International Journal of Computer Vision

DOI: 10.1007/s11263-006-8323-9

3D Reconstruction by Shadow Carving: Theory and Practical Evaluation

SILVIO SAVARESE AND MARCO ANDREETTO
California Institute of Technology, Mail stop 136-93, Pasadena, CA 91125
savarese@vision.caltech.edu
marco@vision.caltech.edu

HOLLY RUSHMEIER AND FAUSTO BERNARDINI
IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598
holly@acm.org
fausto@watson.ibm.com

PIETRO PERONA
California Institute of Technology, Mail stop 136-93, Pasadena, CA 91125
perona@vision.caltech.edu

Received May 20, 2005; Revised March 16, 2006; Accepted March 16, 2006

First online version published in June, 2006

Abstract. Cast shadows are an informative cue to the shape of objects. They are particularly valuable for discovering object's *concavities* which are not available from other cues such as occluding boundaries. We propose a new method for recovering shape from shadows which we call *shadow carving*. Given a conservative estimate of the volume occupied by an object, it is possible to identify and *carve* away regions of this volume that are inconsistent with the observed pattern of shadows. We prove a theorem that guarantees that when these regions are carved away from the shape, the shape still remains conservative. Shadow carving overcomes limitations of previous studies on shape from shadows because it is robust with respect to errors in shadows detection and it allows the reconstruction of objects in the round, rather than just bas-reliefs. We propose a reconstruction system to recover shape from silhouettes and shadow carving. The silhouettes are used to reconstruct the initial conservative estimate of the object's shape and shadow carving is used to carve out the concavities. We have simulated our reconstruction system with a commercial rendering package to explore the design parameters and assess the accuracy of the reconstruction. We have also implemented our reconstruction scheme in a table-top system and present the results of scanning of several objects.

Keywords: shape recovery, shape from shadows, 3D reconstruction, computer vision, shape from silhouettes, shape from contours

1. Introduction

A number of cues, such as stereoscopic disparity, texture gradient, contours, shading and shadows, have been shown to carry valuable information on surface shape, and have been used in several methods for 3*D* reconstruction of objects and scenes. Techniques based on shadows have the advantage that they do not rely on

correspondences, on a model of the surface reflectance characteristics, and they may be implemented using inexpensive lighting and/or imaging equipment. Past methods to recover shape from shadows, however, have proven to be sensitive to errors in estimating shadow boundaries. Moreover, their are mostly restricted to objects having a bas-relief structure.

We propose a novel $3D$ reconstruction method for using shadows that overcomes previous limitations. We assume that we have, as a starting point, a conservative estimate of object shape; that is, the volume enclosed by the surface estimate completely contains the physical object. We analyze images of the object illuminated with known light sources taken from known camera locations. We assume that we are able to obtain a conservative estimation of the object's shadows—that is, we are able to identify image areas that we are certain to be in shadow region, and do not attempt to find exact shadow boundaries. We do not make assumptions about the surface topology (multi-part objects and occlusions are allowed), although any tangent plane discontinuities over the objects surface are supposed to be detected. Our basic idea is that we compare observed shadows to those expected as if the conservative estimate were correct and adjust the current shape to resolve contradictions between the captured images and the current shape estimate. In this process, we incrementally remove (carve out) in a conservative way volume from the current object estimate in order to reduce the inconsistencies. Thus, a closer objects shape estimate is computed at each step. We call this novel procedure *shadow carving*. We provide a proof that the carving process is always conservative.

Shadow carving improves previous results on shape from shadows in two main aspects: (i) it is more robust with respect to the classification of shadow regions; (ii) it gives the possibility of recovering objects in the round (rather than just bas-reliefs).

Our motivation for pursuing this work relies in applications where the user often has a very limited budget, and is primarily concerned with visually, rather than metrically, accurate representations. Furthermore, because users are often not technically-trained, the scanning and modeling systems must be robust and require minimal user intervention.

We validate our theoretical results by implementing a scanning system based on shape from silhouettes and shape from shadows. First, the silhouettes are used to recover a conservative estimate of the object shape. Then, a series of images of the object lit by an array of

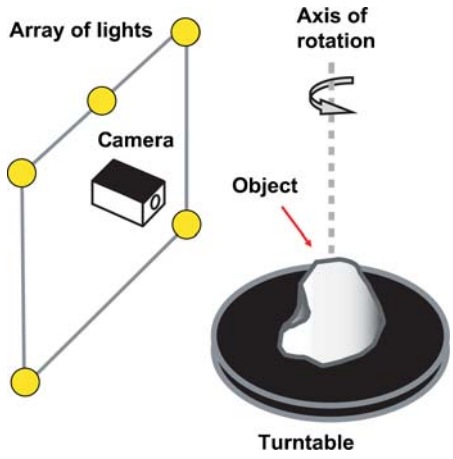

Figure 1. Setup for shape from self-shadowing: using each of several lights in turn, and a camera in front, allows multiple sets of shadow data to be obtained for each object position.

light sources are recorded with a setup shown schematically in Fig. 1. Such images are examined and the shadows that the object casts on itself are detected. The current shape estimate is then refined by the shadow carving procedure. Eventually, the improved estimate of shape can be further refined by methods that work well locally, such as photometric stereo (Horn and Brooks, 1989).

Our system is designed to be inexpensive as other recently proposed schemes (Rocchini et al., 2001; Bouguet and Perona, 1999; Andreetto et al., 2004). It uses a commodity digital camera and controlled lighting systems composed of inexpensive lamps. Moreover, since our method relies on substantial variations in the intensities in acquired images, it does not requires precise tuning, hence it minimizes the user intervention. Finally, since our technique progressively improves conservative estimates of surface shape, it prevents small errors from accumulating and severely deteriorating the final results.

The rest of this paper is organized as follows: we begin by reviewing previous work on shape from shadows and identifying the strengths and weaknesses of these methods in Section 2. We present the geometry of our approach and demonstrate that it always produces a conservative estimate of the object shape in Section 3. We propose our prototype system for shape recovery from silhouettes and shadows in Section 4. We test the performance of the method with different configurations of lights and camera positions and assess the accuracy of the reconstructiondue to error in

the shadow estimates in Section 6.1. Finally, we present results we have obtained from a small table-top implementation of our system in Section 6.2.

2. Brief History of Shape-from-Shadows

Computing shape from shadows—or shape from darkness—has a long history. Shafer and Kanade (1983) established fundamental constraints that can be placed on the orientation of surfaces, based on the observation of the shadows one casts on another. Hambrick et al. (1987) developed a method for labelling shadow boundaries that enables inferences about object shape. Since then, several methods for estimating shape from shadows have been presented. Because we aim at designing a $3D$ scanner, we focus on reconstruction schemes where the light source position is known, rather than the case of unknown light source direction (e.g., Kriegman and Belhumeur, 2001). Also, we limit our analysis to methods using self-shadows (i.e., shadows cast by the object upon itself) rather than shadows cast by external devices as in Bouguet and Perona (1999). Hatzitheodour and Kender (1988) presented a method for computing a surface contour formed by a slice through an object illuminated by a directional light source casting sharp shadows. Assuming that the contour is defined by a smooth function—and that the beginning and end of each shadow region can be found reliably—each pair of points bounding a shadow region yields an estimate of the contour slope at the start of the shadow region, and the difference in height between the two points as shown in Fig. 2. The information for shadows from multiple light source positions is used to obtain an interpolating spline that is consistent with all the observed data points.

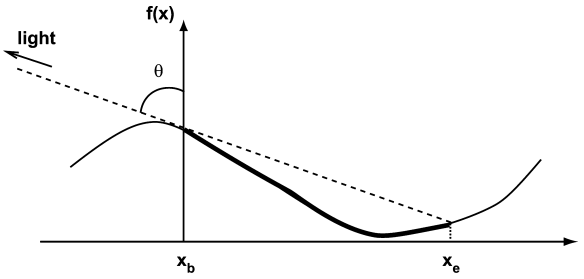

Figure 2. Shape from Shadows: for terrain surfaces $f(x)$ and a known light source direction θ, $f'(x_b) = tan\theta$, and $f(x_b) - f(x_e) = f'(x_b)(x_e - x_b)$. Using data for many angles θ an estimate of the continuous function $f(x)$ can be made.

Raviv et al. (1989) developed an extended shape from shadows method. The object is set on a known reference surface with a camera directly above. A series of images is captured as a collimated light source moves in an arc over the surface. For the $2D$ reference surface, a volume of data is then obtained with the third coordinate being the angle of the light source to the reference surface, and the volume recording whether the reference surface was in shadow for that angle. A slice through this volume is referred to as a *shadowgram*. Similar to Hatzitheodorou and Kender, by identifying beginning and ending shadow points for each light position, the height difference between the points can be computed. Also, by observing the change in shadow location for two light source positions, the height between the start of the shadow at one position relative to the other can be found by integration. As long as shadow beginnings and endings can reliably be detected, the top surface of the object can be recovered as a height field. Furthermore, by detecting splits in the shadowgram (i.e., positions that have more than one change from shadowed to unshadowed), holes in the surface below the top surface can be partially recovered.

Langer et al. (1995) extended the method of Raviv et al. for computing holes beneath the recovered height field description of the top surface for two dimensions. They began with the recovered height field, an $N \times N$ discretization of the two dimensional space, and the captured shadowgram. Cells in this discretization are occupied if they are in the current surface description. Their algorithm steps through the cells and updates them to unoccupied if a light ray would have to pass through the cell to produce a lit area in the captured shadowgram.

Daum and Dudek (1998) subsequently developed a method for recovering the surface for light trajectories that are not a single arc. The estimated height field description is in the form of an upper bound and lower bound on the depth of each point. The upper and lower bounds are progressively updated from the information obtained from each light source position.

All of these methods rely on accurate detection of the boundaries of shadow regions. This is particularly problematic for attached shadows that are the end of a gradual transition of light to dark. Height estimates that use gradients derived from the estimate of the start of attached shadows are particularly prone to error. Yang (1996) considers the problem of shape from shadows with error. He presents a modified form of Hatzitheodorou and Kender approach, in which linear

programming is used to eliminate inconsistencies in the shadow data used to estimate the surface. While the consistency check does not guarantee any bounds on the surface estimate, it does guarantee that the method will converge. He shows that the check for inconsistences is *NP*-complete. While more robust than Hatzitheodorou and Kender's method when applied to imperfect data, Yang's technique is still restricted to $2.5D$ terrains. Finally, Yu and Chang (2005) give a new graph-based representation for shadow constraints.

Our method does not require a restriction to $2.5D$ terrains. Rather, it allows a fully $3D$ reconstruction of the object. Moreover, we do not rely on knowing the boundaries of shadow regions to compute surface shape. Similar to Yang's approach, we use the idea of consistency to avoid misinterpreting data. However, rather than comparing multiple shadow regions for consistency, we check that observed shadow regions are consistent with our current surface estimate.

Our proposed approach—shadow carving—is similar in spirit to the space carving approach of Kutulakos and Seitz (1999). Our approach differs from Kutulakos and Seitz (1999) in that we consider consistency between a camera and light views, rather than multiple camera views. Consistency can be tested robustly by detecting shadows, without requiring the hypothesis of Lambertian surface. We begin with a conservative surface definition, rather than a discretized volume. Inconsistent regions can be carved out by moving surface points at the resolution of the captured images, rather than being limited to a set of fixed resolution voxels. Most importantly, we provide a proof of correctness that well defined portions of volume can be removed in a conservative manner from the current object estimate, instead of iteratively removing single voxels until all the inconsistencies are solved.

This paper gathers our own previous work (Savarese et al., 2002, 2001) and presents an extended experimental analysis in that: (i) performance of the method with different configurations of lights and camera positions is tested; (ii) accuracy of the reconstruction due to errors in the shadow estimates is assessed; (iii) throughout experiments with real objects are shown.

3. Shadow Carving

We introduce a formal definition of shadow carving and present its main property of yielding conservative object estimates in Section 3.1. Then we show a simple example of shadow carving in the context of the epipolar slice model described in Sections 3.2 and 3.3. Finally, we prove that shadow carving always yields conservative estimates in Sections 3.4, 3.5, 3.6 and 3.7.

3.1. The Shadow Carving Theorem

Consider an object in $3D$ space and a point light source L illuminating it. See Fig. 3 . One or more shadows are cast over the object by parts of the object itself. The scene is observed by a pin-hole camera whose center is located in O_c. Let us introduce the following definitions:

Definition 3.1. A *conservative object estimate* is any volume $\hat{R}$ that contains the object R.

Definition 3.2. A portion of a surface is *visible* from a point X if every point of the portion of the surface can be connected to X without intersecting the surface itself.

Notice that in this definition the surface may be either the surface of the object or the surface of the conservative object estimate. Furthermore, the point X may be either the center of the camera O_c or the light source L.

Definition 3.3. A *shadow* is a portion of the surface of the object that is not visible from the light source L.

Definition 3.4. The *shadow volume* V_o is the set of lines through the center of the camera projection O_c and all of the visible points (from O_c) of the object's surface that are in shadow.

Definition 3.5. The *inconsistent shadow* is the portion of the surface of the conservative object estimate $\hat{R}$ that intersects V_o and is visible from the camera and the light source.

Definition 3.6. The *light volume* V_L is the set of lines through the light source L and the points of the inconsistent shadow.

Notice that V_L is dependent on the particular choice of $\hat{R}$.

Definition 3.7. The *carvable volume* V_C is $V_o \cap V_L \cap \hat{R}$.

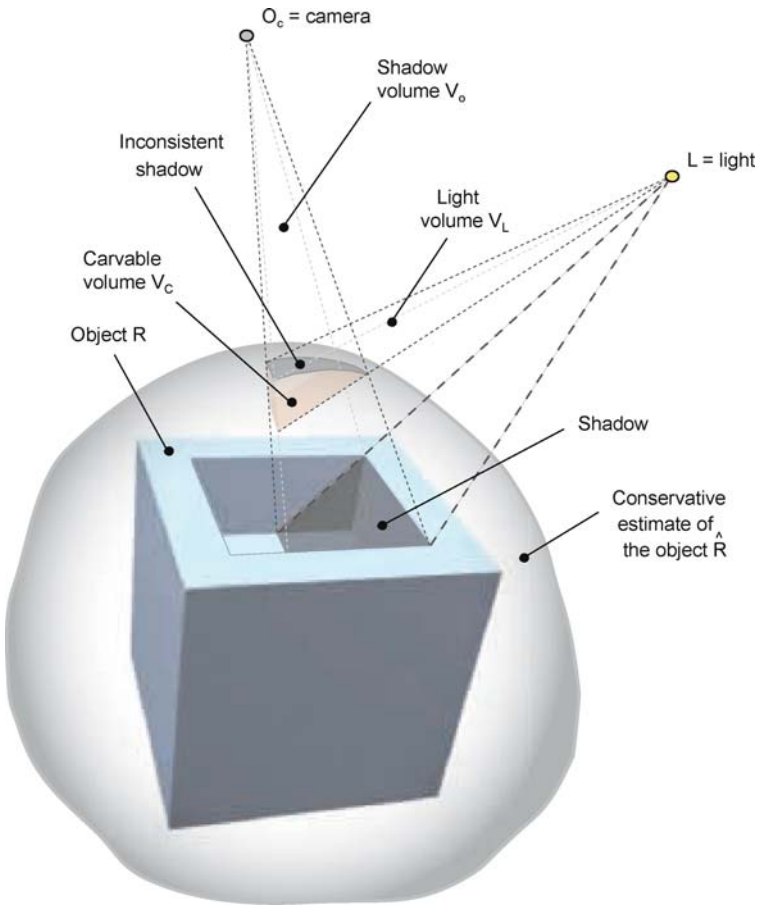

Figure 3. Example of shadow carving in $3D$.

Theorem 3.1. *If the object surface is smooth, $\hat{R}$ minus V_C is a conservative object estimate.*

Notice that all the quantities (i.e., L, O_c, $\hat{R}$ and the image points of the object's surface that are in shadow) are available either from calibration or from measurements in the image plane. Therefore Theorem 3.1 suggests a procedure to estimate the object incrementally: (i) start from a conservative object estimate; (ii) measure in the image plane all of the visible points of the object's surface that are in shadow and compute the shadow volume; (iii) compute the corresponding inconsistent shadows; (iv) compute the corresponding light volume; (v) intersect the conservative object estimate, the shadow volume and the light volume and calculate the carvable volume; (vi) remove the carvable volume from the conservative object estimate to obtain an new object estimate. Theorem 3.1 guarantees that the new object estimate is still a *conservative* object estimate. The process can be iterated by considering a new light source or by viewing the object from a different vantage point. This procedure will be developed in details in Section 4.

As we shall see in Section 3.4, the theorem still holds if there are errors in detecting the visible points of the object's surface that are in shadow. These errors, however, must be conservative; namely, a piece of shadow can mislabeled as non-shadow, but a non-shadow cannot be mislabeled as a shadow.

3.2. The Epipolar Slice

In order to prove Theorem 3.1 we examine the problem in an appropriately chosen $2D$ slice of the $3D$ space, *the epipolar slice*. As we shall discuss in more details

in Section 3.6, the results that we prove for a given slice hold in general and do not depend on the specific choice of the slice. Thus, the extension to the $3D$ case is immediate by observing that the epipolar slice can sweep the entire object's volume.

Consider the family of planes Π_L passing through O_c and L (see Fig. 4). Each plane $\pi_L \in \Pi_L$, intersects the image plane π_i and the object. In other words, each π_L defines an epipolar slice of $3D$ space. For each slice, we have the *image line* (i.e., the intersection of π_i with π_L), the *image shadow segment* (i.e., the intersection of the estimated image shadow with π_L), the *object contour* P (i.e., the intersection of the object's surface with π_L) and the *conservative object contour* $\hat{P}$ (i.e., the intersection of the conservative surface estimation with π_L). Additional quantities are the *object area* A_P (that is, the area bound by P) and the *conservative object area* $A_{\hat{P}}$ (that is, the area bound by $\hat{P}$).

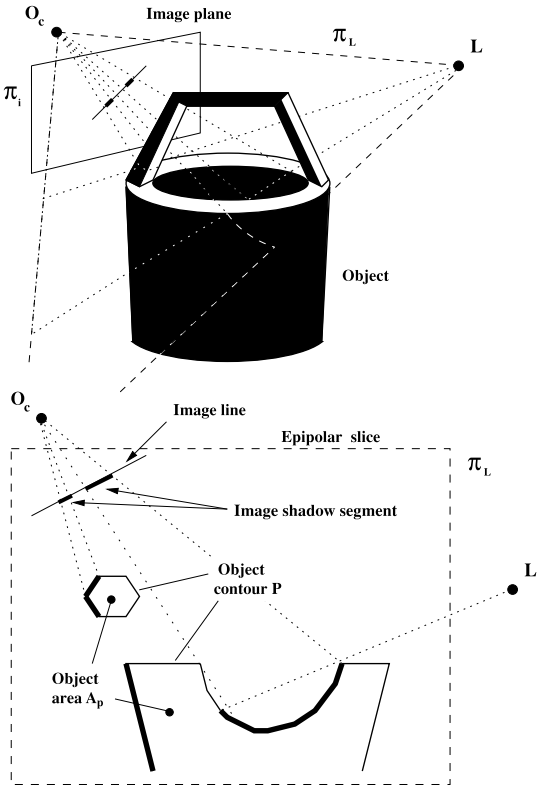

Figure 4. Top: an object in $3D$ space and a point light source L illuminating it. The scene is observed by a camera whose center is located in O_c and whose image plane is called π_i. Bottom: the plane π_L defines an epipolar slice of $3D$ space.

3.3. Example

Figure 5 shows an example of shadow carving in the epipolar slice model. The shadow $\bar{s}$ is cast by the object contour P over itself. s is the image of $\bar{s}$. The *shadow area* A_o is the set of lines through O_c and all of the points along s that are visible from O_c. The portion of $\hat{P}$ that is visible from the camera and the light, and intersects A_o is the *inconsistent shadow* $\hat{s}$. This shadow is called inconsistent because it is visible from the light source L. Thus, the conservative estimate $\hat{P}$ has to be re-adjusted in order to explain this inconsistency. The *light area* A_L is the set of lines through L and all of the points on $\hat{s}$. Finally, $A_o \cap A_L \cap A_{\hat{P}}$ gives the area A_C which we call the *carvable area* (i.e., cross-hatched area in the figure). A_C can be removed from $A_{\hat{P}}$, generating an improved conservative approximation of P. Notice that the new conservative estimate is consistent with the observed shadow s and with the light source position. Thus, no additional carving is required. Finally, notice that A_C can be removed in one shot rather than by means of an iterative process as in Kutulakos and Seitz (1999).

Many questions arise and we may wonder what happens if (i) the topology of P is more complex; (ii) the shadow is occluded by other object parts; (iii) multiple shadows are imaged by the camera; (iv) the object surface contains low-albedo regions that do not allow a correct or complete estimate of the shadows. Can we still define a carvable area? Can we guarantee that the carvable area is always outside the object (i.e., the new estimate is still conservative)?

In Section 3.4, we address these questions; we first introduce the definition of atomic shadow and then describe how the measured shadow can be decomposed into atomic components. Given such decomposition,we formally define the carvable area A_C and prove that A_C is always outside the object area (Sections 3.5 and 3.6). As we shall see, no hypothesis on the topology of object is made. However, an hypothesis on the smoothness of the object surface is required and is discussed at the end of Section 3.4.

3.4. Decomposing the Shadow

Let us consider an epipolar slice of a $3D$ object. In general, the object's contour P on the epipolar slice might comprise more than one separate contour components. For instance, the body and the handle of the bucket (Fig. 4-top) may yield two separate contours

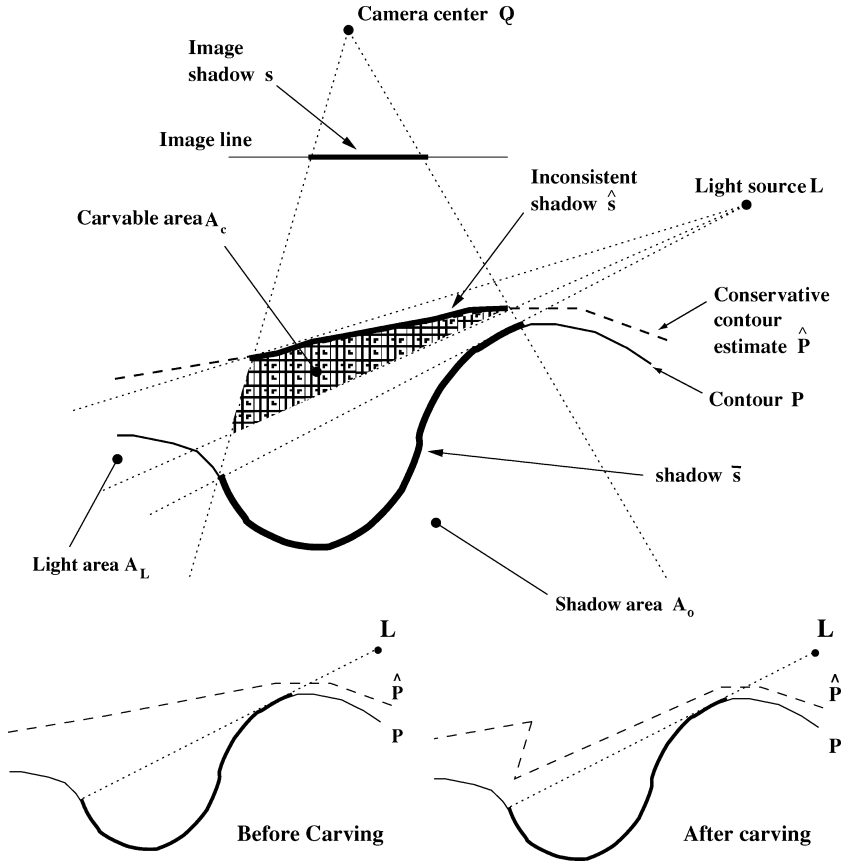

Figure 5. Example of shadow carving.

(Fig. 4-bottom). Thus, depending on the object's topology and the light source position, there will be a certain shadow distribution. Furthemore, some portions of a given shadow might be occluded from the camera view by other parts of the object. A portion of shadow which is visible from the camera (i.e., from the center of projection) is called *unoccluded*. Figure 6 shows examples of unoccluded shadows: the unoccluded shadows are indicated with the black bold lines; the occluded portions of shadow are indicated with black dashed lines. It is clear that whether a shadow is occluded or not only depends upon the contour topology as well as the camera position.

Let us suppose that we have a technique to detect shadows. It is a reasonable assumption that such a shadow detection technique is always *conservative*. That is, a shadow may not be detected, but whatever is labeled as shadow is indeed a shadow. See Section 4.4 for details. Thus, a portion of contour is called *undetectable* if, according to the shadow detection technique, it cannot be classified either as a shadow or as not a shadow. We call *detectable* a shadow which does not lie within an undetectable portion of contour. Figure 7 shows examples of detectable shadows and undetectable portions of contours: the detectable shadows are indicated in bold black; the undetectable portions of contours are indicated in gray.

Definition 3.8. A maximal connected portion of shadow which is both unoccluded and detectable is called *atomic*.

An atomic shadow is indicated with the symbol $\bar{a}$ and its corresponding perspective projection into the image line is indicated by a. We call a an atomic image shadow. As a result, any shadow $\bar{s}_j$ can be decomposed into its atomic components $\bar{a}_{j,1}, \bar{a}_{j,2} \ldots \bar{a}_{j,k}$. See Fig. 8 for an example: the atomic shadows (indicated in bold black) within $\bar{s}_1$ are $\bar{a}_{1,1}, \bar{a}_{1,2}, \bar{a}_{1,3}$ and $\bar{a}_{1,4}$. The perspective projection of the atomic shadows into the

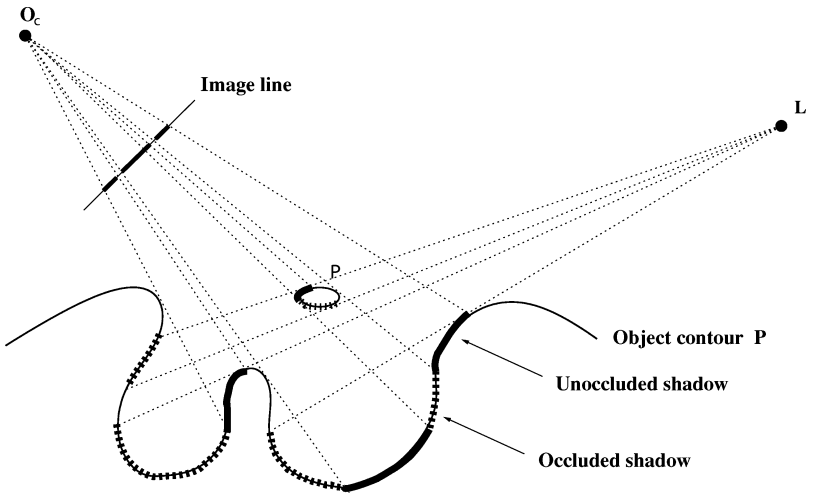

Figure 6. Example of occluded and unoccluded shadows.

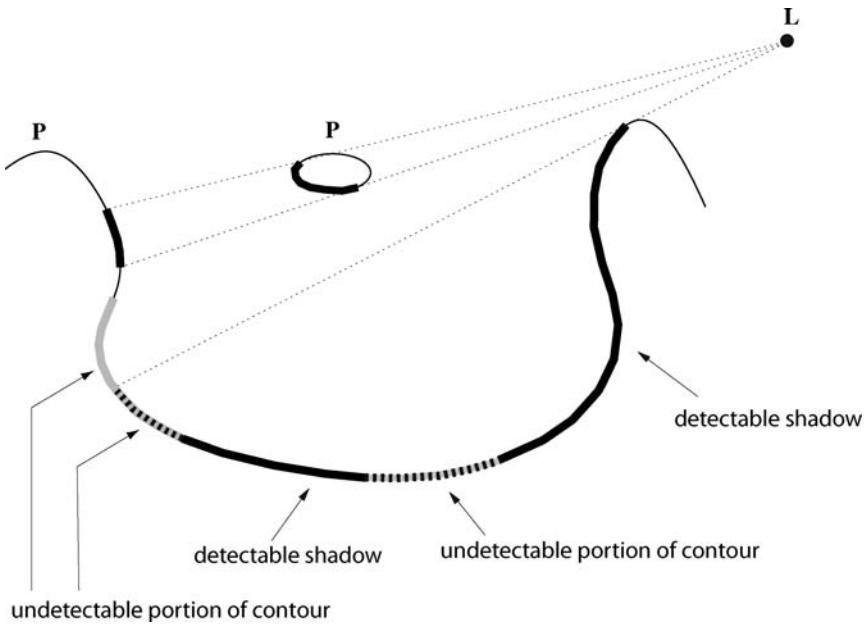

Figure 7. Example of detectable shadows. Undetectable portions of contours are indicated in bold gray. Detectable shadows are indicated in bold black. Shadows within undetectable portions of contours are indicated in black dashed.

image line yields the atomic image shadows $a_{1,1}$, $a_{1,2}$, $a_{1,3}$ and $a_{1,4}$. Occluded shadows and shadows falling into undetectable portions of contour are indicated in dashed bold black.

We assume that the shadow detection technique gives us an estimation e_u of the perspective projection into the image line of the complete set of unoccluded shadows and an estimation e_d of the perspective projection of the complete set of detectable regions. The intersection of e_u with e_d gives the estimated shadow distribution e along the image plane of both unoccluded and detectable shadows. The estimated shadow e can be described as the union of maximal connected components which we call estimated shadow component e_i. It is easy to show that there is no unique correspondence between estimated shadow components and

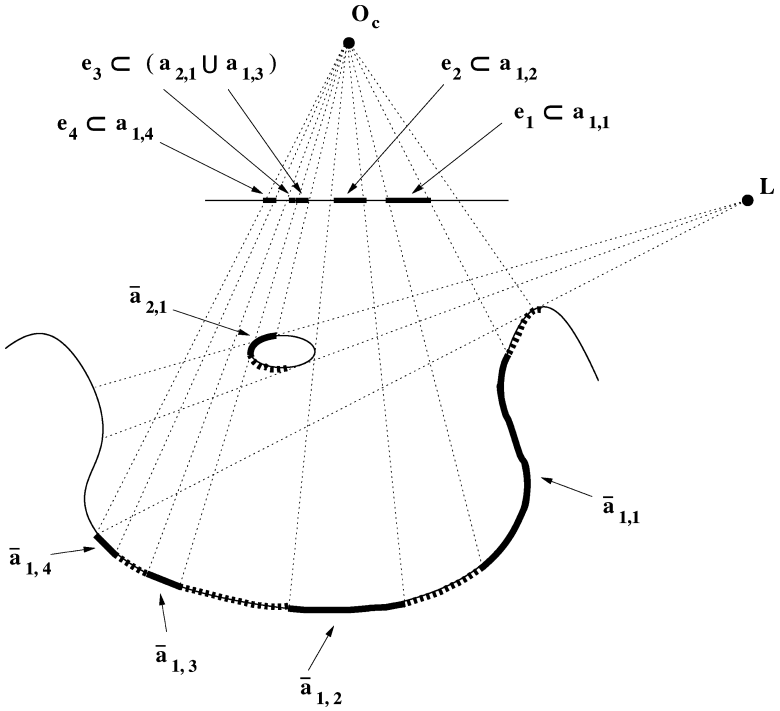

Figure 8. Example of atomic shadows.

atomic image shadows. For instance, see Fig. 8. The estimated shadow component e_3 corresponds to atomic image shadows $a_{2,1}$ and $a_{1,3}$.

Lemma 3.1. *If the object contour's first order derivative is continuous (i.e., the object's contour is smooth) and if the shadow detection is conservative (i.e., a piece of shadow can be mislabeled as non-shadow, but a non-shadow cannot be mislabeled as a shadow), an estimated shadow component is always a lower bound estimation of either an atomic image shadow or the union of two or more atomic image shadows.*

Proof: The lemma just follows from the definitions. □

An example is shown in Fig. 8: the estimated shadow e within the image line can be decomposed into its estimated shadow components e_1, e_2, e_3 and e_4. In particular, e_1 is a lower bound estimation of $a_{1,1}$. e_3 is a lower bound estimation of the union of $a_{1,3}$ and $a_{2,1}$. Notice that e_3 appears as a connected shadow component, although $a_{1,3}$ and $a_{2,1}$ are the images of atomic shadows generated over two different contours.

In the following we want to show that if the hypothesis of smoothness is removed, Lemma 3.1 is no longer verified. Let us consider the example depicted in Fig. 9. The surface P casts two shadows over P itself. $\bar{a}_1$ and $\bar{a}_2$ are the corresponding atomic shadows. In particular, $\bar{a}_1$ is cast by the point $\bar{p}$. Hence, $\bar{p}$ cannot be in shadow. Furthermore, $\bar{p}$ is visible from O_c. Thus, the corresponding atomic image shadows a_1 and a_2 do not constitute a connected component. However, if the edge is sharp enough, the camera might not be able to resolve $\bar{p}$, which will be estimated as being in shadow. The shadow decomposition fails; the estimated shadow component is no longer a lower bound conservative estimate of a_1 and a_2. In other words, a_1 and a_2 are estimated to be a unique connected component instead of two disconnected shadow components e_1 and e_2. As a result, Lemma 3.1 does not necessarily hold when the contour smoothness hypothesis is not verified.

We can remove the hypothesis of smoothness if we suppose that we have available a technique to identify points whose first order derivative is not continuous. We call such points *singular* and we label them as undetectable. Let us consider again the example of Fig. 9. If $\bar{p}$ can be identified as singular and therefore its image

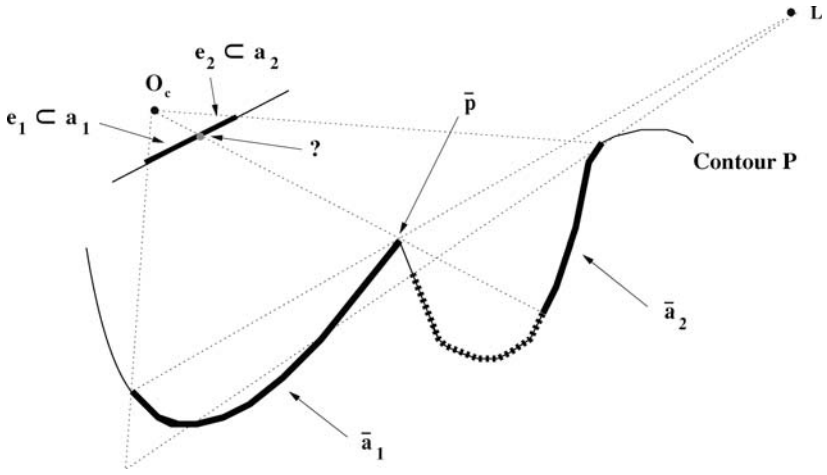

Figure 9. Contour with singular point.

p labeled as undetectable, a_1 and a_2 are no longer estimated as a unique shadow component, but as two disconnected estimated shadows components e_1 and e_2. The shadow decomposition is again conservative and Lemma 3.1 is verified. From now on, we assume either to deal with smooth contours or to have a technique to identify singular points so that we can use the property stated in Lemma 3.1.

In the next section we formally define the carvable area and prove that each carvable area attached to a generic e_i does not belong to the object area (conservative carving). By means of Lemma 3.1 we decompose the problem into two cases: an estimated shadow component corresponds to an atomic image shadow; an estimated shadow component corresponds to the union of two or more atomic image shadows.

3.5. Atomic Shadow

Let us consider an estimated shadow component e and let us assume that e is exactly the atomic image shadow a (see Fig. 10). We call $\bar{a}$ the corresponding atomic shadow over the object contour. Let A_o be the area defined by the set of lines through O_c and e. The following lemma holds:

Lemma 3.2. *Given an estimated shadow component e and the corresponding area A_o, all the points belonging to P and within A_o either belong to $\bar{a}$ or are not visible from O_c. In particular, if there is a point $p \in P$ and $\in A_o$ not visible from O_c, there must exist at least one point $q \in \bar{a}$ between p and O_c.*

Proof: The lemma follows from the definition of atomic shadow and Lemma 3.1. □

Given Lemma 3.2, A_o is the $2D$ slice of the shadow volume V_o in Definition 3.4 and it is called *shadow area.*

In the following, we introduce additional geometric quantities that will lead to the definition of carvable area. Let $\hat{P}$ be the current conservative contour estimation and let $\hat{e}$ (light gray line in Fig. 10) be the projective transformation (centered in O_c) of e onto the conservative contour $\hat{P}$ closest to O_c.[1] Since $\hat{P}$ is a conservative estimation of the real P, $\hat{e}$ is closer to O_c then $\bar{a}$. Let $\hat{v}$ (black line in Fig. 10) be the portion of $\hat{e}$ whose points can be connected by a line segment to L without intersecting $\hat{P}$. Thus, any point $\in \hat{v}$ must be visible from L. $\hat{v}$ corresponds to the inconsistent shadow in Definition 3.5. Let A_l be the area defined by the family of lines passing through L and any point along $\hat{v}$. A_l is the $2D$ slice of the light volume V_L in Definition 3.6 and it referred as *light area*.

Definition 3.9. We define the *carvable area* A_C as the area obtained by the intersection of the estimated object area (bounded by $\hat{P}$), the light area A_l and the shadow area A_o.

The carvable area corresponds to the cross-hatched area in Fig. 10 and is the $2D$ slice of the volume V_C in Definition 3.7. The key result in this section is stated in the following lemma:

Lemma 3.3. *The carvable area A_C does not intersect the real object contour P.*

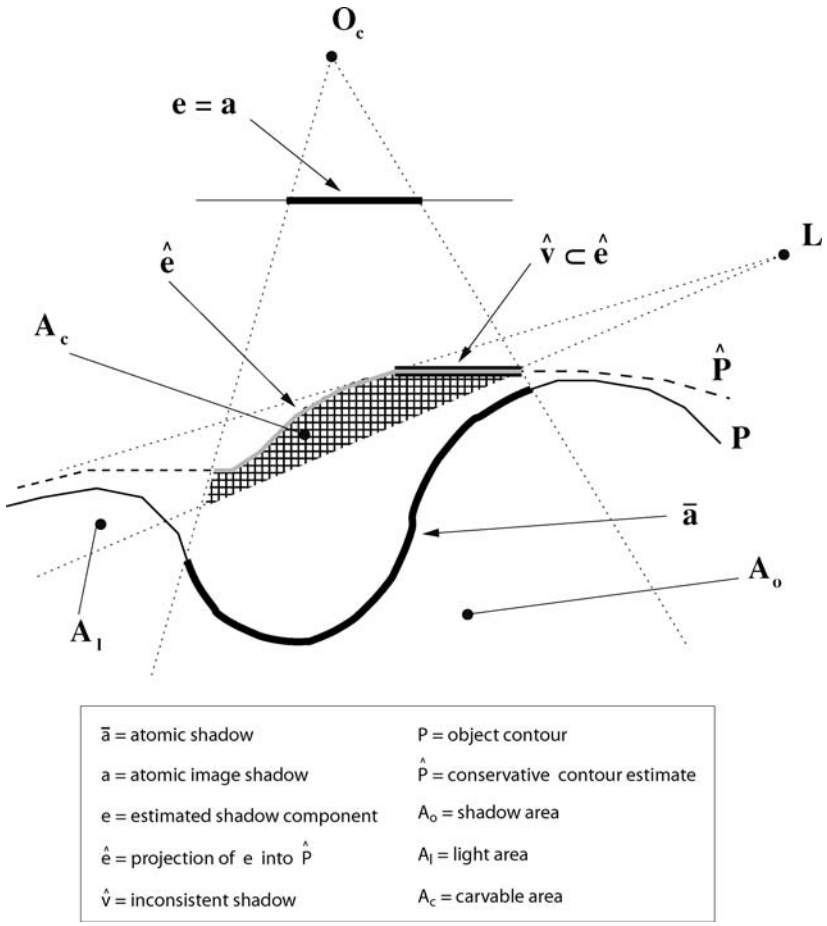

Figure 10. The atomic shadow case.

Proof: Suppose, by contradiction, that there is a portion q of the object contour P within A_C. There are two possible cases: q is fully included within the atomic shadow $\bar{a}$ (case 1); q is not fully included within $\bar{a}$ (case 2). Let us start with case 1. It is easy to conclude that, since A_C is defined by $\hat{v}$ (which is fully visible from L), there must exist at least one point $p \in q \subseteq \bar{a}$ visible from L. The contradiction follows from the definition of shadow. Consider case 2: let q_p be a portion of q not included in $\bar{a}$. Since $\bar{a}$ is atomic, by Lemma 3.2, q_p must eventually be occluded by a portion $\bar{a}_p$ of $\bar{a}$. But since $\hat{e}$ is closer to O_c than $\bar{a}$ and since q_p belongs to A_C, $\bar{a}_p$ must lie within A_C. Hence we can apply case 1 to $\bar{a}_p$ achieving again the contradiction. The lemma is fully proved. □

Lemma 3.4. *The carvable area A_C cannot completely lie within the object.*

Proof: The lemma holds because, by definition, $\hat{e}$ is closer to O_c then $\bar{a}$. □

Proposition 3.1. *Any point within a carvable area cannot belong to the actual object area.*

Proof: The proposition follows from Lemmas 3.3 and 3.4. □

3.6. Composite Shadow

The composite shadow case arises when e is not necessarily attached to a single atomic shadow (see Fig. 11). Let us assume that e is actually composed by the union of J atomic image shadows $a_1, a_2, \ldots a_J$. The corresponding atomic shadows lying over the object contour are $\bar{a}_1, \bar{a}_2, \ldots \bar{a}_J$, respectively. The area A_o is defined by e as in the atomic case. Lemma 3.2 can be modified as follows:

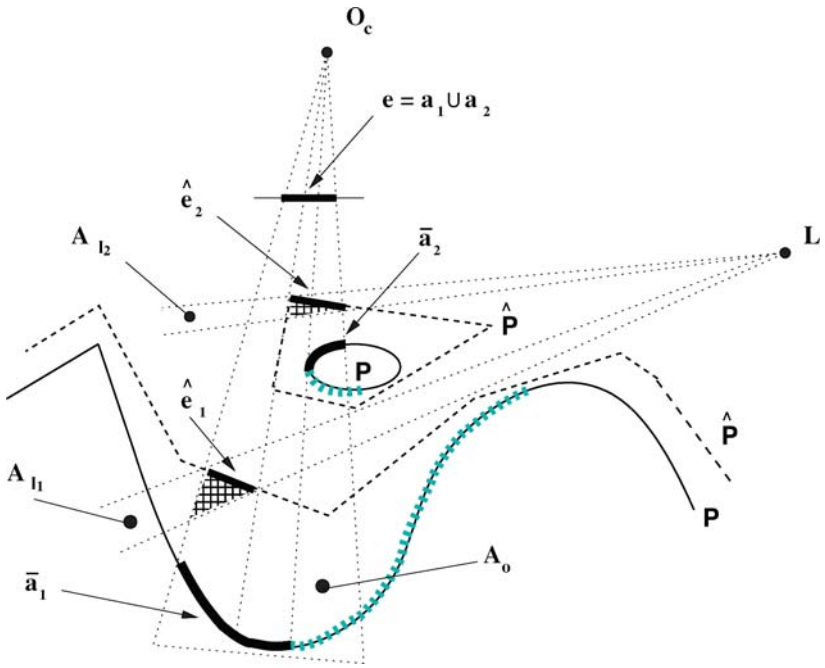

Figure 11. The composite shadow case.

Lemma 3.5. *Given an estimated shadow component e and the corresponding shadow area A_o, all the points belonging to P and within A_o either belong to one of the atomic shadows $\bar{a}_1, \bar{a}_2, \ldots \bar{a}_J$ or they are not visible from O_c. In particular, if there is a point $p \in P$ and $\in A_o$ not visible from O_c, there must exist at least one point $q \in \{\bar{a}_1, \bar{a}_2, \ldots \bar{a}_J\}$ between p and O_c.*

Proof: The lemma directly follows from the definitions of atomic shadows, Lemma 3.1 and the assumption that e is a connected shadow component. □

We can define $\hat{e}$ as in the atomic case. The difference here is that $\hat{e}$ may be decomposed into K different components $\hat{e}_1 \ldots \hat{e}_K$ located in different positions within A_o, depending on the topology of $\hat{P}$. For each k, we can define $\hat{v}_k$ as the portion of $\hat{e}_k$ whose points can be connected by a line segment to L without intersecting $\hat{P}$. Furthermore, for each $\hat{v}_k$ we define a corresponding A_{l_k} as in the atomic case.

Finally:

Definition 3.10. We define the *set of carvable areas* $A_{C_1}, A_{C_2}, \ldots A_{C_K}$ attached to the estimated shadow component e as the set of areas obtained by the intersection among the estimated object area (bounded by $\hat{P}$), the shadow area A_o and the set of light areas $A_{l_1}, A_{l_2}, \ldots A_{l_K}$, respectively.

Proposition 3.2. *Any point within the set of carvable areas $A_{C_1}, A_{C_2}, \ldots A_{C_K}$ cannot belong to the actual object area.*

Proof: The proposition follows from Lemma 3.5 and by modifying accordingly Lemma 3.3, Lemma 3.4 and the corresponding quantities. □

Propositions 3.1 and 3.2 guarantee that points taken from any of the carvable areas, are always outside the real object area. The extension to the $3D$ case is immediate by observing that: (i) the epipolar slice can sweep the entire object's volume; (ii) the interplay of light and geometry take place within each slice and is independent of the other slices. Thus, Theorem 3.1 is fully proven.

Notice that Theorem 3.1 holds under the hypothesis of Lemma 3.1: (i) the actual object's contour is smooth (or a technique is available to identify points whose first order derivative is not continuous—see Section 3.4); (ii) the shadow detection technique is *conservative*—

that is, a shadow may not be detected but whatever is labeled as shadow is indeed a shadow.

3.7. Effect of Errors in the Shadow Estimate

What happens when errors in the shadow estimation occur? We proved Proposition 3.1 with the assumption that an estimated shadow component is equivalent to an atomic image shadow (or the union of two or more atomic image shadows). In the presence of a conservative error in the shadow estimation, the estimated shadow component is always included in the atomic image shadow. Thus, the corresponding carvable areas will be a subset of those obtained from the full atomic shadow. In the limit case where no shadows are detected, no volume is removed from the object. As a result, we still perform a conservative carving. This property makes our approach robust with respect to conservative errors in identifying shadows.

However, if a non-conservative error in the shadow estimation occurs, the estimated shadow components may not necessarily be included in the corresponding atomic image shadows. As a result, some carvable areas may be contained within the actual object's boundaries. The shadow carving procedure is no longer conservative. In Section 6, we will show an example of non conservative carving due to non conservative errors in the shadow estimate.

As a conclusion, an estimated carvable area is always guaranteed to be outside the actual object boundaries as long as errors in the shadow estimation are conservative and singular points along the object contour are detectable.

4. Shape from Silhouettes and Shadow Carving

In this section, we present the design of a system for recovering shape from shadow carving. The system combines techniques of shape from silhouettes and shadow carving. We first briefly review techniques based on shape from silhouettes in Section 4.1. Then we present our implementation of shape from silhouettes. We describe a new hardware configuration that tackles the difficult problem of extracting the object image silhouette in Section 4.2. The outcome of shape from silhouettes is an initial conservative estimate of the object. We review hybrid approaches in Section 4.3 and show that the initial estimate can be refined by shadow carving in Section 4.4.

4.1. Brief History of Shape-from-Silhouettes

The approach of shape from silhouettes—or shape from contours—has been used for many years. An early shape from silhouette method was presented by Martin and Aggarwal (1983) and has subsequently been refined by many other researchers. The approach relies on the following idea. The silhouette of the object in the image plane and camera location for each view forms a cone containing the object. See Fig. 12. All space outside of this cone must be outside of the object, and the cone represents a conservative estimate of the object shape. By intersecting the cones formed from many different viewpoints, the estimate of object shape can be refined. Different techniques for computing the intersection of cones have been proposed. Martin and Aggarwal considered a volume containing the object and uniformly divided such volume in sub-volumes. For each view, each sub-volume—or voxel—is examined to see if it is outside of the solid formed by the silhouette and the view point. If it is outside, the voxel is excluded from further estimates of the object shape. Subsequent research has improved on the efficiency of this approach by alternative data structures (Szeliski, 1993) for storing the in/out status of voxels. Standard surface extraction methods such as

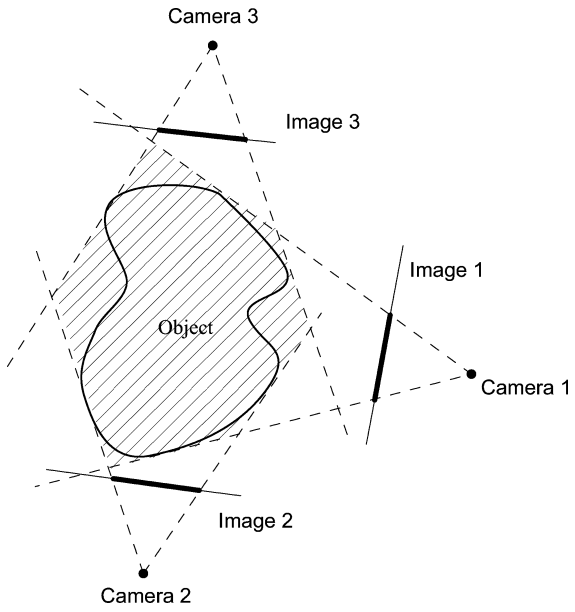

Figure 12. Shape from Silhouettes: The silhouette and camera location for each view forms a cone containing the object. The intersection of multiple cones is a conservative estimate of object shape.

Marching Cubes (Lorensen and Cline, 1987) can be used to compute the final triangle mesh surface. It is also possible to model the cones directly as space enclosed by polygonal surfaces and intersect the surfaces to refine the object estimate, similar to the method used by Reed and Allen to merge range images (Reed and Allen, 1999).

Laurentini (1995) introduced the idea of *visual hull*, i.e., the closest approximation to the object surface that can be estimated from n views. He also showed that the accuracy of a shape from silhouettes approach is limited. For instance, some concavities in the object are never observed from the silhouettes. Koenderink (1984) studied the differential properties of occluding contours and the relationship between contours and local shape. Such analysis was extended by Giblin and Weiss (1986), and Vaillant and Faugeras (1992). In particular, the idea of using a series of silhouettes obtained from very small angular rotations of the object was explored. As illustrated by Cipolla and Blake (1992), and Zheng (1994), the depth of a point on the silhouette can be computed by analyzing the change in silhouette for a small change in orientation. The set of $3D$ points obtained from many views form a cloud that can be integrated into a single surface mesh. The method presented by Zheng has the advantage of the unobserved areas that have unmeasured concavities being identified automatically. It has the disadvantage that many more silhouettes are required to estimate the object shape. The method is also not as robust as the cone intersection methods, because reconstruction errors inherent in meshing noisy points may result in holes or gaps in the surface. Major recent contributions on shape from silhouettes are Lazebnik et al. (2001), Fitzgibbon et al. (1998), Boyer and Berger (1997) and Sullivan and Ponce (1998).

Because they are robust and conservative, volume-based space carving techniques similar to Martin and Aggrawal's original method have found success in low-end commercial scanners. The shape error caused by concavities not apparent in the silhouettes is often successfully masked by the use of color texture maps on the estimated geometry.

Although they are simple and relatively robust, shape from silhouettes approaches fail when they are unable to accurately segment the object from its background. Many systems use a backdrop of a solid, known color—i.e., the backdrops used in chroma-key systems for video compositing. This approach can fail when the object itself has the same color as the background. More frequently, it fails for objects with some specularity that reflect the backdrop into the direction of the camera view. Diffuse white objects may also reflect the color of the backdrop towards the camera through multiple self-inter-reflections. This reflection of the backdrop color can cause the segmentation to fail in two ways. The object boundary may be estimated as entirely inside the actual boundary, resulting in a general shrinkage. Areas in the middle of the object may be classified as backdrop. This results in the more serious error of tunnels in the object. A simple approach to correct tunnelling errors is to have the user inspect the images being segmented, and paint in areas that have been misclassified. Another approach is to use a large, diffusely emitting light source as the backdrop. This can often prevent areas in the middle of the object from being misclassified, but does not guarantee that areas near the silhouette edge that scatter light forward will be properly classified. It also prevents texture images from being acquired simultaneously with the silhouette images.

Recently, Leibe et al. (2000) developed a shape from silhouettes approach that avoids the segmentation problem by using cast shadows. A ring of overhead emitters cast shadows of an object sitting on a translucent table. A camera located under the table records the shadow images. The emitter positions and shadows form the cones that are intersected to form a crude object representation. Because only one object pose can be used, the representation cannot be refined. The crude representation is adequate, however, for the remote collaboration application being addressed by Liebe et al..

To avoid the segmentation problem inherent in many shape from silhouette systems, we adopt an approach similar to that of Leibe et al. (2000). We rearrange the set up, however, to allow for multiple object poses and better refinement of the object shape.

4.2. First Phase—Shape from Silhouettes

Our proposed new set up for shape from silhouettes is shown in Fig. 13. A point light source is placed in front of the object to be measured sitting on a turntable, casting a shadow on a translucent panel. A camera on the opposite side of the panel records the image of the shadow cast on the panel. The locations of the camera, light source, and panel relative to a coordinate system fixed to the turnable are found by calibration. In order

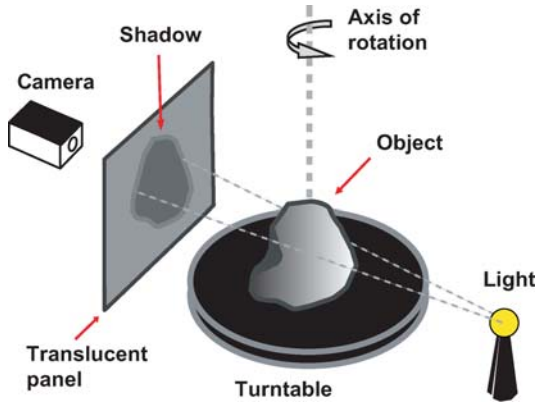

Figure 13. A novel setup for shape from silhouettes: A camera observes the shadow cast by a point light source on a translucent panel.

to be considered a "point" light source, the lamp simply needs to be an order of magnitude or more smaller than the object to be measured, so that the shadow that is cast is sharp. The lamp needs to have an even output, so that it does not cast patterns of light and dark that could be mistaken for shadows. The translucent panel is any thin, diffusely transmitting material. The panel is thin to eliminate significant scattering in the plane of the panel (which would make the shadow fuzzy) and has a forward scattering distribution that is nearly uniform for light incident on the panel, so that no images are formed on the camera side of the panel except for the shadow. The positions of the light source and camera are determined by making sure that the shadow of the object falls completely within the boundaries of the translucent panel for all object positions as the turntable revolves, and that the camera views the complete translucent panel.

By using a translucent panel, the camera views an image that is easily separated (e.g., by performing a k-means clustering) into two regions to determine the boundary intensity value between lit and unlit areas. Because the camera and panel positions are known, the shadow boundary can be expressed in the world coordinate system. The cone that fully contains the object is formed by the light source position and the shadow boundary. A volume can be defined that is initially larger than the object. Voxels can be classified as **in** or **out** for each turntable position. This classification can be done by projecting the voxel vertices along a line starting at the point light source onto the plane of the panel and determining whether they are in or out of the observed shadow. Voxels for which the result of this test is mixed are classified as **in**.

A more accurate estimate of the surface can be obtained by computing the actual crossing point for each in-out edge.

By using the projected shadow, problems such as the object having regions the same color as the background, or reflecting the background color into the direction of the camera are eliminated. The configuration also allows for some other interesting variations. Multiple light sources could be used for a single camera and panel position. One approach would be to use red, green and blue sources, casting different color shadows. In one image capture, three shadows could be captured at once. Another approach would be to add a camera in front, and use several light sources. For each turntable position, several shadow images could be captured in sequence by the camera behind the translucent panel, and several images for computing a detailed texture could be captured by the camera in front. See Fig. 14.

Either configuration for using multiple light sources would give information equivalent to using multiple camera views in a traditional shape from silhouettes set-up. The result of this is a more complete object description from each full rotation of the turntable, and a reduction in the number of positions of the object required to obtain a well-defined surface.

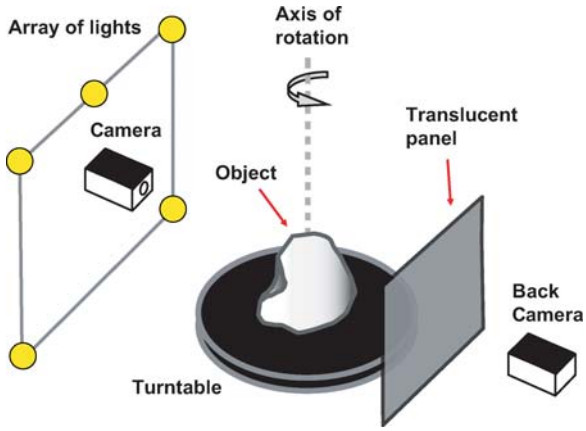

Figure 14. Alternative setup for shape from silhouettes and shape from self-shadowing: shadow images on the translucent panel are obtained for multiple point light sources in front of the object and captured by a camera placed behind the translucent panel. At the same time a camera in front of the object is used to take images of the front of the object and the shadows it casts onto itself from the same light sources.

4.3. Combining Approaches

It has become evident that to produce robust scanning systems it is useful to combine multiple shape-from-X approaches. A system is more robust if a shape estimated from shape-from-A is consistent with shape-from-B. An example is recent work by Kampel et al. (2002) where cues from silhouettes and structured light are combined. An alternative approach is photo-consistency, as introduced by the work of Seitz and Dyer (1997), Kutulakos and Seitz (1999) and numerous extensions and/or alternative formulations (Broadhurst and Cipolla, 2001; Bhotika et al., 2002; Szeliski and Golland, 1999; de Bonet and Viola, 1999; Eisert et al., 1999; Slabaugh et al., 2004; Treuille et al., 2004). The philosophical importance behind this work is summarized by the idea that a surface description is acceptable only if it is consistent with all the images captured of the surface. We use this basic concept in combining shape from silhouettes with shape from shadows. We refine our initial shape estimate obtained from shape form silhouettes with the conservative removal of material to generate shapes that are consistent with images of the object that exhibit self-shadowing.

4.4. Second Phase—Shadow Carving

As discussed in Section 4.1, shape from silhouettes cannot capture the shape of some concave areas which never appear in object silhouettes. The new configuration described in Section 4.2 does not overcome this limitation. Object concavities, however, are revealed from shadows the object casts on itself. In our second phase of processing, we analyze the object's self-shadowing and use shadow carving to adjust our surface estimated in the first phase.

Our implementation of shadow carving is organized in three parts: shadow detection, check for contradiction, and adjustment to resolve contradiction. First of all, we need to obtain multiple images with potential self-shadowing. We propose a hardware set up as shown in Fig. 1. A camera in front of the object is used to take images of the object and the shadows it casts onto itself from an array of light sources. We use these images to refine the object estimate. An alternative hardware configuration is shown in Fig. 14. Shadow images on the translucent panel are obtained for an array of light sources located in front of the object and captured by a camera placed behind the translucent panel. At the same time, a camera in front of the object is used to take images of the object and the shadows it casts onto itself from the same light sources. While all of the front and back images are taken in the same rotation of the turntable, the translucent panel images are processed first to obtain a first surface estimate (shape from silhouettes). We use the images obtained by the front camera in a second phase to refine this estimate (shadow carving).

Detecting shadows is not easy. In objects that have significant self-inter-reflection and/or spatially varying surface albedo, lit portions of the object often have low intensities—sometimes even lower than those attached to portions that do not have a direct view of the light source—but are lit by inter-reflection. A full solution to the shadow detection problem is not the objective of this work. In fact, it is permissible with our approach to avoid finding the exact shadow boundaries—we can misclassify shadow pixels as lit without jeopardizing the conservative property of our estimate. See Section 3.4. Our goal then is to identify areas we are very certain to be in shadow region. At that end we make use of multiple light sources for each object position. We combine the images captured with these different light positions into a unique reference image by taking the maximum value for each pixel. Each image is then analyzed for areas that are dark, relative to the reference image, and also have an intrinsic low value of intensity. We select a threshold value that gives an estimation safely within the dark region for all of the images. The rest of our method is designed to make use of these conservative shadow estimates to continue to refine our initially conservative object estimate.

In the next step, we test whether the shadow regions thus identified can be explained by the current object estimate (check for contradiction). Notice that not all shadows observed on an object indicate that the initial surface estimate is incorrect. Consider the case of a coffee mug shown in Fig. 15. Two shadows would be observed: the attached shadow B and the shadow cast by the handle C. Ray-tracing would show that both of these shadows are explained by the current surface. A ray to the light source in the attached area B would immediately enter the object itself, indicating the light source is not seen. A ray from area C would intersect the handle before reaching the source.

The problem remains of what to do to resolve unexplained shadows (adjustment to resolve contradiction). The surface estimate is conservative, so we cannot add material to the object to block the light source. We can

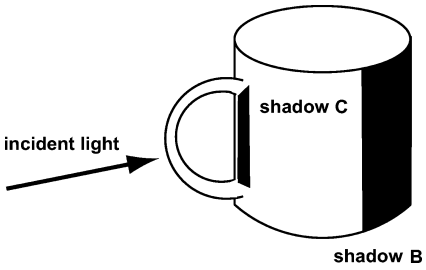

Figure 15. Not all shadows on an object indicate a contradiction with the current object estimate. For the mug, the shadow C cast by the handle and the attached shadow B are both explained by the current object estimate.

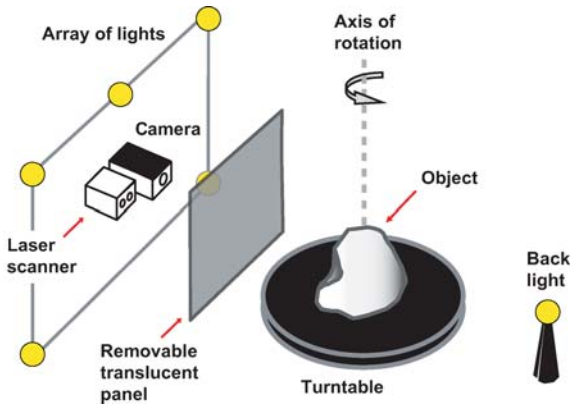

Figure 16. Setup of our proposed system. The system is composed of a turntable, a removable translucent plane, a single 150 W halogen light source on the back of the object. Behind the movable panel, a rigid panel contains a digital camera, an array of five halogen light sources surrounding the camera and a laser scanner. Notice that the laser scanner is only needed to acquire a ground-truth model of the object and is not required for our reconstruction algorithm.

only remove material. Removing material anywhere outside of the unexplained shadow will not form a block to the light source. The only option is to remove material from the unexplained shadow region. This can be done to the extent that the surface in that region is pushed back to the point that its view of the light source is blocked as explained in Theorem 3.1.

As a final remark, in order for shadow carving to work, we assume that there are no features smaller than our image pixel size. See Section 3.4 for details. As noted in Raviv et al. (1989), and Hambrick et al. (1987), this interpretation of shadow fails when there are features sharper than the image resolution.

5. Implementation

We have built a table-top system to test our approach. Our modeling software has two major components: space carving using the silhouette images and shadow carving using the images acquired with an array of light sources.

5.1. Hardware Setup

Our table-top setup is shown in Fig. 16. The system is composed of a *Kaidan MC-3* turntable, a removable translucent plane, a single 150 W halogen light source on the back of the object. Behind the movable panel, a rigid panel contains a *Fuji FinePix S1 Pro* digital camera, an array of five halogen light sources surrounding the camera and a *ShapeGrabber*^{TM} laser scanner which was used to obtain a ground truth. We capture 1536×2304 resolution images and crop out the central 1152×1728 region from each image. Control software allows us to automatically capture a series of N steps of images in rotational increments of $360/N$ degrees and turn the light sources on and off when necessary. The single halogen light source on the back of the object is used to project the shadow on the translucent plane. The translucent plane can be removed when necessary. It is made of frosted glass standing on a rigid support. A sheet of copier paper sandwiched between panes of glass can be used as well. For additional information about the setup and how to obtain the ground truth see (Farouk et al., 2003).

The camera is calibrated with respect to a coordinate system fixed to the turntable using a calibration artifact attached to the turntable and a Tsai camera calibration algorithm (Tsai, 1987). The location of laser scanner is calculated by scanning the artifact used to calibrate the camera. The location of the translucent panel and the position of the back light source are measured using the laser scanner. This is done for convenience—alternative calibration algorithms may be used as well. Since the panel surrounding the camera has known geometry, the location of the array of five light sources is known as well.

Because we use a single camera system, we need to take data in two full rotations of the turntable. In the first rotation, the translucent panel is in place and the silhouette images are obtained. These data are used by the space carving algorithm. In the second rotation of the turntable, the panel is removed without moving the object, and the series of five images from the five

Figure 17. Sample data from table-top system: Shadow panel image on the left, and self-shadow image on the right.

camera-side light sources are captured. These data are used by the shadow carving algorithm.[2]

Cropped images from the sample data acquired by our system are shown in Fig. 17. The left image shows the shadow cast by the object on the translucent panel when it is lit by the back light. The right image shows one of the five self-shadowing images captured for one object position. Even if the original images are captured in color, shadow carving method only requires gray scale. Color images can be used if a final texture mapping step is needed.

5.2. *Software—Space Carving*

Our goal is to recover the shape of the object from the silhouettes of the shadows cast into the translucent panel. Thus, the first step is to extract such shadow silhouettes. This task is very simple. We begin by doing k-means analysis on the images to determine the pixel intensity value dividing the lit and unlit regions for the images of the panel. The boundary of each shadow is then computed with sub-pixel accuracy. The estimation of the shadows is slightly dilated to account for errors in the calibration of the camera (e.g., errors in estimating the camera center and back light position).

After that the silhouettes are extracted, the space carving algorithm is organized as follows. A single volume that completely encloses the working space of the acquisition system is defined. In order to accelerate the computation, an octree scheme is used as proposed by Szeliski (1993), and Wong (2001). The initial volume is subdivided in eight parts (voxels), each one marked as containing the surface of the object (**boundary** label). Each **boundary** voxel is projected into every shadow image. If there is at least one projection in which the voxel lies completely outside a shadow image than the corresponding voxel is marked as being outside the object (**out** label). If all the projections of a voxel lie completely inside the corresponding shadow images then the voxel is classified as being inside the object (labeled as **in**).

If some projections of a voxel partially overlap the shadow edges and some are fully inside other shadows, then the voxel keeps the **boundary** label. After all the voxels are inspected, those labeled **out** are discarded, those labeled **in** are kept but processed no further, and each **boundary** voxel is subdivided into eight new sub-voxels. All of the new voxels are processed as before to improve the estimate of the object volume.

When a suitable subdivision level is reached, each vertex of each **boundary** voxel is projected into the shadow images and classified as being **in** or **out** of the object. Moreover, all of the voxels that were labeled as **in** are subdivided to the finest level, and their vertices are labeled as **in**. Finally, all of the **boundary** and **in** voxels are saved to a file that provides the volume description used for the shadow carving algorithm.

5.3. *Software—Shadow Carving*

Our implementation of the shadow carving algorithm consists of three steps: shadow detection, test for contradiction, and surface update. These three operations are performed for each camera position, with the cumulative results for all positions $k-1$ carried forward to the processing of position k. The pseudocode of the algorithm is in Fig. 18. At the end of the processing for the $k-1$ position, the object is represented in two ways: as a voxel data structure, that describes its volume, and as a triangle mesh, that describes its surface and

Algorithm Shadow Carving

```
1. for each camera view
2.     Threshold shadow images using reference image;
3.     Render depth buffer using current surface;
4.     for each shadow image;
5.         Cast rays from each shadow pixel
               and mark contradictions;
6.         Update depths for pixels in contradiction;
7.     Update volume data using adjusted depth map;
8.     Extract new surface estimate for full object;
```

Figure 18. Skeleton of the shadow carving algorithm.

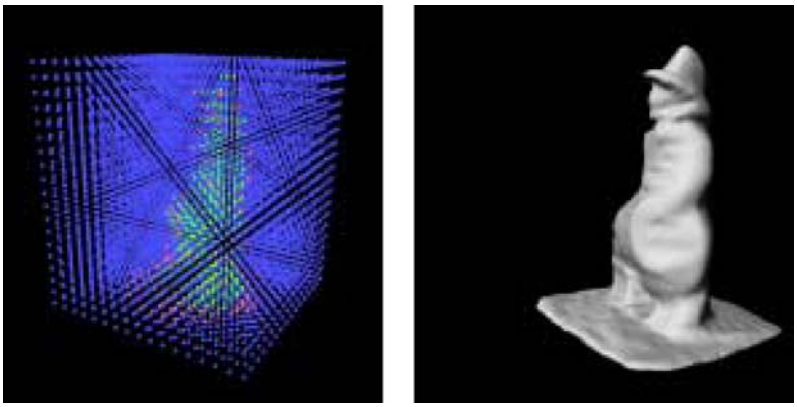

Figure 19. At the start of a shadow carving step, the current object estimation is represented by both a volumetric description (left), and a triangle mesh (right). The voxels of the volume are marked as inside (green), boundary (red), and outside (blue).

is extracted from the current volume using the Marching Cubes algorithm (Lorensen and Cline, 1987). The voxel data structure maintains information for classifying voxels as completely inside, completely outside, or containing the surface of the object (see Fig. 19). We have implemented the algorithm in C++ using the OpenGL library as rendering engine.

Step 1—**Shadow detection.** Because a full solution to the shadow detection problem is not the object of this investigation, we use a simple approach in this implementation. We combine the images captured with five different light positions into a unique reference image by taking the maximum value for each pixel. Each image is then analyzed for areas that are dark relative to the reference image, and also have an intrinsic low value of intensity. We select a threshold value that give an estimation safely within the dark region for all of the images: the goal is to guarantee a conservative estimation of the shadows. Because we acquired objects with uniform albedo, we simply use the same threshold values across all the images of the same object. Examples are shown in Fig. 20.

Step 2—Test for contradiction. The next step is to test for whether points in observed shadow regions are in contradiction with our initial surface or not. We begin by rendering a depth image of our current surface estimate. This is done by simply reading back the depth-buffer after rendering the mesh of triangles that describes the object surface using OpenGL. The depth image has the same resolution as our captured shadow images. An example of depth image is presented in Fig. 21. Higher values of intensity correspond to closer parts of the surface.

We then process each image for each view. For each pixel in the detected shadow region that has a non-background depth, we test if the ray from the corresponding point on the surface to the light source is blocked by the object. We call this test the *contradiction test*. If the ray is not blocked by the object than the shadow pixel is in contradiction and we need to update its depth value. Once that the depth value of every pixel in contradiction is updated, we obtain a new depth image which is consistent with the shadow image.

The algorithm that implements the contradiction test is described in Fig. 22. The algorithm takes as input the current depth map, the shadow image as well as the coordinates of the pixel in shadow we want to test. Call p the point on the surface whose projection into the image plane gives the pixel in shadow we want to test. The point p is moved toward the light and projected into image plane (first while loop). This loop is repeated until the projection of p into the image plane falls outside the shadow region. Then the pixel corresponding to this projection is saved, and the algorithm keeps on moving p toward the light (second while loop). If p reaches the light without intersecting the object—i.e., the depth value of the projection of p is always smaller than the corresponding value of the depth map—then there is a contradiction. [3] The algorithm returns the saved value of the last pixel in shadow and a flag value indicating that the depth map can be updated. If p intersects the

Figure 20. In the first step of shadow carving, shadows are identified in a set of four images all taken from the same viewpoint, but with different light positions (first 4 images in the top row). The reference image is obtained by combining these 4 images into a new one by taking the maximum value for each pixel (right most image). The areas classified as shadow are superimposed in white over the original images (bottom row); the original intensity of the images is modified to highlight the detected shadows.

Figure 21. The surface estimate from the $k-1$ step is rendered as a depth image from the same viewpoint as the shadow images from the k step. The background pixels, in black, are set to the furthest possible depth value.

Algorithm Contradiction Test

```
1.   Initialize the point p to be the point whose
         projection in the image plane
         is the pixel in shadow we want test;
2.   while the projection of p is inside the shadow
3.       Move p along the line that connects p to the light;
4.       Project p into the image;
5.   Save the coordinate of the last projection;
6.   while p is not coincident with the light
7.       Move p along the line that connects p to the light;
8.       Project p on the image and get its depth d;
9.       if d is greater than the corresponding value
                 of the depth image
10.          return a non-contradiction flag;
11.  return a contradiction flag;
```

Figure 22. Contradiction test algorithm.

object then there is no contradiction and no need to update the depth map for the investigated pixel.

Step 3—Surface update. To resolve a pixel in contradiction for a given shadow region, the pixel's height in the depth map is adjusted according to the following algorithm. Consider the $2D$ slice shown in Fig. 23. Call *first unexplained pixel* the last pixel in shadow returned by the contradiction test. Call q its projection into the surface. The depth of each pixel in contradiction is then adjusted along the line of sight from the current camera view point and increased to the point it reaches the ray from the light source through q. This step can be repeated for each pixel in contradiction for a given

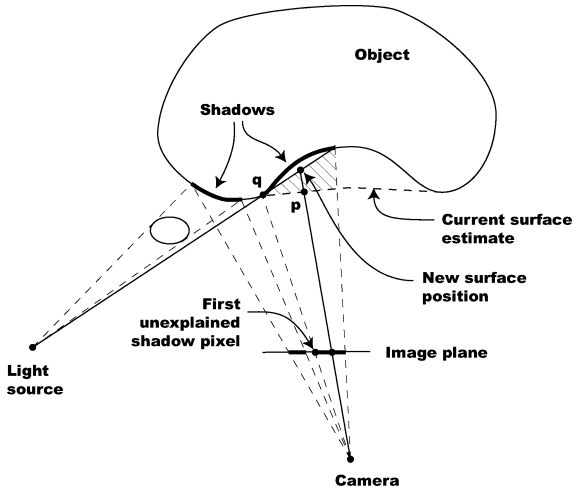

Figure 23. Adjustment of the position of a surface point when a contradiction is detected.

shadow region. A new depth map is thus produced. The difference between the new and the old depth map corresponds (in the $2D$ slice) to the carvable areas introduced in Section 3. Proposition 3.1 guarantees that the depth map adjustment is always conservative.

After resolving all the pixels in contradiction for each shadow region for a given shadow image we obtain a new depth image. The new depth image is used to process the next shadow image acquired from the same viewpoint. The final depth map for the current camera view represents the adjustments for all shadow images from that viewpoint.

The last step for a camera view is to update the volume representation of the surface, so that a new full object representation can be extracted to carry forward to the next view. We test the voxel vertices to see if they lie in front of or behind the depth map surface and update the labeling of vertices which have been carved out. We use these updated values in the Marching Cubes processing to produce a new full surface representation. The surface representation can be rendered into a new depth map for the next camera view.

5.4. Software—Post Processing of the Surface

The final surface may be affected by small artifacts created during the reconstruction. These artifacts may be removed through a final postprocessing step. The post-processing is carried out by low-pass filtering the surface using the algorithm proposed by Taubin (1995) and Taubin et al. (1996). The filter parameters are chosen in order to remove the artifacts while preserving the geometry and the features of the model. In Section 6.1, we will address this issue in more details.

6. Experimental Results

We evaluate the performance of the space and shadow carving algorithms with both synthetic and real objects. As we shall discuss in Section 6.1, a commercial rendering package is used to simulate our system. This allows to test the correctness of the algorithms under a controlled environment, evaluate the performance with different configurations of lights and camera positions, and assess the accuracy of the reconstruction due to errors in the shadow estimates. In Section 6.2, we show examples of reconstruction of real world objects.

6.1. Experiments with Synthetic Objects

We have simulated our acquisition system discussed in Section 5 with a commercial rendering package, *3ds Max 5.1*. The advantage of such experimental methodology is that an exact ground-truth is available for all of the geometrical quantities involved in the reconstruction process: the shape of the object we wish to scan, the object's silhouettes and the self-shadows. Thus, errors due to the calibration and misclassification of shadow boundaries are ruled out. Since the virtual setup is more flexible and the experiments faster to run, such tests are also useful to study the effects of the light sources and camera configuration over the quality of the final reconstruction.

The simulated setup is similar to the scheme depicted in Fig. 16. Two different configurations are tested: the number of camera views (or, equivalently, the number of rotation steps of the turntable) are either 24 or 72; the number of light sources used to cast self-shadows are either 4 or 8. The configuration with 72 views corresponds to an angular rotation step of 5 degrees. Every image is rendered at a resolution of 640×480 pixels.

The configuration with 24 camera views and 4 lights is similar to the one in Section 5. Notice that since each silhouette needed for space carving can be easily extracted from a rendered image of the object, the translucent board to help silhouettes segmentation becomes superfluous. Finally, the shadow detection step

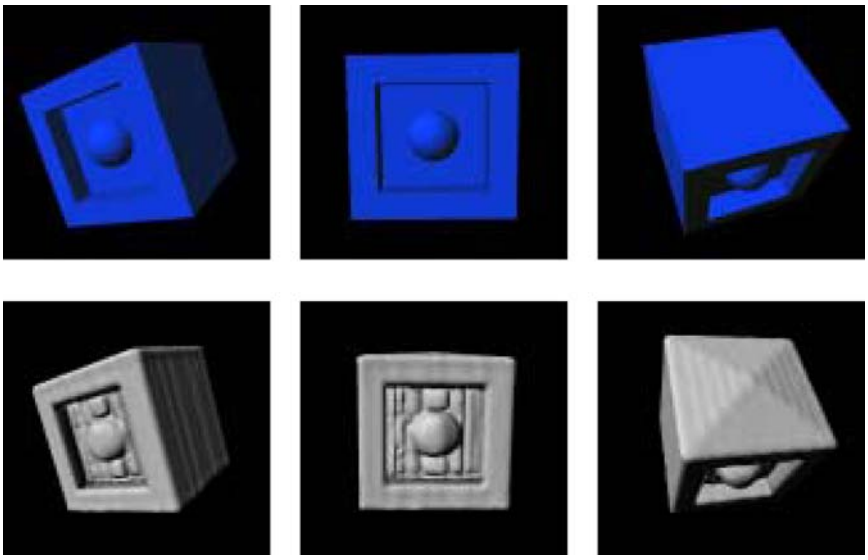

Figure 24. Top: Three views of a synthetic cube. Bottom: Results of shadow carving. 24 views and 4 lights are used in this experiment.

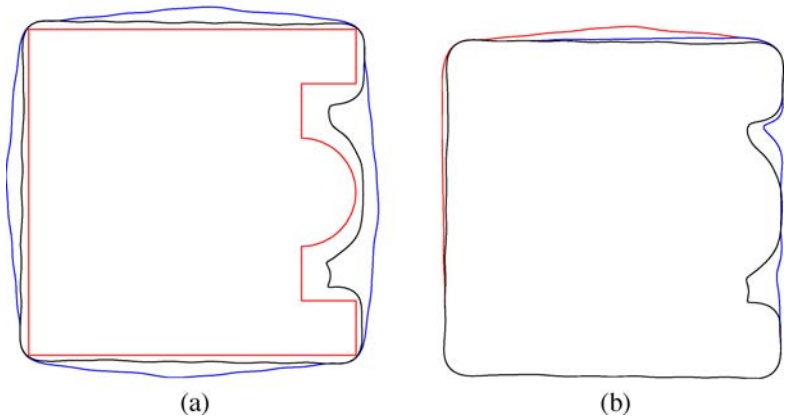

Figure 25. (a) Cross-sections of the real object (red); cross-sections of the result of space carving (blue); cross-sections of the result of shadow carving (black). (b) The progression of the approximations as we increased the number of viewpoints used for the reconstruction: red (8 viewpoints), blue (16) and black (24).

(Section 5.3) is unnecessary since the ray-tracing algorithm allows to identify correctly all of the object's self-shadows in the images.

The first synthetic object used for the experiments is a simple cube with a cavity on one side, as shown in the first row of Fig. 24. The object was reconstructed using 24 views and 4 lights. The reconstruction is shown in the second row of Fig. 24. Notice that a portion of volume is removed from the cavity and that the new surface is still a conservative estimate. However, several visual artifacts are clearly visible. This problem stems from the limited number of views and lights used for carving.

In Fig. 25 a cross section of the original object is presented. The cross section is defined by a horizontal plane passing through the center of the object and is depicted in red. The contours showing the estimate obtained by space carving and shadow carving are superposed in blue and black, respectively. Notice that shadow carving yields an improvement both in the concavity (as expected) and along the side planes of the cube.

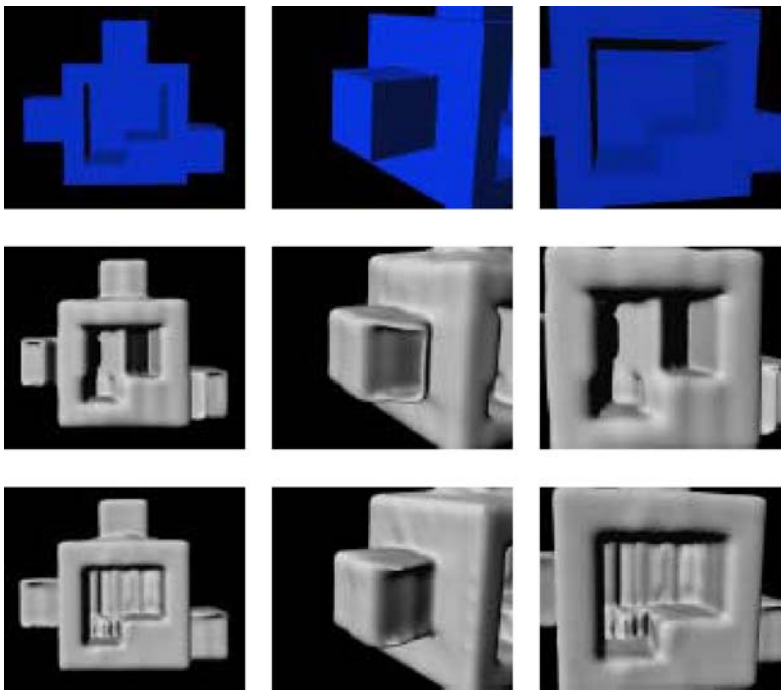

Figure 26. Top row: Three views of the second synthetic object. Central row: Results of shadow carving when 24 viewpoints and 4 lights are used. Bottom row: results of shadow carving when 72 viewpoints and 8 lights are used. Notice that the artifacts (e.g., see central concavity) dwindle away as the number of viewpoints and lights increase.

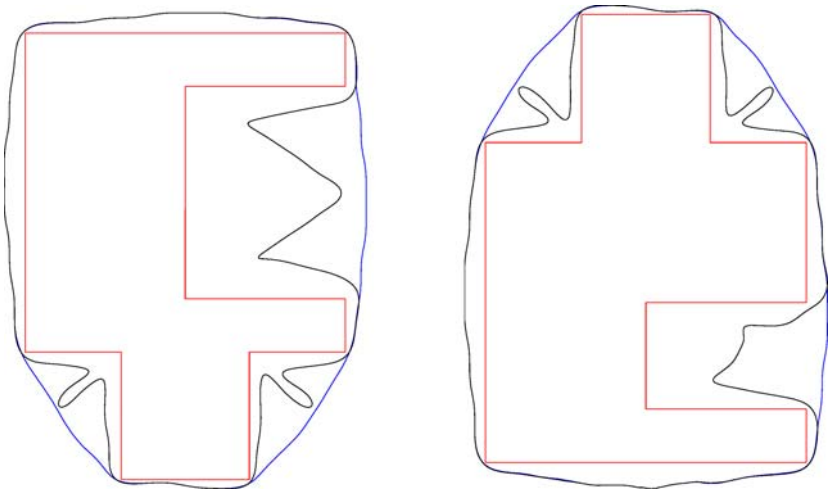

Figure 27. Two cross-sections of the actual object (red); cross-sections of the result of space carving (blue); cross-sections of the result of shadow carving (black). 24 viewpoints and 4 lights are used is this experiment.

In the right side of Fig. 25, sections corresponding to partial improvements produced by shadow carving are presented: the red contour is obtained with 8 views (out of 24), the blue contour is obtained with 16 views (out of 24), the black contour corresponds to the final estimation.

The second synthetic object is a combination of cubes with a non-symmetric cavity on one side and

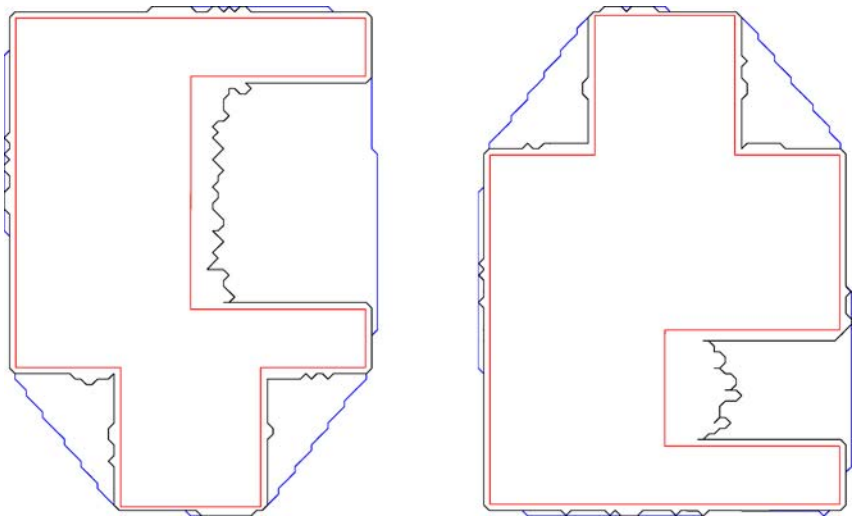

Figure 28. Two cross-sections of the actual object (red); cross-sections of the result of space carving (blue); cross-sections of the result of shadow carving (black). 72 viewpoints and 8 lights are used is this experiment. No post-processing filtering is applied. Since the artifacts' size is smaller the smallest feature of the object, post-processing filtering may be useful to remove the artifacts. See Fig. 29.

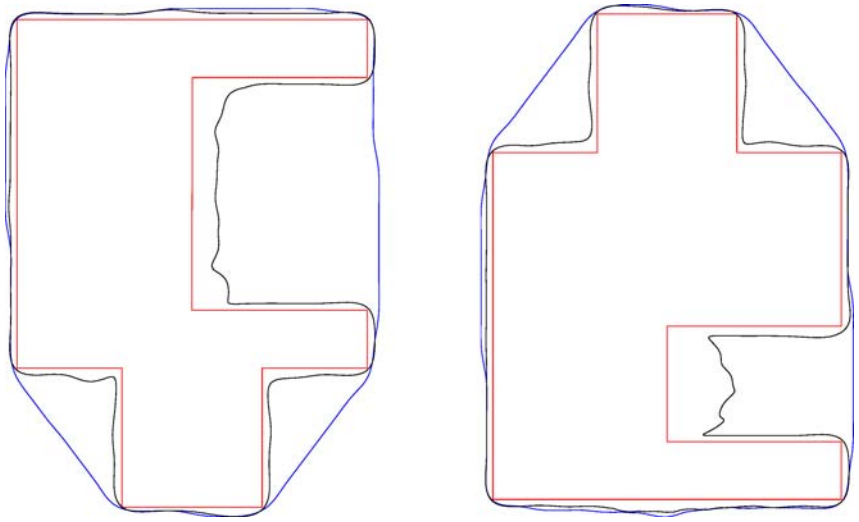

Figure 29. Cross-sections of the object in Fig. 28 after the post-processing filtering.

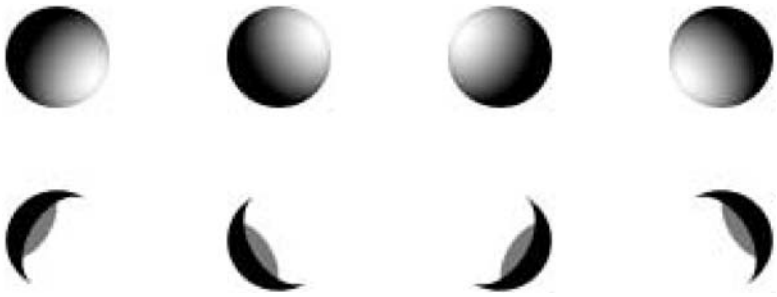

Figure 30. Examples of non-conservative shadow estimates. Upper row: Examples of self-shadows cast over the surface of a sphere. Lower row: The corresponding shadow estimates are shown in black; the lit areas misclassified as shadows are in gray.

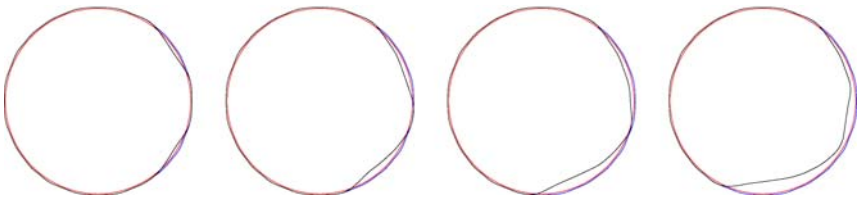

Figure 31. The effect of non-conservative estimate of the shadow.

Figure 32. Left: Captured image of the dog. Center: Results of space carving. Right: Results of shadow carving with 4 lights and 24 views.

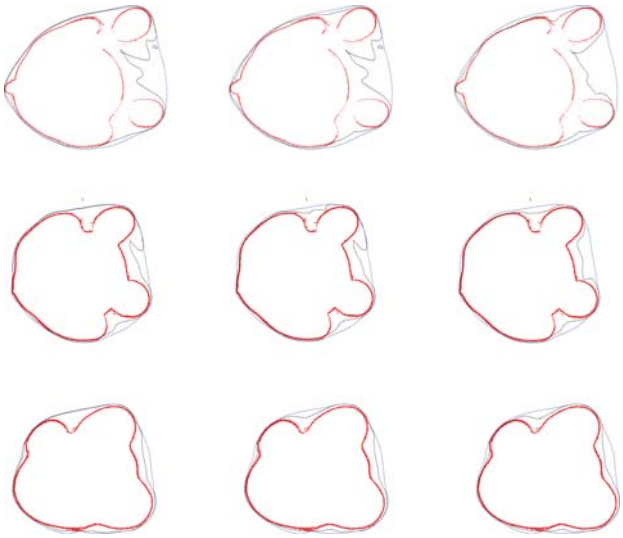

Figure 33. Comparative results: Each row shows a different cross-section of the model of the dog reconstructed with the laser scanner (red), space carving (blue) and shadow carving (black). From left to right: Progression of the approximations as we increased the number of viewpoints used for the reconstruction (8, 16 and 24 viewpoints respectively).

few "extrusions" on the other ones. See the first row of Fig. 26. With only 24 views and 4 lights, shadow carving recovers the concavities only partially as shown in the second row of Fig. 26 and in Fig. 27. Artifacts are left along the two sides of the main body of the object. Notice, however, that the estimation is still conservative and no volume is removed from the actual object.

A more accurate estimate can be obtained by increasing the number views from 24 to 72 and the number of lights from 4 to 8. The results are reported in the last row of Fig. 26. Notice that the volume has been almost completely removed from the cavity and the artifacts are greatly reduced in size and number. The cross sections of the reconstructed object with and without post-processing filtering (see Section 5)

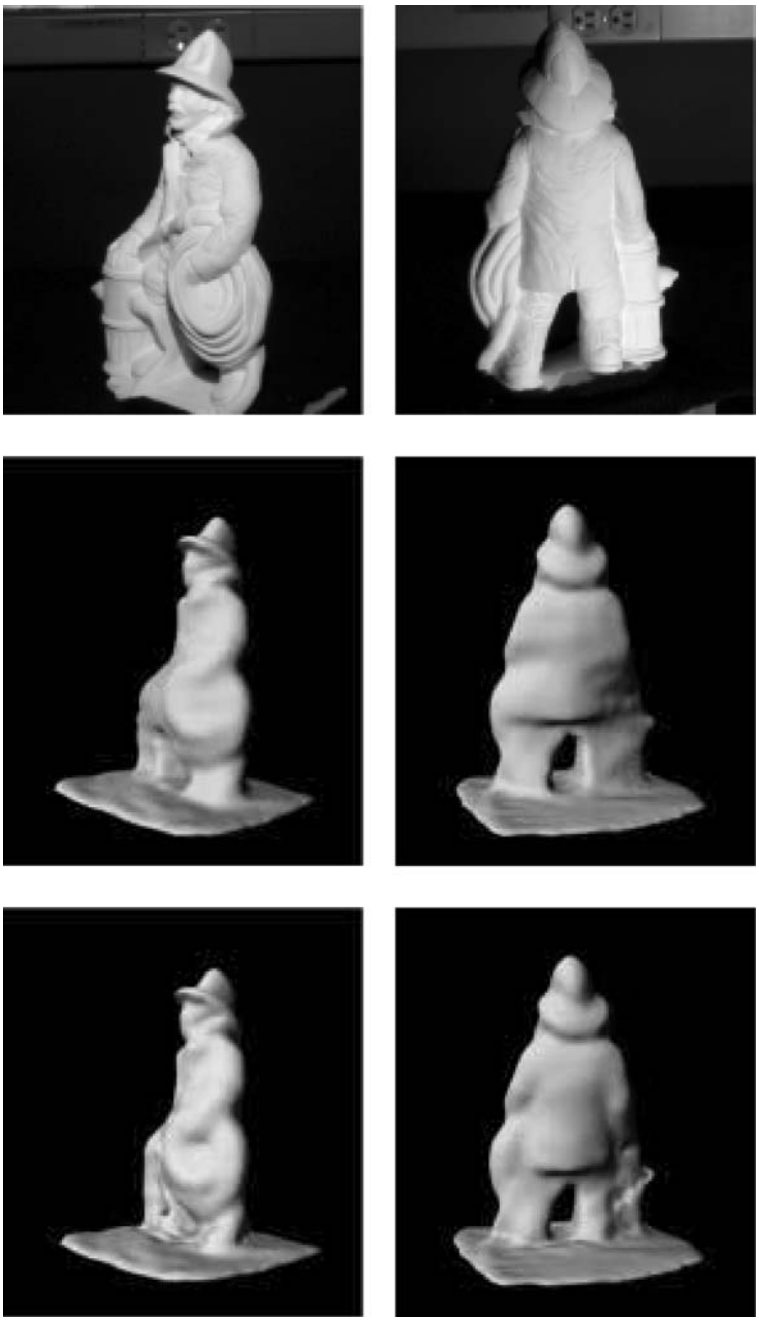

Figure 34. Upper row: Two views of the fireman. Middle row: Results of space carving. Bottom row: Results of shadow carving.

are shown in Figs. 29 and 28 respectively. Notice that since the artifact's scale (size) is smaller than the smallest feature of the object, we can filter out the artifacts while preserving the conservative property of the reconstruction.

The simulated setup is also useful to study the effects of errors in detecting self-shadows. As predicted by our theory, if we violate the hypothesis that any lit pixel is not misclassify as shadow (see Proposition 3.1), the final reconstructed surface is not guaranteed to be

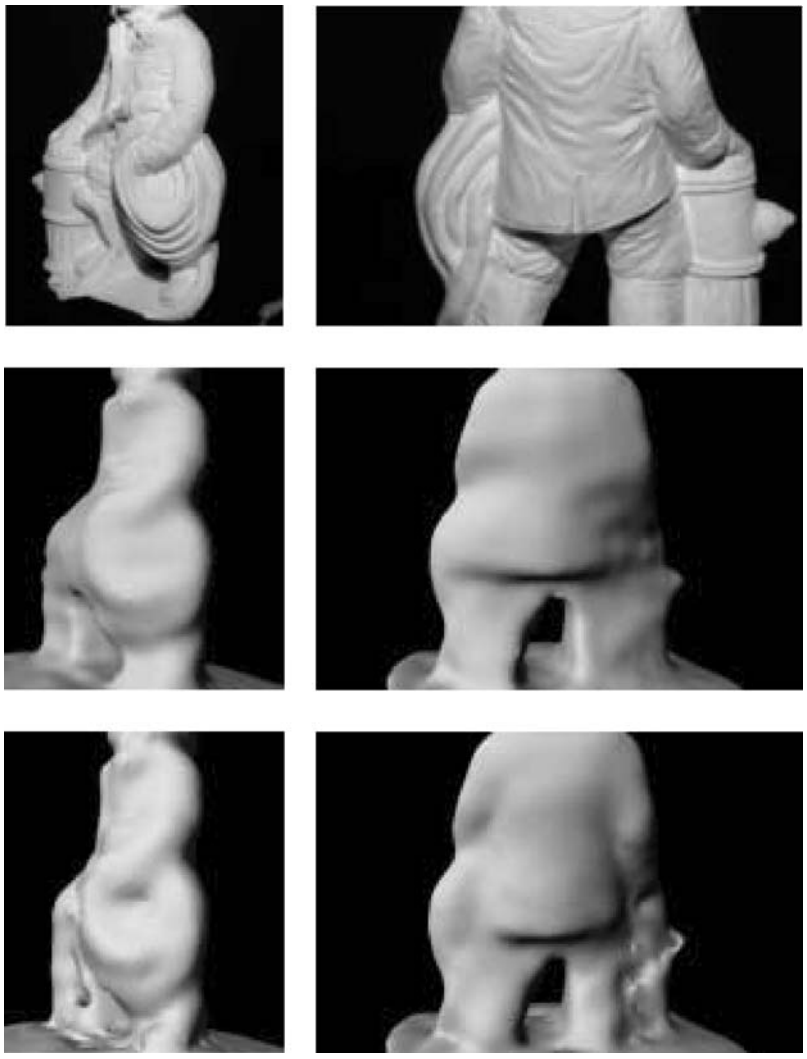

Figure 35. Details of the fireman. Upper row: Two views of the original object. Middle row: Results of space carving. Bottom row: Results of shadow carving.

conservative, i.e., portion of the actual object's volume can be carved out by the shadow carving algorithm.

To test this possibility we increased the size of the estimated shadows. The upper row of Fig. 30 shows examples of self-shadows cast over the surface of a sphere; the lower row shows the corresponding shadow estimates in black. The lit areas misclassified as shadows are in gray. As a result of such a misclassification, the algorithm starts carving out portions of the *actual* volume as shown in Fig. 31, most left panel. As shadow carving keeps on processing the next shadow images, the errors increase more and more. See remaining panels in Fig. 31.

6.2. Experiments with Real Objects

We have tested the carving hardware setup presented in Section 5 and used our shadow carving algorithm to scan several objects. Image resolution was 1152×1728 pixels. Our experiments were designed to test whether conservative estimates, as promised by Theorem 3.1, may be obtained in a practical setting with noisy images and uncertain shadow detections.

Left panel of Fig. 32 shows the first object we reconstructed—a small sculpture of a dog. The central panel shows the results of shape from silhouettes. Notice that the algorithm fails at reconstructing the

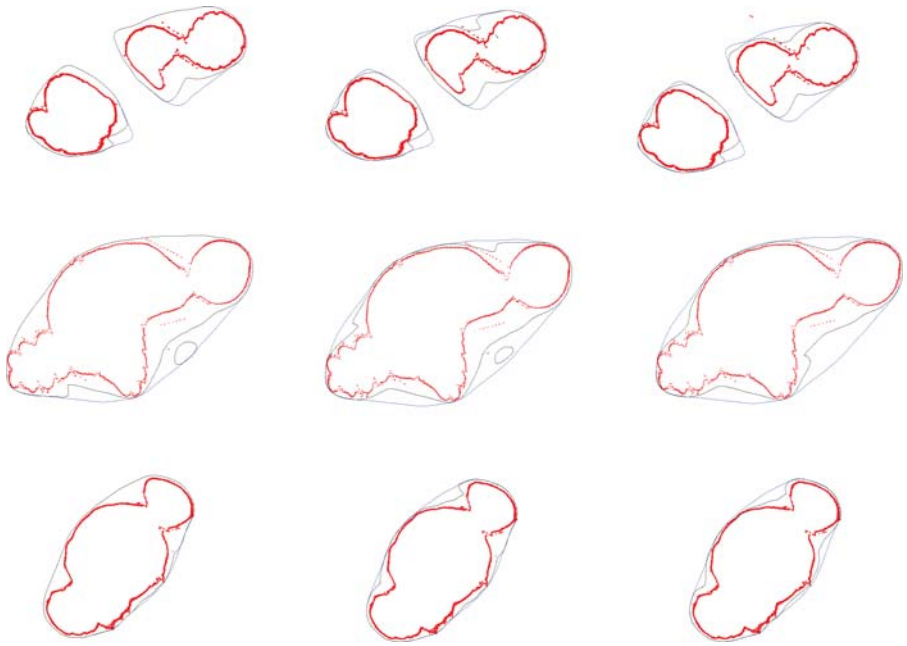

Figure 36. Comparative results: Each row shows a different cross-section of the model of the fireman reconstructed with the laser scanner (red), space carving (blue) and shadow carving (black). From left to right: Progression of the approximations as we increased the number of viewpoints used for the reconstruction (8, 16 and 24 viewpoints respectively).

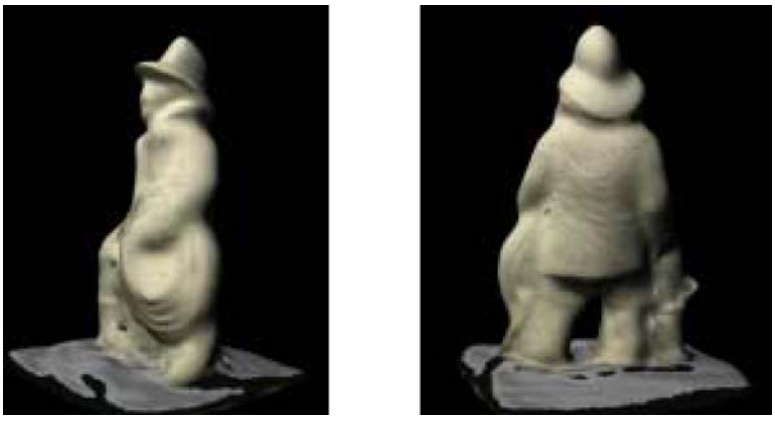

Figure 37. Reconstruction with texture mapping: The object's textures are added to the model captured with shadow carving.

concavities of the object. The right panel shows results of shadow carving. Portions of the concavities are carved out although some artifacts are still visible. These are due to limited number of lights and viewpoints used to carve the model.

To assess the accuracy of our results, we have used the laser scanner to capture the $3D$ shape of the object. Such reconstructed models are visualized as a cloud of points rather than a triangulated mesh. Reconstruction results of space and shadow carving are shown for three different cross sections in Fig. 33.

The second object—a sculpture of a fireman—reconstructed by our carving system is presented in the upper row of Fig. 34. The corresponding results of space carving and shadow carving are shown in the middle and lower rows, respectively. Details of the reconstruction (Fig. 35) highlight the improvement of shadow carving over space carving. The cross-sections

Figure 38. Comparison of the reconstruction with texture mapping. Left: The object's textures are added to the model captured with space carving. Right: The object's textures are added to the model captured with shadow carving.

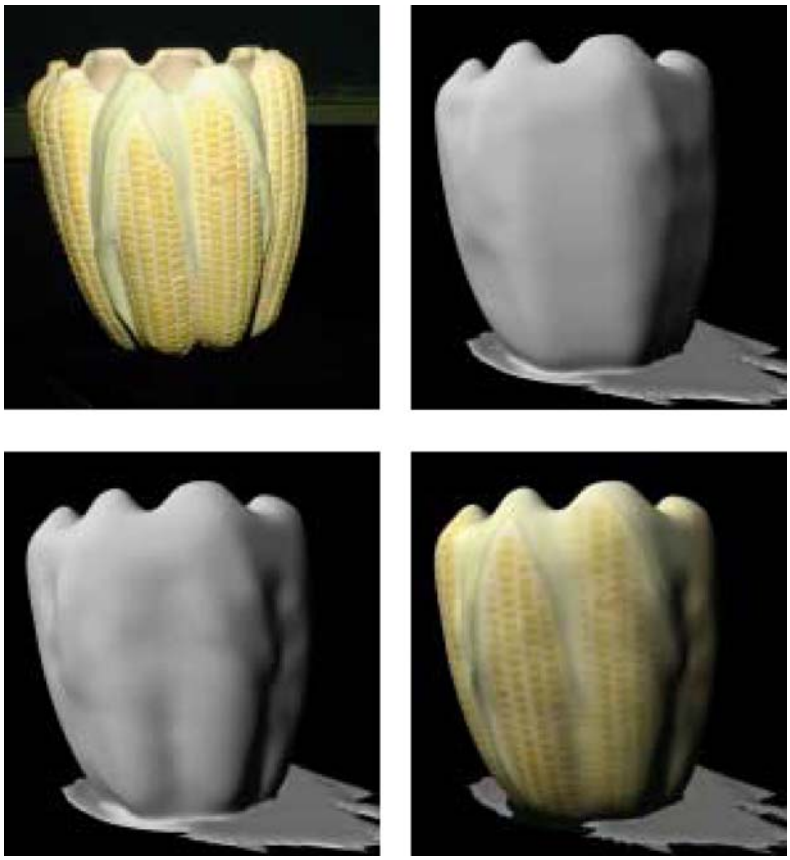

Figure 39. Upper left: Image of the vase. Upper right: Results of space carving. Lower left: Results of shadow carving. Lower right: Textured model.

(Fig. 36 from left to right) show the progressive improvement obtained by shadow carving.

In the left and right panels of Fig. 37 the reconstructed models (obtained by shadow carving) are rendered from two camera views. A texture is superimposed to the models (texture-mapping). Figure 38 shows a comparison between the results of space carving and shadow carving with texture mapping. Notice that the model obtained by shadow carving is visually more compelling than the one by space carving.

A third example of reconstruction is presented in Fig. 39.

6.3. Discussion

Our experiments show that the theoretical framework we proposed and analyzed in Section 3 yields a practical algorithm which can improve upon the well-known space carving methods by carving into concavities of the object. Theorem 3.1 guarantees that the sequence of estimates produced is conservative and our experiments show that this is achievable in practice.

From our experiments it is evident that most concave areas have been carved only partially. This is due to two main reasons. Firstly, we used a small number of views and lights. This was predicted by our experiments with synthetic objects: the reconstruction obtained with 72 views and 8 lights is significantly more accurate than the one with 24 views and 4 lights only. Finding the optimal number and distribution of lights and views is clearly an important open problem. Secondly, our simple and overly conservative shadow detection algorithm reduces the amount of information available to the shadow carving algorithm. The reconstruction in Fig. 39 is an example. Detecting shadows in the vase's concavities is very hard: the edge are smooth, the concavities are brighter due to inter-reflections and some areas have a darker albedo. As a result only few shadows are detected and the shadow carving ability to remove volume is severely impaired. Detecting shadows accurately and yet preserving the conservative property is clearly another important open problem.

7. Conclusions

We presented an analysis of the problem of refining a conservative estimate of an objects shape by observing the shadows on the object when it is lit by a known point light source. We showed that a well defined portion of volume can be removed from the current object estimate. We proved a theorem that guarantees that when this portion is carved away from the shape, the shape still remains conservative. We demonstrated that this insight leads to an algorithm that can work correctly on real images. We called this method *shadow carving*.

Our main assumption is that one can estimate shadows conservatively—i.e., a shadow may not be detected but whatever is labeled as shadow is indeed a shadow. No assumptions about the object topology are necessary. Tangent plane discontinuities over the object's surface have been supposed to be detectable. We showed that shadow carving improves previous work on shape from shadow in that it is more robust with respect to the classification of shadow regions and is not restricted to $2.5D$ terrain surfaces, but rather it may be applied to measuring the objects in the round.

In order to validate our theory, we have implemented a reconstruction system that combines information from silhouettes and shadows. The new system uses inexpensive digital cameras and lamps. Our experiments with real and synthetic objects confirms that the property of conservative carving is achievable in practice and show that shadow carving produces a much better surface estimate than shape from silhouettes alone. This improved estimate is suitable for further refinement by any shape estimation method that works well in local regions of the surface. Future research directions include finding the optimal configuration of lights and cameras that maximizes the amount of volume that can be carved away at each iteration.

Notes

1. We define a segment $\hat{s}$ (e.g., a portion of the conservative estimate of object contour) to be *closer* than another segment $\bar{s}$ (e.g., a portion of object contour) with respect to a point Q, as follows. Let $\hat{p}$ be a point belonging to $\hat{s}$. If the line through Q and $\hat{p}$ intersect $\bar{s}$, the intersection defines a point $\bar{p}$ on $\bar{s}$. If $distance(Q, \hat{p}) \leq distance(Q, \bar{p})$ for any $\hat{p} \in \hat{s}$, then $\hat{s}$ is defined to be closer to Q then $\bar{s}$. Finally, we define a segment $\hat{s}$ to be the *closest to* Q if $\hat{s}$ is closer to Q than any other segment (or point) $\bar{s}$ in the plane.
2. A two camera system (one located behind the translucent panel and other one located, for instance, on the back—see Fig. 14) is more expensive but presents some advantages: (i) only one rotation is needed, and the user is not required to step in and remove the panel; (ii) it is possible to acquire many objects with the same calibration data.
3. Alternatively, for the contradiction test we could render a shadow map (i.e., a scene rendered from the light as if it was a camera) and save the depth map. Once the point in shadow is projected into the shadow map, the check would be if the point is further away than the closest value in the shadow map. The value of the last pixel in shadow has to be computed as in the previous case.

References

Horn, B.K.P. and Brooks, M.J. 1989. *Shape from Shading*. MIT Press.

Rocchini, C., Cignoni, P., Montani, C., Pingi, P., and Scopigno, R. 2001. A low cost optical 3D scanner. *Compute Graphics Forum* 20(3):299–309.

Bouguet, J.-Y. and Perona, P. 1999. 3D photography using shadow in dual space geometry. *International Journal of Computer Vision*, 35(2):129–149.

Andreetto, M., Brusco, N., and Cortelazzo, G.M. 2004. Automatic 3D modeling of textured cultural heritage objects. *IEEE Transactions on Image Processing*, 13(3):354–369.

Shafer, S.A. and Kanade, T. 1983. Using shadows in finding surface orientations. *Computer Vision, Graphics, and Image Processing* 22:145–176.

Hambrick, L.N., Loew, M.H., and Carroll, R.L. 1987. The entry-exit method of shadow boundary segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):597–607.

Kriegman, D. and Belhumeur, P. 2001. What shadows reveal about object structure. *Journal of the Optical Society of America–A*, 18(8):1804–1813.

Hatzitheodour, M. and Kender, M. 1988. An optimal algorithm for the derivation of shape from shadows. In *Proc. of Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 486–491.

Raviv, D., Pao, Y.-H., and Loparo, K.A. 1989. Reconstruction of three-dimensional surfaces from two-dimensional binary images. *IEEE Transactions on Robotics and Automation*, 5(5):701–710.

Langer, M.S., Dudek, G., and Zucker, S.W. 1995. Space occupancy using multiple shadow images. In *Proc. of the International Conference on Intelligent Robotics and Systems*, Pittsburg, PA, pp. 390–396.

Daum, M. and Dudek, G. 1998. On 3D surface reconstruction using shape from shadows. In *Proc. of Computer Society Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 461–468.

Yang, D.K.-M. 1996. *Shape from Darkness Under Error*, Ph.D. thesis, Columbia University.

Yu, Y. and Chang, J. 2005. Shadow graphs and 3D texture reconstruction. *International Journal of Computer Vision*, 62(1–2):35–60.

Kutulakos, K.N. and Seitz, S.M. 1999. A theory of shape by space carving. In *Proc. of the Interational Conference on Computer Vision*, Kerkyra, Greece, pp. 307–313.

Savarese, S., Rushmeier, H., Bernardini, F., and Perona, P. 2002. Implementation of a shadow carving system for shape capture. In *Proc. of 3D Data Processing Visualization and Trasmission*, pp. 12–23.

Savarese, S., Rushmeier, H., Bernardini, F., and Perona, P. 2001. Shadow carving. In *Proc. of the Int. Conf. on Computer Vision*, pp. 190–197.

Martin, W.N. and Aggarwal, J.K. 1983. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158.

Szeliski, R. 1993. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32.

Lorensen, W. and Cline, H. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21:163–169.

Reed, M.K. and Allen, P.K. 1999. 3D modeling from range imagery: An incremental method with a planning component. *Image and Vision Computing*, 17:99–111.

Laurentini, A. 1995. How far 3D shapes can be understood from 2D silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):188–195.

Koenderink, J. 1984. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330.

Giblin, P. and Weiss, R. 1986. Reconstruction of surfaces from profiles. In *Proceedings of the International Conference Computer Vision*, pp. 136–144.

Vaillant, R. and Faugeras, O. 1992. Using extremal boundaries for 3D object modeling. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 14(2):157–173.

Cipolla, R. and Blake, A. 1992. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):99–111.

Zheng, J.Y. 1994. Acquiring 3D models from sequences of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):163–178.

Lazebnik, S., Boyer, E., and Ponce, J. 2001. On computing exact visual hulls of solids bounded by smooth surfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 156–161.

Fitzgibbon, A., Cross, G., and Zisserman, A. 1998. Automatic 3D model construction for turn-table sequences. In *Proceedings of SMILE Workshop on Structure from Multiple Images in Large Scale Environments* Vol. 1506, pp. 154–170.

Boyer, E. and Berger, M. 1997. 3D surface reconstruction using occluding contours. *International Journal of Computer Vision* 22(3):219–233.

Sullivan, S. and Ponce, J. 1998. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(10):1091–1097.

Leibe, B., Starner, T., Ribarsky, W., Wartell, Z., Krum, D.M., Weeks, J., Singletary, B., and Hodges, L.F. 2000. Toward spontaneous interaction with the perceptive workbench. *IEEE Computer Graphics & Applications*, 20(6):54–65.

Kampel, M., Tosovic, S., and Sablatnig, R. 2002. Octree-based fusion of shape from silhouette and shape from structured light. *Intl. Symposium on 3D Data Processing Visualization and Transmission*, pp. 754–757.

Seitz, S.M. and Dyer, C.R. 1997. Photorealistic scene reconstruction by voxel coloring. In *Proc. of Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 1067–1073.

Broadhurst, A. and Cipolla, R. 2001. A probabilistic framework for space carving. In *Proc. of the International Conference on Computer Vision*, pp. 388–393.

Bhotika, R., Fleet, D., and Kutulakos, K. 2002. A probabilistic theory of occupancy and emptiness. In *Proc. of the European Conference on Computer Vision*, pp. 112–132.

Szeliski, R. and Golland, P. 1999. Stereo matching with transparency and matting. *Int. Journal of Computer Vision*, 32(1):45–61.

De Bonet, J. and Viola, P. 1999. Roxels: Responsibility weighted 3D volume reconstruction. In *Proc. of the International Conference on Computer Vision*, pp. 418–425.

Eisert, P., Steinbach, E., and Girod, B. 1999. Multi-hypothesis, volumetric reconstruction of 3D objects from multiple calibrated camera views. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 3509–3512.

Slabaugh, G., Culbertson, W., Malzbender, T., Stevens, M., and Schafer, R. 2004. Methods for volumetric reconstruction of visual scenes. *International Journal of Computer Vision*, 57(3):179–199.

Treuille, A., Hertzmann, A., and Seitz, S. 2004. Example-based stereo with general BRDFs. In *Proc. of the European Conference on Computer Vision*, pp. 457–469.

Farouk, M., El-Rifai, I., El-Tayar, S., El-Shishiny, H., Hosny, M., El-Rayes, M., Gomes, J., Giordano, F., Rushmeier, H., Bernardini, F., and Magerlein, K. 2003. Scanning and processing 3D objects for web display. In *Proc. of the International Conference on 3DIM*, pp. 310–317.

Tsai, R.Y. 1987. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344.

Kenneth Wong, K.-Y. 2001. *Structure and Motion from Silhouettes*, Ph.D. thesis, University of Cambridge.

Taubin, G. 1995. A signal processing approach to fair surface design. *SIGGRAPH*, 351–358.

Taubin, G., Zhang, T., and Golub, G. 1996. Optimal surface smoothing as filter design. Tech. Rep. RC-20404, IBM Research.