Occluding Contours for Multi-View Stereo

Qi Shan[†] Brian Curless[†] Yasutaka Furukawa[◇] Carlos Hernandez[⋆] Steven M. Seitz[†⋆]
[†]University of Washington [◇]Washington University in St. Louis [⋆]Google

Abstract

This paper leverages occluding contours (aka "internal silhouettes") to improve the performance of multi-view stereo methods. The contributions are 1) a new technique to identify free-space regions arising from occluding contours, and 2) a new approach for incorporating the resulting free-space constraints into Poisson surface reconstruction [14]. The proposed approach outperforms state of the art MVS techniques for challenging Internet datasets, yielding dramatic quality improvements both around object contours and in surface detail.

1. Introduction

Object silhouettes provide remarkably strong shape cues; a single silhouette constrains the entire volume of 3D space that projects outside of it to be "empty." Combined with image intensity cues, silhouettes have been shown to greatly enhance the output of multi-view stereo (MVS) methods particularly for low-textured scene regions [10, 20]. However, as the focus of 3D reconstruction research has shifted out of the lab and into "the wild," [1, 6] silhouettes have become less relevant, as it is not clear how to define, much less separate, the "background." E.g., for applications like city modeling (Google Maps and Apple Maps), there is no concept of a silhouette as the goal is to reconstruct the entire scene.

Occluding contours, aka "internal silhouettes," provide similarly powerful shape cues but in a much more general setting, without the need to define a background. An occluding contour corresponds to a boundary in the image between an object surface (e.g., part of a statue) and another surface further away (e.g., wall) that it partially occludes. In principle, occluding contours could be leveraged similarly to silhouettes, to identify regions of empty space between closer and more distant surfaces. The main challenge, however, is that identifying such free-space regions requires accurate reconstructions of both the foreground and the background surface to start with, i.e., it's a chicken-and-egg problem. For example, both statues and walls are hard to reconstruct due to lack of texture, and give rise to incomplete or noisy models, complicating the inference of free space and occluding contours.

Our contributions are 1) a new technique to identify free-space regions arising from occluding contours, and 2) an approach for incorporating the resulting free-space constraints into surface reconstruction. Our approach is based on *extrapolating free space* using the assumption that piece-wise constant image regions have simple (quadric) surface geometry. While this type of assumption is commonly used in stereo methods to interpolate nearby surface geometry, e.g., [19], prior work has not considered the effects that the interpolated geometry has on free-space and how to globally propagate this free-space information across the scene in a globally consistent and noise-robust manner. To this end, we introduce *free-space voting* into the Poisson Surface Reconstruction [13] framework, and demonstrate how this novel extension allows for high quality, free-space aware MVS reconstructions.

We show that incorporating this occluding contour information into an MVS method yields considerably cleaner, more accurate, and more complete reconstructions, especially around object boundaries. Our focus is Internet imagery, which tends to be more challenging than lab-captured datasets. The greatest improvements are in areas of very fine-scale geometry, which tend to be lost using standard regularization approaches due to over smoothing; contour information is critical to retaining these structures. When available, our approach can also incorporate standard silhouettes, and we show results on using external (sky-based) and internal silhouettes together to achieve state-of-the-art reconstructions of large landmarks.

2. Related Work

Multi-view stereo (MVS) and surface extraction techniques are both broad and well-studied fields. We focus here on existing work that is closely related to the technical contributions of the paper.

MVS with occluding contours: Most state-of-the-art MVS techniques rely solely on texture cues in a local image window, which is compared across multiple images. This rather simple cue on local image windows works surprisingly well and has been very successful, but still works poorly at depth discontinuities, where a popular strategy is to just ignore such image regions. There is a vast literature on dynami-

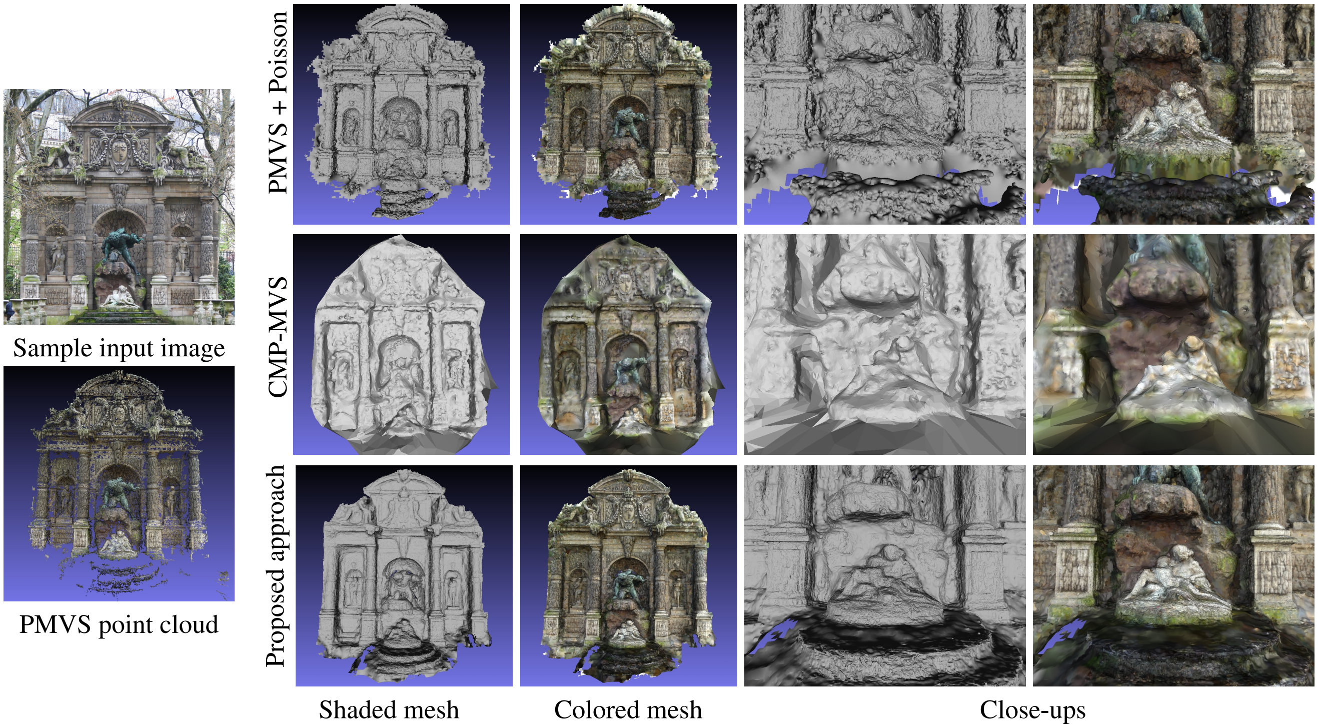


Figure 1. Luxembourg Gardens in Paris. The top row shows the baseline reconstruction by PMVS [8] and Poisson Surface Reconstruction software [13]. The middle row shows another baseline reconstruction by CMP-MVS software [12]. Our results are in the bottom row, illustrating much clearner geometry boundary and more accurate details. PMVS+Poisson and our meshes are colorized from the closest PMVS points. CMP-MVS mesh is colored by the reconstruction software.

cally adapting the local window shape to mitigate these artifacts [23], but these methods make decisions based on local information, and their effectiveness is limited. Occluding contours, on the other hand, are useful 3D cues at such image locations, which are complementary to local image texture, and can be very effective if properly extracted.

Earlier attempts to use occluding contours in MVS formulations have initialized the reconstruction with a visual hull model from image silhouettes (i.e., external occluding contours), which is then refined via a mesh evolution framework that enforces both the photometric and silhouette consistency [7, 10, 21]. The integration of both sources of information is formulated as a convex function minimization problem with constraints in [16]. In all these approaches, image silhouettes are either manually extracted or require a controlled capture setup for easy extraction, which does not easily scale to larger, unconstrained datasets. Several prior approaches have automatically extracted object silhouettes that are consistent in multiple images. While the main focus has been on segmentation rather than reconstruction, methods such as [4] can produce impressive reconstructions with thin structures by exploiting silhouette information. All of the methods described above only handle external occluding contours, which are essentially the foreground/background segmentation of an image. Accordingly, the reconstruction target is always an object, not easily extended to general scenes.

Recent work on combining stereo and segmentation goes beyond object boundaries and uses over-segmentation algorithms to better constrain the stereo problem [3, 15]. These techniques work remarkably well for improving individual depth maps, but do not formulate a full 3D visibility framework and have been tested only on "organized" input images in a narrow baseline stereo setting. To our knowledge, [9] is the only prior work that handled interior silhouettes, by properly modelling visibility changes in a mesh evolution framework. Results were only demonstrated on synthetic examples with highly distinctive colors and on a controlled laboratory example. This paper exploits not just external occluding contours (silhouettes), but also internal contours for for fully automatic 3D reconstruction of complicated scenes from Internet photo collections.

Volumetric data fusion: Almost all the state-of-the-art algorithms formulate the 3D data fusion in a volumetric domain, by first discretizing the 3D space into finite number of cells, then estimating a scalar function field, followed by an isosurface extraction step. The function is typically defined as a signed distance [5] or an indicator function [11, 12, 13] separating the surface interior from the exterior. When the input 3D data is dense and clean (e.g., laser range finders), data fusion can be conducted as a per-voxel operation [5]. To handle input data with more noise and outliers, the data fusion is formulated as an optimization problem to enforce regularization over the domain, such as linear least squares in Poisson Surface reconstruction [13] or a binary Markov Random Field (MRF) [11, 12]. The use of free-space constraints plays an important role in handling outliers, and has been extensively studied in the MRF formulation [11, 12]. In this work, we introduce the free-space constraints into the continuous optimization framework of Poisson Surface reconstruction, which tends to achieve smooth interpolation.

3. Overview

Here we present a high level view of our proposed framework. A detailed description appears in Section 4.

Starting from an internet photo collection for a given site, we recover the camera positions with freely available structure-from-motion software [18] followed by multi-view stereo using PMVS [8]. PMVS is well-suited to recovering geometry from photo collections, but only gives a semi-dense reconstruction, and can be very sparse in low texture regions. In a typical reconstruction pipeline (Figure 2 (a)), these PMVS points are then used directly for surface reconstruction. Instead, we modify the pipeline (Figure 2 (b)) to include construction of dense depth maps that respect occluding contours and provide additional shape and visibility information.

We construct an initial depth map for each input view, starting with the PMVS points that are visible in that view. Each initial depth map can be quite sparse, inadequate for providing dense visbility information. To address this, we interpolate the depth maps by optimizing for depths that are smooth between PMVS samples, with an important modification. Observing that depth discontinuities (occluding contours) tend to coincide with image discontinuities, we relax the smoothness near apparent image contours and edges during optimization.

Next, we prepare a free space volume and an augmented point set that will drive the surface reconstruction. We observe that the optimized depth maps are useful both for visibility constraints and for adding new surface points. Though not accurate everywhere, the depth maps tend to be good proxies for space carving. Thus, we create a free space volume directly from the depth maps. In addition, we use the same depth-map visibility constraints to cull spurious PMVS points, e.g., those that can often appear near occluding contours.

In some interpolated areas, the depth maps can also provide reasonable approximations to the true geometry. The intuition is that within regions that do not contain much image detail, including constant albedo regions with little shading variation, PMVS does not recover much, if any, geometry, but that geometry is likely fairly smooth. Our image-guided interpolation algorithm tends to propagate depths well within such textureless regions. In the end, we consider each depth map point as a candidate for surface reconstruction, while retaining only the points that are consistent with the all the depth maps. We combine the surviving depth map points with the PMVS points (that are not in visibility conflict) to construct an augmented point set.

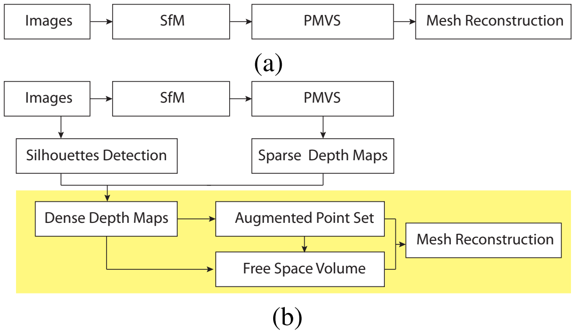


Figure 2. System overview. Our contributions are highlighted in the yellow rectangle. (a) A typical scene reconstruction system pipeline. (b) Our system pipeline.

Finally, we solve for a surface that fits to the augmented point set and performs fair hole-filling, while respecting the free space volume. To achieve this, we employ screened Poisson surface reconstruction [14] with a simple modification: the addition of a soft, free-space volume constraint. The resulting surface has smooth hole fills in under-sampled areas but does not "baloon" out into free space in the way standard Poisson reconstructions often do. In addition, the free space volume provides, in effect, a tighter boundary condition on the screened Poisson problem, which improves the overall quality of the result.

4. Algorithms

4.1. Densifying depth maps

PMVS recovers oriented points and provides a list of views used to reconstruct each point; we say a point is (conservatively) visible in these listed views. We can therefore construct an initial, conservative depth map for each view by projecting its visible PMVS points into a depth buffer for that view. As shown in Figure 3(d), these depth maps can be fairly sparse.

To estimate a dense depth map for a given view, we define an energy function that encourages depths to be (1) close to the PMVS points where available and (2) smooth between PMVS points in a spatially adaptive way, i.e., respecting image contours and color discontinuities. Let (x, y) be a pixel location in an image, $d_{x,y}$ be the unknown depth values, and Ω be a set of pixels with depth values $\hat{d}_{x,y}$ derived from projected PMVS points. Our energy definition is then:

$$\begin{aligned} E_d \quad = \quad & \sum_{x,y\in\Omega} \left(d_{x,y} - \hat{d}_{x,y}\right)^2 + \qquad (1) \\ & \lambda \sum_{x,y} w_x(x,y) \left(\frac{\partial^2 d_{x,y}}{\partial x^2}\right)^2 + w_y(x,y) \left(\frac{\partial^2 d_{x,y}}{\partial y^2}\right)^2 \end{aligned}$$

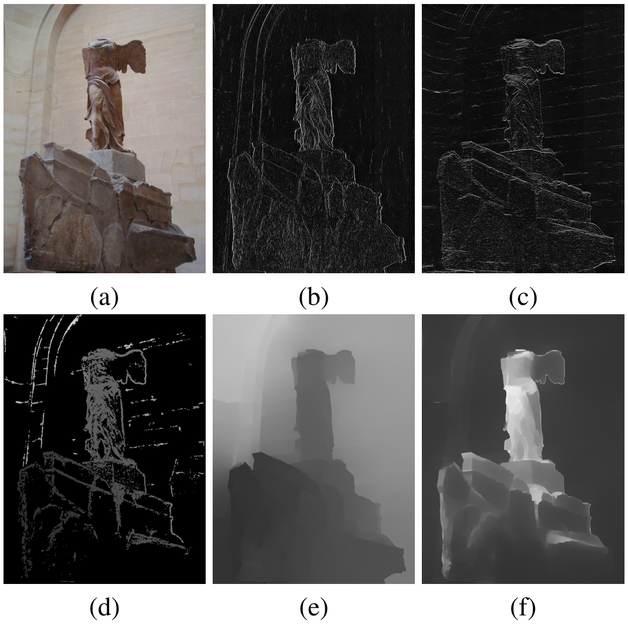


Figure 3. Densifying depth maps. (a) Input color image. (b) gPb contour response along horizonal direction. (c) gPb contour response along vertical direction. (d) Initial depth map based on visible PMVS points. (e) Our estimated dense depth map. (e) Our estimated confidence map.

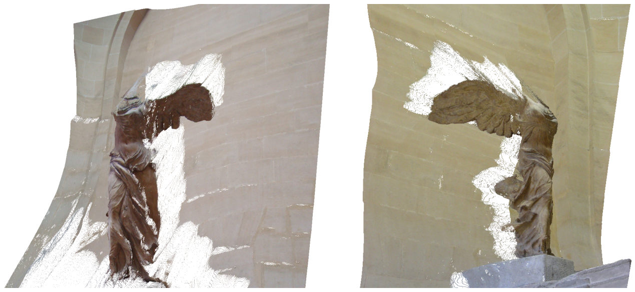


Figure 4. Two single view depth point clouds. The depth interpolation does not create accurate background depth values, as there are not enough samples. But it does create a depth-discontinuity boundary that aligns well with images and recovers geometry where PMVS points are dense.

where λ is a global smoothness weight and $w_x(x,y)$ and $w_y(x,y)$ are spatially varying smoothness weight functions along horizontal and vertical directions, respectively (described below). We set λ to a relatively large value (50) in our experiments to boost the regularization due to the fairly noisy PMVS points that arise when reconstructing from internet photo collections.

Note that we choose to use second order derivatives in order to encourage low curvature reconstructions. We approximate these derivatives as

$$\begin{aligned} \frac{\partial^2 d_{x,y}}{\partial x^2} &= 2d_{x,y} - d_{x-1,y} - d_{x+1,y}, \\ \frac{\partial^2 d_{x,y}}{\partial y^2} &= 2d_{x,y} - d_{x,y-1} - d_{x,y+1}. \end{aligned}$$

We construct the smoothness weighting functions $w_x(x,y)$ and $w_y(x,y)$ to have values close to 1 in visually smooth regions, to encourage depth propagation, and close to 0 on contours and color boundaries, to stop depth propagation and encourage depth discontinuities at visual boundaries.

To quantify visual smoothness and proximity to boundaries, we employ two measures. Recent work in computing image contours has shown significant progress. We leverage the work of gPb [2], which computes oriented contour strength at each pixel, measured in 8 directions. For our purposes, we use just the horizontal and vertical contour strengths, $g_x(x,y)$ and $g_y(x,y)$, which we show for one example in Figure 3(b) and (c). The second measure we use is simply the second derivative of image intensity in the x and y directions. Putting these together, we define the smoothness weighting functions to be:

$$\begin{aligned} w_x(x,y) &= exp\left(-\left\|\frac{\partial^2 I(x,y)\|}{\sigma_1 \partial x^2}\right\|\right) exp\left(-\frac{g_x(x,y)}{\sigma_2}\right), \\ w_y(x,y) &= exp\left(-\left\|\frac{\partial^2 I(x,y)}{\sigma_1 \partial y^2}\right\|\right) exp\left(-\frac{g_y(x,y)}{\sigma_2}\right), \end{aligned}$$

where $\sigma_1 = \sigma_2 = 0.1$ in our implementation.

Minimizing Eq. 1 is a linear least squares problem for which the global optimum is readily computed. Figure 3(e) illustrates an example of dense depth map estimation.

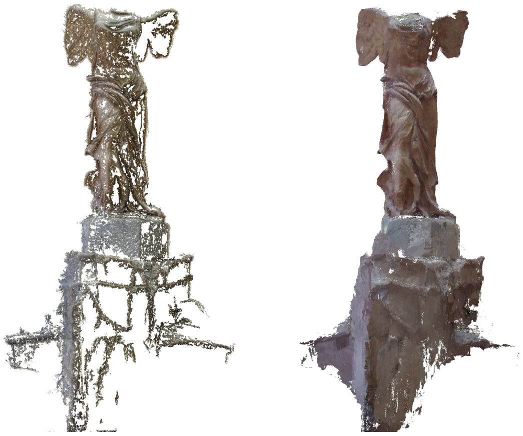
In later steps, having a confidence measure for each depth estimate is important. Depths near projected PMVS points should have relatively high confidence, whereas depths far from these points should have low confidence. Applying nearly the same framework we used for depth estimation (Eq. 1), we estimate per-pixel confidence $c(x,y)$ by minimizing an objective:

$$\begin{aligned} E_c &= \sum_{x,y} (c_{x,y} - \hat{c}_{x,y})^2 + \qquad (2) \\ & \lambda \sum_{x,y} w_x(x,y) \left(\frac{\partial c_{x,y}}{\partial x}\right)^2 + w_y(x,y) \left(\frac{\partial c_{x,y}}{\partial y}\right)^2 \end{aligned}$$

In this case, we define the data $\hat{c}(x,y)$ across all pixels: $\hat{c}(x,y) = 1$ at projected PMVS points, otherwise $\hat{c}(x,y) = 0$. Here, the smoothness weights allow confidences to “diffuse” without crossing color contours; thus, for example, a high confidence foreground does not raise the confidence of a low confidence background. Note that the smoothness term now uses first order instead of second order derivatives, since $c(x,y)$ does not have a geometric meaning that requires second order smoothness. We approximate these derivatives as:

$$\frac{\partial c_{x,y}}{\partial x} = c_{x+1,y} - c_{x,y}, \qquad \frac{\partial c_{x,y}}{\partial y} = c_{x,y+1} - c_{x,y}.$$

Again, we set $\lambda = 50$.



(a) (b)

Figure 5. Augmenting the PMVS point clouds. (a) The direct output from PMVS. (b) Our augmented point cloud.

Figure 4 shows two depth maps visualized as 3D point sets. Note how these point sets capture the occluding contours of the foreground against the background as depth discontinuities. The point locations are accurate at regions with good PMVS point coverage. They are less accurate at areas with very sparse coverage, for example, the wall in the back. We use the confidence map (Figure 3(f)) to measure this accuracy and later to limit the effect of bad depth values on our final reconstructions.

4.2. Augmenting the PMVS point cloud

The reconstructed depth maps provide useful visibility information, as well as new points that can potentially fill in geometry in less textured regions that are not well-covered by PMVS. In this section, we describe a method for enhancing the PMVS point set using the dense depth maps.

First, we consider each depth map point with confidence greater than 0.2 to be a candidate for inclusion in the augmented point set $\mathcal{P}$, which is initially an empty set. Denote the location of this point in world coordinates as q. We will add q to $\mathcal{P}$ if q (1) is not in significant visibility conflict with the all the depth maps and (2) is near other depth map points and thus likely on the surface.

Let $\pi_j : R^3 \rightarrow R^2$ be the function that projects a 3D point into viewpoint j and let $\pi_j^d : R^3 \rightarrow R$ be a function that computes the projected depth of that point. $d_j(x, y)$ represents the depth stored in the j-th depth map at location (x, y). We compute two confidence-weighted visibility votes for each point q:

$$K_f(q) = \sum_j c(\pi_j(q)) \cdot \delta(\pi_j^d(q) \leq l_b d_j(\pi_j(q)))$$

$$K_s(q) = \sum_j c(\pi_j(q)) \cdot \delta(l_b d_j(\pi_j(q)) \leq \pi_j^d(q) \leq u_b d_j(\pi_j(q))),$$

where
$$\delta(x) = \begin{cases} 1, & x = \text{true} \\ 0, & x = \text{false} \end{cases}.$$

l_b and u_b are tolerance bounds to determine if a point lies near the surface; we set $l_b = 0.99$ and $u_b = 1.01$. $K_f(q)$ then measures the degree to which q lies in the free space of all of the depth maps. $K_s(q)$ measures the amount of support for q being with range of some set of depth map points. We only add q to $\mathcal{P}$ if $K_f(q) < \gamma$ and $K_s(q) > \gamma$, where we set $\gamma = 6$ in our experiments.

We additionally add the original PMVS points to $\mathcal{P}$, except those points that are in significant visibility conflict with the depth maps, i.e., for which $K_f(q) < \gamma$.

Figure 5 illustrates the greater coverage provided by the augmented point set.

4.3. Computing free space volume

We additionally compute a free space volume that will later constrain the surface reconstruction. First, we form a finely sampled grid of voxels. Given a voxel u, we project its center (which, with some abuse of notation, we will also denote as u) into each view j to compute the accumulated free-space vote for the voxel as:

$$K'_f(u) = \sum_j c(\pi_j(u)) \cdot \delta(\pi_j^d(u) \leq l'_b d_j(\pi_j(u))),$$

where l'_b determines how close to the surface to carve. Here we are more conservative and let $l'_b = 0.97$, effectively assuming most high confidence values on the depth map are within 3% error.

To prevent carving through well-supported regions of space, we compute the number of augmented points in $\mathcal{P}$ that lie within a voxel – call this number $K_n(u)$ – and finally define the free space volume V_f as:

$$V_f = \{u : (K'_f(u) > \gamma') \cap (K'_f(u) > 10 \cdot K_n(u))\} \qquad (3)$$

In our experiments, we conservatively set $\gamma' = 15$.

4.4. Occluding contours against the sky

Sky regions in images provide strong occluding contour cues for bounding foreground structures. Suppose we have fairly conservatively detected some (but not all) sky pixels in a given view. We incorporate this information into our framework by assigning very large depth values (10^8) to those pixels, and then proceed with densifying the depth map for that view as before. In later steps, depth map pixels with very large depths are not considered as candidates for inclusion in the augmented point set $\mathcal{P}$ – we are not actually reconstructing the sky geometry – but they are used for free space computations.

Our procedure for conservatively identifying sky pixels is as follows. First we reconstruct a Poisson surface just based on the original PMVS points. We then project this surface into each view. Pixels are initially labeled sky if they are not covered by this surface. Assuming sky pixels tend to be blue or gray, we then narrow this set to pixels with (r, g, b) colors that satisfy $r+g \leq 2b$. Finally, we conservatively erode this set in each view with a disk of size 11. Note that this procedure is only to initialize a conservative sky seed. The accurate sky mask is computed through depth densifying using interior contours.

4.5. Poisson reconstruction with free space

We modify the screened Poisson surface reconstruction alrogithm [14] to handle free-space volumes. In particular, we minimize the following objective:

$$E(\chi) = \int \|\nabla\chi(u) - V(u)\|^2 du + \alpha \frac{A}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} c(p)\chi^2(p) + \beta \int_{u \in V_f} \|\chi(u) - 1\|^2 du. \quad (4)$$

The first two terms give rise to the screened Poisson equation [14]. In the first term, u is a point in the volume, V is the vector field constructed from the point normals, and χ is an indicator function; $\chi(u) > 0$ means that u lies outside the surface, and $\chi(u) < 0$ means u is in the interior of the surface. In the second term, α trades off matching point normals and point locations over the point set $\mathcal{P}$ (of size $|\mathcal{P}|$), and A is an area term automatically computed in [14]. This second term encourages values of χ to be zero near the input points. The third term is our contribution: a soft constraint that encourages the indicator function to take on values near 1 in free space. We build this modification into the released source code of [14], version 4.0. Finally, solving for the χ that minizes Eq. 4 results in an implicit function from which the surface can be extracted as the zero level set.

In our experiments, we set $\alpha \in [0.5, 2]$ and $\beta = 1$. We also supply normals for the vector field V. For PMVS points, we use the original PMVS normals. For depth map points, we compute the normals directly from the depth maps in the standard way.

5. Experiments and Evaluations

We evaluate the proposed system on 6 datasets, consisting of images collected from the Internet. The computational bottleneck is in the pre-processing, namely, image contour calculation by gPb [2], SfM execution [22], and PMVS execution [8]. These pre-processing steps may take days for large datasets, in particular, nearly a week for our largest dataset *Colosseum*, which started from more than tens of thousands of images and ended up with an SfM model consisting of 3276 images. Conversely, the new steps we contribute – depth map densification, point set augmentation, free space construction, and constrained Poisson reconstruction – finish within an hour for all the datasets on a 40-core cluster, not including file I/O time.

To evaluate the effectiveness of our approach, we have made comparative evaluations. In Figures 1 and 6, our reconstructions are compared against two other state-of-the-art methods: 1) screened Poisson surface reconstruction [14] operating on the original PMVS point-cloud [8]; and 2) CMP-MVS [12], which improves reconstruction near silhouettes and compares favorably to other state-of-the-art techniques such as [21]. Also note that the screened Poisson surface reconstruction software “hallucinates” large pieces of geometry in an attempt to produce a water tight model. Such geometry consist of large triangles; thus, we remove triangles whose edge length is longer than 10 times the average edge length of the entire mesh for Poisson reconstructions. The figures show that our results have much cleaner geometry boundaries showing the effectiveness of the enhanced depthmaps and their carving power along the occluding boundaries. Notice, in particular, the precise outline of the winged dragon in our 3D model at Place Saint-Michel (Figure 6) as well as sample depthmaps produced by our system.

The top two rows of Figure 7 further illustrate improvements over the standard PMVS+Poisson approach, with tighter bounds on the geometry. We can also see the effect of various components of the system in the bottom row. The bottom left sub-figure shows that augmenting the point set does a better job of completing one of the wings, but the reconstruction is still noisy and does not have a tight silhouette. Alternatively, as shown in the bottom-middle sub-figure, not adding these points, but performing the carving on the original PMVS points does tighten the silhouette, but leaves Poisson to just smoothly fill in missing pieces. We note that the reconstruction is much less noisy, due to the free space volume providing tigher constraints on the Poisson reconstruction. Finally, the combination of augmenting the point set and using the free space constraint gives the best of both worlds as shown in the bottom-right sub-figure.

Figure 8 provides another example where, in particular, PMVS creates a noisy point set and leads to poor surface reconstruction using the standard Poisson approach. Our method significantly improves the mesh quality.

Finally, we highlight the results of incorporating the occluding contours against the sky in depth map densification and free-space volume construction, as shown in Figure 9. The top row (San Marco Basilica) shows how PMVS and Poisson tend to inflate the boundary into sky regions. After carving out sky regions in the free space volume using the improved depth maps, we are much better able to resolve the outlines of statues and even the fine cross structure atop the dome. In the middle and bottom rows (Colloseum), we can see how sky carving has both removed spurious geometry along the top of the Colloseum and also enabled carving through some of the portals around the structure.

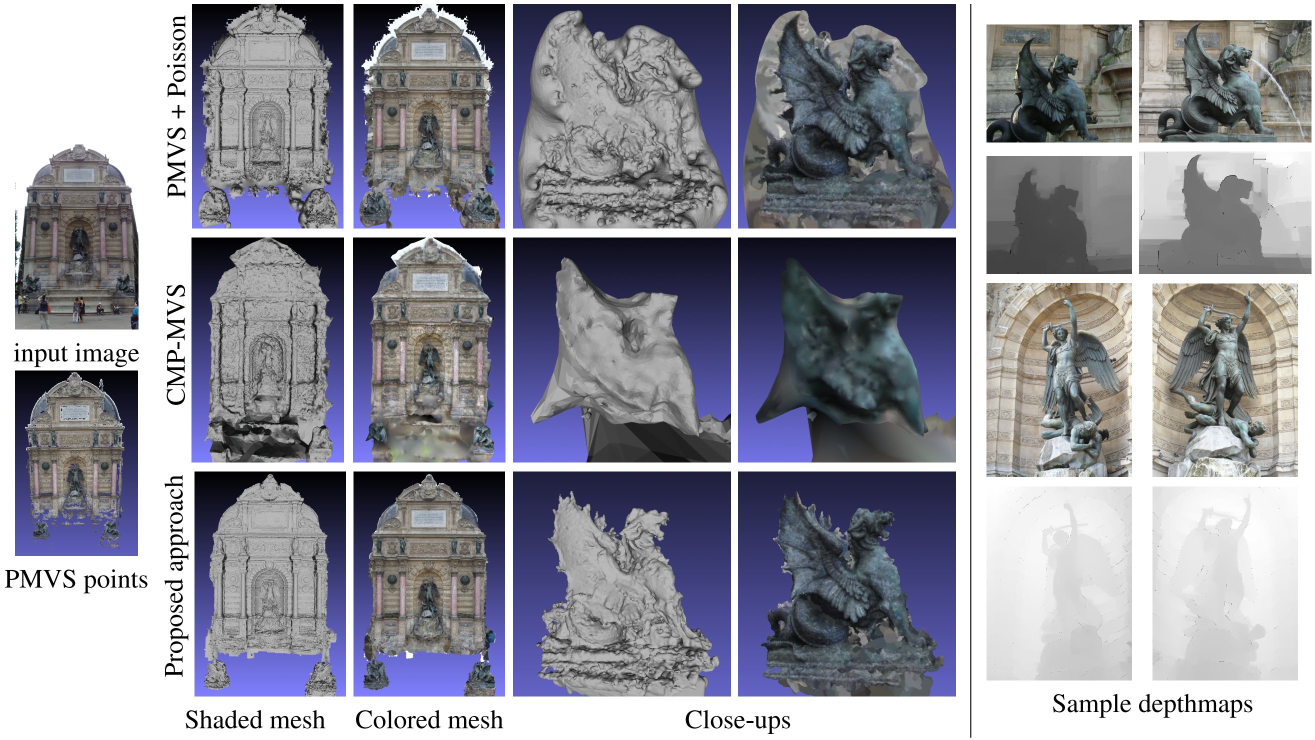


Figure 6. Reconstruction of Place Saint-Michel (583 images after SfM).

6. Conclusion

This paper presents two contributions to the problem of MVS from photo collections: 1) a new technique to identify free-space regions arising from occluding contours, and 2) a new approach for incorporating the resulting free-space constraints into surface reconstruction. We propose a new dense depth map interpolation from 3D point clouds and occlusion boundaries, and a Poisson formulation that incorporates free space constraints. The free space constraints effectively modify the space of solutions with a tigther bound around the initial point cloud. This new formulation retains the regularization/hole filling properties of Poisson [14] while improving its accuracy with noisy data. It outperforms state-of-the art MVS techniques on Internet photo collections, and results show dramatic quality improvements.

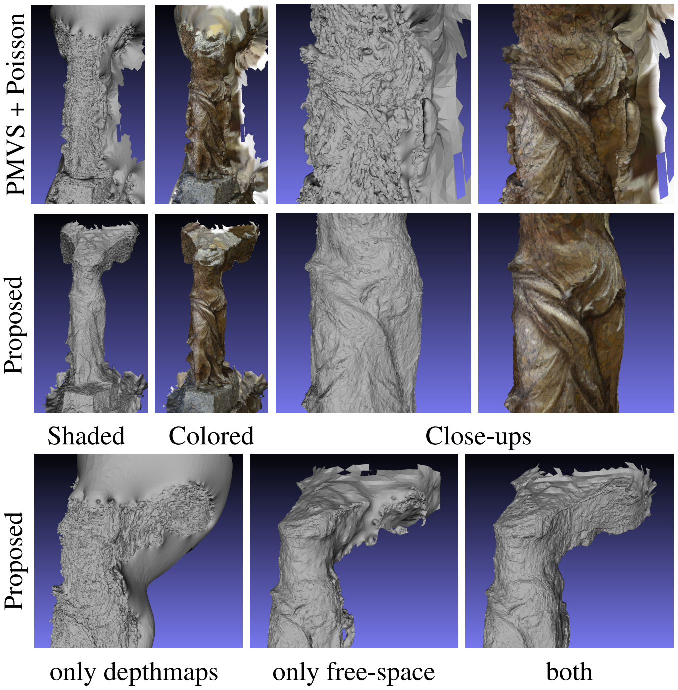
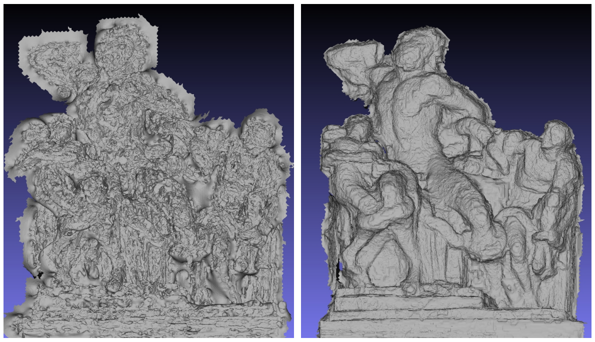


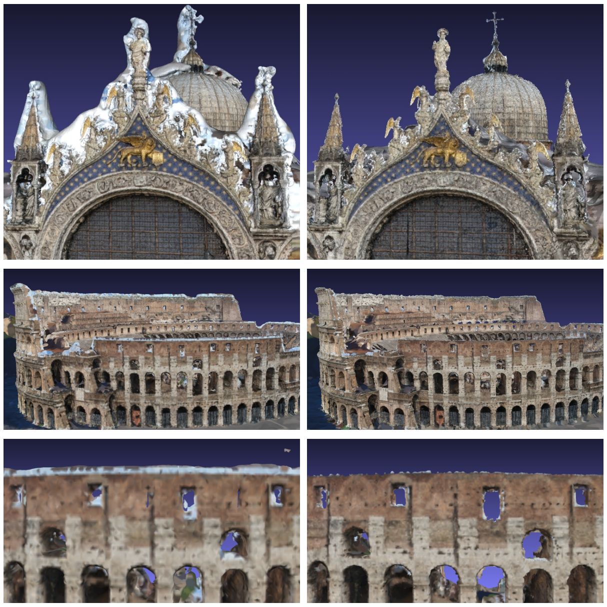
Figure 7. Reconstruction of Winged Victory of Samothrace at Louvre (274 images after SfM).

The method we have described does have limitations. First, it depends on time-consuming pre-processing steps, though these can be difficult to avoid (e.g., SfM and PMVS). Faster implementations of gPb are now available, which should accelerate this part of the process. Second, errors in depth map interpolation have the potential to do too much carving. Third, occluding contours can give better bounds on parts of a scene, but screened Poisson surface reconstruction is still left to fill in parts that were not observed, e.g., the backs of objects. In some cases, the filled-in regions behave unexpectedly, such as the wings of Winged Victory, which are merged across the top, rather than being filled around the back. Finally, 3D volumetric contouring artifacts are visible in some reconstructions, likely due to



PMVS + Poisson Proposed approach

Figure 8. Reconstruction of Laocoon and his Sons at Vatican Museums (303 images after SfM).



[17] Proposed approach

Figure 9. Our reconstructions for San Marco Square (top) and Colloseum (middle and bottom) datasets showing the improved geometry boundaries over [17]. The numbers of images after SfM are 2687 and 3276, respectively.

the influence of the binary free space volume on the Poisson reconstruction. We believe the results could further be improved with a soft voting scheme or perhaps signed distances from depth maps.

Acknowledgment

We would like to thank Riley Adams for video creation, Changchang Wu and Sameer Agarwal for in-depth discussions on SfM. This work was supported by funding from National Science Foundation grant IIS-0963657, Google, Intel, Microsoft, and the UW Animation Research Labs.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[3] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo joint stereo matching and object segmentation. In *CVPR*, 2011.

[4] N. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic object segmentation from calibrated images. In *8th European Conference on Visual Media Production*, 2011.

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[6] J.-M. Frahm, P. 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*, 2010.

[7] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *ECCV*, 2006.

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

[9] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *ICCV*, 2007.

[10] C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. In *Computer Vision and Image Understanding (CVIU)*, 2004.

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

[12] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR*, 2011.

[13] M. Kazhdan, M. Bolitho, , and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006.

[14] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013.

[15] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. In *SIGGRAPH*, 2013.

[16] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *ECCV*, 2008.

[17] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz. The visual turing test for scene reconstruction. In *Joint 3DIM/3DPVT Conference (3DV)*, 2013.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. In *SIGGRAPH*, 2006.

[19] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, volume 1, pages 532–539 vol.1, 2001.

[20] G. Vogiatzis, C. Hernández, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *TPAMI*, 29(12):2241–2246, 2007.

[21] H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009.

[22] C. Wu. VisualSFM. http://homes.cs.washington.edu/~ccwu/vsfm.

[23] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *TPAMI*, 28(4), 2006.