Beyond Silhouettes: Surface Reconstruction using Multi-Flash Photography

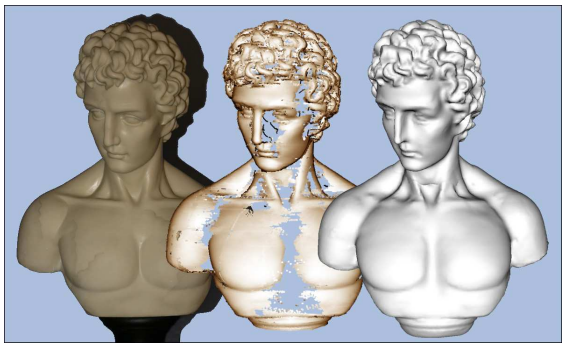
Daniel Crispell* Douglas Lanman* Peter G. Sibley† Yong Zhao* Gabriel Taubin*
Brown University, Providence, Rhode Island, USA

Abstract

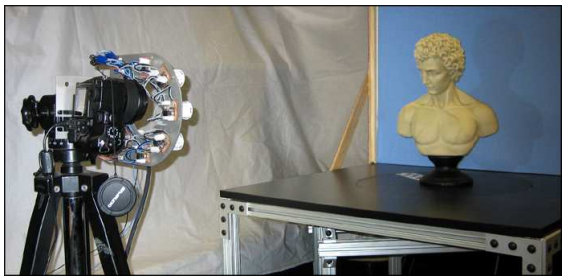
This paper introduces a novel method for surface reconstruction using the depth discontinuity information captured by a multi-flash camera while the object moves along a known trajectory. Experimental results based on turntable sequences are presented. By observing the visual motion of depth discontinuities, surface points are accurately reconstructed – including many located deep inside concavities. The method extends well-established differential and global shape-from-silhouette surface reconstruction techniques by incorporating the significant additional information encoded in the depth discontinuities. The reconstruction method uses an implicit form of the epipolar parameterization and directly estimates point locations and corresponding surface normals on the surface of the object using a local temporal neighborhood of the depth discontinuities. Outliers, which correspond to the ill-conditioned cases of the reconstruction equations, are easily detected and removed by back-projection. Gaps resulting from curvature-dependent sampling and shallow concavities are filled by fitting an implicit surface to the oriented point cloud's point locations and normal vectors.

1 Introduction

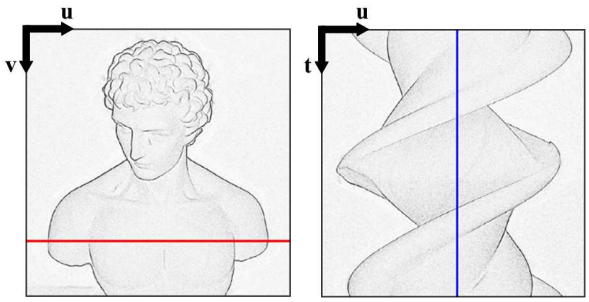
Many methods have been proposed to reconstruct the surface of an object from its occluding contours while it undergoes motion. Space carving and visual hull algorithms follow a global volumetric approach. The whole silhouette of the object from a viewpoint defines a solid *viewing cone* in 3D. The intersection of the viewing cones of an object from all possible viewpoints is called the visual hull. In practice the computed visual hull is an approximation obtained from a few viewing cones, and it is a volume containing the object [16, 13]. Although robust, the quality of the results are somewhat limited, especially for complex objects containing concavities and curved surfaces. An alternative differential approach uses the local deformation of the silhouettes as the camera moves relative to the object to estimate the depth of the points [6, 21]. Related methods use a dual-space approach, where tangent planes to the object surface are represented as points in dual space, and surface estimates can be obtained by examining neighboring points in the space [7, 14, 4, 15]. These systems provide a direct method of estimating depth based on information



(a) Reconstruction results. Left: an input image captured by the multi-flash camera. Center: the estimated oriented point cloud rendered in Pointshop3D [22]. Right: implicit surface fills in gaps.



(b) Experimental configuration: 8 Mpix 8-flash camera and computer controlled turntable.



(c) Depth edge confidence (d) Epipolar slice of confidence

Figure 1: Method Overview.

*{daniel_crispell, douglas_lanman, yong_zhao, taubin}@brown.edu
†pgs@cs.brown.edu

based only on a local region of camera motion, but are subject to singularities in degenerate cases. They also are not capable of modeling surface contours that do not appear as part of the object silhouette for any view e.g., structures protected by concavities.

More recently, algorithms have been proposed to combine image texture and color information with silhouette information [9, 11]. These methods use more information captured in the images, are capable of producing very accurate results, even recovering shape in areas protected by concavities, but at a high cost in algorithm complexity and running time. They are also highly non-linear and subject to local minima.

Our method extends the differential approach described above, using the visual motion of contours relative to camera motion. It uses, however, information about the motion of all visible depth discontinuities which can be estimated from image data, not just those occurring on the object silhouettes. This enables us to reconstruct structures protected by concavities that do not appear as part of the object silhouette in any view. Although silhouettes can be estimated in many cases using passive illumination (blue screen), most accurate estimation results from active illumination (backlighting). Our system uses active illumination to estimate the depth discontinuities from image data. It utilizes a camera with multiple flashes at known positions relative to the camera center, similar to the camera used by Feris et al. [10] to enhance dense stereo matching. In the experimental results presented in this paper, the location of the flashes with respect to the camera were known only in a rough sense. We plan to do a detailed error analysis in the near future.

1.1 Contributions

The method introduced in this paper integrates enhancements to a number of known results in a novel way. Its main features are its simplicity, and the fact that it can be trivially parallelized. Contributions include:

- A refined method to estimate depth discontinuities from images of an object undergoing motion along a known trajectory, captured using the multi-flash non-photorealistic camera proposed by Raskar et al. [19]. The output of this process is a space-time volume resulting from stacking up the depth discontinuity images in the order of capture. We analyze the properties of these images and discuss their relation to traditional silhouettes obtained by foreground segmentation.
- An algorithm to estimate point locations and surface normals from differential properties of smooth space-time curves fitted to ridges in epipolar slices of the space-time volume of depth discontinuities. This algorithm extends Cipolla's traditional method of depth recovery [6] to the data encoded in depth discontinuity edges not associated with silhouettes.

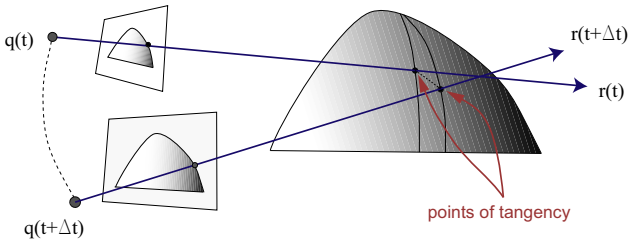


Figure 2: The tangent ray from the camera to the object slides over the surface as the camera moves. Depth can be estimated based on the apparent motion of the contour in the image plane relative to the camera motion in space.

- Surface points which do not produce an observable depth discontinuity cannot be estimated with this method, resulting in an uneven distribution of sample locations. We use Sibley's oriented point cloud implicit surface fitting method [20] to fill the resulting sampling gaps.

The experimental data presented shows that the new reconstruction method is simple, robust, and capable of reconstructing structure not recoverable using silhouette information alone. We review the results it is built upon in Section 2. In Section 3 we describe the new algorithm. In Section 4 we present experimental results, and our conclusions in Section 5.

2 Foundations

This paper builds upon a number of known concepts and results contributed by others. We describe them here.

2.1 Depth from visual motion of curves

The properties of surface shapes based on the apparent motion of their contours in images are well-studied [12, 6, 21]. In general, we represent a surface point p on a depth discontinuity edge as

$$p = q + \lambda r \tag{1}$$

where q is the camera center, r is the camera ray vector corresponding to a pixel $[u,v]$, and λ is the scaling factor that determines the depth. Cipolla and Giblin [6] showed that the parameter λ can be obtained from the following equation

$$\lambda = -\frac{n^t \dot{q}}{n^t \dot{r}} \tag{2}$$

where n is normal vector to the surface at the point p, and $\dot{r}, \dot{q}$ are derivatives in time as the the camera moves with respect to the object and the camera ray r "slides over" the object (Figure 2). This method assumes that the functions $q(t)$, $r(t)$, and $n(t)$, as well as their derivatives with re-

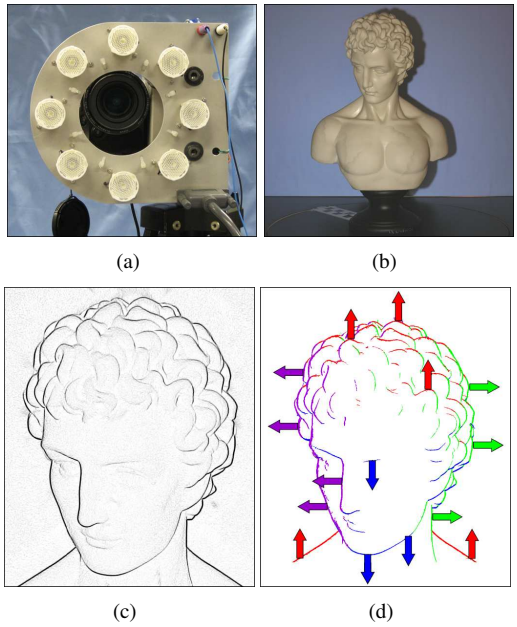


Figure 3: (a) Multi-flash camera. (b) Sample image acquired with flash located to the left of the camera's center of projection. (c) Depth edge confidence image produced by method in [19], with darker pixels representing a higher likelihood of a depth edge. (d) Approximate edge orientation corresponding to the flash with a maximum depth edge response. Up, down, left, and right edge orientations are shown in red, blue, purple, and green, respectively.

spect to t are known. The epipolar parameterization is used in [6] to construct these curves from multiple silhouettes. The main drawbacks to using Equation 2 to estimate depth are: its dependence on usually noisy derivatives; and its ill-conditioning close to frontier points, where $n(t)^t \dot{r}(t) \approx 0$.

2.2 Multi-Flash Photography

The non-photorealistic (NPR) camera introduced by Raskar et al. [19] was designed to detect depth discontinuities in a scene by using multiple point illumination sources. In general, an NPR camera is composed of a single image sensor and a set of flashes evenly distributed about the camera's center of projection, as shown in Figure 3(a). In order to distinguish depth edges from material edges, a single image is taken for each flash position (typically, four to eight flashes are used). If the separation of the flashes is small compared with the distance to the scene, then a narrow shadow will be observed adjacent to each depth discontinuity (see Figure 3(b)).

As presented in [19], a simple method exists to extract both the position and orientation of the depth edges from the multi-flash sequence. First, a *maximum composite* is formed by taking the largest intensity observed in each pixel over the multi-flash sequence. In general, this composite

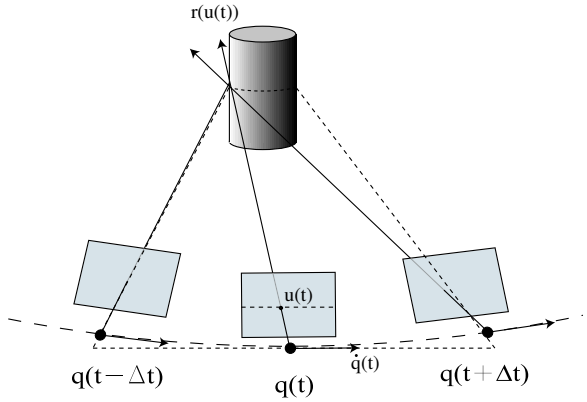


Figure 4: The epipolar plane (dotted line) used for curve parameterization is spanned by the viewing ray, r, and the camera's velocity vector, $\dot{q}$. The images are rectified such that the epipolar lines correspond to scan lines in the image. Unless the camera motion is linear, this plane is only an approximation for finite Δt, since the neighboring camera centers are, in general, not contained in the plane.

should be free of shadows created by the flashes. In order to amplify the shadowed pixels in each flash image (and attenuate texture edges), a *ratio image* is formed by dividing (per pixel) each flash image by the maximum composite. Afterwords, the depth edges can be detected by searching for negative transitions along the direction from the flash to the camera center (projected into the image plane) in each ratio image. With a sufficient distribution of flash positions and under some limiting assumptions on the baseline and material properties of the surface [19], this procedure will estimate a considerable subset of all depth discontinuities in the scene. An intermediate output of this process is the *depth edge confidence image* corresponding to the likelihood of a pixel being located near a depth discontinuity (see Figure 3(c)).

2.3 Camera Model

We use the standard pinhole camera model with projection matrix

$$P = K \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

where R is a 3x3 rotation matrix and T is a 3x1 translation vector relating the world coordinate frame to that of the camera. K is a 3x3 matrix containing the camera's intrinsic projection parameters. We recover these parameters along with 5 radial and tangential distortion coefficients using Bouguet's camera calibration toolbox [3]. We project image points in homogeneous coordinates to vectors in world space using the "inverse" projection matrix, $\hat{P}$.

$$\hat{P} = \begin{bmatrix} R^t & -R^t T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} K^{-1} \tag{4}$$

2.4 Epipolar Parameterization

The *epipolar parameterization* for curved surfaces has been extensively studied in the past [2, 12, 6, 21]. For two cameras with centers q_1 and q_2, an epipolar plane is defined as the plane containing q_1, q_2, and a world point X being imaged. The epipolar planes slice the image planes, forming a pencil of *epipolar lines* in each image, and each point in one image corresponds to an epipolar line in the other. A point x_1 along an apparent contour in one image is therefore matched to a point x_2 in the second image by intersecting the epipolar line defined by q_1,q_2, and x_1 with the corresponding apparent contour in the second image. For a continuous path of camera centers, $q(t)$, an epipolar plane at time t is spanned by the tangent vector $\dot{q}(t)$ to $q(t)$ and a viewing ray $r(t)$ from $q(t)$ to a world point p. So called *frontier points* occur when the epipolar plane is identical to the tangent plane of the surface. In these cases, the denominator of Equation 2 approaches zero, causing unreliable depth estimates. Giblin and Weiss [12] have presented an alternate expression for depth that avoids this mathematical instability, but in our experiments the depth estimates remained unstable at frontier points. This is most likely due to the imprecision of matching when the epipolar lines are tangent to the surface contours.

We rectify each image so that the camera velocity at the time of capture is parallel to the image x axis. By stacking the images from a sequence and "slicing" across a single scanline, we have an approximation to the epipolar constraint in local regions (Figure 4). We refer to these images containing a scanline from each image as *epipolar slices*. By tracking the motion of apparent contours in the slices, we are in effect implicitly utilizing the epipolar constraint for curve matching.

3 Algorithm

3.1 Data Capture and Pre-processing

We use a turntable and stationary 8 megapixel digital camera to acquire data from 670 viewpoints in a circular path around the object. For each turntable position, we capture four images using illumination from the top, left, right, and bottom flashes of the camera, respectively. We experimented with using all eight flash positions of the camera, but found that it did not provide significant improvements over using only four. The camera is assumed to be intrinsically calibrated, and its position and orientation with respect to the turntable is determined using a calibration grid placed on the table. Once the data has been captured, we rectify each of the images to remove any radial distortion, and to align the camera x axis with the direction of camera motion (i.e. perpendicular to the turntable axis of rotation and with zero translation in the x direction). Once the images are rectified, we then execute the algorithm described below to compute images of depth discontinuity for each of the camera positions. These discontinuity images are finally converted into m epipolar slices, where m is the number of scan rows in the images. The slice images are $m \times l$ pixels in size, where m is the number of columns in the original input images, and l is the number of camera positions captured. Our camera is high resolution, producing images of size 3200×2400. The data capture and preprocessing steps are by far the most time-consuming steps of the algorithm, taking on the order of 5 hours per object. Downloading the image data through the camera's USB 1.0 port is the most time consuming part.

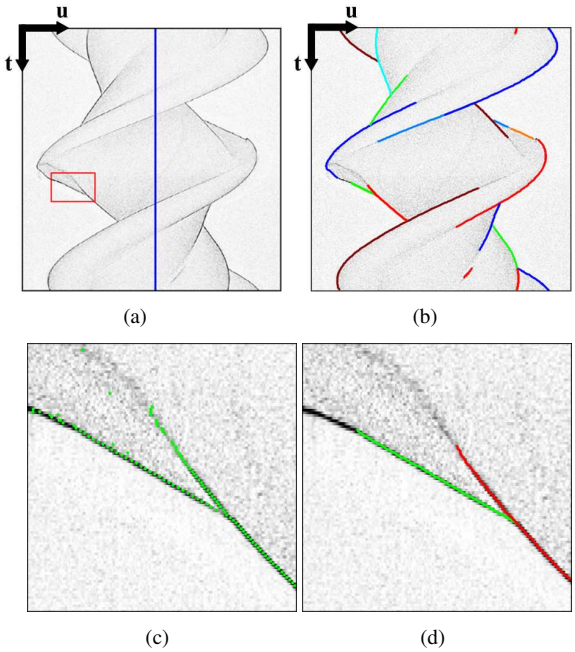


Figure 5: (a) Epipolar slice with axis of rotation in blue and inset region in red. (b) Estimated depth contours. (c) Subpixel depth discontinuities shown in green. (d) Edge linking performance at junctions. Each color represents a different edge chain.

3.2 Depth-Discontinuity Estimation

For this paper, we introduce several modifications to the depth edge detection algorithm presented in [19]. The NPR camera developed by Raskar et al. was originally used to generate stylized non-photorealistic imagery. For such applications, pixel-level accuracy in the depth edges is sufficient. In order to reconstruct surfaces, however, sub-pixel accuracy is required (see Section 3.3). In addition, an estimate of the depth edge normal is required. At a course level, the direction of the depth edge normal (oriented from foreground to background) can be inferred from the flash which produces the strongest depth edge at a given point.

That is, if a certain flash has the largest negative transition in the ratio image at a given point, then the depth edge normal, projected into the image plane, is opposite the direction from the camera center to this flash. As an example, consider the estimate of depth edge orientation generated using four flashes in Figure 3(d).

3.3 Contour Tracking in Epipolar Slices

As previously discussed, the proposed reconstruction method requires tracking the motion of apparent contours in epipolar slices. This problem can be solved using a form of edge following optimized for this task. In particular, we decompose the contour tracking problem into three stages: (1) subpixel edge detection, (2) edge linking, and (3) polynomial curve fitting. Since the epipolar slices can be evaluated independently, we accelerate contour tracking through parallel processing.

As shown in Figure 5(a), the epipolar slices represent the confidence that a certain pixel contains a depth discontinuity for any given camera position. We begin by detecting the pixel-level position of the depth discontinuities by applying a two-level hysteresis threshold. Afterward, we estimate the subpixel position of each depth discontinuity by fitting a quadratic polynomial to the neighboring confidence values. Non-maximum suppression is applied to ensure that a single subpixel position is assigned to each depth edge. The output of the subpixel depth edge detection stage is illustrated in Figure 5(c).

As shown in Figure 5, the epipolar slices are complex and typically contain many junctions, indicating points of bitangency. These junctions emerge for a variety of reasons, including when external silhouettes becomes internal contours (and vice versa). Our edge linking algorithm follows edges through such transitions. We initialize the tracking process by finding the first detection to the left of the axis of rotation in an epipolar slice. Next, we search for the closest detection in the neighboring views within a small window of about ± 5 columns. If any match is found, then we initiate a track using a linear prediction based on these two observations. We proceed to search for new detections within a neighborhood of the predicted edge position (generally three views ahead/behind and ± 5 columns). The closest detection (if any) to the prediction is added to the track and neighboring detections are removed from future consideration. Once three or more detections have been linked, we predict the next position using a quadratic model. In general, the prediction model is fit using a sliding window of the last 15 detections. If a track ends, a new edge chain is initiated using the first available detection either to the left or right of the axis of rotation. This process continues until all detections have been considered. While simple, this tracking method consistently and accurately links depth discontinuities through junctions. For example, consider the edge chains shown in Figure 5(d).

Once the subpixel detections have been linked, a quartic polynomial is fit to each chain – providing an analytic model for the motion of depth discontinuities as a function of viewpoint. Typical results achieved using this method are shown in Figure 5(b).

3.4 Oriented Point Cloud Generation

Once the curves in an epipolar slice have been extracted, we are able to robustly and directly estimate the depth of the points on the curve. For a given epipolar slice image, we have constant $v = v_s$ and image axes corresponding to u and t, where, for a given contour, u is function of t. We therefore express Equation 1 as:

$$p(u(t),t) = q(t) + \lambda\, r(u(t),t) \tag{5}$$

and Equation 2 as

$$\lambda = -\frac{n^t(u(t),t)\,\dot{q}(t)}{n^t(u(t),t)\,\frac{d}{dt}\{r(u(t),t)\}} \tag{6}$$

where

$$\frac{d}{dt}\{r(u(t),t)\} = \frac{\partial r}{\partial u}(u(t),t)\,\dot{u}(t)\ . \tag{7}$$

We can obtain $\frac{\partial r}{\partial u}(u(t),t)$ directly from the inverse projection matrix (Equation 4) associated with camera position $q(t)$:

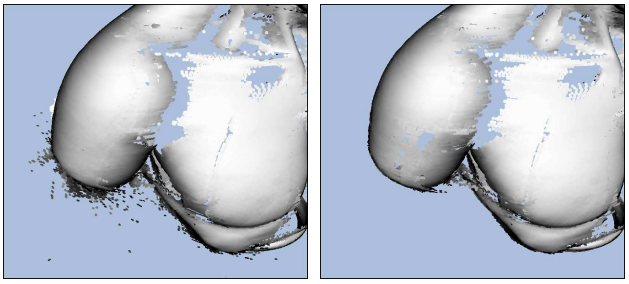
$$\frac{\partial r}{\partial u}(u(t)) = \begin{bmatrix} \hat{P}_{1,1}(t) \\ \hat{P}_{2,1}(t) \\ \hat{P}_{3,1}(t) \end{bmatrix} \tag{8}$$

The contour path's motion in the u direction, $\dot{u}(t)$, can be obtained directly from the coefficients of the curve fit to the contour path (Section 3.3) in the slice image. We estimate the image normal $m(u(t),t)$ by performing principal component analysis (PCA) on a local region about the point $(u(t), v_s)$ in the original depth edge image corresponding to time t. To determine consistent normal orientations we compare with the coarse normal information given by the flash with the maximum depth edge response (Section 2.2). The surface normal $n(u(t),t)$ in 3-D must then be perpendicular to the viewing ray $r(u(t),t)$, and contained in the plane spanned by $r(u(t),t)$ and the projection of the $n(u(t),t)$ onto the image plane, $m(u(t),t)$.

$$n(u(t),t) = (\hat{P}(t) \begin{bmatrix} m(u(t),t) \\ 0 \end{bmatrix} \times r(u(t),t)) \times r(u(t),t) \tag{9}$$

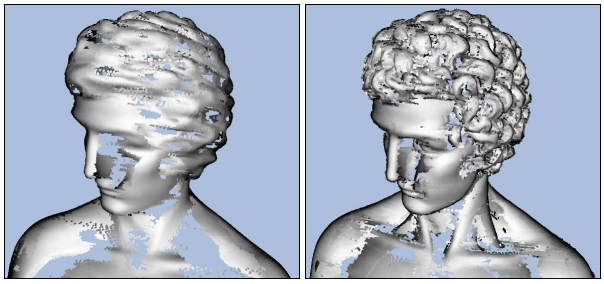
Plugging back in to Equation 6, we can now recover the depth of any point on the contour path, assuming known camera motion $\dot{q}(t)$. Ours is the simple case of circular motion, so $\dot{q}(t)$ is well defined for all t.

Each curve in each slice is processed independently, and sampled uniformly in t. This sampling in t causes the re-



(a) (b)

Figure 6: (a) A portion of the bust point cloud, generated with no outlier rejection. An area of instability can be seen under the arm, where the surface is nearly perpendicular with the axis of rotation. (b) Outliers removed by back-projection validation using a small set of segmented images.



(a) (b)

Figure 7: (a) The generated point cloud using our algorithm with silhouette information only. (b) Reconstruction using all depth discontinuities. Notice increased detail in the eyes, hair, and neck concavities.

constructed points to be sampled very densely in areas of high curvature (since the viewing ray moves slowly over these regions) and conversely, very sparsely in areas of very low curvature, i.e. planes. The dense sampling in areas of high curvature can be dealt with by using decimation as a post-processing step, but the sparse areas provide more of a challenge. We will address this in future work.

3.4.1 Outlier Detection

To deal with instability near frontier points, we perform the simple validation proposed by Liang and Wong [15]. We segment the object from the background in a small subset (15 views) of the original input images. We then back-project the reconstructed points into the images, making sure that each point lies within the image foreground. For the bust data set, 3.7% of points were removed in this way (Figure 6).

3.5 Surface Reconstruction

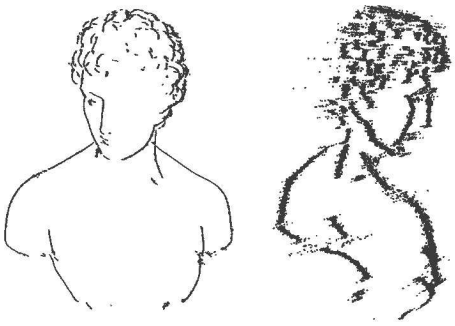
Dense point sets have been previously used as surface representations for rendering and interactive modeling applications [1, 22]. Most systems focus on uniformly sampled data, however irregularly sampled point clouds such as those generated by our method usually require hole filling. Diffusion-based hole filling methods for meshes [8] and point clouds [18] have been developed. Other hole-filling approaches instead produce an approximating surface, frequently in the form of a polygonal mesh or an implicit function. Methods that generate an implicit surface frequently cope with gappy data and irregular sampling patterns more gracefully [17] than mesh-based algorithms. Two representative methods are those proposed by Carr et al. [5] and Ohtake et al. [17]. Carr et al. fit implicits composed of radial basis functions to dense range data, while Ohtake et al. fit an implicit consisting of blended quadratics defined on an adaptive octree. These methods produce high quality surfaces, but are complex. Instead, we implemented Sibley's method [20], which reduces to solving a linear least squares problem. Given an oriented point cloud $\mathscr{D} = \{(p_1,n_1),\ldots,(p_m,n_m)\}$ sampled from a surface M, the method computes an implicit surface $M' = \{p|f(p) = 0\}$ where $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a scalar function, such that ideally $\nabla f(p_i) = n_i$, and $f(p_i) = 0$. If p_α denotes the position of a grid node, the problem reduces to the minimization of the following quadratic energy

$$E = \sum_i \|\nabla f(p_i) - n_i\|^2 + \lambda \sum_{(\alpha,\beta)} \|\nabla f(p_\alpha) - \nabla f(p_\beta)\|^2$$

where (α,β) are edges of the grid, and $\lambda > 0$ is a regularization constant. The scalar field f is represented as a linear combination of basis functions (e.g., trilinear) defined on a uniform Cartesian grid, $f(p) = \sum_\alpha f_\alpha \phi_\alpha(p)$, where $f_\alpha = f(p_\alpha)$. The gradient is approximated with finite differences. Finally, a polygonal mesh is extracted using Marching Cubes for visualization purposes.

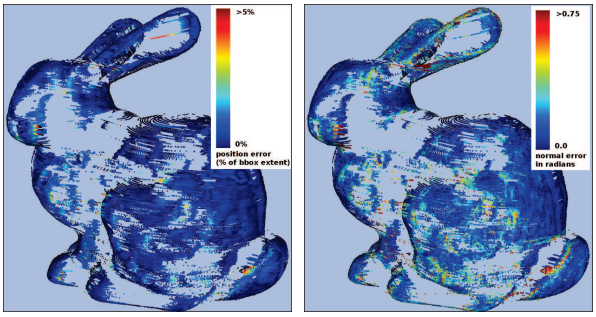
4 Experimental Results

Figure 10 shows the generated oriented point clouds and surface fits for two objects of considerable complexity. The point clouds were rendered in Pointshop3D [22] using the splatting technique. The bust reconstruction contains $1,088,757$ points, and the implicit surface reconstruction was computed using a pair of stacked grids of size 132^3. The extracted polygonal mesh has $221,998$ faces. The hand reconstruction contains $584,721$ points, fit using a grid of size 132^3. The extracted mesh contains $35,316$ faces. In both cases, the surface fitting algorithm has successfully filled in the missing regions of the point cloud where the objects have low surface curvature. The point cloud generation took on the order of 20 minutes for each dataset, distributed over 16 processors. The surface fitting took on the order of 15 minutes running on a single processor.



(a) (b)

Figure 8: A reconstruction of the bust using only 11 views over a 5.9 degree rotation of the turntable. (a) A viewpoint close to that of the input images. (b) A viewing direction nearly orthogonal to that of the input images.



(a) (b)

Figure 9: Synthetic results, showing 399,862 points reconstructed. (a) Points colored according to position error, red indicating higher error. The mean value is 0.1% of the bounding box. (b) Points colored according to normal error. The mean normal error is 0.17 radians.

In order to understand how much extra information we are able to capture by using the interior depth discontinuities, we captured a similar data set of the bust, but used foreground segmentation to capture the silhouette of the object only. Figure 7 shows the results. Using silhouette information only does not allow us to capture nearly as much detail in areas of concavity such as the eyes and hair.

Although our current algorithm achieves accurate results by using information from a dense set of viewpoints over a wide variation of views, significant information can be extracted using only local information from a relatively small number of viewpoints. Figure 8 shows a set of reconstructed points using information from 11 very similar views taken over a 5.9 degree rotation of the turntable.

As a preliminary estimate of algorithm accuracy, we generated a set of synthetic 1024×768 depth edge images of the Stanford bunny (69,451 face) mesh. We then extract curves from the epipolar slices and construct an oriented point cloud as described in Section 3. Figure 9 shows the results, with the point cloud colored according to (a) distance to mesh, and (b) normal error. Position errors are normalized by the extent of the bounding box. Mean position error is 0.11%, with a standard deviation of 9.40e-4. The median position error is more than ten times smaller than the mean, indicating that some outliers remain, even after back-projection filtering (Section 3.4.1). The mean normal error is 0.1714 radians, with a standard deviation of 0.2838. The median error is 0.1043 radians. In the future, we will compare our reconstructions of real data with results produced using a laser scanner.

5 Conclusions and Future Work

We have presented a novel method for surface reconstruction using depth discontinuity images generated by a multi-flash camera. The method accurately reconstructs points on objects with complex features, including those located within concavities and not part of the object silhouette from any vantage point. The depth estimation procedure is direct and does not require solving any non-linear optimization problems. The generated oriented point clouds tend to have gaps in areas of very low curvature, but we demonstrate a surface fitting algorithm that is able to bridge the gaps in most cases. Our future work will involve a more robust hole-filling solution, either as a post-processing step, or as a re-sampling of the epipolar slice curves. We will also further examine the trade-off between reconstruction accuracy and the density and disparity of the set of input images.

References

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

[2] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1), March 1987.

[3] J.-Y. Bouguet. Complete camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc.

[4] M. Brand, K. Kang, and D. Cooper. Algebraic solution for the visual hull. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

[5] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH 2001*, 2001.

[6] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.

[7] G. Cross and A. Zisserman. Quadric surface reconstruction from dual-space geometry. In *IEEE International Conference on Computer Vision (ICCV'98)*, 1998.

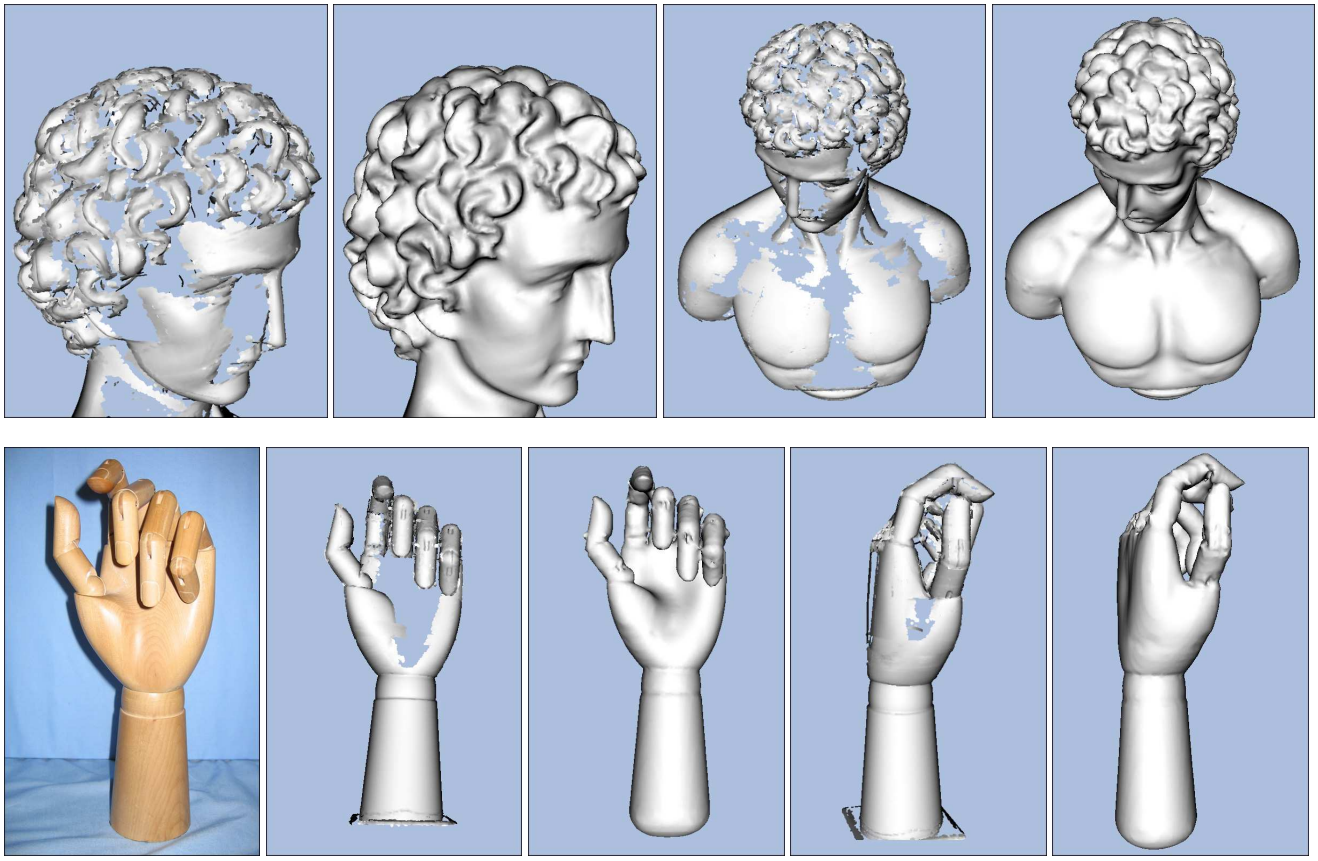


Figure 10: Summary of reconstruction results. Top row from left to right: two aligned pairs of estimated point clouds (1M points) and polygonal mesh reconstructions (220K faces) for the bust model. Bottom row from left to right: an input image of the hand model followed by two aligned pairs of estimated point clouds (600K points) and polygonal mesh reconstructions (35K faces).

[8] J. Davis, S. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *3DPVT 2002*, 2002.

[9] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. In *International Conference on 3-D Digital Imaging and Modeling (3DIM'03)*, 2003.

[10] R. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk. Discontinuity preserving stereo with small baseline multi-flash illumination. In *IEEE International Conference in Computer Vision (ICCV'05)*, 2005.

[11] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer vision 2006*, 2006.

[12] P. J. Giblin and R. S. Weiss. Epipolar curves on surfaces. *Image and Vision Computing*, 13(1):pp. 33–34, February 1995.

[13] K. Grauman, G. Shakhnarovich, and T. Darrell. A bayesian approach to image-based visual hull reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

[14] K. Kang, J.-P. Tarel, R. Fishman, and D. Cooper. A linear dual-space approach to 3d surface reconstruction from occluding contours using algebraic surfaces. In *IEEE International Conference on Computer Vision (ICCV'01)*, volume I, pages 198–204, 2001.

[15] C. Liang and K.-Y. K. Wong. Complex 3d shape recovery using a dual-space approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.

[16] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH 2000*, 2000.

[17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3), 2003.

[18] S. Park, X. Guo, H. Shin, and H. Qin. Shape and appearance repair for incomplete point surfaces. In *IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, 2005.

[19] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph.*, 23(3):679–688, 2004.

[20] P. G. Sibley and G. Taubin. Vectorfield Isosurface-based Reconstruction from Oriented Points. In *SIGGRAPH '05 Sketch*, 2005.

[21] K.-Y. K. Wong. *Structure and Motion from Silhouettes*. PhD thesis, University of Cambridge Department of Engineering, 2001.

[22] M. Zwicker, M. Pauly, O. Knoll, and M. Gross. Pointshop3D: an interactive system for point-based surface editing. In *SIGGRAPH 2002*, 2002.