

Shadow Carving

Silvio Savarese[†] Holly Rushmeier[‡]

Fausto Bernardini[‡] Pietro Perona[†]

[†]California Institute of Technology
Mail stop 136-93, Pasadena, CA 91125
{savarese,perona}@vision.caltech.edu

[‡]IBM T. J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
{holly,fausto}@watson.ibm.com

Abstract

The shape of an object may be estimated by observing the shadows on its surface. We present a method that is robust with respect to a conservative classification of shadow regions. Assuming that a conservative estimate of the object shape is available, we analyze images of the object illuminated with known point light sources taken from known camera locations. We adjust our surface estimate using the shadow regions to produce a refinement that is still a conservative estimate. A proof of correctness is provided. No assumptions about the object topology are made, although any tangent plane discontinuities over the object's surface are supposed to be detectable. An implementation and some experimental results are presented.

1 Introduction

Shape from shadows, also known as shape from darkness, is an approach for scanning 3-D objects that can be implemented using inexpensive lighting and imaging equipment. Shape from shadows has the advantage that it does not rely on surface texture for establishing correspondence, or on a model of the surface reflectance characteristics. However, past methods for shape from shadows can give either poor results or fail to converge when applied to physical data that contains error. We introduce a method for using shadows to estimate shape that is robust with respect to a conservative classification of shadow regions. We assume that we have as a starting point a conservative estimate of object shape – that is, the volume enclosed by the current surface estimate completely contains the physical object. We analyze images of the object illuminated with known point and/or collimated light sources taken from known camera locations. We assume that we have a method for labelling image pixels that we are confident are in shadow, although there may also be additional shadow pixels that we have mislabelled as being lit. No assumptions about the surface topology are made (multi-part objects and occlusions are allowed), although any tangent plane discontinuities over the object's surface are supposed to be detected. Our basic idea is that we adjust the current surface estimate using the shadow regions to produce improved shape estimates that remain conservative. A proof of correctness is provided. Our motivation for pursuing this work is the

construction of an inexpensive and robust scanning system based on shape from silhouettes and shape from shadows. Furthermore, the method can be located within a similar volumetric framework as that employed by traditional shape-from-silhouette and recent multiple view algorithms.

We begin by reviewing previous work in the area of shape from shadows. We describe our shadow carving approach and present a proof of its correctness. Finally, we present some results from a simple implementation of the method in a table-top scanning system.

2 Background

Computing shape from shadows has been studied for many years. Shafer and Kanade [1] established fundamental constraints that can be placed on the orientation of surfaces based on the observation of the shadows that one casts on another. Hambrick et al. [2] developed a method for labelling shadow boundaries that enables inferences about object shape. Yang [3] presents a review of many shape from shadows algorithms, and compares them to other shape from X techniques. Here we summarize the work most relevant to our approach. Hatzitheodorou and Kender [4] presented a method for computing a surface contour formed by a slice through an object illuminated by a directional light source casting sharp shadows. 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 points at the beginning and end of the shadow. Raviv et al. [5] developed a method where 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. The idea of shadowgram is introduced. Langer et al. [6] extend the method of Raviv et al. for computing holes beneath the recovered height field description of the top surface for 2 dimensions. Daum and Dudek [7] subsequently developed a method for recovering the surface for light trajectories that are not a single arc.

All of these methods rely on accurate detection of the beginning and ends 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 [3] considers the problem of shape from shadows with error.

In our method we do not rely on knowing beginning and ends 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.

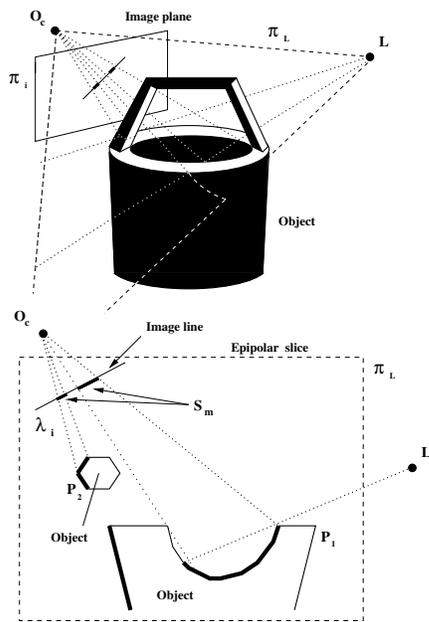


Figure 1. The epipolar slice model.

Our shadow carving approach is similar in spirit to the space carving approach of Kutulakos and Seitz [8]. In their work, voxels are carved out of a region of space enclosing the object if they do not project consistently into a set of captured images. In our approach, we consider consistency between a camera and light views, rather than multiple camera views. Consistency can be tested robustly by detecting shadows, without requiring a 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 a well defined portion of volume can be removed in a conservative manner from the current object estimate, instead of just iteratively removing voxels until all the inconsistencies are solved.

3 Shadow Carving

Suppose that we do not know anything about an object’s surface except an upper bound estimate of the surface which may be thought as a shell surrounding the object itself. Such upper bound surface estimation can be achieved, for instance, using shape from silhouettes. Suppose now that a light source is illuminating the object. What do the observed object self-shadows tell us about the real surface? The main information comes from the inconsistencies between the shadows which would be produced by the estimated surface and the observed shadows which are actually produced by the real surface. Thus, we could remove (carve out) volume from the current object estimate in order to reduce the inconsistencies, and therefore incrementally compute better estimates of the object’s shape, while making sure that at each step an upper bound estimate is maintained (conservative carving). We call this procedure *shadow carving* because we carve out volume from the object by using observed shadows.

To explain our approach, we begin with a simple example of shadow carving in the context of the epipolar slice model. Then we continue our analysis by examining the general case.

3.1 The epipolar slice model

Consider an object in 3-D space and a point light source L illuminating it. A certain number of shadows are cast over the object by parts of the object itself. The scene is observed by a camera whose center is located in O_c and whose image plane is called π_i . An image I is acquired. Let us suppose that we are able to process the image in order to separate the shadows from the rest of the observed object.

We consider now a slice of the scene which we called *the epipolar slice model*. Let us consider the family of planes Π_L passing through O_c and L . 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 3-D space. See Fig. 1. For each slice, we can define the image line λ_i (intersection of π_i with π_L), the shadow segments s_m (intersection of the estimated image shadow with π_L), the object contour P (intersection of the object’s surface with π_L) and the upper bound object contour \hat{P} (intersection between the upper bound surface estimation and π_L .) Within the epipolar slice model, we can also define the object area A_P (that is, the area bound by P) and the upper bound object area $A_{\hat{P}}$ (that is, the area bound by \hat{P} .) The extension to the 3-D case is immediate by observing that the epipolar slice sweeps the entire object’s volume. Therefore if we solve the shadow carving problem in the epipolar slice, we solve the 3-D problem as well.

3.2 Example

Figure 2 shows an example of shadow carving. The shadow \bar{s} is cast by the the object contour P over itself.

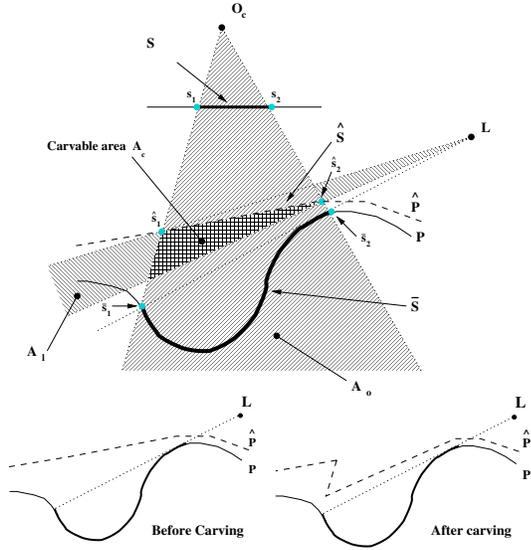


Figure 2. Top. Example of shadow carving.

The end points of \bar{s} are \bar{s}_1 and \bar{s}_2 . s is the image of \bar{s} and is bound by s_1 and s_2 — images of \bar{s}_1 and \bar{s}_2 respectively. The line defined by O_c and s_1 intersects \hat{P} in \hat{s}_1 . The line defined by O_c and s_2 intersects \hat{P} in \hat{s}_2 . \hat{s} is the segment along \hat{P} whose extrema are \hat{s}_1 and \hat{s}_2 . We call A_l the infinite area defined by the family of lines passing through L and any point along \hat{s} . We call A_o the (infinite) area defined by the family of lines passing through O_c and any point along \hat{s} . The intersection between A_o , A_l and A_P gives an area A_C called the *carvable area* (i.e. cross-hatched area in the figure). A_C can be removed from the current upper bound estimate of the object's contour, generating an improved conservative approximation. However, many questions arise. What happens if the topology of P is more complex, if the shadow is occluded by other contour components, if multiple shadows are imaged by the camera, if the object surface contains low-albedo regions that do not allow a correct or complete estimate of the shadows? Can we define the carvable area while facing all these situations? Can we guarantee that the carvable area is always outside the object (i.e. the new estimate is still conservative)?

In Section 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 (Section 5). 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 4.

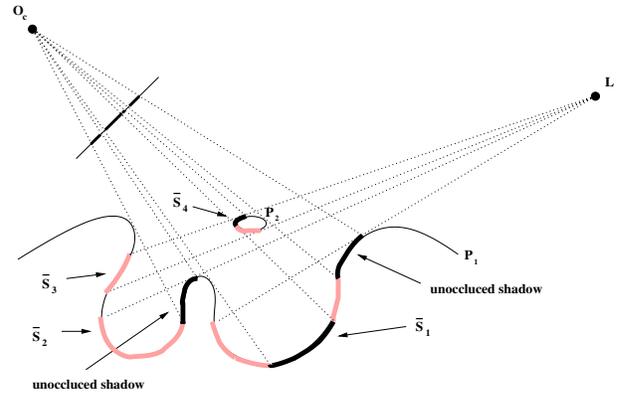


Figure 3. Example of occluded and unoccluded shadows.

4 The Shadow Decomposition

Let us consider an epipolar slice of a 3-D object. In general, the object's contour P on the epipolar slice might comprise more than one separate contour components. For instance the body of the bucket in Fig. 1 produces the contour P_1 and the handle produces the contour P_2 .

Given a point light source L , some portions of the object contour will be visible from the light source whereas other portions will not.

Definition 4.1 A shadow is a portion of contour that is not visible from the light source.

Thus depending on the object's topology and the light source position, there will be a certain shadow distribution. We call s_j an arbitrary shadow lying over the object contour.

In general, 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 point of view is called *unoccluded*. Figure 3 shows examples of unoccluded shadows: the unoccluded shadows are indicated with the bold black lines; the occluded portions of shadow are indicated with bold gray lines. For instance, only 2 portions of the shadow \bar{s}_1 are visible from O_c ; \bar{s}_3 is not visible at all. It is clear that whether a shadow is occluded or not only depends upon the contour topology and the camera position.

Let us suppose that we have a technique to detect shadows. It is not realistic to assume that this technique is perfect¹. However, we assume that the shadow detection technique is always *conservative*. That is, a shadow may not

¹Depending on the reflectance properties of the object's surface, the radiance properties of the light source and the features of the camera model, the task of detecting shadows might be very challenging, if not impossible. For instance, it is difficult to discriminate a shadow where the object's surface exhibits a very low reflectance factor (i.e. low albedo — only a very little portion of the incident light is reflected) or behaves as specular reflector (the incident light is reflected in only one direction).

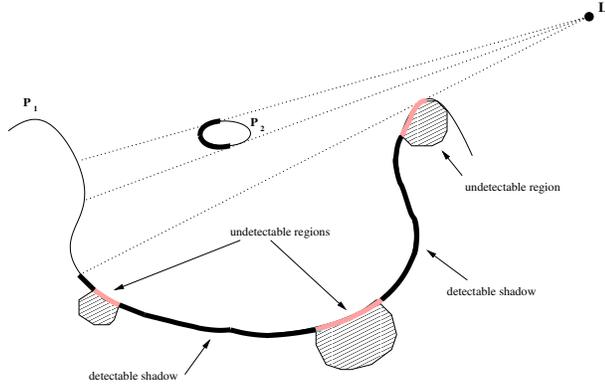


Figure 4. Example of detectable shadows.

be detected but whatever is labeled as shadow is indeed a shadow. 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 define *detectable* a shadow which does not lie within an undetectable portion of contour. Figure 4 shows examples of detectable shadows and undetectable regions: the hatched areas are undetectable contour regions; the detectable shadows are indicated in bold black.

Definition 4.2 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 $\bar{a}_{j,1}, \bar{a}_{j,2} \dots \bar{a}_{j,k}$. See Fig. 5 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 image line yields the atomic image shadows $a_{1,1}, a_{1,2}, a_{1,3}$ and $a_{1,4}$.

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 regions². 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 not difficult to realize that there is no unique correspondence between estimated shadow components and atomic image shadows.

²In our prototype system described in Section 6, we label shadow regions by comparing multiple images of the object lit from different light source locations and taken from the same camera viewpoint. By this comparison we avoid mislabelling low albedo surfaces as being in shadow.

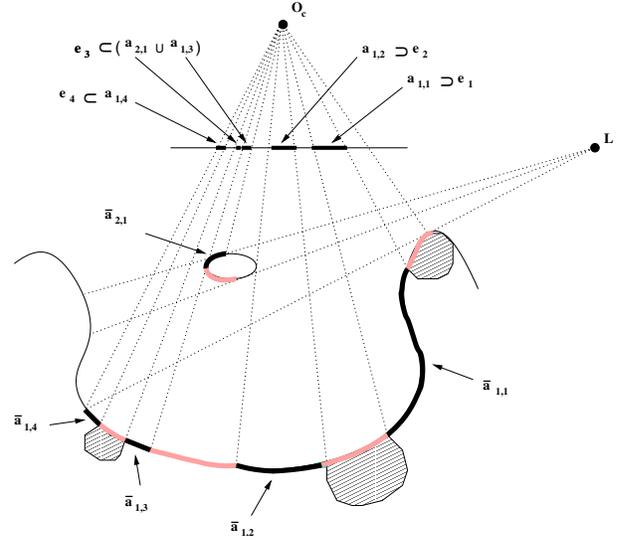


Figure 5. Example of atomic shadows.

Lemma 4.1 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.

An example is shown in Fig. 5: 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 2 different contours.

If we assume that the object contour's first order derivative is continue (contour smoothness hypothesis), Lemma 4.1 just follows from the definitions and the hypothesis of conservative shadow detection technique (a piece of shadow can mislabeled as non-shadow, but a non-shadow cannot be mislabeled as a shadow). In the following we want to show that if the hypothesis of smoothness is removed, Lemma 4.1 might be no longer verified.

Let us consider the example depicted in Fig. 6. 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 does 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 re-

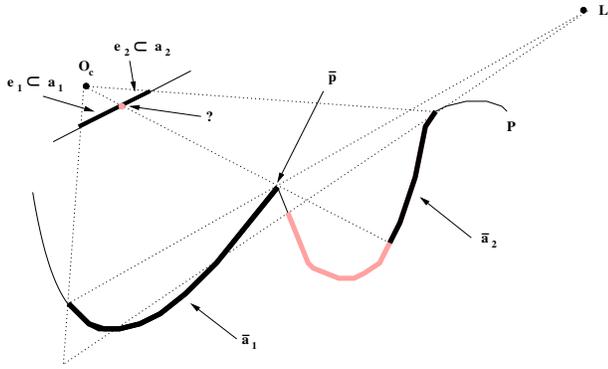


Figure 6. Contour with singular point.

sult, Lemma 4.1 does not necessarily hold when the contour smoothness hypothesis is not verified.

In reality, we can remove the smoothness hypothesis if we suppose that we have a technique to identify points whose first order derivative is not continue. We call such points *singular* and we label them as undetectable. Let us consider again the example of Fig. 6. If \bar{p} can be identified as singular and therefore its image p labeled as undetectable, a_1 and a_2 are no longer estimated as a unique shadow component but as 2 disconnected estimated shadows components e_1 and e_2 . The shadow decomposition is again conservative and Lemma 4.1 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 4.1.

5 Shadow carving analysis

In Section 1 we introduced the idea of carvable area A_C . By removing the carvable area from the current object area estimation we remove the inconsistency between the shadows which would be produced by the estimated object's contour and the observed shadows which are produced by the actual object's contour. As a result, a better approximation of the object's contour is achieved. However, since we are interested in making sure that an upper bound estimate is maintained (conservative carving), we introduce the following Proposition:

Proposition 5.1 *Any point within a carvable area cannot belong to the actual object area.*

This Proposition guarantees that by removing a carvable area from the current object area estimation, we always remove a portion of area outside the real object area.

As described in Section 4, the estimated shadow e is decomposed into a set of the estimated shadow components e_i . Each e_i may correspond to one or more carvable areas. Our aim is to first formally define the carvable area and then

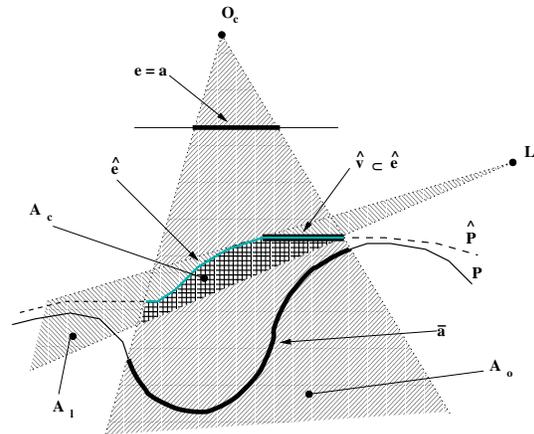


Figure 7. The atomic shadow case.

to prove that each carvable area attached to a generic e_i satisfies Proposition 5.1. By means of Lemma 4.1 we can 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.

5.1 The atomic shadow case

Let us consider an estimated shadow component e and let us assume that e is exactly the atomic image shadow a . See Fig. 7. We call \bar{a} the corresponding atomic shadow over the object contour. Let A_o be the (infinite) area defined by the family of lines passing through O_c and e . The following lemma holds:

Lemma 5.1 *Given an estimated shadow component e and the corresponding 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 .*

Such property follows directly from the definition of atomic shadow. \square

In the following, we introduce the geometric quantities necessary to formally define the carvable area. Let \hat{P} be the current upper bound contour estimation and let \hat{e} (light gray line in fig.) be the projective transformation (centered in O_c) of e onto the upper bound contour \hat{P} closest to O_c ³. Since \hat{P} is an upper bound estimation of the real P , \hat{e} is

³We define a segment \hat{s} (e.g. a portion of the upper bound 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 $\bar{p} \in \bar{s}$, then \hat{s} is defined to be closer to Q than \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

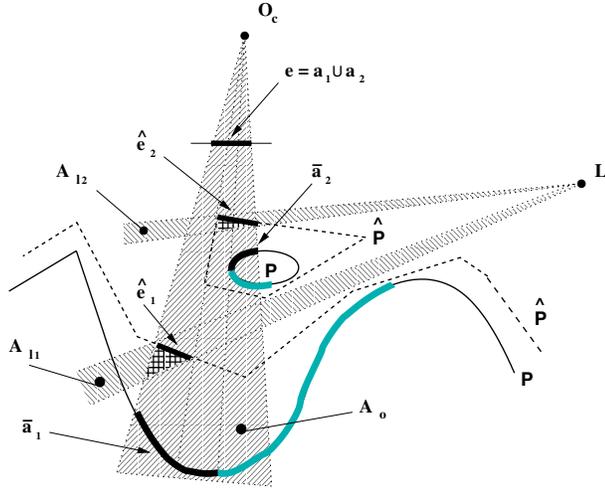


Figure 8. The composite shadow case.

closer to O_c then \bar{a} . Let \hat{v} (black line in fig.) 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 . Let A_{I_i} be the area defined by the family of line passing through L and any point along \hat{v} .

Definition 5.1 We define the carvable area A_C as the area obtained by the intersection of the estimated object area (bounded by \hat{P}), A_{I_i} and A_o .

The carvable area corresponds to the cross-hatched area in Fig. 7. Our main result is stated in the following lemma:

Lemma 5.2 The carvable area A_C does not intersect the real object contour P .

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 4.1. Consider case 2. Consider a portion q_p of q not included in \bar{a} . Since \bar{a} is atomic, by Lemma 5.1 q_p must be eventually 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. \square

Lemma 5.3 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 than \bar{a} . \square

By combining Lemma 5.2 with Lemma 5.3, Proposition 5.1 is proved for the atomic case.

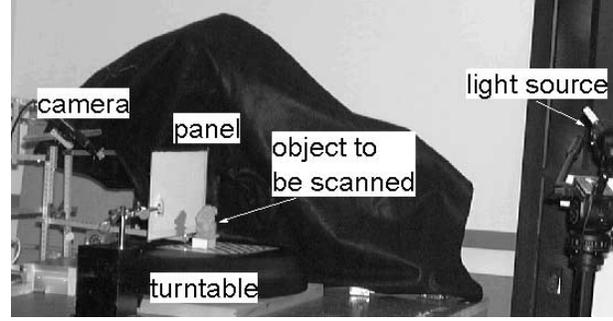


Figure 9. Our proposed setup.

5.2 The composite shadow case

The composite shadow case arises when e is not necessarily attached to a single atomic shadow. See for instance Fig. 8. Let us assume that e is actually composed by the union of J atomic image shadows a_1, a_2, \dots, a_J . The corresponding atomic shadows lying over the object contour are $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_J$, respectively. The area A_o is defined by e as in the atomic case. Lemma 5.1 can be modified as follows:

Lemma 5.4 Given an estimated shadow component e and the corresponding A_o , all the points belonging to P and within A_o either belong to one of the atomic shadows $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_J$ 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 $q \in \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_J\}$ between p and O_c .

The lemma directly follows from the definitions of atomic shadows and from the assumption that e is a connected shadow component. \square

We can define \hat{e} as in the atomic case. The difference here is that \hat{e} may be decomposed into K different components $\hat{e}_1 \dots \hat{e}_K$ 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_{I_k} as in the atomic case. Finally:

Definition 5.2 We define the set of carvable areas $A_{C_1}, A_{C_2}, \dots, A_{C_K}$ attached to the estimated shadow component e as set of areas obtained by the intersection among the estimated object area (bounded by \hat{P}), A_o and the set $A_{I_1}, A_{I_2}, \dots, A_{I_K}$.

It is not difficult to show that each A_{C_k} is empty, by using Lemma 5.4 and by modifying accordingly Lemma 5.2 and 5.3 and the attached quantities. As a conclusion, Proposition 5.1 holds in presence of composite shadows as well.

5.3 Effect of errors in the shadow estimate

In this section we want to analyze the behaviour of the shadow carving technique when errors in the shadow estimation occur. We proved Proposition 5.1 with the as-

sumption that an estimated shadow component is equivalent to an atomic image shadow (or the union of 2 or more atomic image shadows). In presence of a conservative error in the shadow estimation, the estimated shadow component is included in the atomic image shadow. 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. If a non conservative error in the shadow estimation occurs, the estimated shadow components might not be necessarily included in the corresponding atomic image shadows. As a result, there might exist carvable areas within the object's area. The shadow carving procedure is no longer conservative.

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

6 Implementation and Results

We have implemented shadow carving in a simple trial system shown in Fig. 9. An initial conservative estimate of shape is obtained using shape from silhouettes. A camera images shadows of an object cast on a translucent panel. Knowing the light source and panel positions, each shadow defines a cone containing the object. The object surface is then estimated by finding the intersection of the cones using a volumetric method similar to the technique originally proposed by Martin and Aggarwal [9]. The method of segmenting the object silhouette from background is similar to the method described in [10], with the difference that a vertical plane and turntable are used allowing a wider range of object poses to be captured. The output of this first phase is a triangle mesh extracted from the volume using a modified Marching Cubes algorithm [11].

In the second, shadow carving phase, the translucent panel is removed, and images are captured with the object illuminated by each of five light sources in turn. Sets of five images are obtained for multiple poses of the object using the turntable. For each set of five images for a single pose, shadows are detected by comparing the intensities for each region in the entire set. For each object pose, we then render a depth image of the current surface estimate (by simply reading back the z-buffer after rendering the surface in OpenGL). The depth image has the same resolution as our captured shadow images. We then process each shadow image for this view. For each pixel in the detected shadow region that has a non-background depth, we test if a ray from the point in the depth map to the light is blocked by the object. Any ray casting algorithm could be used, in par-

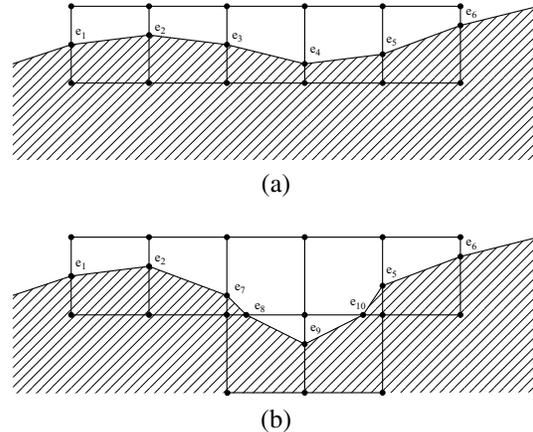


Figure 10. (a) The depth map.

ticular casting a ray through an octree data structure built in the initial surface estimation would be efficient. For this trial implementation however, since our test object had no handles, we just rendered the ray to the light source into the same depth buffer and checked that it was always in front of the surface. Any shadow pixel that has a clear view of the light source from the current estimated surface is marked as a contradiction. Following the projection of the ray in the image from the shadow pixel towards the light source, we can find the point in the shadow region closest to the light source.

To resolve pixels in contradiction, we adjust their height in the depth map. Referring to Fig. 2, all image pixels in \hat{P} before carving, that are in the carvable area, have their depth adjusted so that they lie on the ray formed by the light source L and the point on the shadow edge closest to the light source (i.e. s_2), that is, so that they lie on \hat{P} after carving.

All adjustments for the shadows for lights for a single pose can be made on a single range image. The surface formed by this range image needs to be integrated into the full surface estimate. This could be done by treating the results as a point cloud, and using the ball pivoting algorithm [12] to generate the new mesh. For simplicity in our initial implementation, we use the range map to update a volume representation as shown in Fig. 10. In Fig. 10a a depth map is rendered using a surface extracted from volume data. In Fig. 10b the depth map is adjusted to explain shadows. We do not simply include or exclude voxels, but rather record where the range map intersects the voxel edges. The revised surface is then found using modified Marching Cubes, and the process is repeated for the next pose.

In Fig. 11 the results of our processing are shown. The top row of images shows a figure from the front, and the second row of images shows the figure from the back. The leftmost image in each row is the figure as captured with

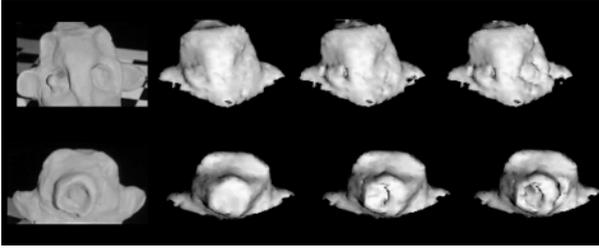


Figure 11. Experimental results.

the light on closest to the camera view point. Images on the shadow panel were obtained for 24 positions of the turntable, spaced 15 degrees apart. The second image in each row of Fig. 11 shows the result of the initial space carving. The third and fourth images in each row show the object after carving using the shadows from 1 and 2 camera positions respectively – i.e. after using 4 and 8 self-shadowing images. The concavities around the eyes and ears have become visible. The large and steep concavity has been significantly carved out. Clearly, our results are not a finely detailed object representation. However, the surface after shadow carving is a much better estimate, and a much better starting point for the application of an additional method such as photometric stereo which is good at estimating shape locally.

7 Conclusions

We have presented an analysis of the problem of refining an upper bound estimate of an object's shape by observing the shadows on the object when it is lit by a known point or collimated light source. We have provided a proof of correctness that a well defined portion of volume can be removed in a conservative manner from the current object estimate. Our main assumption have been that the shadow detection technique is always conservative – that is, a shadow may not be detected but whatever is labeled as shadow is indeed a shadow. No assumptions about the object topology have been made, although any tangent plane discontinuities over the object's surface have been supposed to be detectable. We have implemented the method and validated the theory with experimental results.

8 Acknowledgments

Part of this work was performed during an internship by S. Savarese at the IBM T. J. Watson Research Center and part funded by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726)

References

- [1] S. A. Shafer and T. Kanade, "Using shadows in finding surface orientations," *Computer Vision, Graphics and Image Processing*, vol. 22, pp. 145–176, 1983.
- [2] L. N. Hambrick, M. H. Loew, and R. L. Carroll, "The entry-exit method of shadow boundary segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 597–607, Sept. 1987.
- [3] David Kwun-Moo Yang, *Shape from Darkness Under Error*, Ph.D. thesis, Columbia University, 1996.
- [4] M. Hatzitheodour and M. Kender, "An optimal algorithm for the derivation of shape from shadows," in *Proc. of Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 1988, pp. 486–491.
- [5] D. Raviv, Y.-H. Pao, and K. A. Loparo, "Reconstruction of three-dimensional surfaces from two-dimensional binary images," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 701–710, Oct. 1989.
- [6] M. S. Langer, G. Dudek, and S. W. Zucker, "Space occupancy using multiple shadow images," in *Proc. of the International Conference on Intelligent Robotics and Systems*, Pittsburg, PA, August 1995.
- [7] M. Daum and G. Dudek, "On 3-D surface reconstruction using shape from shadows," in *Proc. of Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 1998, pp. 461–468.
- [8] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," in *Proc. of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, Sept. 1999, pp. 307–313.
- [9] W. N. Martin and J. K. Aggarwal, "Volumetric descriptions of objects from multiple views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 150–158, Mar. 1983.
- [10] B. Leibe et al., "Toward spontaneous interaction with the perceptive workbench," *IEEE Computer Graphics & Applications*, vol. 20, pp. 54–65, November–December 2000.
- [11] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, pp. 163–169, 1987.
- [12] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, October–December 1999.