The geometry of point light source from shadows

Bo Hu and Christopher Brown and Randal Nelson

The University of Rochester
Computer Science Department
Rochester, New York 14627

Technical Report 810

June, 2004

Abstract

Shadows provide valuable information about the scene geometry, especially the whereabout of the light source. This paper investigates the geometry of point light sources and cast shadows. It is known that there is redundancy in the object-shadow correspondences. We explicitly show that no matter how many such correspondences are available, it is impossible to locate a point light source from shadows with a single view. We discuss the similarity between a point light source and a conventional pinhole camera and show that the above conclusion is in accordance to traditional camera self calibration theory. With more views, however, the light source can be located by triangulation. We proceed to solve the problem of establishing correspondences between the images of an object with extended size and its cast shadow. We prove that a supporting line, which, put simply, is a tangent line of the image regions of the object and its shadow, provides one correspondence. We give an efficient algorithm to find supporting lines and prove that at most two supporting lines can be found. The intersection of these two lines gives the direction of the point light source. All this can be done without any knowledge of the object. Experiment results using real images are shown.

Keywords: shadow geometry, cast shadow, light source locating, convex hull, pinhole camera, self calibration

This work is supported by the NSF Research Infrastructure grant No. EIA-0080124
Chapter 1

Introduction

Shadows reveal the spatial layout and surface topology of objects and thus play an important role in visual perception [1, 2, 3]. Past research focuses on how to use shadow to infer the shape of an object (the so-called shape from shadow methods [4, 5].) The light source location is usually given in such methods. In many computer vision and image processing tasks, shadows are actually considered an annoyance, rather than as beneficial. Hence a large body of work deals with how to remove shadows [6, 7].

This research instead utilizes shadows and tackles the problem of locating the light source from the shadow of an object. We solve the common case of cast shadow, i.e., an object or occluder, casting shadow on a surface, which we call shadow surface. The surface does not have to be a plane. We take the image segmentation and labeling of the occluder and its cast shadow for granted, though they are by no means easy problems. We start with the simple case in which there is one camera looking at a scene with one small object illuminated by a point light source. The small object is treated as a particle, so we can discount both the shape of the object and more importantly the shape of the shadow. We show that, if the shadow surface is a plane, the geometry of such a configuration is quite similar to the well-studied two-camera setting, with analogous notions of epipolar plane and epipole. The point light source is a projective device just like a pinhole camera, with the planar shadow surface being its retina plane. In fact, this analogy can be pushed as far as one wants and we compare the problem of light source from shadow to the conventional camera calibration problem. We prove that with only one view, even with the shadow surface known, we can not fully recover the 3-D location of a point light source without knowing an item of metric information, which can be a length or an angle, of the occluder. However, for a directional light source (a point light source at infinity), one image is sufficient to determine the direction of the source. We then show that with two or more cameras, we can obtain the location of a point light source without any metric knowledge of the occluder. This is a well-known fact [4, 8], but we give an
explicit proof and show when such computation can be accomplished. We also make
connection between this conclusion and traditional camera self-calibration theory.

Establishing the correspondences between the image of an occluder and the shadow
is easy in the above simple case. However in practice, there are seldom such simple
cases. We thus investigate the problem of finding occluder-shadow correspondence
for general objects with spatial extents or shape. The central concept is supporting
line, which is, put simply, a tangent line to the image regions of the occluder and
the shadow. We extend the definition of tangency, which we call R-tangency, so that
non-smooth boundary points have tangent lines too. We prove that a supporting
line provides one occluder-shadow correspondence. To appreciate this result, let’s
look at an example. Fig.1.1(a) shows an image of a simple scene with an object, a
semi-sphere on top of a cylinder, sitting on a plane. The light source is at infinity and
on the left side of the object. The image is formed by orthogonal projection. The
line $l$ is tangent to both the image of the object and its shadow. In other words, $l$
is a supporting line. We claim that $l$ provides an object-shadow correspondence, or
the shadow of point $P$ is $Q$. At first look, the result does not seem to be generally
true. The argument against our result could go as follows. $P$ is on a great circle of
the sphere, so its shadow $Q'$ must be on the symmetry axis $s$ of the shadow region.
 Apparently, $Q$ and $Q'$ cannot be possibly be the same point. A more convincing and
more complicated case is that the object has an arm that is pointing away from the
view point and thus cannot be seen. But the shadow of the arm can be seen. The
line $l$ in Fig. 1.1(b) is a supporting line by definition, but apparently $P$ and $Q$ again
are not an object-shadow correspondence.

If we examine the two cases more closely however, we will find what is wrong
with the arguments. In the first case, if we are able to see the shadow at all, we
are looking down at the scene from some angle. The great circle that $P$ is on is the
occluding contour under this view direction. The great circle corresponding to $Q'$
is the one passing $P'$. So indeed $Q$ is the shadow of $P$. Moreover, $PQ$ and $P'Q'$
are parallel, because the light source is at infinity. In the second case, the drawing
(Fig. 1.1(b)) is plainly wrong. If the arm is long enough so that its shadow protrudes
as it is drawn, it cannot be totally obscured by the body (Fig. 1.1(c)). If instead the
arm is short and cannot be seen at all, the shadow of the arm must be bounded by
$l$ (Fig. 1.1(d)). This is quite remarkable, considering that if we cannot see the arm,
it can be of arbitrary shape, implying that its shadow can take arbitrary shape. Yet
our results reveal that there are interesting constraints on the size and position of the
shadow.

We also give an efficient algorithm to find supporting lines, based on the convex
hull of the image of the object and the shadow.
Figure 1.1: Understanding that a supporting line provides an object-shadow correspondence.
1.1 Related work

The early work by Shafer [4] touched the topic of light source location from shadows. He noticed that the shadow information is redundant in that the illumination vectors, which are the vectors from a shadow point to its corresponding occluder point, intersect at a single point. He however did not pursue this observation. Funka-Lee [9] elaborated Shafer’s observation. He gave an informal proof of a conclusion roughly equivalent to our Theorem 1. He also tried to locate an area light source using shadow information. The central theme of this work is to establish occluder-shadow correspondences without prior knowledge about the occluder, whereas Funka-Lee used a known object (a square), which he called a shadow probe, to generate the shadow and essentially circumvented the hard problem of finding correspondences. Similar work done by Pinel et.al. [10] estimates the direction of a directional light source from shadows. They had the notion of Separating Line, which does not give an occluder-shadow correspondence directly but only confines the projection of the light source to one of the half planes divided by such a line. Ikeuchi and Sato [11] use the shadow information differently. They compute the irradiance at a scene point by comparing the difference of the pixel values at that point with and without the occluder. They then use the computed irradiance to generate subtle soft shadows. The disadvantage of their work is that only the illumination of a single point (or its vicinity) can be recovered. By computing the location of the light source in a scene, we can exactly compute the illumination at any point.

1.2 Applications

One of the applications of the method proposed in this paper is in Augmented Reality (AR) systems, in which graphics objects are inserted into a real scene. To render the graphics objects, we need the light source information. Debevec [12] and Powell [13] both placed polished balls, termed light probes, in the scene to locate the light sources. In a sense, Ikeuchi and Sato’s object is a light probe too. If however one is not allowed to manipulate the scene, e.g., only images of the scene are available, the light probe approach will not work. There is a large body of work under the category of shape from shading that finds the light source from the shading of objects [14]. But applications of these methods are limited by the surface type of objects they can deal with. The advantage of our method lies in that no restriction is imposed on the objects. In fact, except for the segmentation in the image, we do not need any geometric or photometric information about the objects.

**Notation** We use capital letters for 3-D points and lower case letters for 2-D points and 2-D lines. The meaning should be clear from the context. We use Greek letters, e.g., $\Gamma$ for 2-D regions and use $\partial \Gamma$ for the boundary of the region.
Chapter 2

The shadow geometry of a point light source

2.1 Single view

Fig. 2.1 shows the basic geometry of cast shadow under a point light source. The point light source is at $L$. Particles $A$ and $B$ cast shadow $A'$ and $B'$ on the plane $\Pi$. Point $O$ is the center of projection of a camera whose retina plane is $I$. The image of $A$, $B$, $A'$ and $B'$ are $a$, $b$, $a'$ and $b'$, respectively. The vectors $\overrightarrow{a}a$ and $\overrightarrow{b}b$ are what Shafer called illumination vectors. One instant observation is that the two vectors intersect at point $l$, which is the image of the point light source. In fact, if we consider the light source as another camera, $l$ is the epipole and the planes $OLA$ and $OLB$ are epipolar planes. Now if we set a coordinate system on the camera and if the camera is internally calibrated, we know the light source is on the line $Ol$. The natural question to ask is that if the plane $\Pi$ is calibrated regarding to the camera, meaning we know its equation in the camera coordinate system, can we recover the exact location of the light source without knowing the 3-D positions of the occluders? Apparently, two particles with their shadows (or two images of a moving particle) only confine the light source to lie on $Ol$ and are not enough to locate the light source. If we have the image of one more point, we have a triangle and its image. We know its shadow cast on a known surface, too. We prove that this one more point is not, however, sufficient to determine the point light source.

Theorem 1. Without metric information of the occluder, the position of a point light source cannot be recovered from a single view.

Proof. We have already seen that two correspondences are not sufficient to determine the position of a point light source. We show that adding more correspondences does not add more constraints. Specifically we prove that any point $L$ on the line $Ol$ (Fig. 2.1) admits three points $A,B$ and $C$ (not drawn) whose shadows on $\Pi$ are
Figure 2.1: The geometry of cast shadow created by a point light source.

$A', B'$ and $C'$ and whose images are $a, b$ and $c$. In other words, we cannot determine the exact position of the point light source along that line. The key observation is that the two projectivities induced by the camera $O$ and the point light source $L$ will enforce collinearity in 3-D space. Although in general two 3-D lines such as $Oa$ and $LA'$ don’t intersect, they do in this case. Examining the plane $OLA'$, we will see that the line $Oa$ is in this plane. In fact, because both $l$ and $a'$ are in $OLA'$ and $a$ is collinear with $l$ and $a'$, $a$ is in $OLA'$, too. So $Oa$ is in $OLA'$. Apparently, $Oa \parallel LA'$, so it intersects with $LA'$. The intersection is the point $A$ we are looking for. In the same way, we can locate point $B$ and $C$ to account for the given shadows. So more points don’t provide more constraints on the location of the light source.

From the proof, we can see if we know the length between two points, say $A$ and $B$, $B$ cannot be located independently to $A$. In other words, the length of $AB$ gives one more constraint we need to compute the position of the light source. Other metric information of the occluder, e.g., the angle between $AB$ and $BC$, will also give us the needed constraint.

Let’s look at Theorem 1 more carefully with a counting argument. For the two-particle case, there are nine unknowns, the position of the light source and the positions of the two particles. There are eight known parameters, which are the four image points, each providing two constraints. So we have one degree of freedom left, which accounts for the exact location of the light source along the line $Ol$. If we have one more particle, we have three more unknowns. We have two more image points too: one of the particle, the other of the shadow cast by the particle. It looks like we have four more known parameters, so we should be able the determine the system.
But these two sets of image points are collinear with the point \( l \), which brings down one degree of freedom. So no matter how many particles we have, we are one degree of freedom short.

We can however determine the light source from a single view if it is at infinity (directional light source). In this case, the light source lies on the plane at infinity, which eliminates one unknown. The epipole is the vanishing point of the light source direction.

### 2.2 Multiple views

If we have two or more calibrated views, the position can be determined by intersecting the lines from the center of projection of each camera and the image of the light source. In practice, the lines will not intersect. But we can always compute the closest point to the lines using singular value decomposition (SVD).

### 2.3 Point light source as a pinhole camera

If the shadow surface is a plane, the point light source can be modeled as a pinhole camera. The shadow surface is its retina plane. The focal length is the orthogonal distance from the light source to the retina plane. The principal point is the orthogonal projection of the light source to the retina plane (Fig. 2.2). Evidently, we can
write down an intrinsic matrix for a point light source, just like we do for regular pinhole cameras:

\[
K_{\text{light source}} = \begin{pmatrix}
  l_z & 0 & l_x \\
  0 & l_z & l_y \\
  0 & 0 & 1
\end{pmatrix},
\]

where \((l_x, l_y, l_z)\) is the position of the point light source.

The problem of locating a point light source naturally turns into a camera calibration problem. More specifically, if we can find the “principal point” \((l_x, l_y)\) and the “focal length” \(l_z\) of a point light source, we know its position regarding the shadow surface (plane). The technique that calibrates cameras just by correspondences, without metric information about the scene, is called camera self calibration. The original self calibration method used the Kruppa equations [15]. For each pair of views, the Kruppa equations provide two independent constraints. Since the camera is already calibrated and we have three unknown (intrinsic) parameters for the point light source. We need one more constraint to calibrate (locate) the light source, which is the same as Theorem 1 states.
Chapter 3

Locating a point light source from general objects

In Chapter 2, we showed how to locate a point light source from point correspondences. In practice, objects are seldom small enough to be treated as particles. In this section, we show how to establish object-shadow correspondences for general objects.

We define a shadow ray to be the imaginary 3-D segment that originates from the point light source, passes through an occluder point and ends at the generally non-planar shadow surface. Its 2-D projection in an image is called a shadow line. There is a causal relationship among the three points on a shadow ray. Namely, the shadow is the result of the light source and the occluder. This causality reflects in geometry as the order of the three points being the light source, the occluder point and the shadow point. Because projection does not change sidedness, the order is kept on the shadow line, too. This argument proves the following lemma.

**Lemma 1.** The images of the point light source, an occluder point and the shadow the occluder point casts are collinear and in that order on the line.

All the shadow rays form a developable surface, which we call the shadow cone. The apex of the shadow cone is the point light source. The convex hull of the occluder plus its shadow is called the 3-D shadow hull (Fig. 3.1). The main assumption of this section is that the whole 3-D shadow hull can be seen from every view point. To be exact, we are assuming that the whole 3-D shadow hull is in the field of view of every camera and it is not occluded. Interestingly, the shadow surface doesn’t have to be a plane. It can be curved surface with bumps or hills, as long as the bumps on the surface do not occlude the 3-D shadow hull.

**Assumption** (Visibility). The 3-D shadow hull is visible at all view points.
In plain words, the above assumption can be understood in a less rigorous way as that each camera can see the whole occluder and its cast shadow, although the shadow could be wholly or partly occluded by the occluder itself. The assumption is logically very strong, as we will see in the following arguments. Yet it is usually satisfied in general scenes. So the algorithm we give below is indeed general.

The shadow surface breaks the 3-D space into two parts. The camera has to be on the same side as the light source and outside the 3-D shadow hull to meet the visibility assumption. The following lemma is immediate given the assumption and that projection does not change coincidency.

**Lemma 2.** All shadow lines intersect at a point outside the shadow hull. The intersection point is the projection of the point light source.

We define the 2-D region that consists of the image points of the occluder to be the *object region* and that of the image points of its shadow to be the *shadow region*. Image points of the object region and shadow region are called *object point* and *shadow point*, respectively. We call the projection of a 3-D shadow hull a *shadow hull*. Since projection does not change convexity, the shadow hull is the convex hull of the object region and the shadow region. Note, however, projection does change concavity. Think about a top down view of a Mexican hat.

We define *R-tangent line* of a 2-D region as follows.

**Definition 1.** Let $\Gamma$ be a compact 2-D region and $l$ be a line in the same plane. The line $l$ divides the plane into two disjoint parts $\Phi$ and $\Psi$. The line $l$ is said to be *R-tangent* to $\Gamma$ if
Figure 3.2: The definition of R-tangent line. (a) Not every point has a R-tangent line. (b) R-tangent lines of a region $\Gamma$. $l_2$ and $l_3$ are both R-tangent lines at a non-smooth point. (c) The region $\Gamma = \Gamma_1 \cup \Gamma_2$. $l_1$ is a R-tangent line to $\Gamma$ but not $l_2$ which is a R-tangent line to both $\Gamma_1$ and $\Gamma_2$, individually. HUF4.eps.

1. $l \cap \Gamma \neq \emptyset$;
2. $\Gamma \cap \Phi \neq \emptyset$ or $\Gamma \cap \Psi \neq \emptyset$;

The compactness of the region is required for the existence of R-tangent line. E.g., an open disk and a closed region enclosed by a two-side hyperbola do not have R-tangent lines. The first condition states that $l$ intersects $\Gamma$ and the second condition states that $\Gamma$ is at one side of $l$. The tangency as defined here is a global property, concerning the convexity of the whole region, whereas the usual definition in differential geometry is a local property, concerning only instantaneous direction at a point. E.g., the usual tangent line at a point of inflection [16] is not a R-tangent line (Fig. 3.2(a)). Also, R-tangent line at a non-smooth point is not unique (Fig. 3.2(b)). Note that $\Gamma$ is not required to be connected (Fig. 3.2(c)). Our definition of the R-tangency is in fact tied to the convexity of the region.

We continue by defining supporting line of the object region plus the shadow region as follows.

**Definition 2.** Let $\Gamma$ be the object region and $\Omega$ be its shadow region. Let $l$ be a line in the image plane. Let $l_\Gamma = l \cap \Gamma$ and $l_\Omega = l \cap \Omega$. The line $l$ is said to be a supporting line of $\Gamma \cup \Omega$ if

1. $l$ is R-tangent to $\Gamma$, $\Omega$ and $\Gamma \cup \Omega$;
2. $l_\Gamma \neq l_\Omega$.

Note that because of the first condition, $l_\Gamma$ and $l_\Omega$ are not empty sets. The second condition excludes the degenerate case that $\partial \Gamma$ and $\partial \Omega$ are adjacent and form a non-smooth point (Fig. 3.3).
The main theorem of this section is that a supporting line is a shadow line. Therefore a supporting line provides one correspondence between the occluder and the cast shadow. To prove that, we need one more lemma.

**Lemma 3.** *Every shadow point has to be accounted for.*

What this lemma means is that for every visible shadow point in an image, if we draw a shadow line, this line must intersect at least one occluder point. This is the consequence of the visibility assumption and the fact that cast shadow does not occlude its occluder. The converse is not generally true though. I.e., not every object point is accounted for by a shadow point, because the occluder may occlude its shadow.

**Theorem 2.** A supporting line is a shadow line.
Proof. By contradiction. $\Gamma$ is the object region and $\Omega$ is the shadow region (Fig. 3.4). Let $\mathcal{H}$ be the shadow hull (not drawn). Let line $go$ be a supporting line, where $g \in \Gamma$ and $o \in \Omega$. If $go$ is not a shadow line, let $o'$ be the shadow point cast by $g$. So $go'$ is a shadow line. Apparently, the segment $go' \subset \mathcal{H}$. Note that $o'$ might be occluded by $\Gamma$. But $go'$ is still in $\mathcal{H}$ in this case. By Lemma 1, the image of the light source $i$ is on the line $go'$ and $g$ is in between of $i$ and $o'$. Because $go$ is a supporting line, $i$ and $\mathcal{H}$ are at different side of the line $go$. In other words, $i$ is outside $\mathcal{H}$. So the segment $io$ is outside $\mathcal{H}$. But $io$ is another shadow line. By Lemma 1 and Lemma 3, there must an object point between $i$ and $o$. This contradicts the conclusion that $io$ is outside $\mathcal{H}$.

From the proof, we can see that supporting line is intimately connected with the shadow hull. In fact, a supporting line overlaps part of the shadow hull. The following algorithm to find supporting lines is based on the above observation.

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th>Labeled object region and shadow region, with the number of points in both regions being $N$.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td>Supporting lines of the object region and the shadow region.</td>
</tr>
</tbody>
</table>

1. Find boundary points by inspecting the 4-neighbor of each point. If all its 4 neighbors are region points, this is an inside point and remove it. Otherwise, it’s a boundary point. This is an $O(N)$ operation. The resulting number of boundary point is $O(\sqrt{N})$.

2. Use Graham’s algorithm [17] to find the shadow hull. Graham’s algorithm works by maintaining a stack of vertices of the shadow hull. The time complexity is $O(\sqrt{N} \log N)$.

3. For every pair of adjacent points in the stack, if the labels (object or shadow point) are different and the distance between the two points is above a threshold, the line defined by the two points is a supporting line. This is an $O(h)$ operation, where $h$ is number of vertices of the shadow hull.

**Algorithm 1:** Finding supporting lines.

Step 1 of the algorithm is for speeding up the second step and it is not necessary for the correctness of the algorithm, which is asserted by the following theorem. The threshold is small to exclude the degenerate case where the object point and the shadow point effectively coincide. (e.g., see Fig. 5.1(c)).

**Theorem 3.** A line found by Algorithm 1 is a supporting line.

**Proof.** We show that the line satisfies the two conditions of Definition 2. The line is an edge of a convex polygon (the convex hull). So both the object region and the
shadow region are at one side of the edge. Each of the two points that define the line belongs to the object region and the shadow region, respectively. So the line is R-tangent to both regions. The threshold ensures the second condition of Definition 2 is satisfied. 

Algorithm 1 by itself does not reveal how many supporting lines can be found with one object. The upper bound is given by the following theorem.

**Theorem 4.** Each object-shadow pair provides at most two supporting lines.

*Proof.* By contradiction. Assume there were more than three supporting lines, which are edges of a convex polygon. By Lemma 3, these supporting lines must intersect at a single point or be parallel if the light source is at infinity. However, we know that three or more edges of a convex polygon do not intersect at a single point. Nor can they all be parallel.

The theorem tells us that without further knowledge of the geometry of the occluder, at most two correspondences can be found. If we call the edges formed by two object points *object edges* and those by two shadow points *shadow edges*, we have the following corollary.

**Corollary.** Non-collinear object edges and shadow edges do not alternate.

In Fig. 3.5(a), if line \(ab\) and \(ef\) were object edges and line \(cd\) and \(gh\) were shadow edges not collinear with \(ab\) and \(ef\), we would have three supporting lines, \(bc\), \(de\) and \(fg\), which contradicts Theorem 4. If object edges and shadow edges are collinear, however, they can alternate in order (Fig. 3.5(b)).
Chapter 4

Experiment

We have done some indoor experiments to verify the method. The setup consists of four cameras and a light source (Fig. 4.1(a)). We use a swing-arm lamp to simulate a point light source. The four cameras are calibrated relative to the floor. There are two objects (a cup and a cube) in the scene (Fig. 4.1(b)). Each object and its cast shadow on the floor provide one supporting line. The intersection of the two supporting lines specifies the direction of the light source in one view (Fig. 4.1(c)). With four views, we can compute the exact 3-D location of the light source using triangulation. The recovered light source is shown in Fig. 4.1(a). It is not easy to obtain the ground truth of the light source position, especially since the lamp is an approximation of a point light source. To verify the result, we create computer graphics renderings using the recovered light source to shade the objects. By comparing the size, shape and orientation of the rendered shadows and the real shadows in a different scene (Fig. 4.1(d)), we can see the computed light source position is very close to the real position.
Figure 4.1: Light source from shadows experiment. (a) Experiment setup with computed light source. (b) Images taken by the four cameras. (c) Computing supporting lines in one view. The left pane shows the segmentation of the objects and shadows. The green dotted lines and the blue dash lines in the right pane are object edges and shadow edges, respectively. The solid red lines are computed candidate supporting lines. Two of the red lines are too short to be real supporting lines. (d) Real vs. virtual scene rendered with computed light source.
Chapter 5

Discussion and Conclusion

The definitions of shadow region and object region are usually clear. But if an object casts a shadow on itself, our method may not be able to find valid shadow lines. The crown of the hat (in fact a Gaussian) in Fig. 5.1(a) casts a shadow on the brim. This shadow region is inside the object region and no supporting lines can be established. On the other hand, the shadow region of the shadow cast by the brim on the ground provides us one supporting line. Theorem 4 claims that every object and shadow pair can provide at most two correspondences. It sounds absurd to assert that no more than two correspondences can be established for scenes like the one in Fig. 5.1(b). However, unless we know the exact relationship between a leaf or a branch and its cast shadow, which is difficult to find and which is what this paper tries to overcome, indeed we can only obtain two object-shadow correspondences. The theorem only gives an upper bound of the number of supporting lines can be computed from a single view. There are cases that no supporting lines can be found at all. In Fig. 5.1(c), the camera is lined up with the light source and the two supporting lines degenerate into two points. If the camera is moved to one side, we can find at least one supporting line. One drawback of our method is that, like any simple triangulating scheme, it is sensitive to noise. A little error of the direction of the support line will be amplified when determining the location of the light source. How to make the method more robust is a subject for future work.

The theorems given in this report are built on the fact that the projection transformation preserves convexity. This implies that the results are more general than their statements here. In fact, the theorems apply to any projective device, as long as the imaging surface of the device can be seen by other cameras. In the case of a point light source, the shadow surface is the imaging surface. Another application of the theorems is to locate a mirror if the mirror image is in the field of view of another camera. Depending on the nature of the mirror, the exact number of parameters to describe its location varies. E.g., if the mirror is flat, the center of projection is at infinity and three orientations suffice to specify the projective properties, its intrinsic matrix so to speak, of the mirror. If we have a picture of an object and its mirror
image, we can use the algorithm described here to compute the orientations of the mirror. With the mirror calibrated in this way, we have an extra view of the object without an extra camera and consequently can recover the geometry of the object using appropriate multi-view 3-D reconstruction methods. If the mirror is spherical, the center of projection is the center of the sphere, which again can be described using three parameters. For general quadric mirrors, there are similar results.

We fully explored the geometry of point light sources and cast shadows on planar and general surfaces. We showed how to recover the location of a point light source from shadows with multiple views. We proved an unintuitive result regarding how to establish correspondence between the object and the shadow. We showed that without prior geometric knowledge about the object, the best we can do is to find at most two such correspondences. The results are of practical value as well as theoretic interest.
References


