Multiview Reconstruction of Complex Organic Shapes

Jasenko Zivanov
jasenko.zivanov@unibas.ch
Thomas Vetter
thomas.vetter@unibas.ch

Department of Mathematics and
Computer Science
University of Basel
Basel, Switzerland

Abstract

We propose a novel narrow baseline multiview stereo surface reconstruction method that is specifically aimed at complex shapes of biological origin that show many thin protrusions and curved occluding contours. Our method is built around fitting local quadrics to occluding contours and it thus avoids any planarity assumptions that are common to other state of the art methods.

We describe a complete pipeline that begins with calibrated noisy images and produces a final watertight surface. We present a novel technique to detect pixel precise internal contours and to fit local quadrics to them. This procedure is designed to deal with curved occluding contours, and it is very robust to noise and to slow changes in surface radiance. Our method can even reconstruct shapes from sequences where the illumination is attached to the observer and not the scene. We demonstrate the potential of our method by reconstructing the intricate shape of a tiny insect from images taken under a scanning electron microscope.

1 Introduction

Multiview stereo, the estimation of shapes from sets of images, has been one of the core problems of computer vision for decades. The field has seen continuous progress over the years, producing methods able to reconstruct shapes of increasing complexity with increasing precision under increasingly challenging conditions. By complexity, we specifically refer to the density of contours seen in the images. While simple, convex objects are defined by their silhouettes, complex shapes feature many internal contours that outline many protruding, often thin parts of the object. In recent years, a number of methods have been proposed that, in addition to photoconsistency cues, exploit the information contained in the internal contours [7, 8, 11, 13]. This enables those methods to reconstruct highly complex shapes at a precision approaching the resolution of the original images.

Still, most current state of the art methods make a number of different, either explicit or implicit planarity assumptions along the way that make them inadequate to deal with complex organic geometry. By organic geometry we refer to shapes that have been created in a natural growth process, instead of being engineered by humans. Such geometry rarely shows any sharp hinges, but instead many curved areas that give rise to curved contours. Such contours do not correspond to static lines on the surface of the object - instead, the corresponding

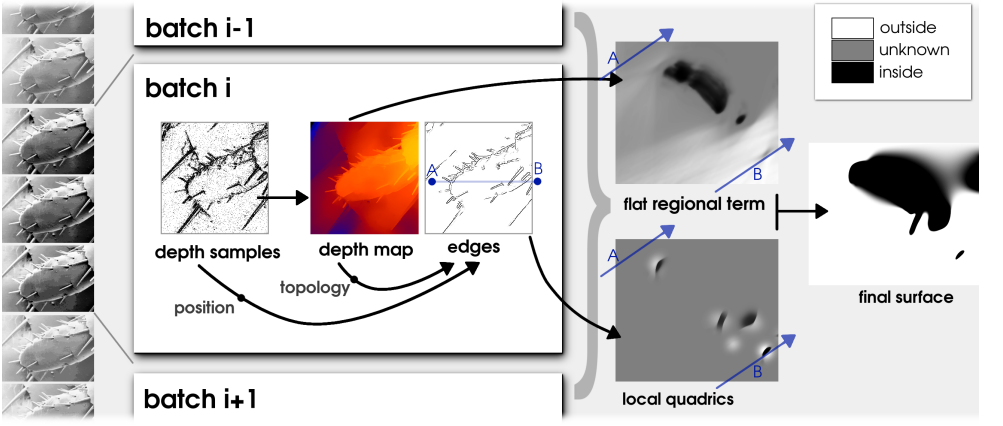

Figure 1: An overview of our method. The image sequence is split into batches, and for every batch, we compute a set of initial sparse depth samples. From those, we estimate a dense depth map that conforms to the image structure and we use it to detect contours. The contours from all the batches are combined to fit a set of unconstrained local quadrics in space. The final surface is then optimized so as to conform to the quadrics in the curved areas and to the depth maps in the flat areas.

lines shift along the surface as the vantage point changes. Furthermore, organic objects are often partially untextured, so that the contour of a protruding curved feature can be the only evidence of that feature.

The method we are proposing aims to reconstruct such complex organic surfaces at the level of precision allowed for by the resolution of the images. We assume that that level of precision can only be attained if the internal contours are considered as a source of information. Our method is built around the explicit estimation of local quadrics from the observed contours. We only assume that the quadric parameters vary smoothly across space, but we make no further assumptions on the curvature of that quadric. This ensures that our model will treat arbitrarily curved shapes as equally likely, and it allows us to reconstruct even very thin cylindrical structures as long as we can detect their contours.

Our contribution is a complete reconstruction pipeline for the image-based reconstruction of complex organic surfaces. That pipeline comprises (1) a novel initial per-pixel depth estimation technique that can deal with curved contours and that generates both a sparse depth map and a denoised version of the image, and (2) a novel method to detect contours and to fit quadrics to them. Our method then computes the final surface by combining the quadrics with a traditional regional term obtained from dense depth maps as input to a Poisson reconstruction method. The result is a watertight surface that conforms to the quadrics in the highly curved areas and to the depth maps in the even areas.

To detect the contours, we first compute a dense depth map that conforms to the structure of the image and that interpolates between the sparse depth samples. The dense depth map computation is equivalent to a process of inference that propagates depth information from the edges of the image into the smooth areas. Only through that process can we learn which edges in the image correspond to contours and which to lightness changes on the surface. It also tells us the orientation of the contours.

In order to estimate the initial per-pixel depth values in the presence of curved contours,

our method can only rely on a small subset of the images that have been taken from a similar vantage point. As a consequence, our algorithm is also very robust to deviations from Lambertian reflectance, and it can even work under conditions where the illumination is static with respect to the observer and not the scene. Note that in certain imaging modalites (*e.g.* electron microscopy) and circumstances (*e.g.* deep sea exploration), this is the only feasible configuration.

Finally, we demonstrate the abilities of our method by reconstructing the shape of a cat flea (ctenocephalides felis) from images taken under a scanning electron microscope.

2 Previous Work

Traditional multiview stereo (MVS) algorithms that rely solely on photoconsistency information have difficulties reconstructing complex geometry, i.e. geometry with a large number of internal contours. The reason is that such algorithms rely on comparisons between image areas in different images, be it by fitting planar patches in 3D space [6] or by estimating a depth map using matching windows [9, 15]. This is only sensible if the entire area being considered belongs to the same suface of the object - close to the contours, this estimation is unreliable.

It has been shown that the inclusion of additional silhouette information can greatly improve the precision around the external contours of the object, but it requires either an a priori segmentation of the image into a foreground and a background [4, 5, 14], or at least a capture setup where the two areas are roughly distinguishable by color [10]. It was only very recently that the first methods have been published that also exploit the information contained in the internal contours [7, 8, 11, 13].

In [7], the autors reformulate the reprojection error functional to include internal contours, to enable local surface optimization algorithms to adhere to the observed contours. Kim et al. [8] propose a method of estimating pixel precise depth maps using an iterative scheme. Their method begins with the estimation of the optimal depth of every pixel. Next, a set of inlier images is determined in which that point is visible and from that set, the next expected color of the pixel is averaged, leading to the set of inliers for the next iteration. Shan et al. [13] extend the patch based PMVS framework [6] to include information from internal contours. They do so by estimating depth maps from planar PMVS patches. Finally, Liu and Cooper [11] propose an MRF based method that explicitly solves an inverse ray tracing problem. Their method is currently not applicable to large scenes due to its extraordinary memory footprint.

Although much progress has been made on the reconstruction of complex geometry, there has been comparatively little recent work on curved, organic geometry. In 2000, Cross and Zisserman [3] have written about the reconstruction of quadrics from contours and point correspondences. Their work assumes the entire object to be one single quadric. In 2003, Cipolla et al. [2] proposed a quadric based surface model to track human hands. In the same year, Xie et al. [16] describe a method of fitting smoothly varying quadrics to a noisy point cloud in order to reconstruct the original surface. Their method is aimed at arbitrary point clouds and it does not take images into consideration.

3 Reconstruction Pipeline

In the following section, we will describe the individual processing steps that begin with a set of calibrated images and in the end produce the final watertight surface.

We assume that we are given the calibration in the following form. For every image i, we are given a 3×4 matrix $P_i = (p_1, p_2, p_3)^T$ that maps points in homogeneous 3D coordinates to points in the projective plane, such that point $x = (x_1, x_2, x_3, 1)^T$ in space corresponds to point (y_1, y_2) in the image, with $y_k p_3^T x = p_k^T x, \ k \in \{1,2\}$.

3.1 Initial Depth Samples

Our reconstruction pipeline begins with the estimation of intial sparse depth maps for a subset of the images. Most MVS methods begin with a similar depth estimation step, but almost all of them consider image areas larger than one pixel. The method by Kim et al. [8] is the only recent attempt known to us to estimate the depths of individual pixels independently of each other. Unlike theirs, our method is non-iterative and it estimates the depths simultaneously to a denoised version of the image. Our method can therefore exploit image edges that are too faint to be detected amid the noise of one single image.

The purpose of those depth maps is to capture both the actual depths of surface features and the momentary depths of curved contours. Curved contours do not project onto static curves on the surface of the object, as do contours generated by hinges. Instead, those curves slide across the surface as the vantage point changes. In order to capture the momentary positions of the curves, we can thus only consider a small selection of images taken from a similar vantage point. If we were to use the entire sequence, then our estimates at the ends of the the sequence would be biased towards the momentary position around the middle of the sequence.

For this reason, we split the image sequence into batches, and we use only images from one batch to compute a sparse depth map corresponding to the central image of that batch. The batch needs to cover a large enough range of viewing directions so as to allow for a reasonable depth estimation, and it must contain a sufficient number of images to overcome the noise - otherwise, we want it to be as short as possible. Although we tacitly assume that we are dealing with an ordered sequence of images, that assumption is only needed to separate the set of images into batches. In the case of an unordered set of images, the batches could be replaced by clusters of images. From this point forward, it is no longer assumed that the images belong to an ordered sequence.

The sparse depth map estimation is designed to be very robust to pixel noise and to still produce pixel precise results. It works on each pixel independently, and it consist of computing both the most likely color and the most likely depth value there. We assume that the observed color components are mutually independent, and that the value of each component c is normally distributed around the actual value c_0. We further assume that the standard deviation of that distribution is an affine linear function of c_0:

$$c \sim \mathcal{N}(c_0, \sigma(c_0)^2), \qquad \sigma(c) = \sigma_0 + c\sigma_m \tag{1}$$

The parameters σ_0 and σ_m are estimated from the smooth areas of the images.

For every pixel (x,y), we discretize the corresponding ray over the area of interest. This yields a sequence of spatial positions r_z, that correspond to color values $c_{z,i}$ in all the images i. We now also discretize over the color domain, and for every color value c, we compute its likelihood given the observed values $c_{z,i}$.

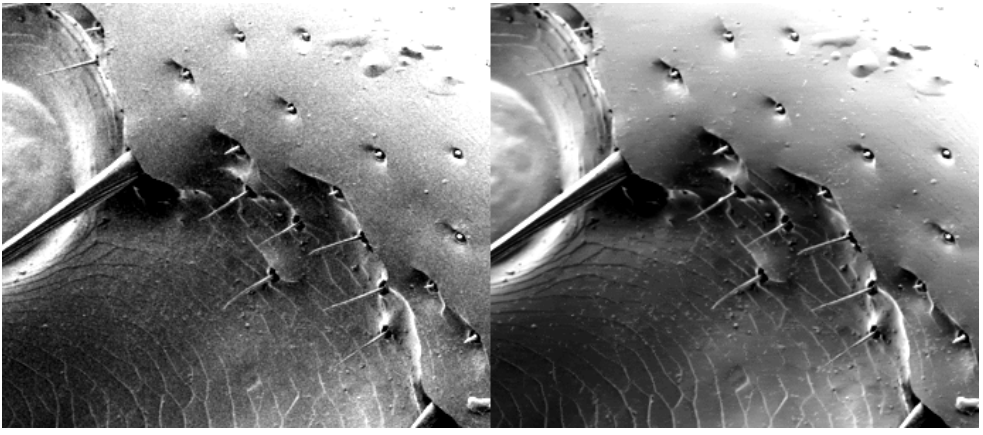

Figure 2: The denoising effect of our depth estimation process. The original image (left) and the denoised version (right). The corresponding batch contained 50 images. The contrast of the displayed images has been enhanced to emphasize the noise.

Since we are dealing with complex geometry, we need to assume that the surface point is occluded in some images. For that reason, we make our estimation robust to outliers by assuming that the probability of observing a certain color c can be no less than a threshold value p_τ, no matter how different c is from the true color c_0. This allows our model to explain away features of arbitrary color that can occlude the point in any image. For color values close to c_0, we still assume the aforementioned noise model. In log-space, this truncated normal distribution is then equivalent to a truncated parabola as the cost function.

The likelihood of color c given a set of observations $c_{z,i}$ is then equal to,

$$p(c|c_{z,i}) = \prod_i e^{-E(c,c_{z,i})} \tag{2}$$

$$E(c|\hat{c}) = \sum_{k\in\{r,g,b\}} \max((c_k - \hat{c}_k)^2/\sigma^2(c_k), \log(p_\tau)) \tag{3}$$

Only for the center image of the batch, the one for which we are computing the depth map, we set $p_\tau = 0$, since we know that the point cannot be occluded in the reference frame - the point visible there is always the point we mean. For all other frames, we have assumed $p_\tau = e^{-4}$, which is equivalent to cutting off the normal distribution at two standard deviations.

Evaluating this probability $p(c)$ at all discrete color values c allows us to pick the most likely color c_z^*. We assume the probability of the most likely color, $p(c_z^*|c_{z_i}) =: \bar{p}_z$, to be the probability of the depth z itself. Next, we normalize that distribution over the range of all considered depths z, which yields the values p_z . This is equivalent to assuming that one of the depths must be the correct one. We determine the most likely depth z^*, and we compute the variance around that depth, $\sigma_d^2 = \sum_z p_z(z-z^*)^2$. This implies our confidence of the depth, $\kappa_0 = \sigma_d^{-2}$. We have found the variance to be a very informative measure of the confidence of a depth, since it considers the spatial relation of the different depths with respect to each other. A large p_z at a depth close to z^* will increase the variance far less than the same value at a vastly different depth.

Finally, we store the optimal depth values z^* as the depth estimate of pixel (x,y), κ_0 as its confidence and the optimal color value at depth $z^*(x,y)$, $c_{z^*}^*$, as the denoised image value, $\bar{c}$.

Figure 2 demonstrates the striking denoising effect of this procedure. Note specifically that faint patterns are being restored that are originally buried in noise.

3.2 Dense Depth Map

Next, we compute a dense depth map for every batch. That depth map is intended to fit the estimated sparse pixel depth values weighted by their confidence, and to be smooth in areas where the denoised image is smooth. The approach is similar to the one presented in [13]. We compute the dense depth map $u(x,y)$ by minimizing

$$\sum_{x,y} \kappa_0(x,y)(u(x,y)-z^*(x,y))^2 + \mu_D \int_\Omega (g(x,y)\cdot\nabla u(x,y))^2 ds, \tag{4}$$

with the regularization weight g derived from the denoised image values $\bar{c}$

$$g(x,y)_i = \exp(-\frac{1}{\lambda^2}\sum_{k\in r,g,b}(\frac{\partial}{\partial i}(\bar{c}_k(x,y)))^2), \quad i\in\{x,y\}, \tag{5}$$

where λ describes the sensitivity to image structure, and μ_D the amount of regularization. Analogously to the depth map $u(x,y)$, we also compute the confidence map $\kappa(x,y)$, by replacing $z^*(x,y)$ in equation 4 by the confidence itself, $\kappa_0(x,y)$.

The optimization is performed using a multiscale Jacobi method. This procedure generates a depth map with discontinuities that coincide with those in the denoised color value. It also carries a strong fronto-planarity bias, and thin protruding features are often shifted towards the background. It does, however, tell us the locations of the depth discontinuities in the image.

3.3 Contour Detection

This is the final operation on one single batch of images. We seek to detect a set of image points that are likely contour candidates, and we wish to estimate their momentary depth. Later, we will use this contour candidates to fit local quadrics.

To detect the contours, we apply the Canny edge detector [1] to the dense depth map u, which provides us with a set of potential contour pixels (x_c,y_c) of maximal local curvature. For every such contour candidate, we estimate its momentary depth from the initial sparse depth samples, and not from the dense depth map. This way, we prevent the planarity bias of the depth map to enter into our quadric estimates. Also, it helps to preserve small features that are pulled towards the background by the depth map regularization. We estimate the momentary depth of every contour as,

$$d(x_c,y_c) = \frac{1}{w_t(x_c,y_c)}\sum_{x,y\in N(x_c,y_c)} w(x,y)z^*(x,y) \tag{6}$$

$$w_t(x_c,y_c) = \sum_{x,y\in N(x_c,y_c)} w(x,y) \tag{7}$$

$$w(x,y) = \kappa_0(x,y)\exp(-\frac{1}{\sigma_c^2}((x-x_c)^2+(y-y_c)^2))\ \mathrm{H}(u(x_c,y_c)-G_{\sigma_c}*u(x,y)), \tag{8}$$

where $N(x,y)$ is a small neighborhood of (x,y), H is the Heaviside function and $G_{\sigma_c}*$ is a convolution with a Gaussian. The first part of eq. 8 weights the pixels by their distance to

the center pixel, while the second part selects pixels that are closer to the observer than the local area mean of the depth map. This selects the foreground at each contour. The estimated momentary depth d_c then implies the position of the contour, to which we will refer as p_c.

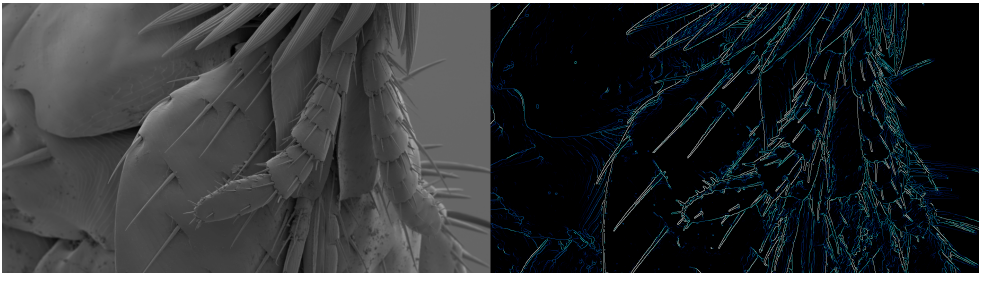

Figure 3: Left: a denoised input image; right: contour candidates and their confidences.

We also compute the w-weighted variance of $z^*(x,y)$ at (x_c,y_c), $v(x_c,y_c)$, and from it the total confidence c_c as $|\nabla u|(1-\exp(-w_t^2/\sigma_w))\exp(-v^2/\sigma_v)$. This excludes candidates that are either insufficiently supported by evidence (w_t is small) or contradicting too much evidence (v is large).

For every contour candidate, we finally determine the surface normal at that point. We define the image space line l as $(\nabla u(x_c,y_c)^T, -(x_c,y_c)\nabla u(x_c,y_c))$ that expresses the position and direction of the contour. It is equivalent to a plane in space $\pi = lP_i$. The normal of that plane is our surface normal, n_c.

3.4 Local Quadrics

The next operation takes place in a discretized 3D volume. There, we aim to estimate local quadrics from the contour candidates of all images. An arbitrary quadric can be written as the level set $f(x) = x^T Cx = 0$, where $x = (x_1,x_2,x_3,1)^T$ is a homogeneous position vector and C is a real, symmetrical 4×4 matrix that holds the coefficients of the quadric.

We are now looking for a matrix-valued field $C(x,y,z)$ that varies smoothly across space, and that corresponds to quadrics that are consistent with the detected contour candidates. To that end, we look at points in space on a discrete grid, and for every such point p_G, we estimate a quadric C, relying on information provided by nearby edge candidates.

We have found that much more reliable results can be obtained by estimating the gradient of $f(x)$ first, and then the integration constant. Estimation of the full quadric at once can lead to unwanted folding-over effects if the contours are misaligned. The gradient of $f(x)$ is parallel to the surface normal of the quadric and it is given by $\nabla f(x) = 2C_3 x$. C_3 refers to the first three rows of C.

To estimate the gradient of the quadric, we minimize the following expression over the contours in a neighborhood N of grid point p_G,

$$\sum_{p_c\in N(p_G)} w_c(2C_3\begin{pmatrix}p_c\\1\end{pmatrix} - n_c)^2, \qquad w_c = c_c\exp(-\frac{1}{\sigma_q^2}|p_c-p_G|^2) \tag{9}$$

This yields the gradient of a quadric that conforms to the contour normals n_c at the locations of the contours p_c, weighted by their confidence c_c and their distance to p_G. This computation is performed in closed form. We also remember the sum of all contour weights w_c, to which we will refer as w_q, the weight of the quadric.

Since C is symmetrical, we are now only missing one last coefficient, $c_{4,4}$. We compute it as the median of $-p_c^T C_0 p_c$ over all contours p_c in $N(p_G)$; C_0 is equal to C with its unknown last coefficient set to zero. We will refer to the quadric estimated at grid point p_G as C_G.

3.5 Watertight Surface

Finally, we estimate a watertight surface that conforms to the quadrics in the areas surrounding the detected contours and to the dense depth maps elsewhere. This is done by solving a Poisson problem using specific regional terms. We find an optimal 3D scalar field f in volume V by minimizing

$$\int_V \mu_q c_q(x)(f(x)-q(x))^2 + \mu_\phi c_\phi(x)(f(x)-\phi(x))^2 + \mu_r |\nabla f(x)|^2 dx, \tag{10}$$

where $q(x)$ and $\phi(x)$ are the quadric and the flat regional terms, $c_q(x)$ and $c_\phi(x)$ are their confidences, and μ_q and μ_ϕ are the respective weights, while μ_r controls the amount of regularization.

The quadric regional term $q(x)$ and its confidence $c_q(x)$ are computed using the following quadric voting scheme,

$$\psi(x) = \sum_{p_G \in N(x)} w_q \exp(-\frac{1}{\sigma_q^2}|x-p_G|^2)\,\mathrm{sign}(x^T C_G x), \tag{11}$$

$$q(x) = \mathrm{sign}(\psi(x)), \qquad c_q(x) = |\psi(x)|. \tag{12}$$

This can be understood as follows. For every point x, every surrounding grid point p_G casts a vote on whether x belongs inside or outside the quadric C_G. Those votes are weighted based on the distance between x and p_G and proportionally to the weight of the quadric, w_q.

The flat regional term $\phi(x)$ is computed from the dense depth maps $u(x,y)$ and the confidence maps $\kappa(x,y)$. For every depth map, we compute a flatness term $h(x,y)$ from the contour confidence c_c as $h(x,y) = \exp(-(G_{\sigma_h} * c_c)^2(x,y)/\eta_h^2)$, where the threshold η_h is set very low. This term selects from the depth maps those areas that are sufficiently far away from any detected contours.

For every 3D point x and image i, let $\theta_i(x)$ be the depth difference between x and the depth map, with a positive sign if x is located behind the surface. We define the flat regional term and its confidence as follows,

$$\chi(x) = \sum_i \begin{cases} \gamma h(P_i(x))\kappa(P_i(x))\exp(-\theta_i(x)^2/\tau^2), & \text{if } \theta_i(x) > 0 \\ h(P_i(x))\kappa(P_i(x))(1-\exp(-\theta_i(x)^2/\rho^2)), & \text{otherwise} \end{cases} \tag{13}$$

$$\phi(x) = \mathrm{sign}(\chi(x)), \qquad c_\phi(x) = |\chi(x)|. \tag{14}$$

If x is located behind the depth map, then we assume it to be inside the material with a certainty that decays as the distance increases. If x is located in front of the depth map, then the certainty of being outside increases with the distance. The parameter τ describes the expected thickness of the material, while ρ describes the uncertainty of the depth map. The parameter γ is intended to increase the weight of the inside region to counter the minimal surface bias introduced by unreliable depth maps. In our experiments, γ was set to 10.

Given those regional terms, we compute the optimal f using the Jacobi method. Finally, we extract a watertight mesh from f by applying the marching cubes algorithm [12].

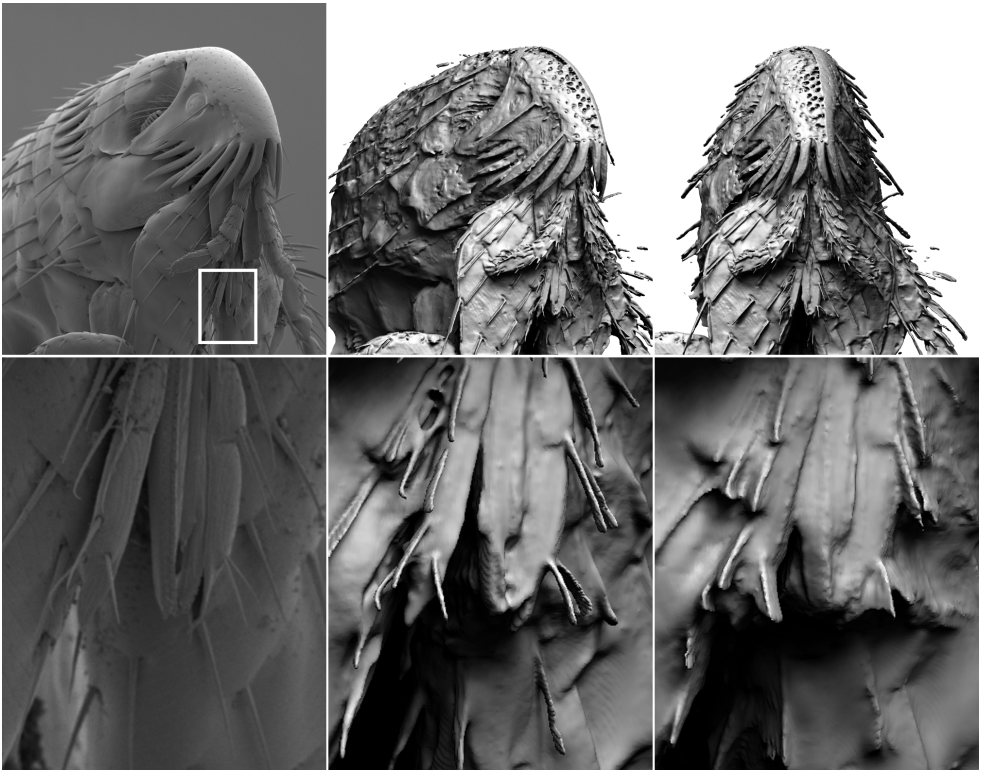

Figure 4: Reconstruction of a cat flea (ctenocephalides felis) from 1400 scanning electron microscope images over an angle of 70°. Top, from left to right: one of the input images; two views of the full reconstruction. Bottom: enlarged area of the input image; a reconstruction obtained through our full quadric based method; a reconstruction based only on our depth maps.

4 Experiments

We have applied our method to a set of 1400 grayscale images of a cat flea (ctenocephalides felis) captured by a scanning electron microscope using secondary electron imaging. In such images, the pseudo-illumination is always static with respect to the viewer. During the sequence, the camera moves along a 70° arc, so the images are spaced roughly 0.05° apart, and the movement is very jittery. The standard deviation of the image noise ranges from 1.1% of the value range for black pixels to 7.3% for white pixels. We have split the sequence into batches of 50 images with 25 images between two batches.

In addition to our full method, we have also reconstructed a shape using only the flat regional term ϕ, for which the flatness term h was set to one everywhere. That way, the entire shape is reconstructed from the depth maps only. The results of both the full method and the purely depth map based method are shown in fig. 4.

5 Conclusion

We have presented a method that is able to reconstruct very complex organic surfaces from images. The method fits local quadrics to detected internal and external contours, and it can deal with changing illumination. We have shown that our method is able to reconstruct very thin features that would be corrupted by a purely depth map based reconstruction.

At the same time, we have observed that our method tends to hallucinate contours in flat areas that are never seen tangentially. Those false contours are already visible in the dense depth maps, so we assume that a more advanced form of depth map regularization might alleviate the problem. More abstractly, it would be necessary to consider the observed shading patterns as an additional constraint on the allowed amount of surface curvature. This would form a bridge into the field of shape-from-shading.

Further, we have shown a way to detect and approximate surface features that are as small as one pixel in size and that are delineated by edges fainter than the noise level of the image. Although we assume quadrics as our local surface model, other models could also be fitted to our depth values.

References

[1] J Canny. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, 8(6):679–698, 1986.

[2] Roberto Cipolla, Bjorn Stenger, Arasanathan Thayananthan, and Philip HS Torr. Hand tracking using a quadric surface model and bayesian filtering. In *Mathematics of Surfaces*, pages 129–141. Springer, 2003.

[3] Geoffrey Cross and Andrew Zisserman. Quadric reconstruction from dual-space geometry. In *Computer Vision, 1998. Sixth International Conference on*, pages 25–31. IEEE, 1998.

[4] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[5] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. In *Computer Vision–ECCV 2006*, pages 564–577. Springer, 2006.

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[7] Pau Gargallo, Emmanuel Prados, and Peter Sturm. Minimizing the reprojection error in surface reconstruction from images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[8] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus H Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73, 2013.

[9] Kalin Kolev, Thomas Brox, and Daniel Cremers. Robust variational segmentation of 3d objects from multiple views. In *Pattern Recognition*, pages 688–697. Springer, 2006.

[10] Kalin Kolev, Thomas Brox, and Daniel Cremers. Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):493–505, 2012.

[11] Shubao Liu and David B Cooper. A complete statistical inverse ray tracing approach to multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 913–920. IEEE, 2011.

[12] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

[13] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz. Occluding contours for multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 4002–4009. IEEE, 2014.

[14] Son Tran and Larry Davis. 3d surface reconstruction using graph cuts with surface constraints. In *Computer Vision–ECCV 2006*, pages 219–231. Springer, 2006.

[15] George Vogiatzis, Carlos Hernández, Philip HS Torr, and Roberto Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2241–2246, 2007.

[16] Hui Xie, Jianning Wang, Jing Hua, Hong Qin, and Arie Kaufman. Piecewise c1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 13. IEEE Computer Society, 2003.