

# Robust Recovery of Shapes with Unknown Topology from the Dual Space

Chen Liang, *Student Member, IEEE*, and Kwan-Yee K. Wong, *Member, IEEE*,

## Abstract

In this paper, we address the problem of reconstructing object surface from silhouettes. Previous works by other authors have shown that, based on the principle of duality, surface points can be recovered, theoretically, as the dual to the tangent plane space of the object. In practice, however, the identification of tangent basis in the tangent plane space is not trivial given a set of discretely sampled data. This problem is further complicated by the existence of bi-tangents to the object surface. The key contribution of this paper is the introduction of epipolar parameterization in identifying a well-defined local tangent basis. This extends the applicability of existing dual space reconstruction methods to handle fairly complicated shape without making any explicit assumption on the object topology. We verify our approach with both synthetic and real-world data, and compare it both qualitatively and quantitatively with other popular reconstruction algorithms. Experiment results demonstrate that our proposed approach produces more accurate estimation, whilst maintains reasonable robustness towards shapes with complex topologies.

## Index Terms

Reconstruction, duality principle, tangent envelope, epipolar parameterization, surface extraction

## I. INTRODUCTION

**S**ILHOUETTES are the profiles of objects in the images. They offer rich information about the shape of the objects, and can often be extracted reliably. Reconstruction algorithms based on silhouette information are often capable of producing relatively complete 3D models.

Each silhouette defines a family of planes tangent to the object surface. Granted enough viewpoints with good coverage around an object, the families of tangent planes from all the images form a tangent

plane space, which is a dual representation of the object surface. Previous works such as [1] and [2] have demonstrated theoretically how points on the primal surface can be recovered from such a dual space sampled from the silhouettes. Due to the existence of singularities and irregularities in the dual space of complicated shapes, however, the major difficulty involved in such a dual space reconstruction method is how to obtain a reliable tangent basis in the dual space.

In this paper, we contribute to the literature by utilizing the *epipolar parameterization* in identifying reliable tangent basis on the dual surface. It allows us to overcome the aforementioned difficulties and arrive at a closed-form solution without the need of searching in the dual space. We focus on more complicated and generic shapes which challenge previous dual space approaches. We also analyze in detail common extreme cases of shapes, such as creases and self-occlusions, and the ways to handle them. As shown in the experiments, our method is generic and practical for robustly reconstructing rather complicated objects.

This paper is organized as follows. In Section II, previous related works are assorted and reviewed. In Section III, we briefly cover the background of the dual space theory and epipolar parameterization. In particular, we show the reason for introducing the latter, and how in theory it strengthens the current dual space approach. Section IV deals with the recovery of contour generators in practice. Extreme cases encountered in a variety of shapes are analyzed in detail. In Section V, we introduce an algorithm to extract a surface from the recovered contour generators, which also preserves the topology observed from the silhouettes. Results of both qualitative and quantitative experiments are shown in Section VI, as well as comparisons with existing approaches for reconstruction from silhouettes. Finally, Section VII concludes this paper with discussions.

## II. PREVIOUS WORKS

Volumetric representations are simple yet robust, and have been extensively used in reconstruction from silhouettes. The concept of volumetric description was first seen in [3], where Martin and Aggarwal rasterized the space into parallelogram structure. This was later developed into representing the object by volumetric intersection of the viewing cones. OcTrees (Oct-Trees) [4] is one of the most popular representations utilized for this procedure. In [5], Potmesil reconstructed OcTrees from arbitrary viewpoints

under perspective projection. Subsequent variations of the OcTree [6], [7], [8] allow faster yet more accurate approximation to the object surface. The major advantage of volumetric approaches is their robustness, especially when no assumption is made on the topology of the object. However, a common problem shared by these volumetric methods concerns with their accuracy. The trade off for achieving a better reconstruction precision is to decrease the size of spatial partition, resulting in a tremendous increment in complexity of the reconstructed 3D models.

Instead of using volume, the intersection of the viewing cones can also be implemented as polygonal intersection, which results in one or more polyhedra formally known as the visual hull [9]. An early attempt of direct polygonal intersection was made in [10]. In [11], the polygonal visual hull was used to initialize a triangular spline surface which was subjected to further refinement. However, direct intersection of generalized viewing cones is neither efficient nor numerically stable. In [12], Lazebnik et al. derived a topological description of the contour generators for weakly calibrated cameras. This description has facilitated the computation of the visual hull polyhedron. Unfortunately, such an approach does not extend to object with non-zero genus. This method was later extended in [13] to more complicated objects. Matusik et al. [14] proposed an efficient algorithm for computing the polyhedral visual hull directly from silhouettes. Their algorithm is capable of generating visual hull mesh in real time in the case of a few cameras. As the number of camera increases, however, their method may still suffer from numerical instability. The authors of [15] and [16] turned to sampling visual rays instead of polygons from silhouettes. These visual rays were pruned by the viewing cones to form viewing edges, which were subsequently joined together to form the visual hull mesh. In a separate attempt, Cheung et al. [17] adopted color information to pinpoint the tangent points to the surface along each viewing edge. Most of the aforementioned approaches, which approximate the visual hull with polyhedrons, outperform the volumetric approaches in terms of accuracy and computation complexity, yet many of them maintain comparable robustness regarding objects with complex topologies.

As for objects with smooth surface, a more precise estimation of surface points can be achieved by a differential analysis on the deformation of the apparent contours. In [18], Cipolla and Blake demonstrated that the curvature and depth of the contour generator points can be recovered, provided that the camera motion is known. Vaillant and Faugeras [19] showed that the contour generator corresponding to a view

can be computed from a triplet of views. Later, Boyer and Berger [20] gave a closed-form solution for the depth along a visual ray by fitting an osculating quadratic to its two epipolar correspondences. With sufficient (and well-distributed) viewpoints, these approaches, which estimate the differential structure, can give high quality estimation of the surface points, especially for smooth objects. However, few of them share similar robustness as the volumetric approaches regarding objects with complicated topologies.

Some early attempts to recover the shape of an object from its tangent envelope can be found in [21] and [22]. Both algorithms are limited to orthogonal projection only. In [1], the authors introduced the concept of recovering object surface from its dual space. This work has led to a new insight into the duality theory and its applicability to the reconstruction problem. The proposed approach fits algebraic surface patch to a local tangent space resulting from space partitioning. These fitted patches are then resampled for higher order surface fitting. Such an approach requires rather careful tuning of several inter-related parameters, which becomes increasingly difficult as the complexity of the object surface increases. Recently, Brand et al. [2] generalized the dual space reconstruction problem into tangent estimation on the dual surface. Under this framework, a point on the object surface could be independently estimated from its dual space neighbors linearly. However, the ambiguity of matching points between successive contour generators still poses an issue for tangent basis estimation, hence their method may still be affected by singularities and ill distribution of the dual space points.

In this paper, we extend the dual theory for surface reconstruction and present a complete and practical framework for recovering object surface from silhouettes. Particularly, our framework aims at robustly recovering more complex surfaces without explicit knowledge of their topologies.

### III. THEORETICAL FRAMEWORK

We denote a scalar value using italic font  $r$ , a vector or vector-valued function using bold font as  $\mathbf{r}$  or  $\mathbf{r}(s, t)$ , respectively. The homogeneous form of a vector is denoted as  $\tilde{\mathbf{r}}$ . The derivative of a function is denoted by subscription  $\mathbf{r}_s(s, t)$ . The dual counterpart (defined later) of a vector or vector-valued function is denoted by asterisk  $\mathbf{r}^*$ .

### A. Existing Dual Space Theory and Possible Problems

The dual theory introduced in [1] and [2] tackles the problem of reconstructing the original surface  $\mathbf{r}$  from the tangent plane space sampled from silhouettes with known projection matrices  $\mathbf{P}$ . An example illustrating the sampling of tangent planes from silhouettes is given in Fig. 1. Each silhouette defines a family of tangent planes, and the locus of the tangent points corresponds to the contour generator for that particular viewpoint. A collection of viewpoints gives rise to a space of tangent planes in  $\mathbb{R}^3$ , of which the envelope is the original object surface. Based on the principle of duality, these tangent planes are equivalent to points on the dual surface  $\mathbf{r}^*$ .

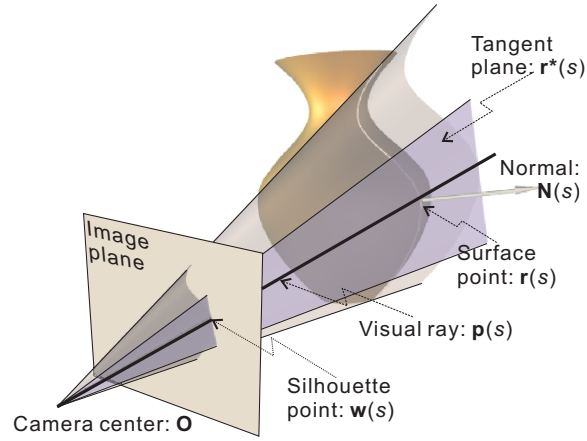


Fig. 1. The back-projection of a tangent line on the silhouette is a plane  $\mathbf{r}^*$  passing through the camera center  $\mathbf{O}$  and tangents to the object surface. The locus of the tangent points is the contour generator for viewpoint  $\mathbf{O}$

In [2], the authors showed that the unknown surface  $\mathbf{r}$  can be recovered by estimating the tangent planes on the dual surface  $\mathbf{r}^*$ , which is formally put as:

$$\tilde{\mathbf{r}}(s, t) \propto [\tilde{\mathbf{r}}_s^*(s, t), \tilde{\mathbf{r}}_t^*(s, t), \tilde{\mathbf{r}}^*(s, t)]^\perp. \quad (1)$$

where  $\tilde{\mathbf{r}}(s, t)$  is a point on the original surface;  $\tilde{\mathbf{r}}^*(s, t)$  is the tangent plane at  $\tilde{\mathbf{r}}(s, t)$ , which is also a point on the dual surface;  $\tilde{\mathbf{r}}_s^*(s, t)$  and  $\tilde{\mathbf{r}}_t^*(s, t)$  form a tangent basis on the dual surface at  $\tilde{\mathbf{r}}^*(s, t)$ . For the mathematical proof of this theory, please refer to [2] and [23]. Note that this theory is based on assuming  $\mathbf{r}$  and  $\mathbf{r}^*$  to be regular. The non-regular cases involving singularities and discontinuities will be discussed later in Section IV of this paper.

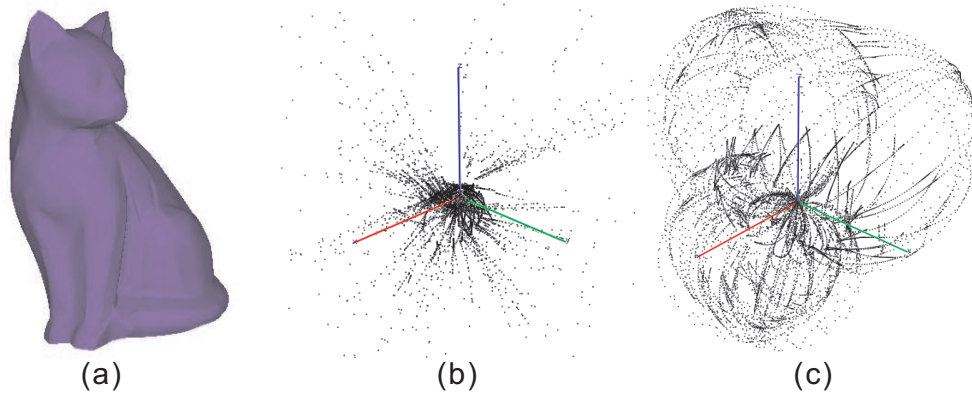


Fig. 2. Surface and its dual; (a) Original object; (b) Points on the dual surface; (c) Pedal surface [24]; The discretely sampled tangent planes correspond to an unorganized point cloud in the dual space and it is nearly impossible to directly recover a local parametric form of the dual surface.

The key challenge of applying this theory in practice is how to reliably obtain the tangent basis vectors  $\tilde{\mathbf{r}}_s^*(s, t)$  and  $\tilde{\mathbf{r}}_t^*(s, t)$ . This originates from the fact that the dual surface is sampled from a discrete set of viewpoints, and no well-defined local parametric form is available. Among the two tangent basis vectors to be estimated,  $\tilde{\mathbf{r}}_s^*(s, t)$  is, as also shown in [2], intuitively computed from successively sampled tangent planes along the same silhouette. Unfortunately, it is not trivial to compute  $\tilde{\mathbf{r}}_t^*(s, t)$  because, unlike  $\tilde{\mathbf{r}}_s^*(s, t)$ , the point-wise connectivity across contour generators of different views is not as apparent.

Spatial proximity in the dual space was exploited in [2] to qualify neighbor points required for estimating  $\tilde{\mathbf{r}}_t^*(s, t)$ . They sought nearby dual space points among those sampled from successive views to reduce the risk of picking bad neighbors. This is because points sharing the same tangent plane (bi-tangent points) are mapped to a single point in the dual space, and the dual surface crosses itself at these singularities. Another practical reason for distance measure in the dual space being unreliable is the evil distribution of the dual space points, despite the fact that they originate from tangents uniformly sampled from silhouettes (see the example given in Fig. 2). The distribution can also be perturbed by changing a few user-specified parameters, such as the chosen origin of the original surface, the sampling density, etc. Therefore, the reliance on spatial proximity in the dual space makes the dual space approach vulnerable to singularities and evil-distribution of the dual space points, preventing it from dealing with more complicated shapes in practice.

On the other hand, if we consider the regular case of a locally smooth parametric surface, the mapping

from the original space to the dual space is simply the tangent operation. This implies we can infer the neighborhood information in the dual space from that of the original space. Although we have no prior knowledge about the actual 3D structure of the original surface, it is still possible to define a local parameterization of the original surface. Epipolar parameterization is such a tool tailored for this purpose.

### B. Epipolar Parameterization

Epipolar parameterization is naturally defined for an object observed by a moving camera. It is a local parameterization and requires no prior knowledge of the 3D structure of the object surface. Let us consider the same setting as that defined in Section III-A. As the camera moves, the contour generator sweeps over the visible surface of the object. Assume the object surface is  $\mathcal{C}^2$  continuous, the object surface can be locally represented as:

$$\mathbf{r}(s, t) = \mathbf{O}(t) + \lambda(s, t)\mathbf{p}(s, t) \quad (2)$$

While the geometric interpretation of the spatial curve  $\tilde{\mathbf{r}}(s, t_0)$  is the contour generator curve observed at time  $t_0$ , the interpretation of the spatial curve  $\tilde{\mathbf{r}}(s_0, t)$  is not uniquely defined. Epipolar parameterization [18] gives a natural definition to the spatial curve  $\tilde{\mathbf{r}}(s_0, t)$ , by defining the tangent vector of this curve to be in the direction of the visual ray:

$$\mathbf{r}_t(s, t) \times \mathbf{p}(s, t) = \mathbf{0}. \quad (3)$$

This definition is particularly useful when  $t$  takes discrete values, i.e., there is only a finite number of cameras/viewpoints. For a point lying on a contour generator, its neighboring points along the  $t$ -parameter are defined as the “epipolar correspondences” of this point (for the sake of simplicity, we abuse the term “epipolar correspondences” by referring the correspondent point of a contour generator point found on the neighboring contour generator as in “epipolar correspondence”, whereas their projections on the images are the actual epipolar correspondence to each other). They are given by the intersection between the neighboring contour generators and the epipolar planes. An example is given in Fig. 3.  $\pi_{t+}$  is the epipolar plane defined by  $\mathbf{O}(t_0)$ ,  $\mathbf{O}(t_0 + \Delta t)$  and the point  $\mathbf{r}(s_0, t_0)$ . The epipolar correspondence  $\mathbf{r}(s_0, t_0 + \Delta t)$  is the intersection between the plane  $\pi_{t+}$  and the contour generator  $\mathbf{r}(s, t_0 + \Delta t)$ , i.e., they satisfy:

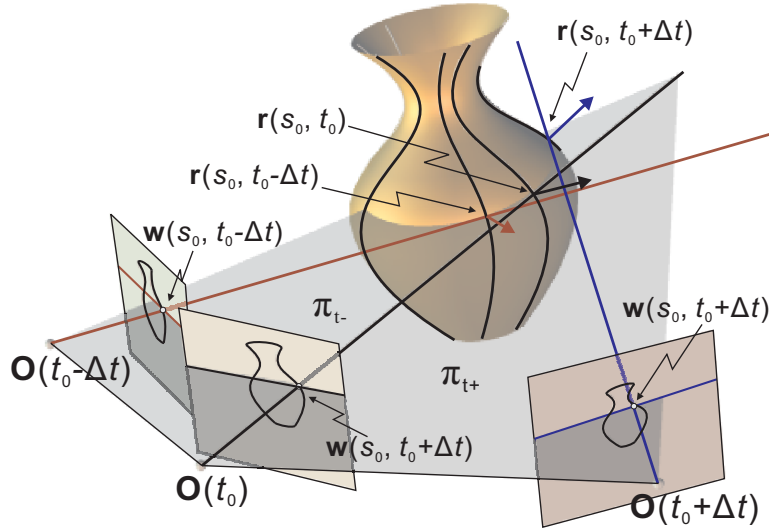


Fig. 3. Epipolar correspondences for a point on the image planes of the neighboring cameras.

$$\mathbf{p}(s_0, t_0) \times \mathbf{p}(s_0, t_0 + \delta t) \cdot (\mathbf{O}(t_0 + \Delta t) - \mathbf{O}(t_0)) = 0, \quad (4)$$

where

$$\begin{aligned} \mathbf{p}(s_0, t_0) &\propto \mathbf{r}(s_0, t_0) - \mathbf{O}(t_0), \text{ and} \\ \mathbf{p}(s_0, t_0 + \Delta t) &\propto \mathbf{r}(s_0, t_0 + \Delta t) - \mathbf{O}(t_0 + \Delta t). \end{aligned}$$

The epipolar correspondences can be identified purely on the image planes without knowing the 3D structure of the original surface. We do this by locating the projections of the matching points as intersections between the silhouettes and the epipolar lines. In above example, the projection of  $\mathbf{r}(s_0, t_0 + \Delta t)$  on the silhouette  $\mathbf{w}(s_0, t_0 + \Delta t)$  is the intersection between the epipolar line and the silhouette. Similar argument applies to  $\pi_{t-}$ ,  $\mathbf{r}(s_0, t_0 - \Delta t)$  and  $\mathbf{w}(s_0, t_0 - \Delta t)$ .

#### IV. RECOVERY OF THE SURFACE POINTS

We extract the silhouettes from images using active B-splines [18]. Each silhouette consists of one or multiple closed cubic B-Splines:

$$\mathbf{w}(s) = \sum_i B_i(s) \mathbf{x}_i, \quad (5)$$



where  $B_i(s)$  are the cubic B-spline basis functions and  $\mathbf{x}_i$  are the control points. The tangent lines along each B-spline can be easily computed as:

$$\tilde{\mathbf{I}}(s) = \left( \left( \sum_i \mathbf{T}_i(s) \mathbf{x}_i \right)^\top, - \sum_i \mathbf{T}_i(s) \mathbf{x}_i \bullet \sum_i B_i(s) \mathbf{x}_i \right)^\top, \quad (6)$$

where

$$\mathbf{T}_i(s) = \frac{\partial B_i(s)}{\partial s} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

There are several advantages of using B-spline representation as well. Most importantly, it grants a closed-form solution for intersecting epipolar lines with silhouettes. For the cubic B-Spline we used, this is equivalent to solving an order 3 polynomial equation. In this way, we do not need to sample every silhouette before estimating surface points, but rather, each surface point is estimated independently. This has also decoupled the accuracy of estimation from the pre-defined sampling density, and simultaneously guaranteed the estimation accuracy for whatever sampling density set. Last but not least, we can achieve sub-pixel accuracy with B-spline representation, which further boosts the estimation accuracy.

#### A. Identify Tangent Basis via Epipolar Parameterization

In Section III-A, we mentioned that surface point estimation can be converted to tangent basis estimation in the dual space. The key challenge involved is the estimation of  $\tilde{\mathbf{r}}_t^*(s, t)$ , and this is where the epipolar parameterization serves its purpose.

Let  $\mathbf{r}(s, t)$  be the point we want to estimate. The projection of this point on the silhouette is  $\mathbf{w}(s, t)$ . The tangent line at  $\mathbf{w}(s, t)$  back-projects to  $\tilde{\mathbf{r}}^*(s, t)$ . To obtain the neighbors of  $\tilde{\mathbf{r}}^*(s, t)$ , we firstly need to locate their duals in the original space, namely, the neighbors of  $\tilde{\mathbf{r}}(s, t)$ . At this stage, we have no knowledge about the depth of  $\tilde{\mathbf{r}}(s, t)$  and its neighbors, however, by using epipolar parameterization (see Section III-B), we can identify the projections of these neighbors on the silhouettes in the neighboring images. Along the  $s$ -parameter curve, the projections of  $\tilde{\mathbf{r}}(s \pm \Delta s, t)$  are simply  $\mathbf{w}(s \pm \Delta s, t)$ . Along the  $t$ -parameter curve, the projections of  $\tilde{\mathbf{r}}(s, t \pm \Delta t)$  are found along the silhouettes on the neighboring images. They can be computed as the intersections between the epipolar lines of  $\mathbf{w}(s \pm \Delta s, t)$  with the silhouettes at time  $t \pm \Delta t$ . Knowing  $\mathbf{w}(s \pm \Delta s, t)$  and  $\mathbf{w}(s, t \pm \Delta t)$ , we can back-project the tangent lines

at these points to obtain  $\tilde{\mathbf{r}}^*(s \pm \Delta s, t)$  and  $\tilde{\mathbf{r}}^*(s, t \pm \Delta t)$ , which can be used to approximate  $\tilde{\mathbf{r}}_s^*(s, t)$  and  $\tilde{\mathbf{r}}_t^*(s, t)$ , and finally  $\tilde{\mathbf{r}}(s, t)$  can be estimated using equation (1).

Note for the case when  $\tilde{\mathbf{r}}_s^*(s, t_0)$  vanishes, which is often characterized by inflections or line segments on the silhouettes, we can incorporate the direction of the visual ray at this point as an additional linear constraint. This is because the visual ray is conjugate to the contour generator [25]. Since similar constraint has also been exploited in [2], this will not be discussed in details here. In fact, this constraint is always incorporated in the actual implementation, as it imposes minimal additional computational cost upon our tangent estimation kernel, which is also linear in its nature.

The above epipolar parameterization based approach naturally extends to more complicated situations, such as when a silhouette consists of multiple curves or non-convex curves. These situations are typical for those objects with complex shape and topology. In these situations, there are usually multiple intersections between the epipolar line and the silhouettes. A few simple strategies can be adopted to pinpoint the actual epipolar match, including similarity of the normal between  $\tilde{\mathbf{r}}^*(s, t)$  and  $\tilde{\mathbf{r}}^*(s, t \pm \Delta t)$ <sup>1</sup>, and *ordering constraint* [26]. Once the correct epipolar match is secured, we can proceed the rest of the steps to obtain  $\tilde{\mathbf{r}}(s, t)$  in the same way as described earlier. In the worse case, no proper epipolar match can be found on the silhouette, and most of the time, this is due to self-occlusions. The way to handle such a case will be discussed in Section IV-B.4.

The merit of the above approach over direct search in the dual space is the geometric interpretation it offers for  $\tilde{\mathbf{r}}^*(s, t \pm \Delta t)$  - they correspond to surface points  $\tilde{\mathbf{r}}(s, t \pm \Delta t)$  in the original space that are actually close to  $\tilde{\mathbf{r}}(s, t)$ . With this guarantee, the choosing of  $\tilde{\mathbf{r}}^*(s, t \pm \Delta t)$  is robust to not only singularities in the dual space but also shape variation of consecutive silhouettes (such as change of topology), which is often observed in more complicated objects. We will see quantitatively in Section VI-A that the deficiency of directly searching in the dual space becomes quite evident as the complexity of the object to be recovered increases.

<sup>1</sup>The normal similarity constraint is also adopted in [2] in the process of estimating the tangent basis vectors.

## B. Extreme Cases

While epipolar parameterization gives a sufficiently good tangent basis most of time, the parameterization itself does have several known extreme cases, such as those studied in [27]. Although these cases usually affect only a small portion of the surface points to be estimated, we introduce some special techniques to handle them to further increase the robustness of our algorithm.

The extreme cases are most likely encountered when recovering complicated objects, include creases, frontier points, cusps, T-junctions and self-occlusions. Some of these cases, such as creases and frontier points can be handled by minor tweaking of our epipolar parameterization based algorithm, while others might need more sophisticated schemes or to incorporate additional information.

1) *Creases (sharp edges)*: A crease on the surface causes discontinuity of the local tangents (Fig. 4(top-left)). This case often happens when the observed object has sharp edges, such as polygonal objects. The silhouette curve produced by a crease over time is equivalent to the projection of a stationary spatial curve. One advantage of the epipolar parameterization is that it unifies the treatment of silhouette produced by rigid spatial curve and by smooth surface [28]. Intuitively, a point on the crease can be treated as having infinite curvature along the direction of the visual ray. The surface normal at this point is similarly chosen to be perpendicular to the direction of the visual ray as in the smooth surface case. In this way, the dual space points produced at different time corresponding to this point do not degenerate and the 3D position of this point can hence be estimated. Experiments of surface with creases can be found in Section VI-B.

2) *Frontier points*: The epipolar parameterization breaks down at frontier points [29] which leads to degenerate tangent basis. While the frontier points themselves can be recovered by a direct triangulation instead (Fig. 4(top-right)), the epipolar match may be ambiguous for the point in proximity of the frontier points. In practical, the projection of the frontier points on the silhouette can be located by locating tangent point of the epipolar line to the silhouette [30], and the ambiguity of the epipolar match can be resolved by applying the ordering constraint (with respect to the projection of frontier points). This is similar to what have been done in [31]. In the example given by Fig. 4, the bottom figures show a portion of the silhouettes at time  $t$  and  $t + \Delta t$ . At time  $t + \Delta t$ , the normal at  $\mathbf{w}_1$  and  $\mathbf{w}_2$  are very similar and they both qualify as the potential epipolar match for the point  $\mathbf{w}(s, t)$  (on silhouette at time  $t$ ) corresponding to the lower epipolar line. However, since we know which side of the frontier point that  $\mathbf{w}(s, t)$  is on, the

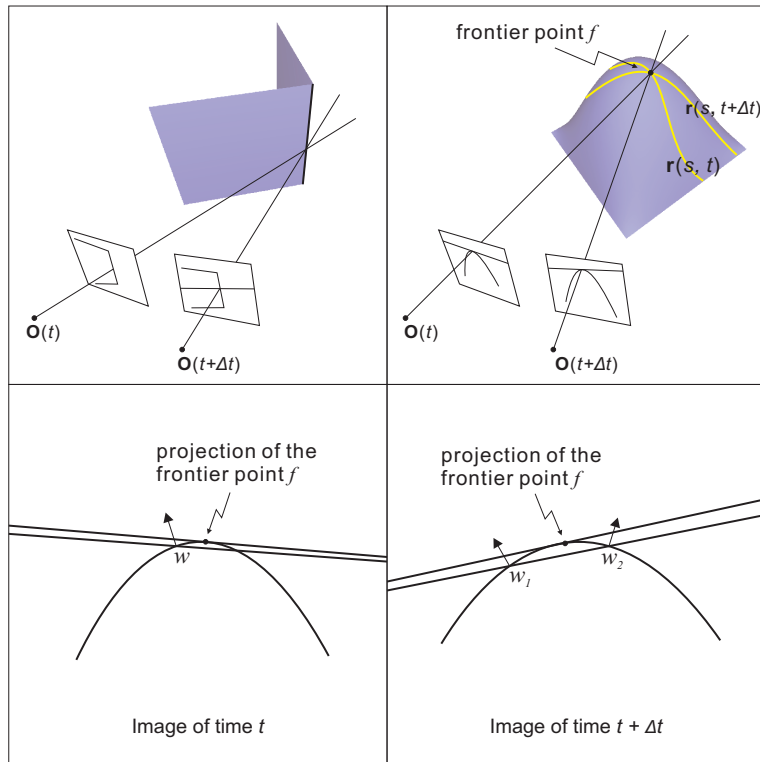


Fig. 4. (top-left) An example of a crease on object surface; (top-right) An example of frontier points; (bottom-left) At time  $t$ , a point  $w$  near the frontier points is to be matched to a point on silhouette of time  $t + \Delta t$  (bottom-right); The ambiguity of choosing either  $w_1$  or  $w_2$  as the correct epipolar match for  $w$  can be resolved by enforcing the ordering constraint.

correct epipolar match should be the one (either  $w_1$  or  $w_2$ ) on the same side of point  $f$ .

In actual implementation, this can be done easily with minimal assumption. As the camera is calibrated, the image of frontier points can be easily tracked by their curve parameter on the silhouettes (represented by closed B-splines). It is rather easy to maintain that the positive direction of the normals on B-spline (see Equation IV) is consistently facing outside (or inside). Assuming that the successive cameras have similar orientation, which is mostly the case for a sequence of images acquired by a moving camera, the order of points near frontier points should also be consistent with the parametric direction of the B-splines.

3) *Cusps and T-junctions*: Cusps and T-junctions are both characterized by abrupt ending of the apparent contours. A cusp occurs on the apparent contour when the viewing ray direction coincides with the tangent vector of the contour generator (Fig. 5(left)). Except for the occluded branch (dotted line in the figure), we usually have no problem to recover the surface points near the cusp in the regular case. T-junctions, on the other hand, are caused by occlusion (Fig. 5(middle)). The dotted line on the surface does not project onto the silhouette, so that their positions cannot be recovered. What is worse, the contour generator points

projecting onto the proximity of the T-junction (the thin line in Fig. 5(right)) are not always recoverable too. This is because their correct epipolar matches may dwell on the “invisible” part of the apparent contour or are completely occluded. To mitigate this problem, we can approximate their positions from nearby recovered contour generator points (the bold line in Fig. 5(right)). Take the Fig. 5(right) for example, we first computed the weighted average of the contour generator points corresponding to  $A$  and  $B$ . Then we pick the point along the visual ray of  $C$  which is closest to the weighted average as the approximated position we wanted.

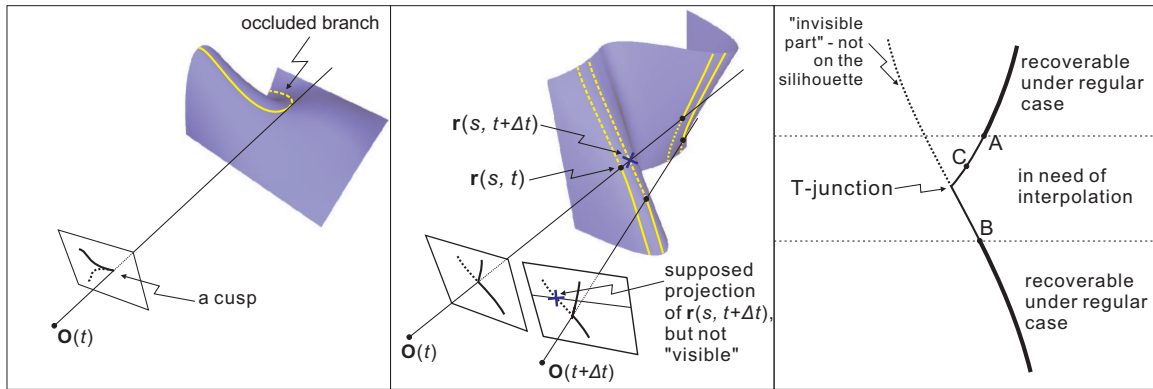


Fig. 5. (left) An example of a cusp; (middle) an example of a T-junction; (right) The proper epipolar match for points in proximity of a T-junction may not be visible on the silhouettes of neighboring cameras, and this leads to low quality or even wrong estimation.

4) *Self-occlusions*: Major occlusion can happen when the surface topology becomes more complicated. The configuration shown in Fig. 6 represents a typical situation. The image of a surface point  $A$  can be seen on the silhouette at time  $t$ . However, its true epipolar correspondence on the surface at time  $t + \Delta t$  is occluded, or lies on the internal contour [25] if the object is transparent. Hence we cannot find its correspondence on the silhouette at time  $t + \Delta t$ . In practice, we have to decide the true epipolar match among the intersections between epipolar lines and silhouettes. Since we have no assumption on the topology of the object observed, it is very difficult to tell whether the best choice among these intersections represents the true epipolar correspondence. In the example shown in Fig. 6, it may well be mistaken that the image of point  $B$  on the silhouette at  $t + \Delta t$  is the true epipolar correspondence.

Incorrect epipolar match leads to low quality estimation. In the worse case, the point will fly out of the true surface. To identify wrong epipolar correspondence caused by occlusions, we can consider the symmetry of epipolar matching, similar to the criterion adopted in classic stereo matching: if the true

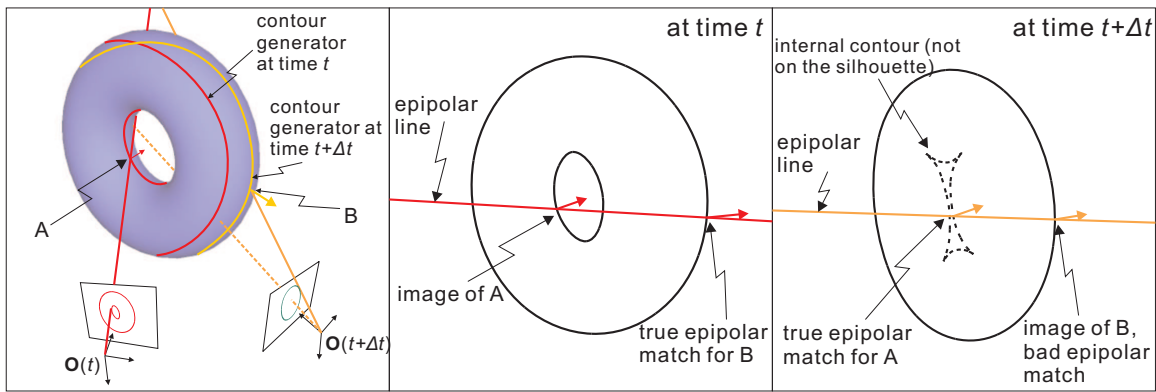


Fig. 6. Self-occlusion on the surface leads to incorrect epipolar correspondence. (left) surface point  $A$  and  $B$  lie on the same epipolar plane spanned by camera  $O(t)$  and  $O(t + \Delta t)$ ; (middle) the epipolar match for point  $B$  is visible on the silhouette at time  $t$ , which is different from the image of  $A$  - this information is later used when evaluating the symmetry of epipolar matching; (right) a potential good epipolar correspondence of  $A$  is occluded on the silhouette at time  $t + \Delta t$ , thus  $B$  may be mistakenly qualified as the epipolar match of  $A$  due to their similar normals. This type of wrong matching can be identified by considering symmetry of epipolar matching.

epipolar correspondence of a surface point  $A$  is  $B$ , then the converse is also true. We illustrate symmetry of epipolar matching also by the example in Fig. 6: suppose  $A$  is the surface point we want to estimate (on the contour generator of time  $t$ ), we follow the steps under the normal circumstance by projecting it into the epipolar line at time  $t + \Delta t$ , and choose the best suited epipolar match  $\tilde{\mathbf{r}}(s, t + \Delta t)$  ( $B$  in above example) using the criteria described earlier. As an additional step, we also intersect the epipolar line for  $\tilde{\mathbf{r}}(s, t + \Delta t)$  with the silhouette at time  $t$  and choose the best suited epipolar match. If the best epipolar match for  $B$  is  $A$  reciprocally,  $B$  can be qualified as the proper epipolar match; if it is not the case, just as in the above example, we can conclude that no proper epipolar match exists.

With insufficient information for determining the epipolar match at time  $t + \Delta t$ ,  $\tilde{\mathbf{r}}(s, t)$  cannot be estimated by the dual space approach directly. There are several possible fallbacks. One possible choice would be applying the method introduced in [15]: the position of  $\tilde{\mathbf{r}}(s, t)$  along its visual ray can be localized to a smaller range which is consistent with all silhouettes. In our implementation, we simply skipped the points failing the symmetric matching test. Doing this has little impact on the shape of the reconstructed model if there are abundant viewpoints with good observation coverage, which is usually the case if we tend to reconstruct more complicated objects.

### C. Summary

A brief outline of our algorithm, together with the proposed counteracts performed for extreme cases, is given in Algorithm 1.

---

#### Algorithm 1 Recovery of the Surface Points

---

```

1: Initialize point array  $\tilde{\mathbf{r}}(s, t)$ 
2: for  $i^{\text{th}}$  silhouette  $\mathbf{w}(s, t_i)$  do
3:   for  $j^{\text{th}}$  point  $\mathbf{w}(s_j, t_i)$  on the silhouette do
4:     Determine the epipolar match  $\mathbf{w}(s_j, t_{i\pm 1})$ 
5:     if the symmetric epipolar match for  $\mathbf{w}(s_j, t_{i\pm 1})$  is  $\mathbf{w}(s_j, t_i)$  then
6:       Back-project to get the tangent planes  $\tilde{\mathbf{r}}_k^*$  at and in the vicinity of  $\mathbf{w}(s_j, t_i)$  and  $\mathbf{w}(s_j, t_{i\pm 1})$ 
7:       Compute the weighted tangent ( $\tilde{\mathbf{r}}^{**}(s_j, t_i)$ ) of  $\tilde{\mathbf{r}}_k^*$  satisfying the visual ray ( $\mathbf{w}(s_j, t_i)$ ) constraint as well
8:       if  $\tilde{\mathbf{r}}^{**}(s_j, t_i)$  projects inside all silhouettes then
9:          $\tilde{\mathbf{r}}(s_j, t_i) \leftarrow \tilde{\mathbf{r}}^{**}(s_j, t_i)$ 
10:      else
11:         $\tilde{\mathbf{r}}(s_j, t_i) \leftarrow \text{*PENDING*}$ 
12:      end if
13:    else
14:       $\tilde{\mathbf{r}}(s_j, t_i) \leftarrow \text{*NULL*}$ 
15:    end if
16:  end for
17:  for each point  $\tilde{\mathbf{r}}(s_j, t_i)$  equals  $\text{*PENDING*}$  do
18:    Interpolate  $\tilde{\mathbf{r}}(s_j, t_i)$  using  $\tilde{\mathbf{r}}(s_{j-m}, t_i)$ ,  $\tilde{\mathbf{r}}(s_{j+n}, t_i)$ . and the visual ray constraint, where  $m$  and  $n$  are bounded minimum possible integers that both  $\tilde{\mathbf{r}}(s_{j-m}, t_i)$  and  $\tilde{\mathbf{r}}(s_{j+n}, t_i)$  have already been estimated
19:  end for
20: end for

```

---

## V. EXTRACTION OF THE SURFACE

The result of the earlier estimation is a set of points lying on the contour generators, and each point is coupled with the tangent plane at that point. This means that the surface normal at each point is known. Unfortunately, these points are not distributed uniformly. This is due to the fact that the sampling density along each contour generator can be set arbitrarily high, while the density across the contour generators

is limited by the number of distinct viewpoints. Direct triangulation, in this case, results in ill-formed triangles. On the other hand, unlike the case of unorganized point cloud, we do know the connectivity of points along the same contour generator, as well as the the spatial order of the contour generators. We therefore propose a simple but robust method for extracting the surface using the connectivity and ordering information available.

#### A. Slicing the contour generators

The proposed way for surface extraction is built on a slice based re-sampling method, and it targets at producing more evenly distributed mesh grids. The earlier computed contour generators are now re-sampled by parallel slicing planes. Each slicing plane contains the intersections between the contour generators and that plane. The normal of each resampled point is interpolated from the normals of the adjacent points along the same contour generator. For a better visual effect, these slices are made parallel to the plane maximally spanning all the camera centers.

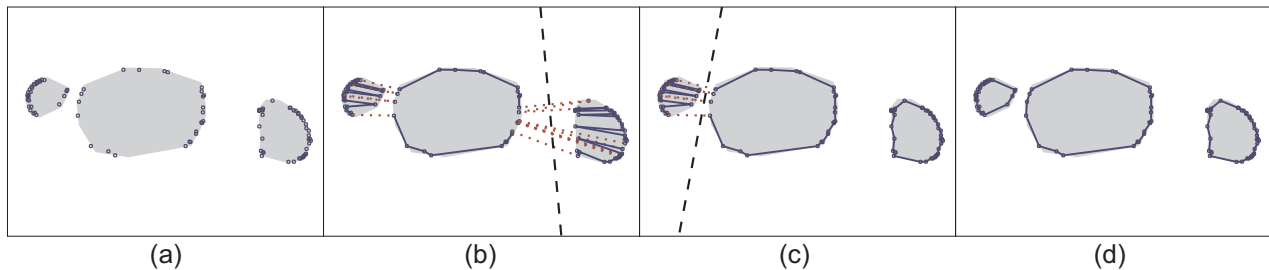


Fig. 7. Forming topologically correct 2D polygons; (a) resampled points on a slice; the shaded area is the cross section of the visual hull on the slice, which is unknown; (b) initial single polygon; the red dotted edges are those violating the projection test; the black dash line represents the first decision plane to separate the points into two groups, each forms a sub-polygon; (c) the sub-polygon on the left hand side, which still contains edges violating the projection test, is further separated into two sub-polygons; (d) the final result after topology correction.

Once the re-sampled points on each slice are obtained, we proceed to connect these points and form topologically correct 2D polygons. To do this, these points are linked together according to their spatial order. Of course, for complex shapes, a single polygon may not reflect the real topology of the original object. We correct the topology of this single polygon by separating it into smaller polygons (see the example given in Fig. 7). The first step involved is to conduct a projection test against all silhouettes for every polygonal edge. An edge violating the projectiong test would partially project outside at least one



silhouette, say the  $i^{th}$  silhouette. We then choose an arbitrary point on this edge which projects outside silhouette  $i$ , and link it with the corresponding camera center  $\mathbf{O}_i$  to create a separation plane. The plane divides the points of the initial single polygon into two groups, each resembles a new sub-polygon which better complies with the true topology suggested by the silhouettes. This process is repeated until no more edge violates the projection test. The final result is a set of sub-polygons that fully conform to the true topology observed.

We perform the topology correction for every slice. An example of slices with their topology corrected is given in Fig. 8.

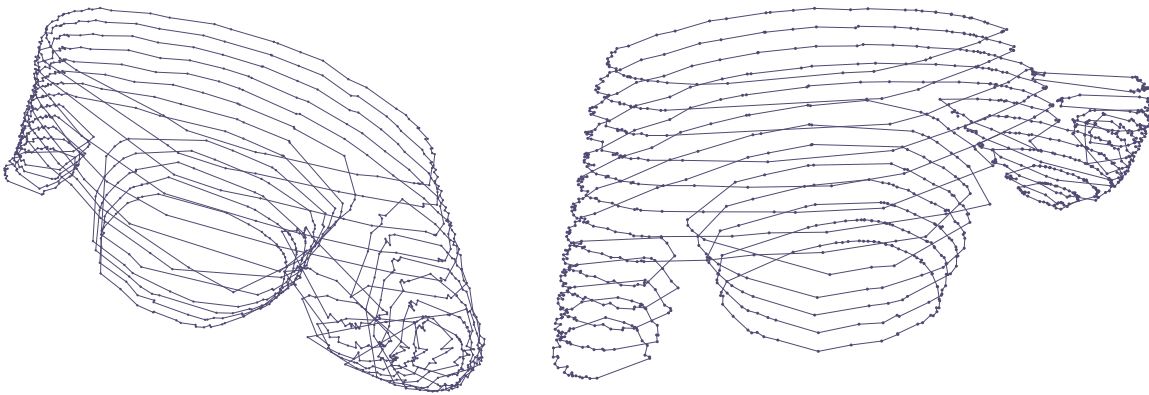


Fig. 8. 2D polygons on the slicing planes generated from the David turn-table sequence.

Note that if we link the contour generator points on each slice directly, we result in a shape generally smaller than the visual hull volume. This becomes apparent if fewer viewpoints are available. To create a better visual effect, we fit a smooth second order parametric curve on the slicing plane replacing long edges of the polygons on each slice.

### B. Generating final surface

The final object surface is extracted from the polygons on each slicing plane using some well-established method. We borrow and modify a method described in [32]. The surface normal at each vertex is simply the interpolated normal from the previous slicing step (see Section V-A).

The overall algorithm outline for extracting the surface from the contour generator points is given below:

**Algorithm 2** Extracting object surface

---

```

1: for each contour generator  $\Gamma_i$  do
2:   for each connected segment along  $\Gamma_i$  do
3:     if has intersection  $p$  with slicing plane  $\pi_k$  then
4:       add  $p$  to  $\pi_k$  along with the interpolated normal
5:     end if
6:   end for
7: end for
8: for each slicing plane  $\pi_k$  do
9:   correct topology and do sorting for the points on  $\pi_k$ 
10:  fit smooth curve if user specified
11:  generate surface triangles with points on  $\pi_{k-1}$ 
12: end for
13: map texture for each triangle

```

---

## VI. EXPERIMENTS

## A. Comparison with naïve dual space approach

In the following set of experiments, we compare our approach (epipolar approach) with the naïve approach. The goal of the comparison is to see quantitatively how much the epipolar parameterization can stabilize the tangent basis estimation.

The two approaches are based on the same implementation except for one key aspect, that is, how the dual space neighbors along the  $t$  direction are located. To be fair, we feed exactly the same sets of silhouettes to both implemented approaches and the sampling rate (so do the sampled points) are also the same on each silhouette. Both approaches estimate  $\tilde{\mathbf{r}}_s^*(s, t)$  by taking finite difference using tangent planes at successively sampled points on the same silhouette. For  $\tilde{\mathbf{r}}_t^*(s, t)$ , our approach locates the dual space neighbors on two immediate neighbor views using the criteria described in Section IV-A, where as the naïve approach searches for the dual space neighbors in tangent space sampled from the silhouettes on the same two views. Based on the idea presented in [2] for estimating  $\tilde{\mathbf{r}}_t^*(s, t)$ , we pick those points lying close to  $\tilde{\mathbf{r}}^*(s, t)$  in dual space as the neighbors. With collected neighbors, both approaches use the same weighted tangent kernel to estimate the surface points. Finally, the estimated points are projected

onto every silhouette, and points falling outside any silhouette are marked as invalid.

We have tested the algorithms over a set of sequences ranging from simplest shapes to objects with richer details and non-zero genus. The comparison result in the form of valid estimation ratio is shown in Fig. 9. The first four sequences (vase, cube, cylinder and cat) are based on synthetic models and the images are simply the duplications of OpenGL's rendering buffer. The remaining sequences are real world image sequences.

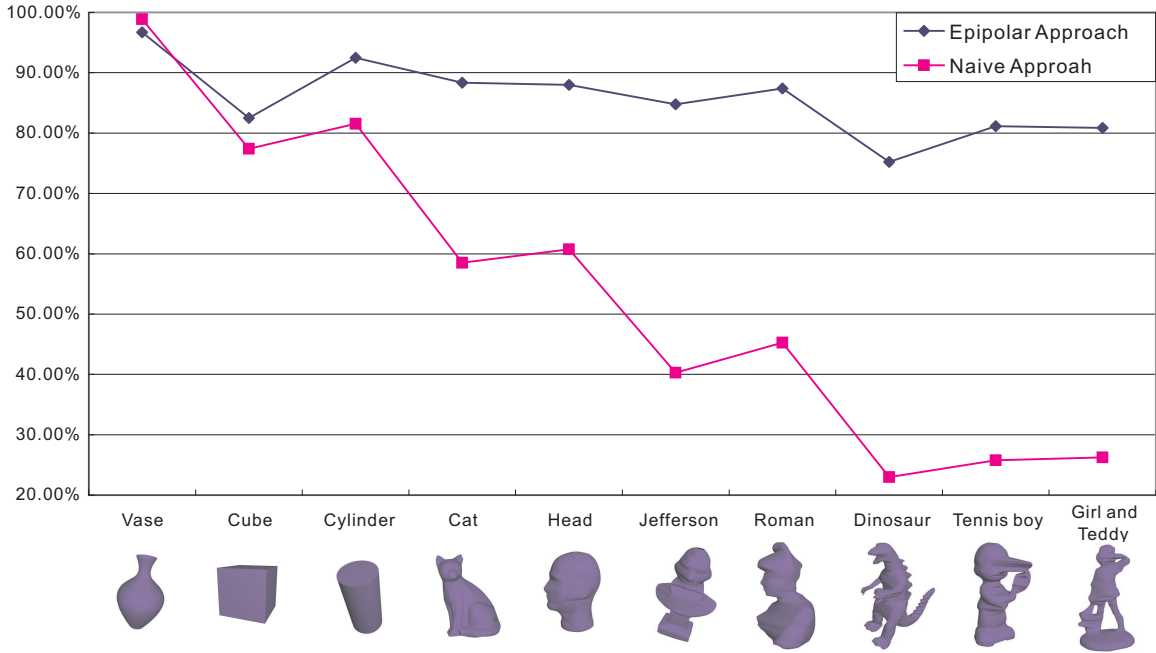


Fig. 9. The comparison between the epipolar approach and the naïve approach on the same set objects of various complexity. The vertical axis shows the ratio of the number of points projected inside all the silhouettes to the total number of points estimated. Higher percentage means more valid estimations and less erroneous estimations.

Both algorithms perform equally well for simple shapes such as vase (Surface of Revolution), cube and cylinder. There is, however, a significant decline in the valid ratio for the naïve approach as the complexity of the object increases. On the other hand, our approach performs reasonably well even for rather complicated objects given sufficient number of viewpoints.

### B. Comparison with other approaches

We have also compared the result of our approach with a typical volumetric approach [8] (denoted as OcTree later) and a polygonal approach [16] (denoted as EPVH later). The OcTree approach is a variation

of [6], which extracts silhouettes up to sub-pixel accuracy by using B-spline Snake. The EPVH approach has been implemented as a library available at <http://www.inrialpes.fr/movi/people/Franco/EPVH/>.

We compared the accuracy of reconstruction in terms of average distance (error) from the reconstructed points to the ground truth surface. To achieve this, we rendered some known parametric or implicit surfaces on the screen. Since we knew the size of the rendering buffer and viewpoints, we could form the intrinsic and extrinsic matrices easily. The rendering results were captured as image sequences together with the computed camera matrices. We show below the reconstruction error for three categories of surfaces, namely, smooth surfaces (a surface of revolution (SOR)), non-smooth surfaces (a unit cube) and surface with complex topology (a knot surface) (see Fig. 10). For each surface, we performed the reconstruction using the two earlier mentioned approaches and our approach. We also adjusted the sampling density of each algorithm in order to maintain a similar number of reconstructed points. We gradually increased the number of images used for reconstruction and compared its impact over the three approaches.

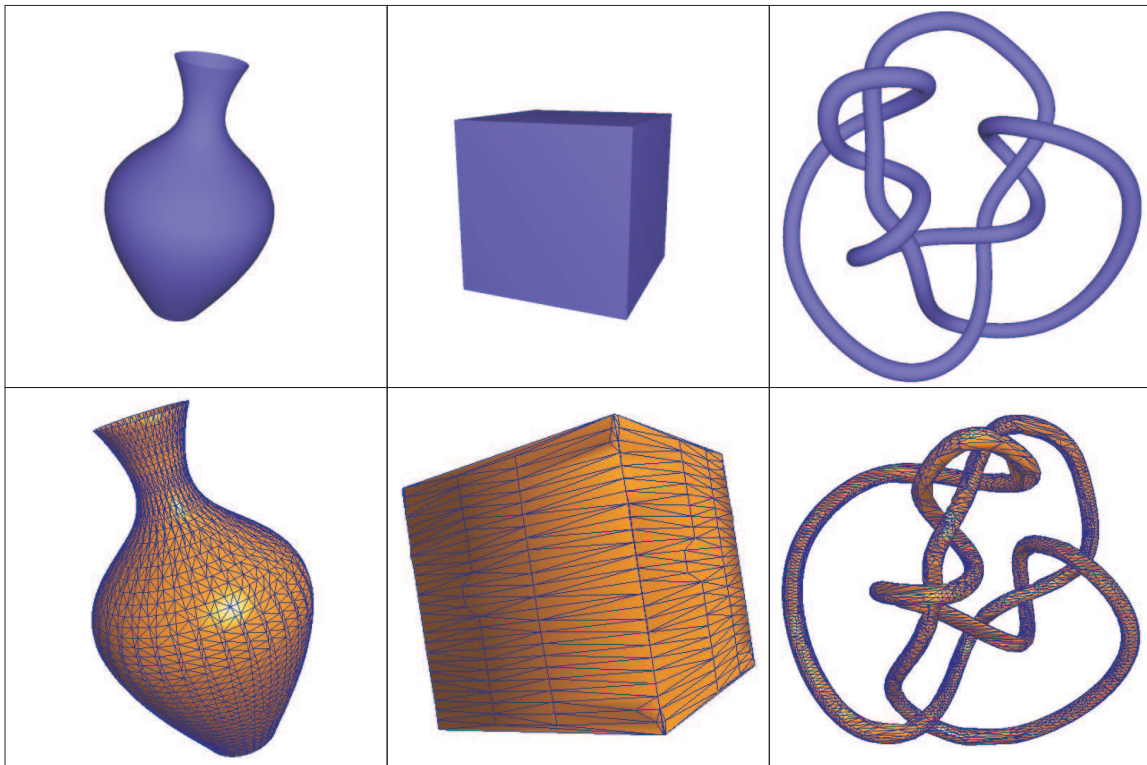


Fig. 10. Examples of known surfaces (upper row) and the reconstruction results of our approach (lower row): a surface of revolution swept by a 2D B-spline (left), a unit cube (middle), and a knot surface produced from known 3D B-spline (right).

Our algorithm performed significantly better than OcTree and EPVH regarding the smooth surface

(see Fig. 11(top-left)). This result is readily foreseeable since OcTree uses parallelepiped-shaped cells to approximate the object surface, while EPVH conservatively approximates the surface as the visual hull. In order to achieve higher accuracy, both OcTree and EPVH requires more images.

Both our approach and EPVH performed reasonably well for non-smooth surface (see Fig. 11(top-right)) given enough images. Our approach performed significantly worse when the number of image was too few (less than 10) due to the fact that our approach is essentially a differential approach.

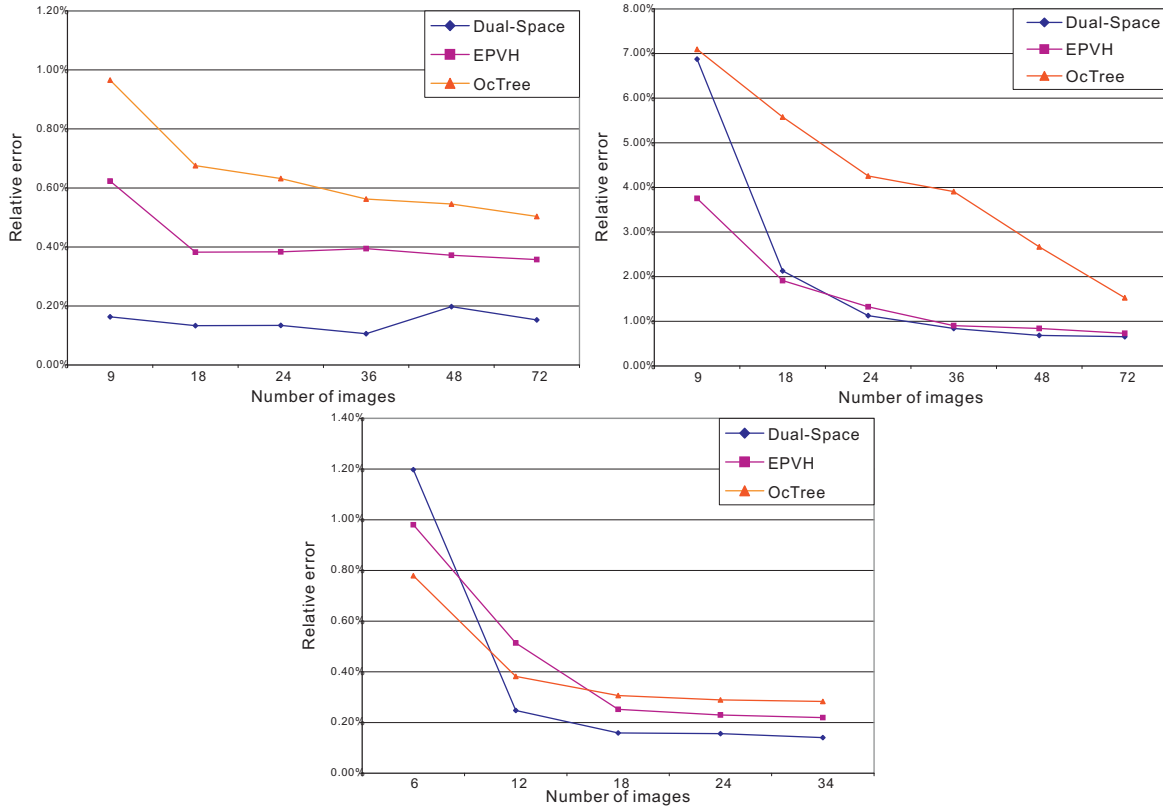


Fig. 11. Average distance of the reconstructed points from the ground truth surface - (top-left) SOR sequence; (top-right) Cube sequence; (bottom) Knot sequence.

As for a surface with very complex topology, when the number of images is too few, our approach produced a significant number of low quality estimations. This is largely due to the fact that with many self-occlusions, too few images leads to incorrect epipolar matching, which greatly affects the quality of the estimation. This situation becomes lessened when more images are adopted (see Fig. 11(bottom)).

### C. Real World Data

We acquired image sequences of real world objects using an electronic turntable, and the cameras were calibrated using the method described in [30]. The purpose of using turntable sequence is solely for the sake of easy camera calibration. In theory, the proposed algorithm should work as long as an ordered image sequence is provided and the camera travels in a relative smooth path (see Fig. 12). This condition is usually satisfied by video sequence captured by digital video recorders.

The Roman Soldier sequence (Fig. 13) consists of 36 colored images, each of size  $800 \times 600$  and the results are shown from novel viewpoints, including the reconstructed contour generator curves, the extracted surface mesh and the shaded surface.

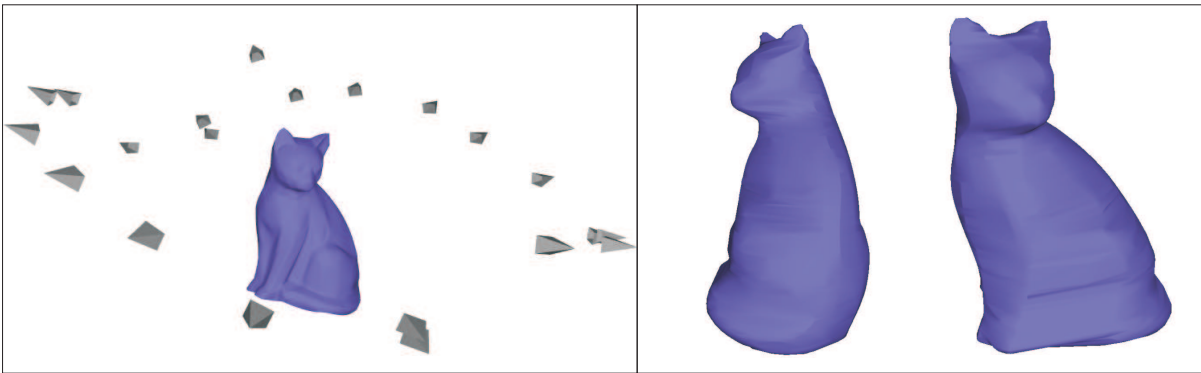


Fig. 12. Cat sequence: our proposed algorithm works as long as an ordered sequence with relative smooth camera path is provided; (left) The viewpoints and the original object - a 3D cat model; (right) The reconstructed cat model using our proposed algorithm.

The Girl and Teddy sequence consists of 20 images and has a fairly complex topology. We visually compared the result of the OcTree algorithm with that of our approach (Fig. 14). By adjusting sampling density, we made both algorithms to produce a similar number of triangles (near 10k). The reconstruction results were registered with one of the silhouette and juxtaposed on each side of that silhouette. The surface are extracted from the OcTree using marching cube algorithm and are slightly smoothed for removing the bumpy effect. While maintaining the similar number of triangles, our method produced points on the surface. Consequently the final mesh perceptibly better resembled the original object.

The David sequence also consists of 20 images. The results and auxiliary data, such as the recovered points on the contour generators and the resampled slices, are shown in (Fig. 15). The result of some more real world sequences are shown in Fig. 16, including a Tennis Boy sequence, a Jefferson statuette

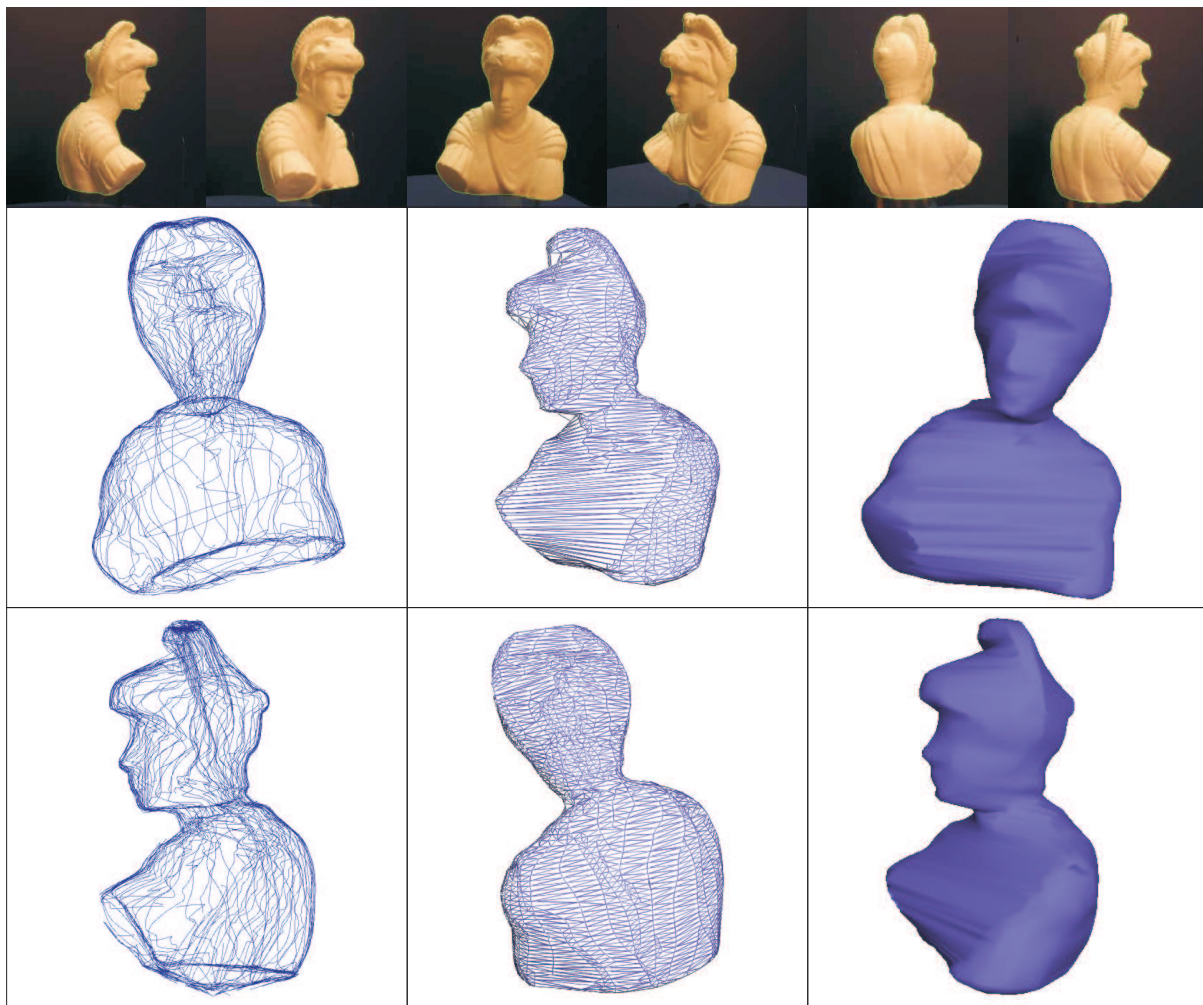


Fig. 13. Roman Soldier sequence; (top) 6 images from the original sequence; (left) reconstructed contour generator curves; (middle) reconstructed surface mesh; (right) shaded reconstructed surface.



Fig. 14. Girl and Teddy sequence; (from left to right) surface generated by OcTree/Marching Cube (10095 triangles); smoothed OcTree model removed the bumpy effect, as well as some shape details; original silhouette; surface generated by our approach (9982 triangles), more consistent with the silhouette compared with the smoothed OcTree model.

sequence and a Statuette of Liberty sequence. All these sequences consist of 20 images captured on an electronic turn-table.

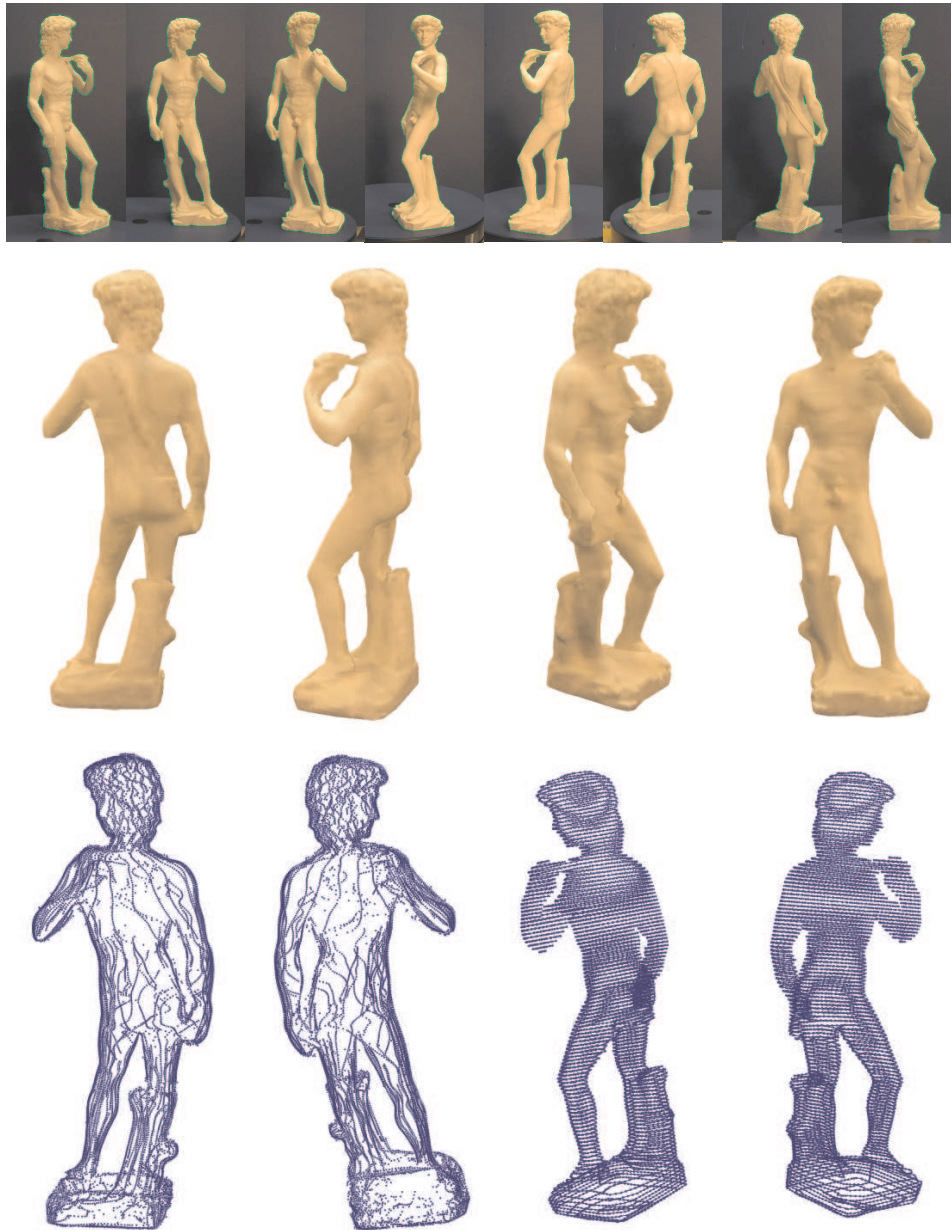


Fig. 15. David sequence; (top) 8 images from the original sequence; (middle) reconstructed final surface; (bottom) auxiliary data, including recovered points on the contour generators and resampled slices.

## VII. CONCLUSION

We have presented in this paper a novel approach for recovering object surface from silhouettes. Our algorithm is motivated by the principle of duality which makes use of the symmetric relationship exhibited





Fig. 16. More experiments on real world data; (top) Images from the original sequence; (left) Tennis boy sequence; (middle) Jefferson statuette sequence; (right) Statuette of liberty sequence.

between the object surface and its tangent plane space sampled from the silhouettes. A crucial problem is how to identify reliable tangent basis in the dual space in the presence of singularities and evil point distributions, which is almost inevitable for fairly complex shapes captured with finite viewpoints. We bring in the epipolar parameterization to solve this problem which results in significant robustness regarding objects with complex topologies. For extreme cases that we may possibly encounter, we have analyzed their nature in detail and proposed counteracts. These have granted our algorithm the ability to cope with fairly complex shapes.

#### ACKNOWLEDGMENT

This project is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project HKU 7185/05E.

#### REFERENCES

- [1] K. Kang, J.-P. Tarel, R. Fishman, and D. Cooper, "A linear dual-space approach to 3d surface reconstruction from occluding contours using algebraic surface," in *International Conference on Computer Vision*, vol. 1, Vancouver, Canada, July 2001, pp. 198–204.
- [2] M. Brand, K. Kang, and D. Cooper, "An algebraic solution to visual hull," in *Computer Vision and Pattern Recognition*, vol. 1, Washington, DC, July 2004, pp. 30–35.

- [3] W. Martin and J. Aggarwal, "Volumetric descriptions of objects from multiple views," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 150–158, 1983.
- [4] C. Chien and J. Aggarwal, "Volume/surface octrees for the representation of three-dimensional objects," *Computer Vision, Graphics and Image Processing*, vol. 36, no. 1, pp. 100–113, October 1986.
- [5] M. Potmesil, "Generating octree models of 3d objects from their silhouettes in a sequence of images," *Computer Vision, Graphics and Image Processing*, vol. 40, no. 1, pp. 1–29, October 1987.
- [6] R. Szeliski, "Rapid octree construction from image sequences," in *CVGIP: Image Understanding*, vol. 58, no. 1, July 1993, pp. 23–32.
- [7] B. García and B. Brunet, "3d reconstruction with projective octrees and epipolar geometry," in *International Conference on Computer Vision*, January 1998, pp. 1067–1072.
- [8] K.-Y. Wong, "Structure and motion from silhouettes," Ph.D. dissertation, University of Cambridge, 2001.
- [9] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 150–162, February 1994.
- [10] B. Baumgart, "A polyhedron representation for computer vision," in *AFIPS National Computer Conference*, 1975.
- [11] S. Sullivan and J. Ponce, "Automatic model construction and pose estimation from photographs using triangular splines," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1091–1096, October 1998.
- [12] S. Lazebnik, E. Boyer, and J. Ponce, "On computing exact visual hulls of solids bounded by smooth surfaces," in *Computer Vision and Pattern Recognition*, vol. 1, Hawaii, December 2001, pp. 156–161.
- [13] S. Lazebnik, "Projective visual hulls," Master's thesis, University of Illinois at Urbana-Champaign, 2002.
- [14] W. Matusik, C. Buehler, and L. McMillan, "Polyhedral visual hulls for real-time rendering," in *Eurographics Workshop on Rendering*, 2001.
- [15] E. Boyer and J.-S. Franco, "A hybrid approach for computing visual hulls of complex objects," in *Computer Vision and Pattern Recognition*, vol. 1, Madison, Wisconsin, June 2003, pp. 695–701.
- [16] J.-S. Franco and E. Boyer, "Exact polyhedral visual hulls," in *British Machine Vision Conference*, vol. 1, September 2003, pp. 329–338.
- [17] G. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo," in *Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003.
- [18] R. Cipolla and A. Blake, "Surface shape from the deformation of apparent contours," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 83–112, 1992.
- [19] R. Vaillant and O. Faugeras, "Using extremal boundaries for 3-d object modeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 157–173, February 1992.
- [20] E. Boyer and M.-O. Berger, "3d surface reconstruction using occluding contours," in *Computer Analysis of Images and Patterns*, 1995, pp. 198–205.
- [21] P. Giblin and R. Weiss, "Reconstruction of surfaces from profiles," in *International Conference on Computer Vision*, London, England, June 1987, pp. 136–144.
- [22] K. N. Kutulakos, "Shape from the light field boundary," in *Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 53–59.
- [23] K. Kang, "Three-dimensional free form surface reconstruction from occluding contours in a sequence images or video," Ph.D. dissertation, LEMS, Div. of Engineering, Brown University, 2004.

- [24] A. Sethi, D. Renaudie, D. Kriegman, and J. Ponce, "Curve and surface duals and the recognition of curved 3d objects from their silhouette," *International Journal of Computer Vision*, vol. 58, no. 1, pp. 73–86, 2004.
- [25] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.
- [26] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 139–154, 1985.
- [27] R. Weiss, "The epipolar parametrization," in *Object Representation in Computer Vision, Int. NSF-ARPA Workshop*, New York City, NY, USA, December 1994, pp. 101–107.
- [28] R. Cipolla, G. Fletcher, and P. Giblin, "Following cusps," *International Journal of Computer Vision*, vol. 23, no. 2, pp. 115–129, 1997.
- [29] R. Cipolla, K. Åström, and P. Giblin, "Motion from the frontier of curved surfaces," in *International Conference on Computer Vision*, Cambridge, MA, USA, June 1995, pp. 269–275.
- [30] K.-Y. Wong and R. Cipolla, "Structure and motion from silhouettes," in *International Conference on Computer Vision*, vol. 2, Vancouver, Canada, 2001, pp. 217–222.
- [31] E. Boyer and M.-O. Berger, "3d surface reconstruction using occluding contours," *International Journal of Computer Vision*, vol. 22, pp. 219–233, 1997.
- [32] J.-D. Boissonnat, "Shape reconstruction from planar cross sections," *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 1–29, October 1988.