Estimation of Multiple Illuminants from a Single Image of Arbitrary Known Geometry

Yang Wang and Dimitris Samaras

Computer Science Department,
State University of New York at Stony Brook, NY 11794, USA
{yangwang, samaras}@cs.sunysb.edu

Abstract. We present a new method for the detection and estimation of multiple illuminants, using one image of any object with known geometry and Lambertian reflectance. Our method obviates the need to modify the imaged scene by inserting calibration objects of any particular geometry, relying instead on partial knowledge of the geometry of the scene. Thus, the recovered multiple illuminants can be used both for image-based rendering and for shape reconstruction. We first develop our method for the case of a sphere with known size, illuminated by a set of directional light sources. In general, each point of such a sphere will be illuminated by a subset of these sources. We propose a novel, robust way to segment the surface into regions, with each region illuminated by a different set of sources. The regions are separated by boundaries consisting of critical points (points where one illuminant is perpendicular to the normal). Our region-based recursive least-squares method is impervious to noise and missing data and significantly outperforms a previous boundary-based method using spheres[21]. This robustness to missing data is crucial to extending the method to surfaces of arbitrary smooth geometry, other than spheres. We map the normals of the arbitrary shape onto a sphere, which we can then segment, even when only a subset of the normals is available on the scene. We demonstrate experimentally the accuracy of our method, both in detecting the number of light sources and in estimating their directions, by testing on images of a variety of synthetic and real objects.

1. Introduction

In this paper we consider the problem of multiple illuminant direction detection from a single image, i.e. using an image of an object or scene to recover all real light sources in that environment. Knowledge of illuminant directions is necessary both in computer vision for shape reconstruction, and in image based computer graphics, in order to realistically manipulate existing images. This problem is particularly hard for diffuse (Lambertian) surfaces and directional light sources and cannot be solved using local information only. Lambertian reflectance is the most common type of reflectance. Previous methods that estimate multiple light sources require images of a calibration object of given shape (typically spheres)

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 272–288, 2002.

which needs to be removed from the scene and might cause artifacts. Instead, our method relies instead on partial knowledge of the geometry of the scene and can be used on objects of arbitrary shape. This allows us to possibly use any diffuse object of the scene for illumination calibration.

The interest in computing illuminant directions first arose from shape from shading applications, and was initially focused on recovering a single light source, [5,12,9,22,17]. However, illumination in most real scenes is more complex and it is very likely to have a number of co-existing light sources in a scene. In the context of shape estimation, the ability to automatically detect the light sources allows for more practical methods. For example, there exist methods that use stereo images to compute a partially correct geometry, and then refine it by Shape from Shading under a single light source [18]. These methods can now be extended to multiple light sources, using the partial geometry obtained by stereo to compute the parameters of the light sources with the methods proposed in this paper. More recently, with the advent of image based rendering methods, it is crucial to have knowledge of the light. An early attempt to recover a more general illumination description [6], modeled multiple light sources as a polynomial distribution. A discussion of the various types of light sources can be found in [8]. More recently, with the advent of image based methods in computer graphics, estimation of illumination parameters from images is necessary, in order to compensate for illumination artifacts, and also to allow super-imposition of synthetic images of new objects into real scenes. Most such methods need to use a calibration object of fixed shape, typically a sphere. In [3] a specular sphere is used as a light probe to measure the incident illumination at the location where synthetic objects will be placed in the scene. Such a sphere though might have strong inter-reflections with other objects of the scene, especially if they are close to the it. Using the Lambertian shading model, Yang and Yuille [20] observed that multiple light sources can be deduced from boundary conditions, i.e., the image intensity along the occluding boundaries and at singular points. Based on this idea, Zhang and Yang [21] show that the illuminant directions have a close relationship to critical points on a Lambertian sphere and that, by identifying most of those critical points, illuminant directions may be recovered if certain conditions are satisfied. Conceptually, a critical point is a point on the surface such that all its neighbors are not illuminated by the same light sources. However, because the detection of critical points is sensitive to noise, the direction of extracted real lights in not very robust to noisy data. In [13] a calibration object that comprises of diffuse and specular parts is proposed.

Inserting calibration objects in the scene complicates the acquisition process, as they either need to be physically removed before re-capturing the image, which is not always possible, or they need to be electronically removed as a post processing step, which might introduce artifacts in the image. Our proposed method can be applied to objects of known arbitrary geometry, as long as that shape contains a fairly complete set of normals for a least-squares evaluation of the light sources. Thus, it would be possible to estimate the illuminants from the image of a scene, using geometry that is part of the scene. Although estimating the object geometry from the image would be more convenient, current single image computer vision methods do not provide the necessary accuracy for precise illuminant estimation. Such accuracy can be achieved only by more complex multi-image methods [10] that require precise calibration. The idea of using arbitrary known shape, can also be found in the approach of Sato et al. [19], which exploits information of a radiance distribution inside shadows cast by an object of known shape in the scene. Recently, under a signal processing approach [15,1] a comprehensive mathematical framework for evaluation of illumination parameters through convolution is described. Unfortunately, this framework does not provide a method to estimate high-frequency illumination such as directional light sources when the BRDF is smooth as in the Lambertian case. Convolution is a local operation and the problem is ill-posed when only local information is considered [2]. Our method uses global information to overcome this problem, and in this sense, it is complementary to the methods of [15], [1], [11].

In this paper, we propose an illuminant direction detection process that minimizes global error (by a recursive least-squares algorithm [4]). In general, each point of a surface is illuminated by a subset of all the directional light sources in the scene. We propose a novel, robust way to segment the surface into regions (virtual "light patches"), with each region illuminated by a different set of sources. The regions are separated by boundaries consisting of critical points (where one illuminant is perpendicular to the normal of the surface [21]). We extract real lights based on the segmented virtual "light patches" instead of critical points that are relatively sensitive to noise. Since there are more points in a region than on the boundary, the method's accuracy does not depend on the exact extraction of the boundary and can tolerate noisy and missing data better. Furthermore, we can further adjust and merge light patches to minimize the least-squares errors. The number of critical boundaries detected is sensitive to the threshold used in the Hough transform, especially when there are noisy or incomplete data. Fewer boundaries will cause lights to be missed, more boundaries will result in spurious lights to be detected. This is not a problem for our method since the spurious lights will be eliminated during the merging stage that follows the Hough transform. The ability of our method to perform well when the data is not perfect is crucial for the extension of the method from spheres (for which we initially develop it) to arbitrary shapes. When the observed shape is not spherical we map its normals to a sphere, although a lot of normals will be missing. However our method works well even for incomplete spheres, as long as there are enough points inside each light patch for the least-squares method to work correctly. Our method to detect multiple illuminant directions consists of the following five steps:

1. We identify critical points (defined in the same manner as [21]) in the image, using two consecutive windows using a standard recursive least-squares algorithm[4].
2. We group critical points into different critical boundaries by the Hough Transform.

3. These critical boundaries segment the image into different separate light patches and we calculate a virtual light for each light patch.
4. We adjust and merge critical boundaries by minimizing the average least-squares error of all light patches. If two critical boundaries are too close to each other, i.e. the angle between them is less than a threshold angle, then we merge these two critical boundaries.
5. We extract real lights based on these segmented virtual light patches instead of critical points that are relatively sensitive to noise. The absolute intensities of real lights can be determined only if the albedo is known.

We performed a series of experiments which demonstrate the superior accuracy of our method. Experiments with synthetic data from a sphere and a vase model allow us to recover accurately up to fifteen light sources from a sphere image, with a 0.5 degree average error. We also show that our method performs well even with added noise (error less than 1 degree for five lights). We successfully detect five lights from the image of a real ball made of rubber, even though the surface is rather rough and at various parts deviates from the Lambertian reflectance assumption. Our results compare favorably with the boundary-based method in [21] for the sphere experiments. We perform experiments on two different non-spherical objects, the synthetic images of a vase and the real images of a toy duck under four illuminants. The algorithm performs accurately, despite the increased levels of noise, both in the reflectance and in the acquisition of the normals.

The rest of this paper is structured as follows: Section 2 describes the notion of critical points and their properties as they pertain to our problem. Section 3 describes our basic algorithm and extensions that make it robust to noise and missing data. These properties of our algorithm allow us to introduce a mapping of normals from arbitrary geometry to the shape of a sphere. This is the key to applying our algorithm to objects of arbitrary shape in Section 4. We present our experiments on synthetic and real data in Section 5 and conclude with future working Section 6.

2. Critical Points

Definition 1. *Given an image, let $\mathbf{L}_i$, $i = 1, 2, \ldots$, be the light sources of the image. A point in the image is called a critical point if the surface normal at the corresponding point on the surface of the object is perpendicular to some light source $\mathbf{L}_i$.*

2.1. Observation Model

We assume that images are formed by perspective or orthographic projection and the object in the image has a Lambertian surface with constant albedo, that is each surface point appears equally bright from all viewing directions:

$$L = \rho \mathbf{I} \cdot \mathbf{n} \tag{1}$$

where L is the scene radiance of an ideal Lambertian surface, ρ is the albedo, $\mathbf{I}$ represents the direction and amount of incident light, and $\mathbf{n}$ is the normal to the surface.

2.2. Sphere Model

Initially, we develop our algorithm using a sphere model. We will show later in this paper how to extend this to objects of arbitrary shape.

- We assume the observed object is a sphere with Lambertian reflectance properties whose physical size is already known.
- For light sources whose direction is co-linear with the lens axis of the camera and the angle with OS is less than ω (Fig. 1(a)) depending on the resolution, the best possible result is their equivalent frontal light source $\mathbf{L}_{frontal}$.

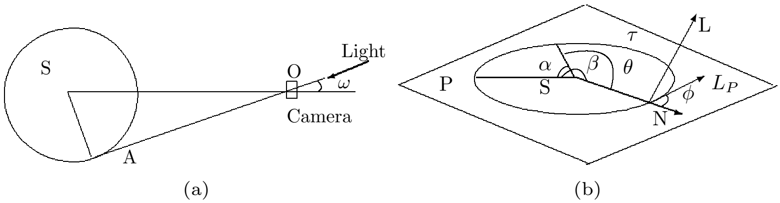


Fig. 1. (a) minimum detectable angle ω. (b) $\mathbf{L}$ and its projection $\mathbf{L_P}$ onto plane $\mathbf{P}$.

It has been proven in [21] that it is not possible to recover the exact value of the intensity of any individual light source among four (or more) pairs of antipodal light sources. However, this kind of situation, i.e. an object illuminated by antipodal light sources, happens rarely, so for simplicity in the rest of this paper, we will make an additional assumption that there are no antipodal light sources.

2.3. Sphere Cross Section with a Plane P

Let $\mathbf{P}$ be an arbitrary plane such that $\mathbf{S}$, the center of the sphere, lies on it (Fig. 1(b)), $\mathbf{L}_i$, $i = 1, 2, \ldots$, be the light sources of the image and $(\mathbf{L})_{iP}$ their projections on $\mathbf{P}$. A point on the arc τ can be specified by its corresponding angle parameter in $[\alpha, \beta]$ using the following proposition [21]:

Proposition 1. *Consider an angle interval* $[\alpha, \beta]$ *of a sphere cross section (Fig. 1(b)). We can always find a partition* $\theta_0 = \alpha < \theta_1 < \ldots < \theta_n = \beta$ *of the interval* $[\alpha, \beta]$ *such that in each* $[\theta_{i-1}, \theta_i]$ *we have* $E(\theta) = b_i \sin\theta + c_i \cos\theta$ *for some constants* b_i *and* c_i, $1 \leq i \leq n$ *(Fig. 2(a)), where* $E(\theta)$ *is the intensity function along the arc* τ.

Intuitively, (b_{i-1}, c_{i-1}) represents the virtual light source of the $[\theta_{i-2}, \theta_{i-1}]$ part, and (b_i, c_i) of the neighboring $[\theta_{i-1}, \theta_i]$ part. These two light sources will be different, if each of these two parts is lit by a different illuminant configuration. More formally, Proposition 2 (from [21]) describes the difference between (b_{i-1}, c_{i-1}) and (b_i, c_i):

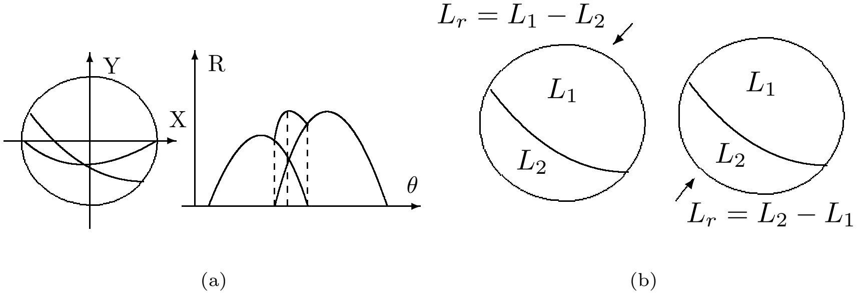


Fig. 2. (a) the xy-space and the θR-space for the case with two sources. (b) Illustration of real light pre-direction. $\mathbf{L}_r$ is the real light direction.

Proposition 2. *In the configuration of Proposition 1, for any $1 \leq i \leq n$, we define Λ_i as the index set of real light sources contributed to the $[\theta_{i-1}, \theta_i]$ part of the arc τ. Then the Euclidean distance between two (b_i, c_i) pairs is*

$$\sqrt{(b_i - b_{i-1})^2 + (c_i - c_{i-1})^2} = \sum_{j \in \Lambda' \cup \Lambda''} \|(\mathbf{L}_j)_P\| \tag{2}$$

where $\|(\mathbf{L}_j)_P\|$ is the Euclidean norm, $\Lambda' = \Lambda_{i-1} \setminus \Lambda_i$ (the index set of elements in Λ_{i-1} but not in Λ_i), $\Lambda'' = \Lambda_i \setminus \Lambda_{i-1}$, and $\sum_{j \in \Lambda_i} \mathbf{L}_j$ is the virtual light source corresponding to [θ_{i-1}, θ_i].

Propositions 1 and 2 show that the difference between (b_{i-1}, c_{i-1}) and (b_i, c_i) will be maximized at a critical point for these two virtual light sources. So, possible critical points can be detected by thresholding $\|(L_j)_P\|$.

2.4. Properties of Critical Points

Let Σ be the set of all critical points and Ω be the space of the sphere image. Then intuitively Σ will cut Ω into a decomposition, i.e.

$$\Omega = (\bigcup_{i \in I} u_i) \bigcup \Sigma \tag{3}$$

where each $u_i \subset \Omega$ is a subset of R^2 which does not contain any critical points and **I** is an index set.

Proposition 3. *Given a decomposition (3), for any u_i there exists an equivalent light source* **L** *such that the image formed by the corresponding surface part of u_i illuminated by* **L** *is exactly the same as the image on u_i.*

Proposition 2 already provides us with a criterion to detect critical points on the sphere based on the distance between (b_i, c_i) pairs. Unfortunately, this criterion greatly depends on the intensities of virtual light sources, $\sum_{j \in \Lambda' \cup \Lambda''} \|(\mathbf{L}_j)_P\|$, which are projected on the plane with respect to each different cross section. To locate the critical points more accurately, we provide another way to detect critical points on each cross section. Instead of using the distance between (b_i, c_i) pairs, we can use the tangent angles defined on the intensity curve (Fig. 3(a)).

Proposition 4. *Along a sine curve, at a critical point θ_c, the inner angle γ between two tangent lines of each side $(\mathbf{T}_1, \mathbf{T}_2)$ will be larger than 180 degrees:*

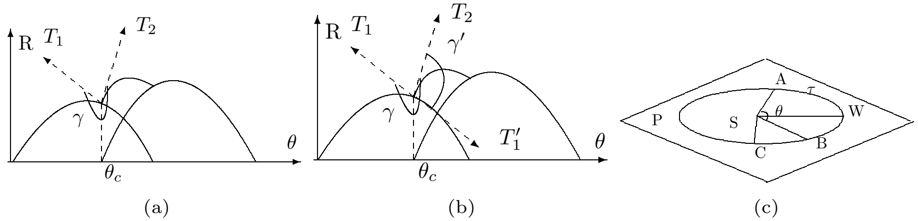


Fig. 3. (a) inner angle γ. (b) Angles between two tangent lines. (c) a part of arc τ, AB, is covered by two consecutive windows AW and WB.

Proof. We define the sine curve contributing to this critical point θ_c is $x(\theta) = \kappa \sin(\xi(\theta - \theta_c)), (\kappa, \xi > 0)$ (Fig. 3(b)).Then the combined curve function will be

$$f(\theta) = \begin{cases} \sin(\theta) & (\theta \leq \theta_c) \\ \sin(\theta) + \kappa \sin(\xi(\theta - \theta_c)) & (\theta > \theta_c) \end{cases} \tag{4}$$

We therefore get the tangent lines:

$$\begin{aligned} T_1' &: f'_{\theta_c^-} = \cos \theta_c \\ T_2 &: f'_{\theta_c^+} = \cos \theta_c + \kappa \cos(\xi(\theta_c - \theta_c)) \cdot \xi = \cos \theta_c + \kappa \xi \end{aligned} \tag{5}$$

So the angle between T_1' and T_2 is $\gamma' = \arctan(\cos \theta_c + \kappa \xi) - \arctan(\cos \theta_c)$. Since $\kappa \xi > 0$,

$$\arctan(\cos \theta_c + \kappa \xi) > \arctan(\cos \theta_c) \Rightarrow \gamma' > 0, \tag{6}$$

that is, $\gamma = 180° + \gamma' > 180°$.

3. Real Light Detection

3.1. Critical Point Detection

From Proposition 1 we know that, for every cross section of the sphere with a plane $\mathbf{P}$ such that $\mathbf{S}$, the center of the sphere, lies on $\mathbf{P}$ (illustrated in figures 2(a),3(c)), there is a partition $\theta_0 = a < \theta_1 < \ldots < \theta_n = \beta$ of the angle interval $[\alpha, \beta]$ such that in each $[\theta_{i-1}, \theta_i]$, we have $E(\theta) = b_i \sin \theta + c_i \cos \theta$ for some constants b_i and c_i, $1 \leq i \leq n$. By applying a standard recursive least-squares algorithm [4], we can use two consecutive windows to detect the local maximum points of inner angles γ and distance defined by Eqn. (2). Starting from an initial point A, any point B on the same arc can be determined uniquely by the angle θ between SA and SB. Then we cover this part AB by two consecutive windows AW and WB (Fig. 3(c)).

With points B and W moving from the beginning point A of the visible part of the circle to its ending point C along the arc τ, we could estimate b_i and c_i from the data in the two consecutive windows AW and WB respectively. Once a local maximum point of Eqn. (2) is detected, it signifies that we have included at least a 'critical point' in the second window WB. Because the inner angle γ defined in Proposition 4 is very sensitive to noise, we use two different criteria simultaneously to detect critical points. First we examine the distance defined in Proposition 2, then if the distance is above a threshold $T_{distance}$, we try to locate the critical point by searching for the maximum inner angle γ along the curve. In practice, for the distance criterion threshold $T_{distance}$, we use a ratio T_{ratio} instead of the direct Euclidean norm to normalize for the varying light intensities. T_{ratio} is calculated from Proposition 2:

$$T_{ratio} = \frac{\sqrt{(b_i - b_{i-1})^2 + (c_i - c_{i-1})^2}}{max\{\sqrt{b_{i-1}^2 + c_{i-1}^2}, \sqrt{b_i^2 + c_i^2}\}} \tag{7}$$

where (b_{i-1}, c_{i-1}) and (b_i, c_i) are calculated from the two consecutive windows AW and WB. Therefore, we can keep growing the first window AW to find critical point p_c using the recursive least-squares algorithm again. Then we fix the initial point A at p_c and keep searching for the next 'critical point' until we exhaust the whole arc τ.

3.2. Segmenting the Surface

Definition 2. *All critical points corresponding to one real light will be grouped into a cut-off curve which is called a critical boundary.*

Intuitively, each critical boundary of the sphere in our model is on a cross section plane through the center of the sphere. Therefore, critical points can be grouped into critical boundaries using the Hough transform in a (ζ, θ) angle-pair parameter Hough space, i.e. we apply the cross-section plane equation in the following form:

$$\begin{cases} x \cdot n_x + y \cdot n_y + z \cdot n_z = 0 \\ n_x = r\cos\theta, n_y = r\sin\theta\cos\zeta, n_z = r\sin\theta\sin\zeta \end{cases} \tag{8}$$

where (x, y, z) is the position of each critical point, (n_x, n_y, n_z) is the normal of the cross-section plane, r is the radius of the sphere and $\zeta, \theta \in [0, 180]$. Typically, we use one-third of the highest vote count in the Hough transform as the threshold above which we detect a (ζ, θ) angle pair as a possible critical boundary.

Although critical points provide information to determine the light source directions [21], they are relatively sensitive to noisy data. Since most real images are not noise free, if we only use the Hough transform to extract critical boundaries, we will very likely find more boundaries than the real critical boundaries, as can be seen from the data in Fig. 4. Noise can either introduce many spurious critical points or move the detected critical points away from their true positions. However, except from the critical points, the non-critical points can also provide important information to determine the light source directions which are much more robust than using critical points only. In this section, we will explore this information in detail.

Light	X Y Z
1	0.03 0.87 0.49
2	0.71 0.66 0.23
3	0.72 0.12 0.68
4	0.30 0.72 0.63
5	0.40 0.61 0.68

(a)

Boundary	Vote #	X Y Z
1	162	0.72 0.65 0.24
2	107	0.37 0.62 0.69
3	86	0.72 0.12 0.68
4	71	0.37 0.71 0.60
5	60	0.31 0.53 0.79
6	56	0.28 0.45 0.85
7	46	0.07 0.88 0.47

(b)

Fig. 4. For an image of a synthetic Lambertian sphere with 5 lights listed in Table (a): We extracted the potential boundaries in (b) by the Hough transform. The critical boundary closest to light 1, is only the seventh boundary to be detected

Definition 3. *Critical boundaries will segment the whole sphere image into several regions, and intuitively, each segmented region is corresponding to one virtual light. Each region is called a virtual light patch.*

Once we get the patches corresponding to each virtual light, the directions of virtual light sources can be calculated.

Let **A**, **B**, **C** and **D** be four points in a patch corresponding to one virtual light source and $\mathbf{n}_A$, $\mathbf{n}_B$, $\mathbf{n}_C$ and $\mathbf{n}_D$ be their normals respectively. From the Lambertian Eqn. (1), augmented by an ambient light term, we have

$$\begin{bmatrix} n_{Ax} & n_{Ay} & n_{Az} & 1 \\ n_{Bx} & n_{By} & n_{Bz} & 1 \\ n_{Cx} & n_{Cy} & n_{Cz} & 1 \\ n_{Dx} & n_{Dy} & n_{Dz} & 1 \end{bmatrix} \cdot \begin{bmatrix} L_x \\ L_y \\ L_z \\ \alpha \end{bmatrix} = \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_D \end{bmatrix} \tag{9}$$

where I_A, I_B, I_C and I_D are brightness of each pixel in the source image corresponding to four points **A**, **B** ,**C** and **D** respectively.

If $\mathbf{n}_A$, $\mathbf{n}_B$, $\mathbf{n}_C$ and $\mathbf{n}_D$ are non-coplanar, we can obtain the direction of the corresponding virtual light source **L**, $[L_x, L_y, L_z]^T$, and the ambient light intensity α by solving the system of equations in (9). Ideally, we would solve for the directions of virtual light sources by using four non-coplanar points from corresponding patches. Due to computation and rounding errors, four non-coplanar points are not always enough for us to get a numerically robust estimate of the direction of a virtual light source. Furthermore, it is not necessary that we can always find several non-coplanar points in an interval of an arc in some plane **P** as described above. To solve these problems we apply our method in two dimensions instead of one dimension. Thus, we scan the image in both x and y directions, and recover the two dimensional patches that are separated by critical boundaries. Then from each two dimensional patch, we use the internal non-critical points of each virtual light patch to solve for the direction of the virtual light source[1].

Proposition 5. *If a critical boundary separates a region into two virtual light patches with one virtual light each, e.g.* $\mathbf{L}_1$*,* $\mathbf{L}_2$*, then the difference vector between* $\mathbf{L}_1$ *and* $\mathbf{L}_2$*,* $\mathbf{L}_{pre} = \mathbf{L}_1 - \mathbf{L}_2$*, is the real light pre-direction with respect to this critical boundary. Since we have already assumed that there are no antipodal light sources, the real light direction will be either the pre-direction* $\mathbf{L}_1 - \mathbf{L}_2$*, or its opposite* $\mathbf{L}_2 - \mathbf{L}_1$ *(Fig. 2(b)).*

To find out the true directions, we pick a number of points on the surface, e.g. $P_1, P_2, ..., P_k$ and their normals, e.g. $\mathbf{N}_1, \mathbf{N}_2, ..., \mathbf{N}_k$, then the true directions will be the solution of:

$$E(P_j) = \sum_{i \in \Lambda} max(e_i \mathbf{L}_i \cdot \mathbf{N}_j, 0) + \mathbf{L}_v \cdot \mathbf{N}_j, 1 \leq j \leq k. \tag{10}$$

where $\mathbf{L}_v$ is the virtual light source of a possible frontal illuminant whose critical boundaries could not be detected and will be checked as a special case.

Selecting points in the area inside the critical boundaries is a robust way to detect real lights. This can be done using standard least-squares methods [4], [14].

3.3. Finding the Real Lights

After we find all the potential critical boundaries, Proposition 5 provides a way to extract real lights by calculating the light difference vector of two virtual light patches of two sides along the critical boundary. However, one real light might be calculated many times by different virtual light patch pairs, and since our data will not be perfect, they will not be necessary exactly the same vector. We introduce an angle threshold to cluster the resulting light difference vectors into real light groups, that can be approximated by one vector.

By minimizing the least-squares errors of virtual light patches, we are able to merge the spurious critical boundaries detected by the Hough transform, by the following steps:

[1] We only use points that are at least 2 pixels away from the critical boundary for increased robustness to noise.

1. Find initial critical boundaries by Hough transform based on all detected critical points.
2. Adjust critical boundaries. We adjust every critical boundary by moving it by a small step, and a reduction in the least-squares error indicates a better solution. We keep updating boundaries using a “greedy” algorithm in order to minimize the total error.
3. Merge spurious critical boundaries. If two critical boundaries are closer than a threshold angle $T_{mergeangle}$ (e.g. 5 degrees), they can be replaced by their average, resulting into one critical boundary instead of two.
4. Remove spurious critical boundaries. We test every critical boundary, by removing it temporarily and if the least-squares error does not increase, we can consider it a spurious boundary and remove it completely. We test boundaries in increasing order of Hough transform votes (intuitively we test first boundaries that are not as trustworthy)
5. Calculate the real lights along a boundary by subtracting neighboring virtual lights as described in Proposition 5.

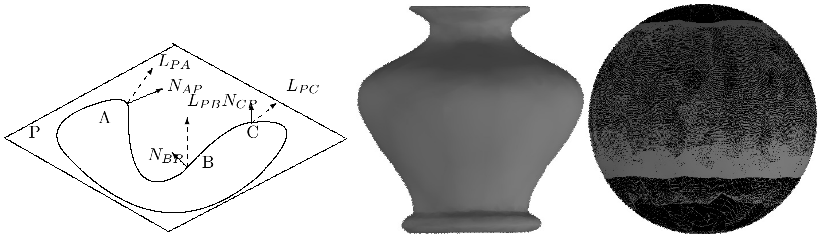
4. Arbitrary Shape

So far, our method requires photographs of a sphere. In this section we extend our method to work with any object of known shape. Obviously, there should exist enough non-coplanar points on the object illuminated by each light to allow for a robust least-squares solution. We assume no inter-reflections and no shadows. These issues will be addressed in future work.

In order to extend our method, we need to address the following issues: unlike the sphere model, we can not uniquely determine a point B on the same arc by the angle θ, which is defined as the angle between two projected normals on plane $\mathbf{P}$, $\mathbf{N}_{AP}$ and $\mathbf{N}_{BP}$, in which A is an initial point. Furthermore, it is possible that in an arbitrary shape model several separated two dimensional patches share a same virtual light which is illustrated in Fig. 5(a).

To avoid such problems, we map the normals from the surface of the arbitrary shape back to a sphere. We detect all potential critical points based on the points mapped on the sphere.

As expected, not every point on the surface of the sphere will be corresponding to a normal on the surface of the arbitrary shape, so there will be many holes on the mapped sphere, e.g. the black area in Fig. 5(b). Thus, many critical points’ locations will be erroneously calculated even for noise-free data. Consequently, the critical boundaries calculated by the Hough transform based on these critical points might not be correct or even far away from their correct positions in some cases. Since we can not recover these missing data from the original image, it is impossible to adjust the critical boundaries detected by the Hough transform itself. On the other hand, as long as the critical boundaries are not too far from the truth, the majority of the points in a virtual patch will still correspond to the correct virtual light (especially after the adjustments steps described in Sec. 3.3. Thus it is still possible, using sparse points on the sphere,



(a) (b)

Fig. 5. (a) two dimensional patches containing points A and C respectively share a same virtual light, i.e. $\mathbf{L}_{pA} = \mathbf{L}_{pC}$. (b) Vase and its sphere mapping. Both image sizes are 400 by 400 and the black color represents points which are not mapped by the normals on the shape's surface.

to calculate the true light for each virtual light patch based on Proposition 5. If two points have the same normal but different intensities, we use the brighter one (assuming that the other is in shadow).

5. Experiments

5.1. Synthetic Sphere

We performed a number of experiments with synthetic sphere images to better understand our method. In order to successfully detect the virtual light patch there should exist enough points to obtain a least-squares solution for the system of equations in (9), using at least 10 points in practice. Obviously, the larger the input image, the more lights can be detected from one image. For comparison purposes, we did experiments on images of five random light sources with up to 0.02 percent maximum noise level and the resulting angle estimation errors are even less than the best reported result in [21] for four light sources (Fig. 6(a)) The following parameter values were chosen for the algorithm: sliding window width $w = 14$ pixels (approximately 8 degrees), distance ratio $T_{ratio} = 0.2$ and angle threshold for boundary merging (described in Sec. 3.3) $T_{mergeangle} = 5.0$ degrees. We have also successfully detected 15 lights on 320 by 320 images with a 0.45 degree average error and 1.64 degree maximum error. Results are shown in Fig. 7. In all the segmented light patch images in this report, detected critical points are denoted with green dots. Typical running times for moderate noise levels and seven light sources are in the order of 5 minutes, on a Pentium4 1.8 Ghz processor. The more merging and adjustment steps the algorithm goes through the slower it becomes. Methods to speed up the computation of the least-squares solution, would improve the performance of the algorithm for noisy environments.

5.2. Real Sphere

We used a blue rubber ball illuminated with five light sources. The original photo, taken by an Olympus C3030 digital camera was cropped to 456 by 456 pixels

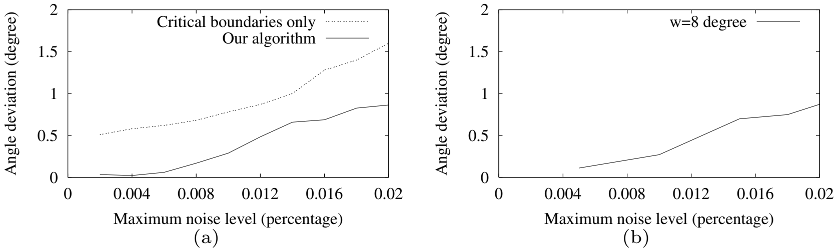


Fig. 6. Error estimation for different noise levels with five random light sources: (a) Synthetic sphere (b) Synthetic vase

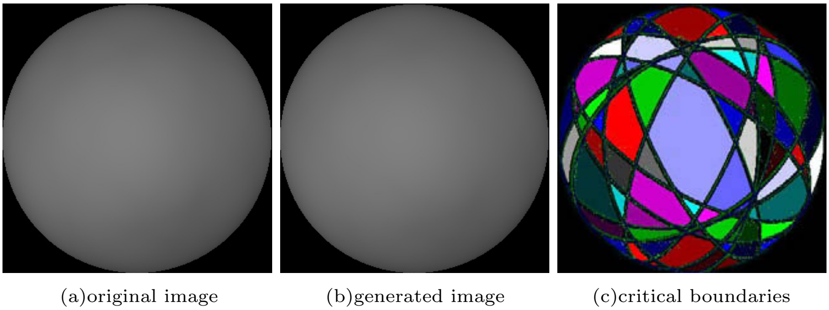


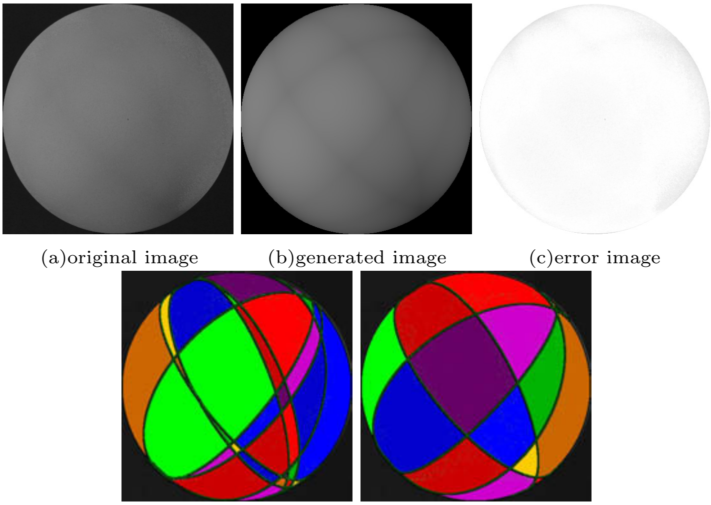
Fig. 7. Synthetic sphere image: a synthetic Lambertian ball illuminated by fifteen light sources. Image size: 320x320. Green dots on (c) represent the detected critical points.

with the ball at the center of the image. Sampling the blue channel as brightness, the color image is converted to gray-scale with 256 levels. The following parameter values were used: sliding window width $w = 30$ pixels (approximately 12 degrees), distance ratio $T_{ratio} = 0.5$ and angle threshold for boundary merging (described in Sec. 3.3) $T_{mergeangle} = 5.0$ degrees. Although the surface did not conform very closely to the Lambertian model, the correct number of light sources was detected and the estimated lights appear to be reasonably close to the truth (no ground data was available).

5.3. Synthetic Arbitrary Shape

We used a vase mess with 14995 vertices and 5004 triangles [2]. The rendered image size is 320 by 320 and the mapped sphere is of the same size. For the vase images we did the same experiments as for the synthetic Lambertian sphere image. The following parameter values were chosen for the algorithm: sliding window width $w = 14$ pixels (approximately 8 degrees), distance ratio $T_{ratio} = 0.2$ and angle threshold for boundary merging (described in Sec. 3.3) $T_{mergeangle} = 5.0$ degrees. We have also successfully detected 15 lights on 320 by 320 images with an 0.56

[2] We used a VRML file downloaded from http://www.vit.iit.nrc.ca/3D/Pages_HTML/3D_Models.html



(a)original image (b)generated image (c)error image

(d)initial (e)resulting critical boundaries

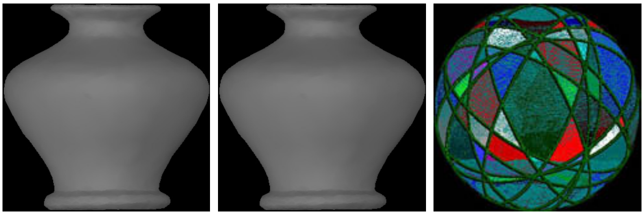
Fig. 8. Real sphere image: an almost Lambertian rubber ball with five light sources. Image size: 456x456. (a) the original image, (b) the generated image of a Lambertian ball with the five light sources extracted from (a), (c) the error image: darker color means higher error, (d) the initial eight boundaries and virtual light patches extracted by the Hough transform, and (e) the resulting critical boundaries and virtual light patches calculated by our algorithm, three out of the initial eight boundaries were automatically merged and the locations of the other five boundaries were automatically adjusted.

degree average error and 2.09 degree maximum error. Experiments on images of five random light sources with up to 0.02 percent maximum noise level are reported in Fig. 6(b).

5.4. Real Arbitrary Shape

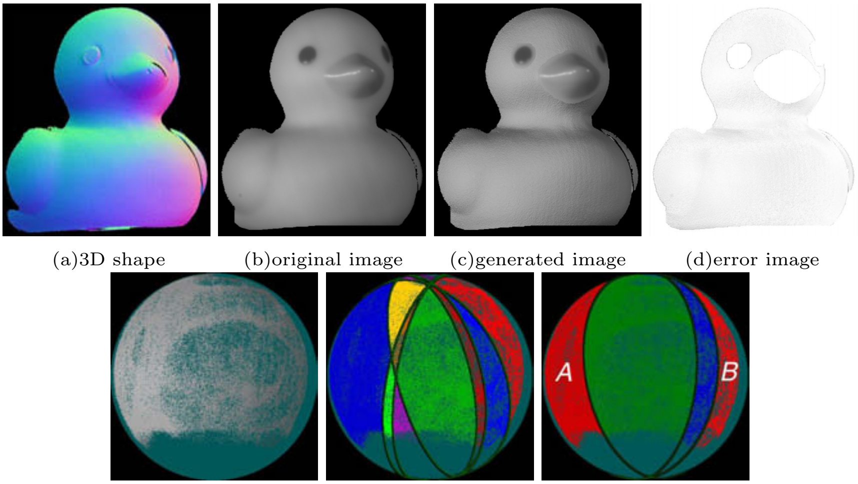
We used a rubber duck illuminated by four light sources. The original image and 3D geometry was captured by the range scanner system described in [7]. In Fig.10(a) we can see that there are some inaccuracies and noise on the recovered 3D shape. The cropped image is 531 by 594 pixels with the duck at the center of the image. Based on the size of the duck, the mapping sphere is a 400 by 400 Lambertian ball. The following parameter values were chosen for the algorithm: sliding window width $w = 30$ pixels (approximately 13.5 degrees), distance ratio $T_{ratio} = 0.5$ and angle threshold for boundary merging (described in Sec. 3.3) $T_{mergeangle} = 5.0$ degrees.

In Fig. 10(e) we can see that in the rightmost region of the mapping sphere one light critical boundary is lost due to data sparseness (as the light is nearly frontal) and cannot be recovered by other critical points or the Hough transform itself. But if we examine the two outmost regions, it is still possible to recover



(a) original image (b) generated image (c) critical boundaries

Fig. 9. Synthetic vase image: a synthetic Lambertian vase with fifteen light sources. Image size: 320x320. (a) the original image, (b) the generated image of a Lambertian vase with the fifteen light sources extracted from (a), (c) the resulting critical boundaries and virtual light patches calculated by our algorithm.



(a)3D shape (b)original image (c)generated image (d)error image

(e)mapping sphere (f)initial (g)resulting critical boundaries

Fig. 10. Real arbitrary shape image experiment: a rubber duck illuminated by four light sources. Image Size: 531x594(duck), 400x400(mapping sphere). (a) the 3D shape of the duck frontal surface, R,G,B color values represent the x,y,z components of the normal. Notice that the recovered 3D shape has a high level of noise, (b) the original image, (c) the generated image of a Lambertian duck with the four light sources extracted from (b), where for completeness purposes the eyes and beak were copied from the original image manually as they have different albedo and non-Lambertian reflectance characteristics. The noise in the generated image is due to the inaccuracies in shape estimation. Nonetheless illuminant estimation is still possible, (d) the error image: darker color means higher error, (e) the initial mapping sphere, where blue color represents points which are not mapped by the normals on the shape's surface, (f) the initial six boundaries extracted by the Hough transform, and (g) the resulting critical boundaries calculated by our algorithm, three of the initial six boundaries were automatically removed and the locations of the other three boundaries automatically adjusted. The two outmost regions A and B are a special case where missing lights are calculated based on virtual light patches.

the missing light(s). For example, in this experiment, after calculating all the real lights: $\mathbf{L}_1 = 77.59 \times (0.81, 0.49, 0.32)$, $\mathbf{L}_2 = 66.81 \times (-0.72, 0.27, 0.64)$ and $\mathbf{L}_3 = 78.06 \times (-0.95, 0.10, 0.30)$ along three critical boundaries, we calculate the virtual light $\mathbf{L}_{vA}$ and $\mathbf{L}_{vB}$ and then calculate for the two outmost virtual light patches A and B: $\mathbf{L}_4 = \mathbf{L}_{vB} - \mathbf{L}_1 = 61.90 \times (-0.11, -0.33, 0.94)$ and $\mathbf{L}_5 = \mathbf{L}_{vA} - \mathbf{L}_2 - \mathbf{L}_3 - \mathbf{L}_4 = 0.00 \times (0.00, 0.00, 0.00)$. $\mathbf{L}_4$ is clearly a missing light whereas $\mathbf{L}_5$ is not.

Notice that the recovered 3D shape in Fig. 10(a) has a high level of noise. The noise in the generated image in Fig. 10(c) is due to the inaccuracies in shape estimation. Nonetheless illuminant estimation is still possible.

High resolution versions of the experiment images in this paper can be found at http://www.cs.sunysb.edu/ samaras/res/multilight.html

6. Conclusions and Future Work

In this paper we presented a method for the estimation of multiple illuminant directions from a single image. We do not require the imaged scene to be of any particular geometry (e.g. a sphere). This allows our method to be used with the existing scene geometry, without the need for special light probes when the illumination of the scene consists of directional light sources. In our experiments with real images, the proposed algorithm performs well when the known geometry is imaged by 300 by 300 pixels at least. Experiments on synthetic and real data show that the method is robust to noise, even when the surface is not completely Lambertian. Future work includes study of the properties of arbitrary surfaces, so that we can avoid the intermediate sphere mapping, speeding up of the least-squares method and extending the method to non-Lambertian diffuse reflectance for rough surfaces. Furthermore, we intend to explore combinations of our method with shadow based light estimation methods and with specularity detection methods.

Acknowledgments. Support for this research was provided by DOE grant MO-068. The authors would like to thank Professors M. Subbarao and P. Huang and their graduate students for help with data acquisition and Professor G. Moustakides for useful discussions on adaptive estimation.

References

1. Basri, R., Jacobs, D.: Lambertian Reflectance and Linear Subspaces. ICCV (2001) 383–390
2. Chojnacki, W., Brooks, M.J., Gibbins D.: Can the Sun's Direction be Estimated prior to the Determination of Shapes? Australian Jnt. Conf. on A.I. (1994) 530–535
3. Debevec, P.: Rendering Synthetic Objects Into Real Scenes. SIGGRAPH (1998). 189–198
4. Haykin, S.: Adaptive Filter Theory. Prentice Hall, Englewood Cliffs (1986)
5. Horn, B.K.P., Brooks, M.J.: Shape and Source from Shading. IJCAI (1985) 932–936

6. Hougen, D.R., Ahuja, N.: Estimation of the Light Source Distribution and Its Use in Integrated Shape Recovery from Stereo and Shading. ICCV (1993) 29–34
7. Hu, Q.Y.: 3-D Shape Measurement Based on Digital Fringe Projection and Phase-Shifting Techniques. M.E. Dept. SUNY at Stony Brook (2001)
8. Langer, M.J., Zucker, S.W.: What Is A Light Source? CVPR (1997 172–178
9. Lee, C.H., Rosenfeld, A.: Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System. AI **26(2)** (1985) 125–143
10. Lin, H.Y., Subbarao, M.: A Vision System for Fast 3D Model Reconstruction. CVPR (2001) 663–668
11. Marschner, S.R., Greenberg, D.P.: Inverse Lighting for Photography. Fifth Color Imaging Conference (1997) 262–265
12. Pentland, A.P.: Finding the Illuminant Direction. JOSA **72** (1982) 448–455
13. Powell, M.W., Sarkar, S., Goldgof, D.: A Simple Strategy for Calibrating the Geometry of Light Sources. PAMI **23** (2001) 1022–1027
14. Press, W.H., Teukolsky, S.A., Vettering, W.T., Flannery, B.P.: Numerical Recipes in C. Cambridge University Press. (1992)
15. Ramamoorthi, R., Hanrahan, P.: A Signal-Processing Framework for Inverse Rendering. SIGGRAPH (2001) 117–128
16. Ramamoorthi, R., Hanrahan, P.: An Efficient Representation for Irradiance Environment Maps. SIGGRAPH (2001) 497–500
17. Samaras, D., Metaxas, D.: Coupled Lighting Direction and Shape Estimation from Single Images. ICCV (1999) 868–874
18. Samaras, D., Metaxas, D., Fua, P., Leclerc, Y.: Variable Albedo Surface Reconstruction from Stereo and Shape from Shading. CVPR (2000) I:480-487
19. Sato, I., Sato, Y., Ikeuchi, K.: Illumination Distribution from Brightness in Shadows. ICCV (1999) 875–883
20. Yang, Y., Yuille, A.: Sources From Shading. CVPR (1991) 534–439
21. Zhang, Y., Yang, Y.H.: Illuminant Direction Determination for Multiple Light Sources. CVPR (2000) 269–276
22. Zheng, Q., Chellappa, R.: Estimation of Illuminant Direction, Albedo, and Shape from Shading. PAMI **13(7)** (1991) 680–702