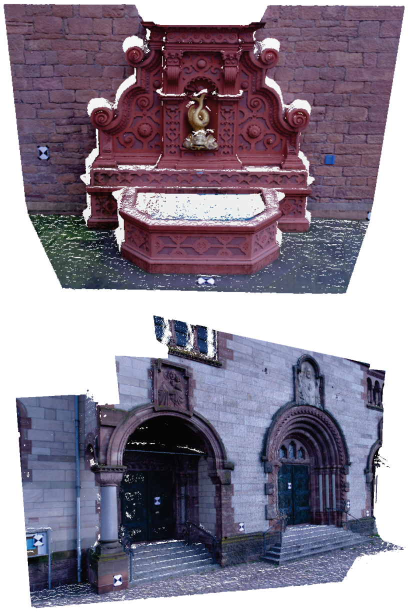


Figure 1. Screenshots of 3D point clouds generated by our method on the data of Strecha et al. [28]. Top: a point cloud from a single depth map of the *fountain-P11*. Bottom: three overlayed point clouds from *Herz-Jesu-P8*.

small number of inliers, they cluster around the correct solution, which can be detected because of this consensus. This suggests a *least commitment strategy* which generates independent hypotheses and infers the final surfaces based on consensus. Regularization in our approach occurs primarily across depth maps and not in pixel neighborhoods. Unlike other methods, we delay making inlier/outlier decisions until we accumulate all evidence for each hypothesis. In fact, we include more than one depth value per pixel in each depth map to account for cases where the true depth was not assigned the maximum photoconsistency score. The difference with other methods that generate multiple candidates per pixel [3, 22, 24] is that they select one of them based on neighborhood and photoconsistency criteria on the same depth map, while our approach allows these candidates to be evaluated on multiple depth maps until the end.

For our method to be applicable to very large image sequences, it should allow the decomposition of the problem into manageable pieces. This can be achieved either by a representation in the form of surface patches in 3D [12, 17, 9] or depth maps [23, 30, 32]. Both types of algorithms, besides depth, also estimate visibility. Patch-based methods often make early hard decisions by assuming that the seeds used to initialize the patches are correct and not considering alternative candidates. Criteria for resolving conflicts between patches include counting the number of images in which a 3D point is visible [9] or selecting the candidate with highest average photoconsistency [12, 17, 9]. Parameters, such as the minimum number of supporting images or the threshold for acceptable photoconsistency, depend on the dataset. We propose *a soft way of enforcing long-range free-space and occlusion constraints*, similar to [23], that does not require parameter tuning and does not make hard decisions before considering all evidence for and against each depth hypothesis.

In order for our approach to succeed, it must overcome errors due to the small triangulation angle and mismatches. The former cause large uncertainty in the coordinates of the reconstructed 3D points, while the latter introduce spurious depth candidates and may prevent the correct depth value from being among the candidates for a pixel. We address these challenges by explicitly modeling *geometric and correspondence uncertainty*. Geometric uncertainty is related to the expected error in a depth estimate's 3D position, given the camera configuration (focal length, resolution, baseline). Here, it is used to determine whether two depth candidates refer to the same part of the surface and, therefore, should be fused. Correspondence uncertainty measures the likelihood of establishing wrong matches on the images. The opposite of correspondence uncertainty will be referred to as *confidence* throughout the paper. Note that the two types of uncertainty are independent: a depth estimate in the near range has low geometric uncertainty, but may have high correspondence uncertainty due to repetitive texture; conversely, a distinctive point far from the cameras can be matched unambiguously, but it has high geometric uncertainty. To the best of our knowledge, ours is the first dense 3D reconstruction method that considers both types of uncertainty.

Our approach is designed to be part of a large-scale 3D reconstruction system capable of generating high-quality reconstructions in the form of point clouds (Fig. 1) without requiring global processing of the entire sequence which may contain thousands of frames. In this paper, we focus on the core depth map fusion module and not on a complete pipeline encompassing for example pose estimation, view selection and mesh generation. We think that our work provides new insights into a central problem in computer vision and demonstrates that very accurate 3D reconstruction is possible with minimal regularization.

2. Related Work

Seitz et al. [27] categorize multi-view reconstruction algorithms according to whether they represent shape using volumetric grids, global surfaces, depth map collections or surface patches. Due to the requirements for scalability and applicability to large-scale, open surfaces, the latter two categories are of more interest to us.

Methods using depth map collections generate the final 3D surfaces by fusing the input depth maps. This can be achieved by computing signed distance functions on a 3D grid [5, 14, 37] or by generating point clouds [2, 22] and reconstructing meshes from them using the method of [18]. Volumetric fusion does not meet our requirements due to the limitations it imposes on the resolution of the final surface, which is limited by the cubic memory requirement of the grid. Recently, volumetric methods have been scaled up using a two-stage algorithm [35, 16] that estimates the surface by solving a minimum s-t cut problem on the 3D Delaunay mesh of a point cloud extracted by matching keypoints. This allows space away from the surfaces to be represented by very small numbers of tetrahedra.

Alternatively, depth maps can be fused by rendering them on common planes: the images [29, 23, 30, 32] or the ground [11]. Operating on surfaces instead of volumes allows processing at high resolution. The methods of [23] and [32] are of particular interest to us because they take into account the correspondence uncertainty of depth candidates rendered onto a reference view.

Other authors have used individual rays as reference on which to link multiple depth estimates [19, 33] under the assumption that each true depth will be supported by multiple depth maps. Depth linking methods keep track of the uncertainty of the current depth hypothesis and reject unreliable candidates, but do not model long range interactions among points.

Patch-based methods integrate information on small [12, 17, 9] or large patches [21, 34]. Most of them use sparse feature correspondences to initialize patches in 3D, which are filtered to reject outliers and grown according to photoconsistency, which is estimated on several images. Visibility constraints, which are necessary for accurate results, are evaluated by projecting on reference planes, images or patches.

3. Initial Depth and Confidence Estimation

In this section, we describe how the inputs for the fusion process are generated. *Depth maps* computed from a set of calibrated images using plane-sweeping stereo implemented on the GPU similar to [10] with normalized cross-correlation (NCC) as the similarity function. NCC is computed in windows defined in the reference image and mapped to several target images via a homography through the hypothesized plane. For all experiments in this paper, we sweep four sets of planes: one horizontal and three vertical. One of the vertical sets is fronto-parallel and the other two are rotated 45^o clockwise and counter-clockwise around the gravity direction. Horizontal planes are very effective for reconstructing surfaces such as the ground, which are heavily distorted when mapped via homographies defined on vertical planes. (An alternative that approximately models horizontal foreshortening only was proposed by Bradley et al. [2].) Furukawa and Ponce [9], among others, argued for the importance of estimating the normal of the hypothesized planes correctly in achieving high matching accuracy. Our experiments are consistent with their finding, but our approach is faster. We have observed that increasing the sweeping directions has very little impact on the results.

To compute the photoconsistency of a candidate depth d_i of pixel (x, y) of the reference view for a given plane, all pairwise NCC scores are averaged, excluding cameras where the candidate is invisible (projects out of bounds), and stored in the matching volume. The candidate with the highest NCC among all sweeping directions is selected for each pixel in winner-take all fashion. When multiple hypotheses per pixel are used in fusion, the local maxima with the highest NCC values are kept regardless of the sweeping direction that generated them.

The second type of inputs to the fusion stage is *confidence maps*, one for each depth map, that represent the correspondence uncertainty of each depth candidate. Based on our evaluation of confidence measures [15], we selected a form of confidence introduced by Merrell et al. [23], and named AML in [15], to obtain a probability mass function for depth given NCC scores, assuming a uniform depth prior. The confidence of a depth candidate d_i is defined as follows:

$$C(d_i) = \frac{e^{-\frac{NCC(d_0)-NCC(d_i)}{2\sigma^2}}}{\sum_j e^{-\frac{NCC(d_0)-NCC(d_j)}{2\sigma^2}}}, \tag{1}$$

where d_0 is the depth with the maximum NCC for that sweeping direction. Taking into account $NCC(d_0)$ is crucial for allowing confidence values to transfer across depth maps. The choice of (1) was made in part because it allows the assignment of confidence values to depth candidates other than the one with the maximum NCC. Confidence is computed separately per sweeping direction, using the corresponding NCC maximum as $NCC(d_0)$. Multiple candidates (different depths) from the same direction compete and decrease each other's confidence, while if approximately the same depth were selected by multiple directions, this depth receives large support during fusion. By converting photoconsistency to a probability mass function over depth, our algorithm is less sensitive to NCC values which depend on image content.

The fact that we allow the depth probability mass function to be multimodal distinguishes our approach from that of Vogiatzis and Hernández [33] who restrict it to being unimodal. Ambiguity in stereo is manifested as multiple distinct depth candidates which often exist for a pixel and which cannot be modeled by a unimodal distribution. The fusion process of the next section is able to handle a large number of candidate depths per ray of the reference view.

4. Depth Map Fusion

Our depth map fusion algorithm is a synthesis of the confidence and stability-based fusion algorithms [23] and new ideas introduced here. In its design we aimed at eliminating failure modes, heuristics and suboptimal choices of the previous algorithms. We begin by discussing the shortcomings of [23] and how they are addressed here.

Both algorithms of [23], as well as ours, begin by rendering the input depth and confidence maps onto the reference view. (All renderings in this paper are done on points.) They consider violations of visibility constraints, namely occlusions and free space violations, which indicate inconsistencies among the depth maps. An *occlusion* occurs when a depth map from view v appears in front of the current depth estimate on the ray of the reference view (Fig. 2(b)), while a *free space violation* occurs when a depth estimate of the reference view appears in front of depth map v on the ray of view v (Fig. 2(c)). Both visibility constraints are required; if only free space violations were considered, the algorithm would select hypotheses far away from the cameras, while the opposite is true for occlusions.

Stability-based fusion [23] seeks a depth that balances the two types of violations. This "median" depth is found by rendering all depth candidates accumulated on the reference view to all other views and evaluating their free space

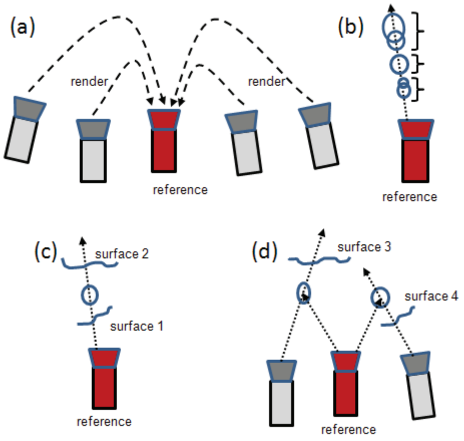


Figure 2. Depth map fusion. (a) All depth and uncertainty maps are rendered onto the reference view. (b) Depth hypotheses are formed by fusing depth values that support each other. (c) Occlusions are detected on the rays of the reference camera. Surface 1 occludes the hypothesis, while there is no conflict with surface 2. (d) Free-space violations are detected on the rays of the target cameras. One hypothesis violates the free space of surface 3; the other does not conflict with surface 4. All four surfaces in (c) and (d) are parts of input depth maps.

violations, while occlusions are evaluated on the reference view. The hypothesis that has an equal number of occlusions and free space violations is selected, as it is not too far neither too close to the reference camera according to all the depth maps. The main disadvantage of stability-based fusion is that *it only selects one of the existing depth candidates*, thus deriving no benefit in terms of accuracy from having multiple measurements of the same depth.

Confidence-based fusion [23] is linear in complexity because it greedily selects the depth candidate with the highest confidence and evaluates the support it receives versus the violations of visibility constraints it causes. Its drawback is that *if the correct depth is not initially selected, the depth for that pixel will be wrong or missing*. Moreover, the support range is heuristically chosen independent of depth.

Both methods fail if the true depth is not among the set of candidates for a pixel. We address this limitation by keeping several candidates with high confidence for each pixel in every depth map. In this paper, *three local NCC maxima are used as candidates for every pixel*. For pixels with clear matches the two additional candidates have minimal impact on the result due to their low confidence values.

4.1. Uncertainty-guided Depth Map Fusion

The inputs to our algorithm are depth and confidence maps, but not images, for a set of calibrated views. Using the estimated depths, this information is rendered onto the reference view. For clarity, readers can initially assume that depth maps include one depth candidate per pixel in the following description.

Let us denote by $H^v(x,y)$ a hypothesized 3D point for pixel (x,y) of the reference view, generated by view v, and by $D^u(H^v(x,y))$ its depth on view u. This is not generally equal to $\hat{D}^u(x,y)$ which denotes the depth estimated for pixel (x,y) of view u. $C(H^v(x,y))$ is the confidence of $H^v(x,y)$ and $S(H^v(x,y))$ is the radius of its support region which is defined below. (These quantities can only be defined with respect to view v that generated $H^v(x,y)$.)

Support. For each pixel (x,y) of the reference view, we examine the depth hypotheses rendered on it and compute the support each of them receives from the other depth hypotheses for the same pixel (Fig. 2(b)). To do this, we need to address a fundamental computer vision problem: how to distinguish whether two samples are from the same or different surfaces. We accomplish this by examining the geometric uncertainty of a triangulated 3D point. Even though our images are not rectified, a rectifying transformation could have been applied and we can introduce *effective disparity*, denoted by δ, which determines the depth of the 3D point given the image coordinates of two corresponding pixels. The relationship between depth and disparity is $Z = bf/\delta$, where f is the focal length and b is the baseline. (We set b equal to the widest baseline between the reference and any of the target views. Determining the effective baseline for multi-view stereo more precisely is outside the scope of this paper.)

Having written depth as a function of disparity, we can propagate the uncertainty of disparity to determine the uncertainty of depth. Unlike uncertainty in depth which is a function of depth itself, uncertainty in disparity can be considered identically distributed over the entire depth map. Its sources include primarily quantization noise due to limited pixel resolution as well as slight mismatches. Assuming that disparity errors are zero-mean and ignoring higher than second-order moments, the variance of the errors in Z can be written as a function of σ_δ as follows [4, 6]:

$$\sigma_Z^2 = \frac{\partial Z}{\partial \delta}\sigma_\delta^2\frac{\partial Z}{\partial \delta} = \frac{-Z^2}{bf}\sigma_\delta^2\frac{-Z^2}{bf} = \frac{Z^4\sigma_\delta^2}{b^2 f^2}. \tag{2}$$

The standard deviation of the error in depth σ_Z grows quadratically with depth. Therefore, the radius within which we will consider two points as being indistinguishable should also grow quadratically with depth.

For each depth hypothesis $H_i(x,y)$ for pixel (x,y), we seek other depth hypotheses on the same ray that support it by testing whether they are within its support region $S(H_i)$, which depends on its σ_{Z_i} and a constant c_s. We omit (x,y) for clarity.

$$C_{supp}(H_i) = \textstyle\sum_j C(H_j), \tag{3}$$

$$B_i = \frac{\sum_j C(H_j)H_j}{\sum_j C(H_j)},$$

$$\text{if } |D^{ref}(H_i) - D^{ref}(H_j)| \le S(H_i) = c_s\sigma_{Z_i},$$

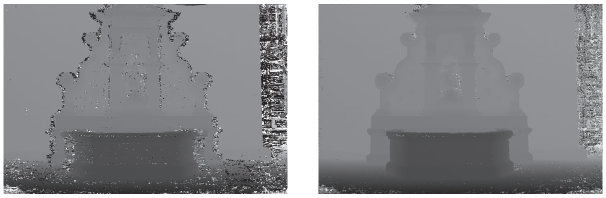


Figure 3. Raw and fused, but not filtered, depth map for the central view of the *fountain-P11* dataset. Notice the improvements especially near depth discontinuities where depth estimation is likely to fail. Wrong depth estimates, however, have low confidence and are corrected by depths estimated from more favorable viewpoints. (The surface on the far right is beyond the specified depth range.)

We update the confidence of $H_i(x, y)$ by adding the confidences of all hypotheses that support it. The depth of the blended hypothesis $B_i(x, y)$ is computed as the confidence-weighted average of the depths of all hypotheses that support it. The number of supporting hypotheses $N_i(x, y)$ is also recorded.

Occlusions. Blended hypotheses are penalized if they are occluded on the ray of the reference view since this is an indication that they may not be visible from that viewpoint, and thus that they may potentially be wrong. For each occlusion, the confidence of the occluding depth estimate is subtracted from the updated confidence of B_i.

$$\begin{aligned} C_{occ}(B_i) = C_{supp}(B_i) - \textstyle\sum_j C(H_j), \\ \text{if } D^{ref}(B_i) - D^{ref}(H_j) > S(B_i) \end{aligned} \tag{4}$$

where $S(B_i)$ is taken equal to $S(H_i)$. Hypotheses that support B_i are not considered to be occluding it.

Free space violations. Blended hypotheses are then rendered onto all depth maps to assess whether they violate the free space of the estimated surfaces. Let us denote the projection of a 3D point onto view v as $P^v(B_i) \doteq (x^v, y^v)$ and its depth with respect to view v as $D^v(P^v(B_i))$. If a blended hypothesis B_i is in front of the current depth estimate $\hat{D}^v(x^v, y^v)$ by more than its support range, then a violation occurs and the blended hypothesis is penalized by the confidence value at (x^v, y^v) in the confidence map for view v. This test is repeated for all views.

$$\begin{aligned} C_{fin}(B_i) = C_{occ}(B_i) - \textstyle\sum_v C^v(x_i^v, y_i^v), \\ \text{if } \hat{D}^v(x_i^v, y_i^v) - D^v(P^v(B_i)) > S(B_i) \end{aligned} \tag{5}$$

Hypothesis selection and hole filling. After the above computations, the blended hypothesis with the highest confidence is selected for each pixel. If its number of supporting hypotheses $N_i(x, y)$, however, is much lower than the maximum observed for that pixel, it is rejected and the next hypothesis is considered. We require that $N_i(x, y) > max\{N_i(x, y)\} - 2$ for a hypothesis to be accepted. If its final confidence $C_{fin}(B_i)$ is negative, then it is also rejected and no depth is assigned to the pixel. Figure 3 shows an example of fusion. The resulting depth map is iteratively filtered by a median filter to fill in the holes. At least 50% of the pixels in the window must have valid depths for a hole at the center to be filled. Valid depths are not altered.

A final, but important, note in this section is that *the algorithm does not change when multiple candidates for each pixel are used.* In fact, we run the code without modification by providing as input three "different" depth, confidence and geometric uncertainty maps for the same camera. Allowing these additional hypotheses to compete and support other hypotheses is precisely what our objective was.

5. Experimental Results

In this section, we present an evaluation of our algorithm on two real outdoor datasets with ground truth[1] [28]. We rendered the provided 3D models to generate ground truth depth maps for *fountain-P11* and *Herz-Jesu-P8*, which contain 11 and 8 images respectively. All images were downsampled to 1536×1024 and experiments were performed with fixed parameters: three adjacent images were used for the computation of each depth map, the NCC window was 7×7, σ in (1) was set to 0.2 and the sweeping planes where distributed in space so that the disparity step between them was no more than 0.2, with disparity defined between the reference view and the farthest target view. The derivative of depth with respect to disparity (2) was used to determine the steps between planes. All depth maps are fused on each view, except for the two extreme views of each dataset. The support range was set to $4\sigma_Z$, i.e. $c_s = 4$ in (4), and median filtering for hole filling was performed in 13×13 windows.

Due to the high resolution of the input images, storing the entire cost volume on the GPU is impossible. While depth can be computed by just keeping track of the current best candidate, computing confidence according to (1) requires the entire volume which has to be computed in parts and transferred to CPU memory. To avoid this very time consuming operation, we approximate confidence computation by explicitly maintaining the three highest NCC scores for every pixel and histogramming the remaining NCC values in histograms of 16 bins. This allows us to only transfer the top depth candidates and their associated confidence values from the GPU to the CPU resulting in a major speedup with negligible loss of accuracy.

We also compare our final fused depth maps with results kindly provided to us by their authors. 3D models were given to us by Furukawa and Ponce [9], Zaharescu et al. [38], Tylecek and Sara [31] and Jancosek and Pajdla [16] and they are denoted by FUR, ZAH, TYL and JAN, respec-

[1] The online evaluation system for these data was unavailable for several months before the submission deadline.

tively, in the remainder. We label our results with LC, for "least commitment". We are grateful to all the authors for making their models available to us.

We use two types of errors to evaluate the depth maps: *absolute errors* which are defined as the absolute values of the difference between depth estimates and the corresponding ground truth depths and *relative errors* which are defined as:

$$e_{rel} = \frac{|Z - Z_{GT}|}{\frac{Z_{GT}^2}{bf}}, \tag{6}$$

where Z is the estimated depth and Z_{GT} is the ground truth, f is the focal length of the reference camera and b is the widest baseline between the reference and either of the target views, as in (2). Relative errors show how well the surface is reconstructed taking into account camera configuration and distance to the scene. They can be viewed as measures of the effective standard deviation of noise in pixel correspondences that would give rise to the observed 3D errors. For example, $e_{rel} = 4$ means that the effective matching error, due to quantization, calibration errors and failures of the matching function, was 4 pixels. We use cumulative histograms, similar to those of [28], of both types of errors to compare the results of all methods. Note, however, that the standard deviation of the LIDAR data used in [28] was not available to us.

Table 1 presents quantitative results comparing various stages of our algorithm, as well as our results with those of other algorithms. Only pixels with ground truth depth are considered. The table contains statistics of absolute and relative errors for: i) raw depth maps (top candidate per pixel according to confidence), ii) fused depth maps using one depth candidate per pixel, iii) fused depth maps using three candidates per pixel and iv) median-filtered fused depth maps using three candidates per pixel (LC), which is our final result.

The error statistics of fused depth maps in which one or three candidates per pixel were used are similar, but always in favor of using more candidates, the qualitative difference is significant. This is because using multiple candidates is more effective near depth discontinuities where stereo is prone to errors. See Fig. 4 for examples from the *fountain-P11* dataset.

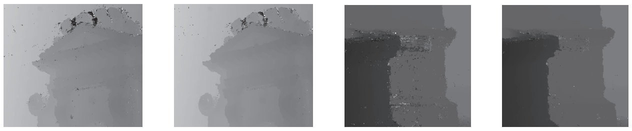
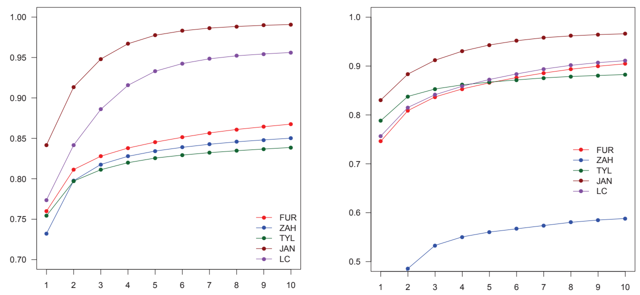


Figure 4. Details of fused depth maps for *fountain-P11* before filtering, using one (left) and three (right) depth candidates per pixel. Boundaries are reconstructed more precisely using more candidates.

Error	2cm	10cm	σ	3σ
fountain-P11				
Raw	0.695	0.874	0.732	0.838
Fuse-1	0.740	0.915	0.760	0.874
Fuse-3	0.749	0.919	0.768	0.876
LC	0.754	0.930	0.774	0.886
FUR	0.731	0.838	0.760	0.828
ZAH	0.712	0.832	0.732	0.818
TYL	0.732	0.822	0.754	0.811
JAN	0.824	0.973	0.842	0.948
Herz-Jesu-P8				
Raw	0.584	0.781	0.695	0.776
Fuse-1	0.638	0.817	0.735	0.811
Fuse-3	0.637	0.822	0.739	0.817
LC	0.649	0.848	0.757	0.841
FUR	0.646	0.836	0.746	0.837
ZAH	0.220	0.501	0.377	0.533
TYL	0.658	0.852	0.788	0.853
JAN	0.739	0.923	0.831	0.912

Table 1. Percentage of pixels with absolute errors below 2 and 10 cm and relative errors below 1 and 3σ in intermediate and final depth maps generated by our method and rendered depth maps generated from models provided to us by the authors of [9, 38, 31, 16], labeled FUR, ZAH, TYL and JAN, respectively. Our final results are labeled with LC. The two extreme images are excluded from each dataset since we do not compute fused depth maps for them, leaving nine fused depth maps for *fountain-P11* and six for *Herz-Jesu-P8*.

To compare our results with those of other methods, meshes were rendered onto the image planes to generate depth maps and all reported errors are computed on the pixels of these depth maps. Figure 5 shows cumulative histograms of relative error for all methods compared to the ground truth. (Cumulative histograms of absolute error are similar and have been omitted due to space constraints.) Table 1 contains statistics for a few thresholds on absolute and relative errors. The most salient conclusion is that the method of Jancosek and Pajdla [16] ranks first. This can be observed in the results published in [35, 16]. Our method ranks clearly second on *fountain-P11*, while the other methods form essentially one cluster. On *Herz-Jesu-P8*, our method forms a cluster with TYL and FUR, outperforming the latter slightly for all thresholds and being better than the former for larger thresholds. It should be noted that the ground truth provided with *Herz-Jesu-P8* does not include the handrails next to the steps and the horizontal bar above the left entrance (see Fig. 6), which our method is able to reconstruct. TYL reconstructs smaller pieces of these parts, but their overall effect on error statistics is hard to assess. Our method produces depths for more pixels and thus achieves higher density at large values of the threshold. Results by other methods can be seen at the currently inactive



(a) *fountain-P11*

(b) *Herz-Jesu-P8*

Figure 5. Cumulative relative error histograms. The x-axis is the distance from the ground truth divided by σ_Z and the y-axis the fraction of points within the threshold.

web site of [28] [2].

Revisiting one of the questions we set out to answer in this paper, our method seems to outperform FUR on both datasets, while it produces denser depth maps by reconstructing areas of high uncertainty that are bypassed by FUR. We speculate that this is due to not initializing patches because reliable sees could not be detected.

For visualization purposes, we generated colored point clouds using the depth maps and captured screen shots shown in Figs. 1 and 6.

All experiments were performed on a PC with a quad-core Intel i7-920 at 2.67GHz, 6GB of RAM and an Nvidia GTX570 with 1.3GB of memory. Computing a 1536×1024 depth map for *fountain-P11* using on average 8309 planes and NCC in 7×7 windows takes 320 sec on the GPU using OpenCL. The same operation for *Herz-Jesu-P8* requires an average of 6717 planes and takes 268 sec. Fusion of 27 depth maps of *fountain-P11* (3 depth maps for each of the 9 views) takes 182 sec, while it takes 73 sec for 18 depth maps for *Herz-Jesu-P8* using C++ on the CPU. Median filtering in 13×13 windows takes 22.6 sec in Matlab. Porting all steps to the GPU, further optimizing the code and investigating the effectiveness of subpixel approximations are among our priorities for future work.

6. Conclusions

We have presented a method for multi-view 3D reconstruction that, compared to other methods, delays the transition to a world-based representation and most importantly delays hard decisions on the correctness of depth hypotheses. We have shown that viewpoint-based approaches can effectively aggregate information and take into account long range interactions, in the form of occlusions and free-space violations, which are harder to implement using 3D representations. Experiments on data with ground truth show that viewpoint-based methods, which have fallen out of favor recently, can compete with the state of the art in multi-view reconstruction while being faster.

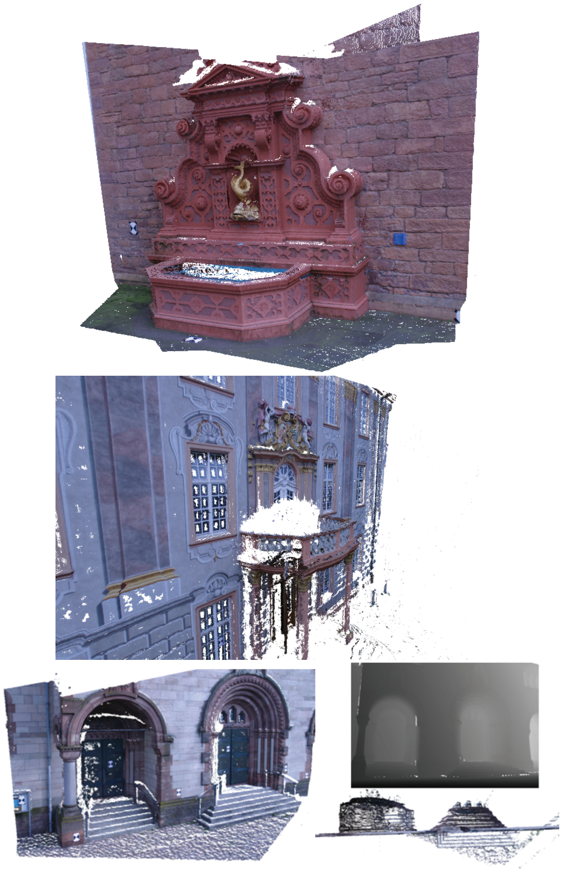


Figure 6. Screen shots of reconstructed colored point clouds: multiple overlayed point clouds from *fountain-P11*; multiple point clouds from *entry-P10*; ground truth depth map, view from above of multiple point clouds and from the side of a single cloud of *Herz-Jesu-P8*. Notice the accuracy of the pediments and the slightly ajar windows in the *entry-P10* model, as well as the missing bar and rails in the ground truth of *Herz-Jesu-P8*, which have been reconstructed by our method.

Our analysis of relative errors can be used to characterize the expected performance of multi-view stereo systems as a function of depth given the configuration parameters (focal lengths and baselines). If the results of ZAH on *Herz-Jesu-P8* are excluded, all methods perform consistently in terms of relative error. The same cannot be said for absolute error since it strongly depends on the actual depth. Extending this preliminary study to more diverse datasets will hopefully lead to empirical sensor models, i.e. expected error distributions for a given depth, for stereovision sensors.

Acknowledgements

We are grateful to the authors of [9, 38, 31, 16] for providing their results to us. This research has been supported in part by the Domestic Nuclear Detection Office of the U.S. Department of Homeland Security, by the National Science

[2] `http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html` is still online

Foundation (grant CNS-0855218) and by Google Inc. via a Google Research Award.

References

[1] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. N. Sinha. Object stereo: Joint stereo matching and object segmentation. In *CVPR*, pages 3081–3088, 2011.

[2] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multiview reconstruction using robust binocular stereo and surface meshing. In *CVPR*, 2008.

[3] N. D. F. Campbell, G. Vogiatzis, C. Hernández Esteban, and R. Cipolla. Using multiple hypotheses to improve depthmaps for multi-view stereo. In *ECCV*, pages 766–779, 2008.

[4] C. C. Chang, S. Chatterjee, and P. R. Kube. A quantization error analysis for convergent stereo. In *ICIP*, pages II: 735–739, 1994.

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *ACM Trans. on Graphics*, 30:303–312, 1996.

[6] W. Förstner. Uncertainty and projective geometry. In E. Bayro-Corrochano, editor, *Handbook of Geometric Computing*, pages 493–534. Springer, 2005.

[7] J. M. Frahm, P. Fite Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*, pages IV: 368–381, 2010.

[8] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

[9] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2010.

[10] D. Gallup, J. M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*, 2007.

[11] D. Gallup, M. Pollefeys, and J. M. Frahm. 3D reconstruction using an n-layer heightmap. In *DAGM*, 2010.

[12] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[13] C. Hernández Esteban and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *CVIU*, 96(3):367–392, 2004.

[14] C. Hernández Esteban, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *CVPR*, 2007.

[15] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *PAMI*, 2012.

[16] M. Jancosek and T. Pajdla. Robust, accurate and weakly-supported-surfaces preserving multi-view reconstruction. In *CVPR*, 2011.

[17] M. Jancosek, A. Shekhovtsov, and T. Pajdla. Scalable multiview stereo. In *3DIM*, pages 1526–1533, 2009.

[18] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.

[19] R. Koch, M. Pollefeys, and L. J. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *ECCV*, volume I, pages 55–71, 1998.

[20] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *ECCV*, pages III: 538–551, 2010.

[21] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *PAMI*, 27(3):418–433, 2005.

[22] Y. Liu, X. Cao, Q. Dai, and W. Xu. Continuous depth estimation for multi-view stereo. In *CVPR*, pages 2121 –2128, 2009.

[23] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. M. Frahm, R. Yang, D. Nistér, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007.

[24] B. Micusik and J. Kosecka. Multi-view superpixel stereo in urban environments. *IJCV*, 89(1), 2010.

[25] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *IJCV*, 78(2-3):143–167, 2008.

[26] J. P. Pons, R. Keriven, and O. D. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV*, 72(2):179–193, 2007.

[27] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, pages 519–528, 2006.

[28] C. Strecha, W. von Hansen, L. J. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.

[29] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *ACM SIGGRAPH*, pages 311–318, 1994.

[30] R. Tylecek and R.Sara. Depth map fusion with camera position refinement. In *Computer Vision Winter Workshop*, pages 59–66, 2009.

[31] R. Tylecek and R. Sara. Refinement of surface mesh for accurate multi-view reconstruction. *International Journal of Virtual Reality*, 9:45–54, 2010.

[32] C. Unger, E. Wahl, P. Sturm, and S. Ilic. Probabilistic disparity fusion for real-time motion-stereo. In *ACCV*, 2010.

[33] G. Vogiatzis and C. Hernández Esteban. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7):434 – 441, 2011.

[34] G. Vogiatzis, P. H. S. Torr, S. M. Seitz, and R. Cipolla. Reconstructing relief surfaces. *Image and Vision Computing*, 26(3):397–404, 2008.

[35] H. Vu, P. Labatut, J. P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multi-view stereo. *PAMI*, 2011.

[36] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[37] C. Zach. Fast and high quality fusion of depth maps. In *3DPVT*, 2008.

[38] A. Zaharescu, E. Boyer, and R. P. Horaud. Topology-adaptive mesh deformation for surface evolution, morphing, and multi-view reconstruction. *PAMI*, 33(4):823–837, 2011.