Multi-View Stereo for Community Photo Collections

Michael Goesele[1,2] Noah Snavely[1] Brian Curless[1] Hugues Hoppe[3] Steven M. Seitz[1]
University of Washington[1] TU Darmstadt[2] Microsoft Research[3]

Abstract

We present a multi-view stereo algorithm that addresses the extreme changes in lighting, scale, clutter, and other effects in large online community photo collections. Our idea is to intelligently choose images to match, both at a per-view and per-pixel level. We show that such adaptive view selection enables robust performance even with dramatic appearance variability. The stereo matching technique takes as input sparse 3D points reconstructed from structure-from-motion methods and iteratively grows surfaces from these points. Optimizing for surface normals within a photoconsistency measure significantly improves the matching results. While the focus of our approach is to estimate high-quality depth maps, we also show examples of merging the resulting depth maps into compelling scene reconstructions. We demonstrate our algorithm on standard multi-view stereo datasets and on casually acquired photo collections of famous scenes gathered from the Internet.

1 Introduction

With the recent rise in popularity of Internet photo sharing sites like Flickr and Google, *community photo collections* (CPCs) have emerged as a powerful new type of image dataset. For example, a search for "Notre Dame Paris" on Flickr yields more than 50,000 images showing the cathedral from myriad viewpoints and appearance conditions. This kind of data presents a singular opportunity: to reconstruct the world's geometry using the largest known, most diverse, and largely untapped, multi-view stereo dataset ever assembled. What makes the dataset unusual is not only its size, but the fact that it has been captured "in the wild"—not in the laboratory—leading to a set of fundamental new challenges in multi-view stereo research.

In particular, CPCs exhibit tremendous variation in appearance and viewing parameters, as they are acquired by an assortment of cameras at different times of day and in various weather. As illustrated in Figures 1 and 2, lighting, foreground clutter, and scale can differ substantially from image to image. Traditionally, multi-view stereo algorithms have considered images with far less appearance variation, where computing correspondence is significantly easier, and have operated on somewhat regular distributions of viewpoints (e.g., photographs regularly spaced around an object, or video streams with spatiotemporal coherence). In

Figure 1. CPC consisting of images of the Trevi Fountain collected from the Internet. Varying illumination and camera response yield strong appearance variations. In addition, images often contain clutter, such as the tourist in the rightmost image, that varies significantly from image to image.

Figure 2. Images of Notre Dame with drastically different sampling rates. All images are shown at native resolution, cropped to a size of 200×200 pixels to demonstrate a variation in sampling rate of more than three orders of magnitude.

this paper we present a stereo matching approach that starts from irregular distributions of viewpoints, and produces robust high-quality depth maps in the presence of extreme appearance variations.

Our approach is based on the following observation: given the massive numbers of images available online, there should be large subsets of images of any particular site that are captured under compatible lighting, weather, and exposure conditions, as well as sufficiently similar resolutions and wide enough baselines. By automatically identifying such subsets, we can dramatically simplify the problem, matching images that are similar in appearance and scale while providing enough parallax for accurate reconstruction. While this idea is conceptually simple, its effective execution requires reasoning both (1) at the image level, to approximately match scale and appearance and to ensure wide-enough camera baseline, and (2) at the pixel level, to handle clutter, occlusions, and local lighting variations and to encourage matching with both horizontal and vertical parallax. Our main contribution is the design and analysis of such an adaptive view selection process. We have found the approach to be effective over a wide range of scenes and CPCs. In fact, our experiments indicate that simple matching metrics tolerate a surprisingly wide range of lighting variation over significant portions of many scenes. While we hope that future work will extend this operating range and even exploit large changes in appearance, we believe that view selection combined with simple metrics is an effective tool, and an important first step in the reconstruction of scenes from Internet-derived collections.

Motivated by the specific challenges in CPCs, we also present a new multi-view stereo matching algorithm that uses a surface growing approach to iteratively reconstruct robust and accurate depth maps. This surface growing approach takes as input sparse feature points, leveraging the success of structure-from-motion techniques [2, 23] which produce such output and have recently been demonstrated to operate effectively on CPCs. Instead of obtaining a discrete depth map, as is common in many stereo methods [21], we opt instead to reconstruct a sub-pixel-accurate continuous depth map. To greatly improve resilience to appearance differences in the source views, we use a photometric window matching approach in which both surface depth and normal are optimized together, and we adaptively discard views that do not reinforce cross-correlation of the matched windows. Used in conjunction with a depth-merging approach, the resulting approach is shown to be competitive with the current top-performing multi-view stereo reconstruction methods on the Middlebury benchmarks [22].

2 Previous Work

Here we describe the most closely related work in multi-view stereo (MVS), focusing on view selection, matching with appearance variations, region growing, and normal optimization. We refer the reader to [22] for a detailed overview of the state-of-the-art in MVS.

Many MVS methods employ some form of global view selection to identify nearby views, motivated by efficiency and the desire to minimize occlusions. Commonly, MVS methods assume a relatively uniform viewpoint distribution and simply choose the k nearest images for each reference view [19, 4, 6]. CPC datasets are more challenging in that they are non-uniformly distributed in a 7D viewpoint space of camera pose and focal length, thus representing an extreme case of unorganized image sets [12]. Furthermore, choosing the nearest views is often undesirable, since many images are nearly identical and thus offer little parallax.

Local view selection has also been used before in MVS techniques to achieve robustness to occlusions. Kang et al. [13] exploit the assumption that the temporal order of images matches the spatial order, and use shiftable windows in time to adaptively choose frames to match. A number of recent stereo matching methods have used outlier rejection techniques to identify occlusions in the matching step [4, 6]. We further develop this kind of approach and demonstrate that it can be generalized to handle many kinds of appearance variations beyond occlusions.

A parallel thread of research in the stereo community is developing robust matching metrics that enable matching with variable lighting [10], non-Lambertian reflectance [11], and large appearance changes [15]. While we have found normalized cross correlation (NCC) to be surprisingly robust to appearance changes, use of more sophisticated techniques may further broaden the range of views that can be compared, and is thus complementary to the problem of view selection. We note, however, that increased invariance can potentially lead to reduced discriminatory power and should be used with care.

In its use of region-growing and normal optimization, our MVS approach builds on previous work in the computer vision and photogrammetry communities. Notably, Zhang et al. [24] present a binocular stereo method that employs normal optimization to obtain high quality results with structured lighting. Hornung and Kobbelt [10] propose a sample-and-fit approach to estimate planes and higher-order surfaces for photo-consistency computations. Concurrent with our work, Habbecke and Kobbelt [9] and Furukawa and Ponce [5] introduced region growing approaches for multi-view stereo that propagate a surface out from initial seed points. These two approaches use careful modeling of visibility to minimize the effects of outliers, whereas we rely solely on robust statistics and adaptive view selection to achieve reconstruction results of similar quality.

Our work builds on the framework of multiphoto geometrically constrained least squares matching (MPGC) from the photogrammetry literature [8, 1]. In particular, it extends the MPGC-based region-growing MVS algorithm by Otto and Chau [20] by imposing consistent surface normals between multiple views. In a related region-growing paper, Lhuillier and Quan [17] acknowledge the accuracy of [20] but point out two major drawbacks: the inability of an MPGC approach to define a uniqueness constraint to avoid bad matches, and the need for large patch sizes to achieve a stable match. In contrast, we show that even small patch sizes are sufficient for high quality reconstruction if we use a good view selection process and a suitable matching model. Other notable region-growing approaches include Zhang and Shan [25], who cast the problem in a probabilistic framework.

Our work is closely related to Kamberov et al.'s automatic geometry reconstruction pipeline for unstructured image sets [12]. The key algorithmic differences are our use of MVS instead of binocular stereo for each reference view and our view selection approach, which accounts for variations in image resolution and avoids matching narrow baselines. In addition, we demonstrate results on large CPCs with considerably more variation in scene content and capture conditions.

3 Algorithm Overview

Our approach to reconstructing geometry from Internet collections consists of several stages. First, we calibrate the cameras geometrically and radiometrically (Section 4). Next, we estimate a depth map for each input image — each image serves as a reference view exactly once. In order to find good matches, we apply a two-level view selection algorithm. At the image level, *global view selection* (Section 5.1) identifies for each reference view a set of good neighborhood images to use for stereo matching. Then, at the pixel level, *local view selection* (Section 5.2) determines a subset of these images that yields a stable stereo match. This subset generally varies from pixel to pixel.

Stereo matching is performed at each pixel (Section 6) by optimizing for both depth and normal, starting from an initial estimate provided by SIFT feature points or copied from previously computed neighbors. During the stereo optimization, poorly matching views may be discarded and new ones added according to the local view selection criteria. The traversal of pixels is prioritized by their estimated matching confidence. Pixels may be revisited and their depths updated if a higher-confidence match is found.

4 Calibrating Internet Photos

Because our input images are harvested from community photo collections, the camera poses, intrinsics, and sensor response characteristics are generally not provided. Therefore we must first calibrate the set of images both geometrically and radiometrically.

First, when feasible, we remove radial distortion from the images using PTLens, a commercially available tool that extracts camera and lens information from the image metadata (EXIF tags) and corrects for radial distortion based on a database of camera and lens properties. Images that cannot be corrected are automatically removed from the CPC unless we know that they contain no significant lens distortion (e.g., in the case of the MVS evaluation datasets [22]). Next, the remaining images are entered into a robust, metric structure-from-motion (SfM) system [2, 23] (based on the SIFT feature detector [18]), which yields extrinsic and intrinsic calibration (position, orientation, focal length) for all successfully registered images. It also generates a sparse scene reconstruction from the matched features, and for each feature a list of images in which it was detected.

In order to model radiometric distortions, we attempt to convert all input images into a linear radiometric space. Unless the exact response curve of the capture system is known, we assume that the images are in standard sRGB color space and apply the inverse sRGB mapping.

5 View Selection

5.1 Global View Selection

For each reference view R, global view selection seeks a set $\mathbf{N}$ of neighboring views that are good candidates for stereo matching in terms of scene content, appearance, and scale. In addition, the neighboring views should provide sufficient parallax with respect to R and each other in order to enable a stable match. Here we describe a scoring function designed to measure the quality of each candidate neighboring view based on these desiderata.

To first order, the number of shared feature points reconstructed in the SfM phase is a good indicator of the compatibility of a given view V with the reference view R. Indeed, images with many shared features generally cover a similar portion of the scene. Moreover, success in SIFT matching is a good predictor that pixel-level matching will also succeed across much of the image. In particular, SIFT selects features with similar appearance, and thus images with many shared features tend to have similar appearance to each other, overall.

However, the number of shared feature points is not sufficient to ensure good reconstructions. First, the views with the most shared feature points tend to be nearly collocated and as such do not provide a large enough baseline for accurate reconstruction. Second, the scale invariance of the SIFT feature detector causes images of substantially different resolutions to match well, but such resolution differences are problematic for stereo matching.

Thus, we compute a global score g_R for each view V within a candidate neighborhood $\mathbf{N}$ (which includes R) as a weighted sum over features shared with R:

$$g_R(V) = \sum_{f \in \mathbf{F}_V \cap \mathbf{F}_R} w_{\mathbf{N}}(f) \cdot w_s(f), \tag{1}$$

where $\mathbf{F}_X$ is the set of feature points observed in view X, and the weight functions are described below.

To encourage a good range of parallax within a neighborhood, the weight function $w_{\mathbf{N}}(f)$ is defined as a product over all pairs of views in $\mathbf{N}$:

$$w_{\mathbf{N}}(f) = \prod_{\substack{V_i, V_j \in \mathbf{N} \\ \text{s.t. } i \neq j,\ f \in \mathbf{F}_{V_i} \cap \mathbf{F}_{V_j}}} w_\alpha(f, V_i, V_j), \tag{2}$$

where $w_\alpha(f, V_i, V_j) = \min((\alpha/\alpha_{\max})^2, 1)$ and α is the angle between the lines of sight from V_i and V_j to f. The function $w_\alpha(f, V_i, V_j)$ downweights triangulation angles below $\alpha_{\max}$, which we set to 10 degrees in all of our experiments. The quadratic weight function serves to counteract the trend of greater numbers of features in common with decreasing angle. At the same time, excessively large triangulation angles are automatically discouraged by the associated scarcity of shared SIFT features.

The weighting function $w_s(f)$ measures similarity in resolution of images R and V at feature f. To estimate the 3D sampling rate of V in the vicinity of the feature f, we compute the diameter $s_V(f)$ of a sphere centered at f whose projected diameter in V equals the pixel spacing in V. We similarly compute $s_R(f)$ for R and define the scale weight w_s based on the ratio $r = s_R(f)/s_V(f)$ using

$$w_s(f) = \begin{cases} 2/r & 2 \leq r \\ 1 & 1 \leq r < 2 \\ r & r < 1 \ . \end{cases} \quad (3)$$

This weight function favors views with equal or higher resolution than the reference view.

Having defined the global score for a view V and neighborhood $\mathbf{N}$, we could now find the best $\mathbf{N}$ of a given size (usually $|\mathbf{N}| = 10$), in terms of the sum of view scores $\sum_{V \in \mathbf{N}} g_R(v)$. For efficiency, we take a greedy approach and grow the neighborhood incrementally by iteratively adding to $\mathbf{N}$ the highest scoring view given the current $\mathbf{N}$ (which initially contains only R).

Rescaling Views Although global view selection tries to select neighboring views with compatible scale, some amount of scale mismatch is unavoidable due to variability in resolution within CPCs, and can adversely affect stereo matching. We therefore seek to adapt, through proper filtering, the scale of all views to a common, narrow range either globally or on a per-pixel basis. We chose the former to avoid varying the size of the matching window in different areas of the depth map and to improve efficiency. Our approach is to find the lowest-resolution view $V_{\min} \in \mathbf{N}$ relative to R, resample R to approximately match that lower resolution, and then resample images of higher resolution to match R.

Specifically, we estimate the resolution scale of a view V relative to R based on their shared features:

$$\text{scale}_R(V) = \frac{1}{|\mathbf{F}_V \cap \mathbf{F}_R|} \sum_{f \in \mathbf{F}_V \cap \mathbf{F}_R} \frac{s_R(f)}{s_V(f)}. \quad (4)$$

$V_{\min}$ is then simply equal to $\arg\min_{V \in \mathbf{N}} \text{scale}_R(V)$. If $\text{scale}_R(V_{\min})$ is smaller than a threshold t (in our case $t = 0.6$ which corresponds to mapping a 5×5 reference window on a 3×3 window in the neighboring view with the lowest relative scale), we rescale the reference view so that, after rescaling, $\text{scale}_R(V_{min}) = t$. We then rescale all neighboring views with $\text{scale}_R(V) > 1.2$ to match the scale of the reference view (which has possibly itself been rescaled in the previous step). Note that all rescaled versions of images are discarded when moving on to compute a depth map for the next reference view.

5.2 Local View Selection

Global view selection determines a set $\mathbf{N}$ of good matching candidates for a reference view and matches their scale. Instead of using all of these views for stereo matching at a particular location in the reference view, we select a smaller set $\mathbf{A} \subset \mathbf{N}$ of active views (typically $|\mathbf{A}|$=4). Using such a subset naturally speeds up the depth computation.

During stereo matching we iteratively update $\mathbf{A}$ using a set of local view selection criteria designed to prefer views that, given a current estimate of depth and normal at a pixel, are photometrically consistent and provide a sufficiently wide range of observation directions. To measure photometric consistency, we employ mean-removed normalized cross correlation (NCC) between pixels within a window about the given pixel in R and the corresponding window in V (Section 6). If the NCC score is above a conservative threshold, then V is a candidate for being added to $\mathbf{A}$.

In addition, we aim for a useful range of parallax between all views in $\mathbf{A}$. Viewpoints in a typical CPC are not equally distributed in 3D space. Most images are taken from the ground plane, along a path, or from a limited number of vantage points. At a minimum, as we did during global view selection, we need to avoid computing stereo with small triangulation angles. In addition, we would like to observe points from directions that are not coplanar. This is particularly important for images containing many line features such as architectural scenes, where matching can be difficult if views are distributed along similar directions. For example, a horizontal line feature yields indeterminate matches for a set of viewpoints along a line parallel to that line feature.

We can measure the angular distribution by looking at the span of directions from which a given scene point (based on the current depth estimate for the reference pixel) is observed. In practice, we instead consider the angular spread of epipolar lines obtained by projecting each viewing ray passing through the scene point into the reference view. When deciding whether to add a view V to the active set $\mathbf{A}$, we compute the local score

$$l_R(V) = g_R(V) \cdot \prod_{V' \in \mathbf{A}} w_e(V, V'), \quad (5)$$

where $w_e(V, V') = \min(\gamma/\gamma_{\max}, 1)$ and γ is the acute angle between the pair of epipolar lines in the reference view as described above. We always set $\gamma_{\max} = 10$ degrees.

The local view selection algorithm then proceeds as follows. Given an initial depth estimate at the pixel, we find the view V with the highest score $l_R(V)$. If this view has a sufficiently high NCC score (we use a threshold of 0.3), it is added to $\mathbf{A}$; otherwise it is rejected. We repeat the process, selecting from among the remaining non-rejected views, until either the set $\mathbf{A}$ reaches the desired size or no non-rejected views remain. During stereo matching, the depth (and normal) are optimized, and a view may be discarded (and labeled as rejected) as described in Section 6. We then attempt to add a replacement view, proceeding as before. It is easy to see that the algorithm terminates, since rejected views are never reconsidered.

6 Multi-View Stereo Reconstruction

Our MVS algorithm has two parts. A region-growing framework maintains a prioritized queue Q of matching candidates (pixel locations in R plus initial values for depth and normals) [20]. And, a matching system takes a matching candidate as input and computes depth, normal, and a matching confidence using neighboring views supplied by local view selection. If the match is successful, the data is stored in depth, normal, and confidence maps and the neighboring pixels in R are added as new candidates to Q.

6.1 Region Growing

The idea behind the region growing approach is that a successfully matched depth sample provides a good initial estimate for depth, normal, and matching confidence for the neighboring pixel locations in R. The optimization process is nonlinear with numerous local minima, making good initialization critical, and it is usually the case that the depth and normal at a given pixel is similar to one of its neighbors. This heuristic may fail for non-smooth surfaces or at silhouettes.

Region growing thus needs to be combined with a robust matching process and the ability to revisit the same pixel location multiple times with different initializations. Prioritizing the candidates is important in order to consider matches with higher expected matching confidence first. This avoids growing into unreliable regions which in turn could provide bad matching candidates. We therefore store all candidates in a priority queue Q and always select the candidate with highest expected matching confidence for stereo matching.

In some cases, a new match is computed for a pixel that has previously been processed. If the new confidence is higher than the previous one, then the new match information overwrites the old. In addition, each of the pixel's 4-neighbors are inserted in the queue with the same match information, if that neighboring pixel has not already been processed and determined to have a higher confidence. Note that, when revisiting a pixel, the set of active views $\mathbf{A}$ is reset and allowed to draw from the entire neighborhood set $\mathbf{N}$ using the local view selection criteria.

Initializing the Priority Queue The SfM features visible in R provide a robust but sparse estimate of the scene geometry and are therefore well suited to initialize Q. We augment this set with additional feature points visible in all the neighboring views in $\mathbf{N}$, projecting them into R to determine their pixel locations. Note that this additional set can include points that are not actually visible in R; these bad initializations are likely to be over-written later.

Then, for each of the features points, we run the stereo matching procedure, initialized with the feature's depth and a fronto-parallel normal, to compute a depth, normal, and confidence. The results comprise the initial contents of Q.

6.2 Stereo Matching as Optimization

We interpret an $n \times n$ pixel window centered on a pixel in the reference view R as the projection of a small planar patch in the scene. Our goal in the matching phase is then to optimize over the depth and orientation of this patch to maximize photometric consistency with its projections into the neighboring views. Some of these views might not match, e.g., due to occlusion or other issues. Such views are rejected as invalid for that patch and replaced with other neighboring views provided by the local view selection step.

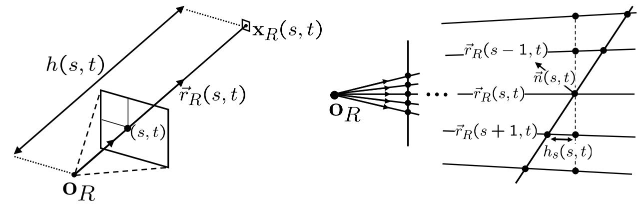


Figure 3. Parametrization for stereo matching. *Left:* The window centered at pixel (s,t) in the reference view corresponds to a point $\mathbf{x}_R(s,t)$ at a distance $h(s,t)$ along the viewing ray $\vec{r}_R(s,t)$. *Right:* Cross-section through the window to show parametrization of the window orientation as depth offset $h_s(s,t)$.

Scene Geometry Model We assume that scene geometry visible in the $n \times n$ pixel window centered at a pixel location (s,t) in the reference view is well modeled by a planar, oriented window at depth $h(s,t)$ (see Figure 3). The 3D position $\mathbf{x}_R(s,t)$ of the point projecting to the central pixel is then

$$\mathbf{x}_R(s,t) = \mathbf{o}_R + h(s,t) \cdot \vec{r}_R(s,t) \tag{6}$$

where $\mathbf{o}_R$ is the center of projection of view R and $\vec{r}_R(s,t)$ is the normalized ray direction through the pixel. We encode the window orientation using per-pixel distance offsets $h_s(s,t)$ and $h_t(s,t)$, corresponding to the per-pixel rate of change of depth in the s and t directions, respectively. The 3D position of a point projecting to a pixel inside the matching window is then

$$\begin{aligned} \mathbf{x}_R(s+i,t+j) = {} & \mathbf{o}_R + \\ & [h(s,t) + ih_s(s,t) + jh_t(s,t)] \cdot \vec{r}_R(s+i,t+j) \end{aligned} \tag{7}$$

with $i,j = -\frac{n-1}{2} \dots \frac{n-1}{2}$. Note that this only approximates a planar window but we assume that the error is negligible for small n, i.e., when $\vec{r}_R(s+i,t+j) \approx \vec{r}_R(s,t)$. We can now determine the corresponding locations in a neighboring view k with sub-pixel accuracy using that view's projection $\mathbf{P}_k(x_R(s+i,t+j))$. This formulation replaces the commonly used per-view window-shaping parameters (see e.g., [8]) with an explicit representation of surface orientation that is consistent between all views, thus eliminating excess degrees of freedom.

Photometric Model While we could in principle model a large number of reflectance effects to increase the ability to match images taken under varying conditions, this comes at the cost of adding more parameters. Doing so increases not only the computational effort but also decreases the stability of the optimization. We instead use a simple model for reflectance effects—a color scale factor c_k for each patch projected into the k-th neighboring view. Given constant illumination over the patch area in a view (but different from view to view) and a planar surface, this perfectly models Lambertian reflectance. The model fails for example when the illumination changes within the patch (e.g., at shadow boundaries or caustics) or when the patch contains a specular highlight. It also fails when the local contrast changes between views, e.g., for bumpy surfaces viewed under different directional illumination or for surfaces that are wet in some views but not others [16].

In practice, this model provides sufficient invariance to yield good results on a wide range of scenes, when used in combination with view selection. Furthermore, the range of views reliably matched with this model is well-correlated to images that match well using the SIFT detector.

MPGC Matching with Outlier Rejection Given the models in the previous section we can now relate the pixel intensities within a patch in R to the intensities in the k-th neighboring view:

$$I_R(s+i,t+j) = c_k(s,t) \cdot I_k(\mathbf{P}_k(\mathbf{x}_R(s+i,t+j))) \quad (8)$$

with $i,j = -\frac{n-1}{2} \dots \frac{n-1}{2}$, $k = 1 \dots m$ where $m = |\mathbf{A}|$ is the number of neighboring views under consideration. Omitting the pixel coordinates (s,t) and substituting in Equation 7, we get

$$I_R(i,j) = c_k \cdot I_k(\mathbf{P}_k(\mathbf{o}_R + \vec{r}_R(i,j) \cdot (h + ih_s + jh_t))). \quad (9)$$

In the case of a 3-channel color image, Equation 9 represents three equations, one per color channel. Thus, considering all pixels in the window and all neighboring views, we have $3n^2m$ equations to solve for $3 + 3m$ unknowns: h, h_s, h_t, and the per-view color scale c_k. (In all of our experiments, we set $n = 5$ and $m = 4$.) To solve this overdetermined nonlinear system we follow the standard MPGC approach [8, 1] and linearize Equation 9:

$$\begin{aligned} I_R(i,j) = {} & c_k \cdot I_k(\mathbf{P}_k(\mathbf{o}_R + \vec{r}_R(i,j) \cdot (h + ih_s + jh_t))) \\ & + \frac{\partial I_k(i,j)}{\partial h} \cdot (dh + i \cdot dh_s + j \cdot dh_t). \end{aligned} \quad (10)$$

Given an initial value for h, h_s, and h_t (which we then hold fixed), we can solve for dh, dh_s, dh_t, and the c_k using linear least squares. Then we update h, h_s, and h_t by adding to them dh, dh_s, and dh_t, respectively, and iterate.

In this optimization we are essentially solving for the parameters that minimize the sum of squared differences (SSD) between pixels in the reference window and pixels in the neighboring views. We could have instead optimized with respect to sums of NCC's. The behaviors of these metrics are somewhat different, however. Consider the case of a linear gradient in intensity across a planar portion of the scene. After removing the mean and normalizing, NCC would permit shifted windows to match equally well, resulting in an unwanted depth ambiguity. Now consider the case of an unshadowed planar region with constant albedo. The SSD optimization, after estimating the scale factor, will converge to a minimum with nearly zero error, essentially fitting to the noise. By contrast, after removing the mean, NCC is essentially measuring the correlation of the noise between two views, which will be low. In this case, NCC provides a good measure of how (un-)confident we are in the solution. As described below, we have opted to use SSD for the parameter estimation, while using NCC to measure confidence, as well as convergence.

While the iterative optimization approach described above tends to converge quickly (i.e., within a couple of iterations given good initial values), matching problems will yield slow convergence, oscillation, or convergence to the wrong answer [7]. We therefore include specific mechanisms into the optimization to prevent these effects.

We first perform 5 iterations to allow the system to settle. After each subsequent iteration, we compute the NCC score between the patch in the reference view and each neighboring view. We then reject all views with an NCC score below an acceptance threshold (typically $\kappa = 0.4$). If no view was rejected and all NCC scores changed by no more than $\epsilon = 0.001$ compared to the previous iteration, we assume that the iteration has converged. Otherwise, we add missing views to the active set and continue to iterate. The iteration fails if we do not reach convergence after 20 iterations or the active set contains less than the required number of views.

In practice, we modify the above procedure in two ways to improve its behavior significantly. First, we update the normal and color scale factors only every fifth iteration or when the active set of neighboring views has changed. This improves performance and reduces the likelihood of oscillation. Second, after the 14th iteration (i.e., just before an update to the color scale factors and normal), we reject all views whose NCC score changed by more than ϵ to stop a possible oscillation.

If the optimization converges and the dot product between normal and the viewing ray $\vec{r}_R(s,t)$ is above 0.1, we compute a confidence score C as the average NCC score between the patch in the reference view and all active neighboring views, normalized from $[\kappa \dots 1]$ to $[0 \dots 1]$. We use this score to determine how to update the depth, normal, and confidence maps and Q, as described in Section 6.1.

7 Results and Conclusion

We computed MVS reconstructions for several Internet CPCs gathered from Flickr varying widely in terms of size, number of photographers, and scale (see Table 1). Additional reconstructions are provided on the project web page [3]. Figure 4 shows for each site a sample view, the corresponding depth map, and a shaded rendering of the depth map. These results demonstrate that the MVS system can reconstruct detailed and high quality depth maps

Dataset	Images	Photographers	Scale range
Pisa Duomo	56	8	7.3
Trevi Fountain	106	51	29.0
Statue of Liberty	72	29	14.2
Notre Dame	206	92	290.2
St. Peter (Rome)	151	50	29.5

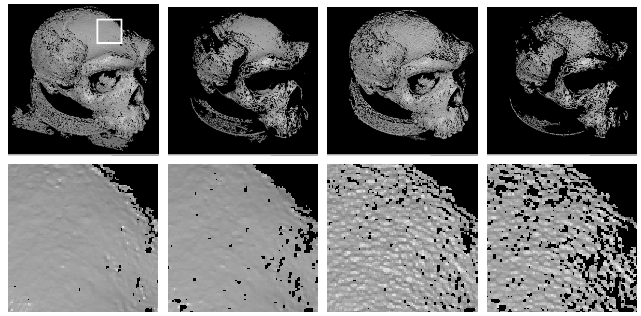
Table 1. Overview of the CPCs used in this paper.

Figure 4. Individual views from the Trevi, Statue of Liberty, St. Peter cathedral, and Pisa Duomo dataset, corresponding depth maps, and shaded renderings of each depth map.

for widely varying input data. The computation time varies with the number of reconstructed depth samples and the speed of convergence of the optimization. The depth map of St. Peter in Fig. 4, for example, contains 320K valid depth samples, and was reconstructed in 1.7 hours of CPU time (3.2 GHz Xeon).

Figure 5 uses the *nskulla* data set to demonstrate the effectiveness of two key ingredients of our approach—local view selection and optimization of normals. Local view selection enables more matches and improves completeness, even for datasets such as this one taken under laboratory conditions. Optimization of normals reduces noise as the patch better models the underlying surface geometry.

The individual depth maps can be combined into a single surface mesh using a variety of techniques. Figure 6 shows merged results for two CPCs using a Poisson surface reconstruction approach [14]. Note that this surface reconstruction approach performs fair hole-filling where no scene geometry is estimated. For objects only partially observed (e.g., only the front side), the hole-fill can extend well beyond the boundary of the observations. As a post-process, we automatically remove these spurious extensions using standard mesh filtering operations.



LVS+ON ON only LVS only neither

Figure 5. Effect of local view selection (LVS) and optimization of normals (ON) on a depth map from the nskulla model. The lower row shows an enlarged version of the marked area of the model.

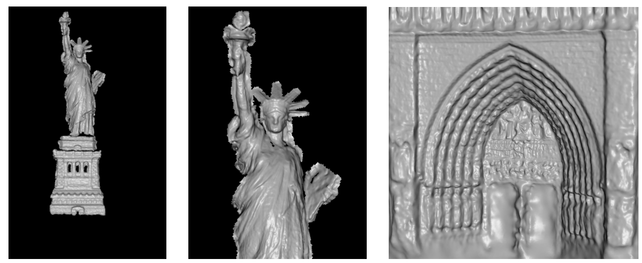


Figure 6. *Left and center:* Full merged model of 72 depth maps of the Statue of Liberty and close-up view. *Right:* Merged model of the central portal of Notre Dame cathedral (206 depth maps).

To compare the performance of our MVS approach with other state-of-the-art methods, we reconstructed two benchmark datasets from the MVS evaluation [22]. As the input images were captured using constant illumination, we fixed the color scale factor $c_k = 1$, excluding it from the optimization. The *templeFull* model achieved an accuracy of 0.42 mm at 98.2 % completeness. The *dinoFull* model achieved an accuracy of 0.46 mm at 96.7 % completeness. Both reconstructions are accurate to within a few hundredths of a millimeter of the top reported results, demonstrating that our approach is competitive with the current state-of-the-art for images captured under lab conditions.

To evaluate the quality of our reconstructions from CPC datasets, we created a merged surface model from the 56 depth maps in the Pisa Duomo dataset and compared it to a partial model of the Duomo acquired with a time-of-flight laser scanning system. Figure 7 shows both models and an overlaid comparison. Using the same accuracy metric described in [22], modified to avoid portions of the model not captured by the laser scanner, we found that 90% of

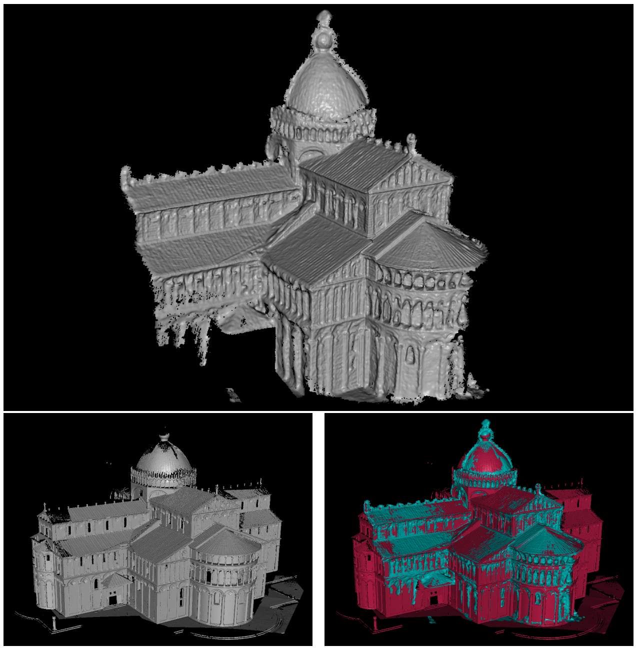


Figure 7. Comparison of the merged Pisa (top) model with a laser scanned model (bottom left). The false color rendering on the right shows the registered models overlaid on top of each other.

the reconstructed samples are within 0.128 m of the laser scanned model of this 51 m high building.

In conclusion, we have presented a multi-view stereo algorithm capable of computing high quality reconstructions of a wide range of scenes from large, shared, multi-user photo collections available on the Internet. With the explosion of imagery available online, this capability opens up the exciting possibility of computing accurate geometric models of the world's sites, cities, and landscapes.

Acknowledgments We thank all photographers who provided their images via Flickr (see the project page [3]), the Visual Computing Group at CNR Pisa for the Duomo model, Yasutaka Furukawa and Jean Ponce for the nskulla dataset, as well as Rick Szeliski for his helpful comments and suggestions. This work was supported in part by a Feodor Lynen Fellowship granted by the Alexander von Humboldt Foundation, NSF grants EIA-0321235 and IIS-0413198, the University of Washington Animation Research Labs, the Washington Research Foundation, Adobe, Microsoft, and an endowment by Rob Short and Emer Dooley.

References

[1] E. Baltsavias. *Multiphoto geometrically constraint matching*. PhD dissertation, ETH Zurich, 1991.

[2] M. Brown and D. G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *Proc. 3DIM*, pages 56–63, 2005.

[3] Project page. http://grail.cs.washington.edu/projects/mvscpc.

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

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proc. CVPR*, 2007.

[6] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *Proc. CVPR*, pages 2402–2409, 2006.

[7] A. Gruen. Least squares matching: a fundamental measurement algorithm. In *Close Range Photogrammetry and Mach. Vision*, chapter 8, pages 217–255. 1996.

[8] A. Gruen and E. Baltsavias. Geometrically constrained multiphoto matching. *Photogrammetric Engineering and Remote Sensing*, 54(5):633–641, May 1988.

[9] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Proc. CVPR*, 2007.

[10] A. Hornung and L. Kobbelt. Robust and efficient photo-consistency estimation for volumetric 3D reconstruction. In *Proc. ECCV*, pages 179–190, 2006.

[11] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo beyond Lambert. In *Proc. CVPR*, pages 171–178, 2003.

[12] G. Kamberov, G. Kamberova, O. Chum, S. Obdrzalek, D. Martinec, J. Kostkova, T. Pajdla, J. Matas, and R. JaBra. 3d geometry from uncalibrated images. In *Proc. ISVC*, pages 802–813, 2006.

[13] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. CVPR*, pages 103–110, 2001.

[14] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. SGP*, pages 61–70, 2006.

[15] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proc. ICCV*, pages 1033–1040, 2003.

[16] J. Lekner and M. C. Dorf. Why some things are darker when wet. *Applied Optics*, 27(7):1278–1280, April 1988.

[17] M. Lhuillier and L. Quan. Match propogation for image-based modeling and rendering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1140–1146, 2002.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60(2):91–110, 2004.

[19] P. J. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc. ICCV*, pages 3–10, 1998.

[20] G. P. Otto and T. K. W. Chau. 'Region-growing' algorithm for matching of terrain images. *Image Vision Comput.*, 7(2):83–94, 1989.

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

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

[23] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH Conf. Proc.*, pages 835–846, 2006.

[24] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *SIGGRAPH Conf. Proc.*, pages 548–558, August 2004.

[25] Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In *SMILE '00 Workshop on 3D Struct. from Mult. Images of Large-Scale Environments*, pages 68–85, 2001.