

Discovering Point Lights with Intensity Distance Fields

Edward Zhang
University of Washington
Seattle, WA
`edzhang@cs.washington.edu`

Michael F. Cohen
Facebook, Inc.
Seattle, WA
`mcohen1@fb.com`

Brian Curless
University of Washington
Seattle, WA
`curless@cs.washington.edu`

Abstract

We introduce the light localization problem. A scene is illuminated by a set of unobserved isotropic point lights. Given the geometry, materials, and illuminated appearance of the scene, the light localization problem is to completely recover the number, positions, and intensities of the lights. We first present a scene transform that identifies likely light positions. Based on this transform, we develop an iterative algorithm to locate remaining lights and determine all light intensities. We demonstrate the success of this method in a large set of 2D synthetic scenes, and show that it extends to 3D, in both synthetic scenes and real-world scenes.

1. Introduction

Estimating the lighting in a scene has a long history in computer vision and graphics, and is a key subproblem in many areas such as reconstruction from shading cues (e.g. [8, 13, 33]), where inferred shape depends heavily on the incident lighting, or augmented reality (e.g. [4, 30]), where virtual objects must be lit consistently with the real world to appear convincing. The majority of prior work assumes *distant illumination*, where lighting incident on a surface does not vary spatially and only depends on the normal of the illuminated surface. Distant illumination, however, is insufficient to represent the lighting in many environments, such as indoor scenes, which are commonly lit by a small number of local light emitters.

Compared with distant illumination, inferring discrete, local lighting models introduces two new unknowns: *cardinality* (the number of lights) and *position* (where the lights are), in addition to *intensity*. Estimating cardinality and position is especially difficult when emitters cannot be directly imaged, such as when a light is located inside an alcove near the ceiling, too high to be seen from human height. Can we resolve the questions of cardinality and position, even in the ideal

case? More concretely, given all the information about the scene except the lighting (i.e. given geometry, materials, and final lit appearance), we seek to completely recover the number of lights, their positions, and their intensities. We call this the light localization problem.

In this paper, we introduce a method for solving the light localization problem. Our key contributions are: (1) a formulation of the problem for an unknown number of local, discrete emitters; (2) a novel scene transform that proposes multiple candidate light positions based on the reflected light in the scene; and, (3) an iterative algorithm that uses the light proposals to recover the full set of scene illuminants, including positions and intensities. We show that the method works on synthetic and real-world scenes; though the solution can be inherently ambiguous (e.g., two very close lights may be recovered as a single light), our approach is generally able to arrive at a low-error solution.

2. Related Work

The distant illumination model introduced by [4] is the most widely used lighting model in inverse problems. We focus our discussion on prior light estimation works that go beyond distant illumination, with methods to estimate light cardinality and position.

Several works attempt to decompose distant illumination into the combination of a discrete number of directional light sources. Earlier works identify occlusion boundaries, referred to as critical points, on a diffuse sphere [29, 31]. Wang and Samaras [26] examine the regions within the boundaries, avoiding the issue of detecting critical points, and subsequently [27] extend this framework for arbitrary reference objects. Lopez-Moreno et al [16] iteratively add directional sources based on reconstruction error. Our algorithm performs a similar iterative light discovery process. However, we do not rely on the presence of critical points and thus we are more robust to missing data.

Many works relax the distant illumination assumption by extending the parameterization of each di-

rectional light to include a distance [7, 24]. This direction-distance formulation, however, remains an object-centric approach, and lacks spatial variation.

A few works directly optimize the location of a single point source in a scene, without the direction-distance formulation, via a least-squares approach [3, 9, 18]. These methods avoid considering occlusions, as hard shadows result in a discontinuous objective function.

Instead of optimizing for the lighting across all observed surfaces, another approach for locating light emitters is to look at strong, discrete lighting effects, such as shadow boundaries, specularities, and symmetries. These methods are often very accurate, but rely on the presence of appropriate conditions in the scene and will not work in general cases. The most common approach is to use specular surfaces (usually light probes), identifying maxima as light sources and triangulating corresponding specularities [1, 10, 12, 14, 15, 20]. Triangulation of specularities can also be used to recover the shapes of area light sources [22, 32]. All of these works rely on both the presence of strongly specular surfaces (manually inserted into the scene or already present in the scene) and on the visibility and distinctness of light sources in the specular surface. Triangulation can also be used when light sources are directly imaged from multiple views [5, 6]. These methods assume that the light sources will be directly visible from the camera viewpoints.

Cast shadows are another strong lighting cue, and have been widely used in distant illumination environments [11, 15, 17, 21], but have also been used to identify near point sources [23] via triangulation. For triangulation-based methods, there must not only exist strong shadow edges in the scene, but they must also be identified and associated with object boundaries.

Other works [19, 25] are able to recover the position of a single point source with varying intensity distribution by analyzing symmetries of the light cast on planes. Although these assume that planar surfaces exist near the light source, this assumption is reasonable in most indoor scenes. Like these methods, our candidate light selection process also analyzes the distribution of direct illumination to directly extract light positions without optimization; however, illuminated surfaces in our work can have arbitrary shape.

3. Candidate Light Proposal

3.1. Formalizing the Light Localization Problem

Consider a scene with known geometry G , lit by a set of unobserved isotropic point lights $L = \{(p_i, I_i)\}$, with positions p_i and intensities I_i . We assume that, at each point in the scene, we are given the BRDF $f(x, \omega_i, \omega_o)$

specifying the proportion of the incident light from direction ω_i scattered in direction ω_o . We make a set of observations $Q = \{(x, \omega)\}$ of points in the scene; each of these observations $q \in Q$ measures the outgoing radiance in a direction ω from a point in the scene x due to reflected direct illumination from L . Let these radiance measurements be $B(q)$; $B(q)$ is thus a surface light field [28]. Note that, while observed radiances will normally include reflected direct and indirect illumination, we can extract $B(q)$ containing only directly reflected light by using the observations themselves to estimate and remove the indirect illumination.¹ The **light localization** problem is as follows: Given $G, f(x, \omega_i, \omega_o)$, and $B(q)$, completely recover L .

3.2. Deriving the Intensity-Distance Function

At a point x on the scene surface, the outgoing radiance in direction ω due to a single, unoccluded, isotropic point light at point p with intensity I is

$$\begin{aligned} B(x, \omega) &= I \frac{V(x, p)f(x, \hat{l}, \omega)(n(x) \cdot \hat{l})_+}{\|l\|^2} \\ &= I \frac{V(x, p)f(x, \frac{p-x}{\|p-x\|}, \omega)(n(x) \cdot (p-x))_+}{\|p-x\|^3} \end{aligned} \quad (1)$$

by expanding $l = p - x$, $\hat{l} = \frac{l}{\|l\|}$. Here, the visibility term $V(x, y)$ is 0 if x and y are mutually occluded and 1 otherwise, $n(x)$ is the normal at x , and $(m)_+ = \max(m, 0)$.

We define the shading function $S(q, p)$ as

$$S(q, p) \equiv \frac{V(x, p)f(x, \frac{p-x}{\|p-x\|}, \omega)(n(x) \cdot (p-x))_+}{\|p-x\|^3} \quad (2)$$

such that $B(q) = IS(q, p)$ for the single light case.

We can now define a function that, for any possible light position p in a single-light scene, tells us what intensity a point light must have if it were at that position, given the information about observation q :

$$I(q, p) \equiv \frac{B(q)}{S(q, p)}. \quad (3)$$

¹That indirect illumination can be directly removed may be surprising. In fact, the observations themselves are rays that comprise the indirect illumination to the rest of the scene. We can thus use the observed radiances to estimate the indirect incident illumination on the rest of the scene. Practically, if the scene is largely diffuse and the origins of the observed rays cover most of the scene, then we can obtain a good estimate of the indirect illumination incident on each point in Q . Then, since we know the BRDF at each point in the scene, we can compute how much of the reflected light was due to this indirect illumination and subtract it from the observation to leave our desired $B(q)$. This process is inspired by the handling of indirect illumination by Zhang et al. [30]

Each of our observations q induces such a function $I(q, p)$. We illustrate an example of $I(q, p)$ for the diffuse case in Figure 1a, and for a specular case in Figure 1b. Note that if the true light parameters were (p_1, I_1) , then by construction, for every observation q , $I(q, p)$ will be equal to the true light intensity at the true light location p_1 : $\forall q, I(q, p_1) = I_1$.

In the case of a multi-light scene,

$$B(q) = \sum_{i=1}^{|L|} I_i S(q, p_i) \quad (4)$$

Expanding and rearranging for I_1 ,

$$\begin{aligned} B(q) &= I_1 S(q, p_1) + \sum_{i=2}^{|L|} I_i S(q, p_i) \\ I_1 &= \frac{B(q)}{S(q, p_1)} - \sum_{i=2}^{|L|} I_i \frac{S(q, p_i)}{S(q, p_1)} \\ &= I(q, p_1) - \sum_{i=2}^{|L|} I_i \frac{S(q, p_i)}{S(q, p_1)} \end{aligned} \quad (5)$$

Both $S(q, p)$ and I_i are nonnegative, so $I_1 \leq I(q, p_1)$. Thus, the $I(q, p)$ derived from the single-light case is an upper bound on the brightness of a light at point p , based on the observation q .

Every observation q induces such a bound, so taking the most restrictive bound gives $I_1 \leq \min_q I(q, p_1)$. We now define $\mathcal{D}(p)$ to be

$$\mathcal{D}(p) \equiv \min_q I(q, p) \quad (6)$$

implying a light at point p can be no brighter than $\mathcal{D}(p)$ given all the observations of the scene.

Since $I(q, p)$ is informally a distance function between points on the scene surface and points inside the scene, taking the minimum distance to the scene boundary results in a distance field, which we refer to as the **Intensity-Distance Field** (IDF), given by $\mathcal{D}(p)$. An example of the Intensity-Distance Field for a single-light case in a diffuse scene is shown in Figure 1c.

Traditional specularity triangulation methods interpret a single bright specular highlight as a strong indicator of a light source in the mirror direction, while our formulation effectively uses the contrapositive of this reasoning: a *lack* of specular highlight means that it is *unlikely* for a light to fall along the mirror direction. In this work we primarily work with diffuse scenes with no occlusions, since these are effectively the hardest case for light localization. More detail on specularity and occlusion is given in the supplementary material.

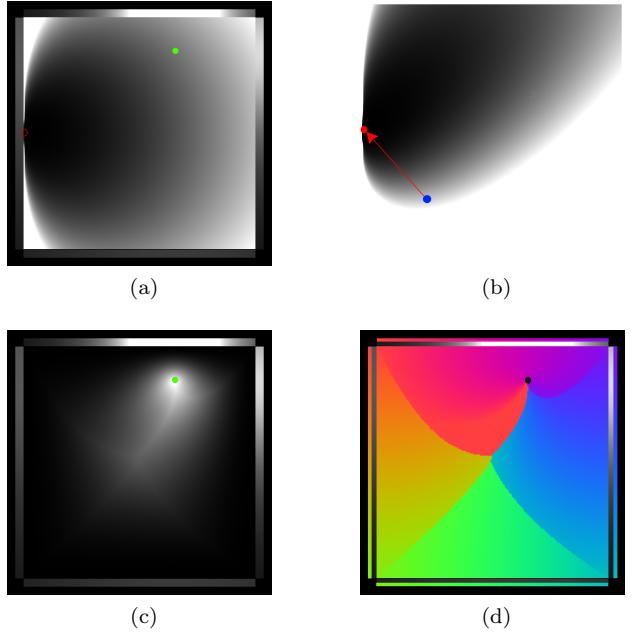


Figure 1: Derivation of the Intensity-Distance Field in a one-light scene. A diffuse box-shaped room in 2D is lit by a single light at the green circle. The lit appearance (i.e. $B(q)$) on each wall is shown as a 1D intensity plot. As input, we are given the intensity plots but not the location of the light. 1a shows $I((x_0, \omega), p)$ for p varying across the interior of the scene, where x_0 is the surface in the red circle (note that $I((x_0, \omega), p)$ is independent of ω for diffuse surfaces). For comparison, 1b shows $I((x_0, \omega), p)$ for a microfacet BRDF if x_0 were a shiny surface, with ω pointing along the red arrow. 1c shows $\mathcal{D}(p)$. 1d shows $\mathcal{L}(p)$, where the color at each point in the scene maps to the single point on the scene boundary bounding $\mathcal{D}(p)$ (mapping shown adjacent to intensity plots).

3.3. Limiter Field

A related function for analyzing the IDF is the **Limiter Field**, which tells us which surface point(s) induced the value of $\mathcal{D}(p)$:

$$\mathcal{L}(p) \equiv \pi_x(\operatorname{argmin}_q I(q, p)). \quad (7)$$

Here, $\pi_x(q)$ extracts the position of the observed point from q . Thus, $\mathcal{L}(p)$ tells us which surface most strictly limits the brightness of a potential point light at p . To visualize $\mathcal{L}(p)$ in 2D, we assign each surface point a different color according to a linear gradient, and then map each p in the scene to the color corresponding to $\mathcal{L}(p)$ (Figure 1d). Examining $\mathcal{D}(p)$ and $\mathcal{L}(p)$ for different scene geometries and light configurations (please refer to the supplementary video for examples) suggests that the medial axis of the distance field is often correlated with light positions. The medial axis is defined as the set of points for which $\mathcal{L}(p)$ has multiple values (visually, discontinuities in $\mathcal{L}(p)$).

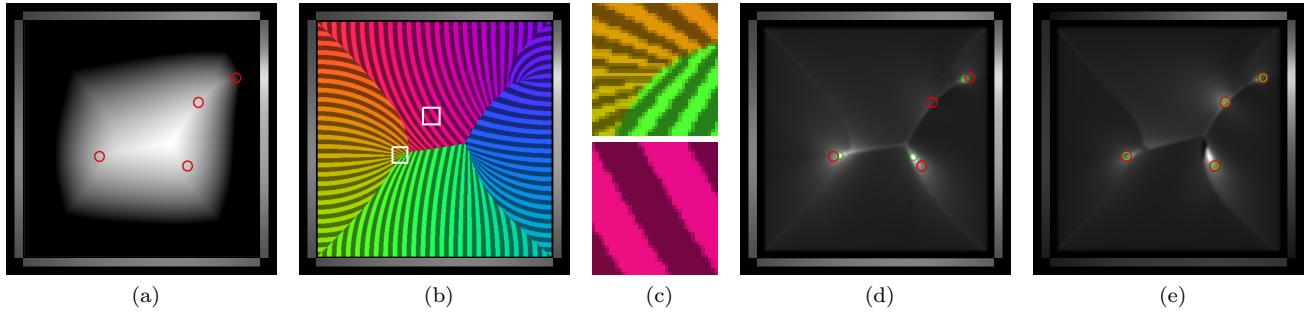


Figure 2: A set of four isotropic point lights (red) lights a diffuse 2D scene. Given only the lighting on the surfaces of the scene shown along the edges of the square, we compute $\mathcal{D}(p)$ (a) and $\mathcal{L}(p)$ (b). A closeup of $\mathcal{L}(p)$ near a light source (c, top) shows about 25 unique colors in a neighborhood, i.e. unique values of $\mathcal{L}(p)$, while away from light sources (c, bottom) only 5 different values exist in the neighborhood. $\mathcal{V}(p)$ plots these counts (d), the peaks of which form our initial estimates for the number of lights and their positions (three green circles). Iterating our refinement algorithm recomputes the $\mathcal{V}(p)$ (e) and reveals the fourth light. The updated estimates then initialize a nonlinear optimization, which successfully recovers the original lighting parameters.

3.4. Voting Function

Finally, to identify candidate light positions, we treat $\mathcal{L}(p)$ as a discrete-valued function (since we measure a finite number of surface radiances at a finite number of points) and construct a voting measure for the light positions. The intuition here is that some surfaces will primarily be lit by a single light, and will thus vote consistently on the position of that particular light. Due to quadratic distance falloff, it is likely that there exists some such set of surfaces for at least one light in the scene (unless the scene lighting is inherently ambiguous, as described in the introduction).

In Figure 2b, we show a refined visualization highlighting the discreteness of $\mathcal{L}(p)$, essentially a Voronoi diagram based on $\mathcal{D}(p)$. To determine the number of surfaces voting for a particular location, we simply count the number of unique values that $\mathcal{L}(p)$ takes in a small neighborhood (Figure 2c). The surfaces corresponding to these values all have similar $I(q, p)$ intensities in the neighborhood, and so are effectively voting for the same intensity of a light at p . The count of locally unique $\mathcal{L}(p)$ values gives us the Voting Function (shown in Figure 2d): for some neighborhood size δ ,

$$\mathcal{V}(p) \equiv |\{\mathcal{L}(p'), ||p - p'| < \delta\}| \quad (8)$$

4. Implementation of Scene Transforms

4.1. Computing the IDF

We construct the Intensity-Distance Transform (both $\mathcal{D}(p)$ and $\mathcal{L}(p)$) directly from Equations 6 and 7, computing $I(q, p)$ for each possible light position p , discretized over a regular grid, and each observation q . This brute force algorithm is trivially parallelizable.

For a 2D scene, we can compute distance fields at interactive frame rates (45ms) at 256×256 resolution.

tion with 400 surface intensity measurements, run on a GTX760M graphics card. Incorporating occlusion computations takes an additional 8ms per primitive.

4.2. Medial Axis

We use the discontinuities in $\mathcal{L}(p)$ to identify the medial axis in a discrete fashion. $\mathcal{L}(p)$ is continuous off of the medial axis; that is, a small change in p will result in a small change in $\mathcal{L}(p)$. More specifically, if $d(x, y)$ is the distance along the scene surface between points x, y , then $d(\mathcal{L}(p), \mathcal{L}(p + \epsilon)) < T$ for a small perturbation ϵ with an appropriate threshold T . Discontinuities resulting in large $d(\mathcal{L}(p), \mathcal{L}(p + \epsilon))$ only occur on the medial axis. Thus, in 2D, we examine the neighbors of a pixel in clockwise order; a pixel is a medial axis pixel if the distance between limiters of any consecutive neighboring pixels is greater than T . In our experiments we set T to 5% of the scene’s radius.

Although we do not directly use the medial axis for light localization, our formulation leads to a method of discretely computing the voting function.

4.3. Voting Function

Our basic definition of $\mathcal{V}(p)$ at a point is the number of different values $\mathcal{L}(p)$ takes in a small neighborhood around that point. With our discrete sampling of $\mathcal{L}(p)$, simply counting unique values in a pixel’s neighbors gives insufficient granularity; for example, examining the immediate neighbors of a pixel can give at most 9 unique $\mathcal{L}(p)$ values. Instead, we note that when $d(\mathcal{L}(p), \mathcal{L}(p + \epsilon)) < T$ (where T is the medial axis discontinuity threshold, as described above), the continuity of $\mathcal{L}(p)$ implies that, for any x in between $\mathcal{L}(p)$ and $\mathcal{L}(p + \epsilon)$, there exists some point p' between p and $p + \epsilon$ such that $\mathcal{L}(p') = x$. Thus, if $d(\mathcal{L}(p), \mathcal{L}(p + \epsilon))$ is large

(but less than T) then $\mathcal{V}(p)$ at p is large. However, if $d(\mathcal{L}(p), \mathcal{L}(p + \epsilon)) > T$, then there is a discontinuity somewhere between p and $p + \epsilon$; not every x between $\mathcal{L}(p)$ and $\mathcal{L}(p + \epsilon)$ will necessarily contribute a vote.

Thus, to compute the vote total at a pixel p , we examine the pixels on the perimeter of a circle with radius r centered on p . The total vote $\mathcal{V}(p)$ is the mean distance between the limiters of consecutive pixels on this circle, as long as the distance is less than the medial axis discontinuity threshold. Larger values of r are required for noisy data, while $r = 1$ pixel usually suffices for the noiseless case.

To extract light proposal candidates, we search for local maxima in the voting function with a value greater than some minimum H . In practice we start with a very high value of $H = |Q|/10$ and reduce until at least one peak is found. We also combine multiple peaks if the distance between them is less than some user-defined threshold (e.g. 10% of the scene radius).

4.4. Moving to Three Dimensions

In 3D scenes, the computation of $\mathcal{D}(p)$ and $\mathcal{L}(p)$ is identical to the computation in 2D. However, this process is no longer real-time. We have to take several orders of magnitude more surface measurements to reduce discretization artifacts in the distance field. Computing the distance field at 512^3 resolution with $|Q| = 60\,000$ took about 1 hour on a Titan X GPU.

Computing $\mathcal{V}(p)$ is slightly more complex in 3D. In 2D, $d(x, y)$ was computed by taking the distance along the perimeter of the scene. In 3D, the analogous approach to compute $\mathcal{V}(p)$ is to triangulate the points on the surface of a sphere centered at p , and then for each triangle with vertices A, B, C let $d(A, B, C)$ be the geodesic area (i.e. area along the surface of the scene) spanned by the triangle $\mathcal{L}(A), \mathcal{L}(B), \mathcal{L}(C)$. In our fairly simple test cases, we approximated the geodesic area with the solid angle spanned by the triangle relative to the center of the scene. While this resulted in some artifacts (e.g. underestimating votes near the corners of boxy scenes) it still gave good localizations.

5. Refinement Algorithm

The candidate light positions proposed by the voting function are often incomplete or inaccurate, for example in Figure 2d. A light might be “hidden” by other lights, such that no surface in the scene is primarily lit only by that light, and thus no peak is produced in $\mathcal{V}(p)$ (e.g. the second light from the right). Even identified lights have inaccurate positions because of the influence of other light sources (e.g. the lower-right light). Therefore we must refine the proposed lights.

We store a set of hypothesis lights, keeping track of how much of the scene’s original appearance remains unexplained by these hypothesis lights. We iteratively increase the intensity of some of the hypothesized lights, refine the positions of all hypothesized lights, and then update the remaining unexplained illumination. We then recompute the voting function, which may reveal new lights not yet in the hypothesis set. Any time a new light is found, we perform a full nonlinear optimization to check for convergence.

5.1. Light Discovery

Light discovery begins by computing the voting function of the current unexplained illumination $B'(q)$,

$$B' = B - \text{RENDER}(G, \text{hypotheses}),$$

where `RENDER` computes the illumination due to direct lighting for all surfaces in the scene.

We then extract a set of candidate light positions from the voting function and associate them with the set of hypothesized lights. Note that not all hypothesized lights will appear in the set of current candidates; for example, if a correctly placed hypothesis light’s intensity reaches the true intensity, then it will completely cancel out the effects of the true light. On the other hand, if there is a candidate light that is not yet in the hypothesis list, then we add it to the hypothesis list with 0 intensity and trigger an optimization step.

Only the lights that are in the current set of candidates will have their intensities incremented in this iteration. Our algorithm terminates in failure if incrementing intensities results in $B'(q) < 0$ for any x .

5.2. Optimization

Our optimization procedure consists of two steps. We first hold the positions of the hypothesized lights constant and then solve for their intensities only. This step is linear and can be directly computed. This step is necessary because the hypothesized intensities for recently discovered lights are likely to be far below their correct intensities. We use these computed intensities and the hypothesized positions as initializations to a Levenberg-Marquardt nonlinear optimization using Ceres Solver [2]. If the resulting error is low enough, then we declare success. This optimization can be accelerated by terminating early if the light positions stray too far from their initial estimates. A failed optimization usually implies that we have not yet found all the lights in the scene.

5.3. Position Refinement

Suppose that all but one of the hypothesis lights were in the correct locations with the correct inten-

sities; if we then take the measured illumination and subtract the hypothesized illumination due to the correct lights, we should be left with the scene illumination as if it were lit only by our incorrect light. But for a single-light scene, our scene transforms will directly give us the true light position.

Based on this intuition, we develop the following position refinement step. For each hypothesis light, we recompute $\mathcal{V}(p)$ based on the scene illumination with the contributions of the other hypothesis lights removed; in other words, we compute the votes using

$$B^i = B - \text{RENDER}(G, \text{hypotheses} \setminus \text{light}_i)$$

We then extract the peaks of $\mathcal{V}(p)$ and set the position of the current hypothesis light to that of its corresponding candidate.

6. Results and Discussion

6.1. 2D Synthetic Data

To validate our method, we generated a synthetic 2D dataset with 400 diffuse square room scenes with varying lighting conditions. We generated 100 random configurations each for two-light, three-light, four-light and five-light scenes, each light having random intensities between 1 and 5. While generating these configurations, we ensured that no light was closer than 5% of the scene’s side length to another light or to a wall.

The scene geometry for our quantitative evaluation is a simple diffuse box scene. This functions as a worst-case setup for light localization, as none of the stronger cues such as specular highlights or occlusions are present. Although scenes comprised of large planar surfaces admit other approaches, our method does not rely on the presence of planar surfaces. Note that we include some examples of our method in non-planar scenes in the supplementary video.

In addition to the noise-free case, we also examine how small amounts of noise affect the performance of our system. We approximate the effects of shot noise in CCD sensors by replacing the true direct intensity value $B(q)$ with a random variable with a Gaussian distribution with mean $B(q)$ and standard deviation $k\sqrt{B(q)}$; here we report results for $k = 0.05$.

For the noiseless case, we compute $\mathcal{V}(p)$ with a radius of 1, and set the threshold for convergence at a MSE of 0.0025. For the noisy case, we compute $\mathcal{V}(p)$ with a radius of 15, and set the threshold for convergence at 105% of the MSE of the ground truth lighting. For both cases, the per-iteration hypothesis light intensity increment was 0.025.

We ran our method on the entire dataset both with and without noise. We show the runtimes and the suc-

$ L $	Success Rates				Runtimes (s)	
	Random	$\mathcal{V}(p)$	Solved	Full	Median	Max
Noiseless						
2	0.77	0.96	1.00	1.00	2	64
3	0.52	0.73	1.00	1.00	3	105
4	0.29	0.35	1.00	1.00	29	250
5	0.26	0.15	0.94	0.97	85	442
Noisy						
2	0.80	0.83	0.78	0.99	2	52
3	0.53	0.66	0.73	0.98	13	492
4	0.32	0.41	0.64	0.95	34	301
5	0.23	0.21	0.38	0.84	80	378

Table 1: Synthetic 2D Dataset results and runtimes. The Random column shows the proportion of randomly initialized nonlinear optimizations which achieved a low-error solution. The $\mathcal{V}(p)$ column gives optimization success rates after a single round of voting, while the Full column corresponds to running our full algorithm. The Solved column gives the proportion of scenes resulting in successful recovery of the ground truth lighting, while the $\mathcal{V}(p)$ and Full columns include low-error solutions that did not recover the ground truth lighting.

cess rates of our algorithm in Table 1. A single computation of $\mathcal{V}(p)$ frequently provides an initialization giving a successful optimization without further iterations of our algorithm. Runtimes were measured on an Intel i7-7700HQ CPU with a GTX1060 GPU.

For the noiseless case, our algorithm can successfully recover the original lighting conditions in all of the two, three, and four light cases. However, it fails in three of the five-light test cases. All of these failures, as well as three other cases in which the ground truth lighting was not recovered, involve clusters of several nearby lights that resulted in some ambiguity. Some of these failure cases are shown in Figures 3-4.

In the noisy case, our algorithm is slightly less successful, although it still matches the original lighting conditions the majority of the time. The winner-take-all nature of the distance field means that noise can cause large constant regions in $\mathcal{L}(p)$ (where the same surface point is the limiting point for a large region), especially closer to the center of the scene. These constant regions hide any useful information from $\mathcal{V}(p)$. This effect, illustrated in Figure 4a-4b, is the cause of all the two- and three-light failure cases, and most of the four- and five-light failure cases. Several of the four- and five-light failure cases are due to spurious peaks, caused by noise, being confused with actual light peaks, while the remaining failures come from scenes similar to Figure 3b. Determining a more robust way of collecting votes, or in general designing a more robust candidate proposal method, is an important area for further investigation. In practice, real-world captures often involve multiple samples of the brightness of each surface; combining these samples can significantly reduce the amount of noise.

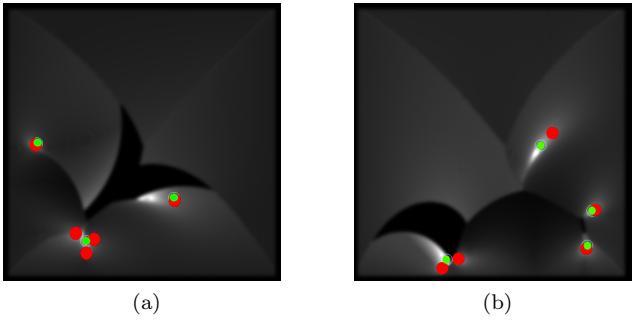


Figure 3: Failure case due to light clustering: In 3a, we show $\mathcal{V}(p)$ where we converged on a low error solution (green) different from the ground truth (red), replacing the lower-left cluster of lights with a single light. In 3b, the cluster of two lights near the bottom of the scene is closer to the wall and a single light cannot fit a low-error solution. $\mathcal{V}(p)$ never manages to distinguish two separate peaks, and we terminate without finding a solution.

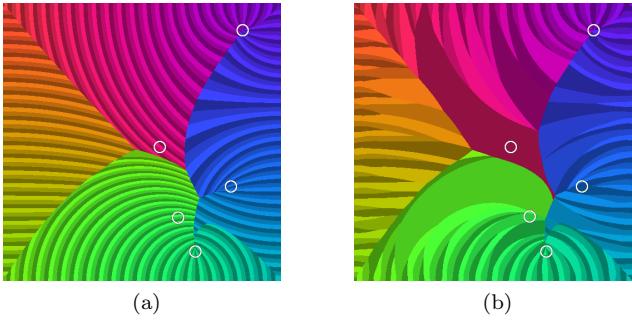


Figure 4: Failure case under noise: We compare a noiseless $\mathcal{L}(p)$, in 4a, with $\mathcal{L}(p)$ of the same scene with noise, in 4b. The region near the leftmost light (near the center of the scene) is completely taken over by just two Voronoi cells. We have no way of computing a meaningful vote in these areas.

As a baseline, we run a Levenberg-Marquardt nonlinear optimizer (using Ceres Solver) with random initialization 100 times on each of the scenes, and compute the proportion of the runs which successfully converge. In this baseline, we assume that the number of lights is known a priori. Despite this extra knowledge, the baseline frequently fails to find a solution reproducing the original illumination conditions.

6.2. 3D Synthetic Data

Beyond 2D experiments, we also ran our algorithm on several synthetic 3D scenes (Figure 5). We informally observe that in 3D, it is more likely for our initial voting function proposals to provide the full solution. This is due to the fact that, in a higher-dimensional space, more points in the interior of a region are close to that region’s boundaries (a well-known implication of the curse of dimensionality). Thus, each light is

more likely to be very close to a scene boundary, and therefore is more likely to be the dominant source of incident illumination for that boundary.

Because of our brute-force IDF approach, each computation of a 3D $\mathcal{V}(p)$ takes about one hour, while each iteration of our algorithm requires numerous $\mathcal{V}(p)$ computations. We expect that investigating alternate methods of computing $\mathcal{D}(p)$, or approaches that avoid computing the full IDF (e.g. focusing on the medial axis), would lead to better performance in 3D scenes.

6.3. Real-World Data

We also demonstrate that our method applies to real-world scenes. We set up two bare light bulbs (approximating isotropic point sources) in a small room. We scanned the scene using the Lenovo Phab2 Pro Tango phone, reconstructing the scene geometry using the Tango software. After using an HDR exposure correction pipeline similar to [30], we project the RGB images onto the geometry, shown in Figure 6a-6b.

To avoid having to recover materials for the entire scene, we take advantage of two properties of our method. First, since we do not rely on having a complete or closed scene, we can compute our scene transforms and optimize over only a subset of scene points. Furthermore, if we choose this subset to consist of diffuse surfaces with identical albedo (e.g. wall points), then $f(x, \omega_i, \omega_o) = \rho/\pi$ for constant ρ , and π/ρ is a global scaling of all $I(q, p)$. This means that the limiter field $\mathcal{L}(p)$ is independent of the actual albedo ρ .

We manually segment out wall vertices to form Q (although this can easily be replaced by an automatic material segmentation pipeline). Because of our diffuse assumption, we can directly take the projected RGB colors of our wall vertices as our $B(q)$.

Using camera images directly for B introduces the issues of noise, indirect illumination, and unmodelled material properties (including specularity). With our diffuse Q , we can ignore ω and combine all the observations of a single wall point by taking the robust minimum (average of the values between the first quartile and the median) of the projected colors. This also removes the erroneous projections of the light stands, bulbs, and glare onto the walls.

For simplicity, in this work we ignore the effects of indirect illumination on our candidate proposal scheme, and assume that incident indirect illumination is negligible relative to direct lighting at the surfaces given by $\mathcal{L}(p')$ for p' near the maxima in $\mathcal{V}(p)$. We model indirect illumination in our optimization as a constant ambient term. Had we instead measured the materials, we would have just removed indirect illumination, as described in Section 3.

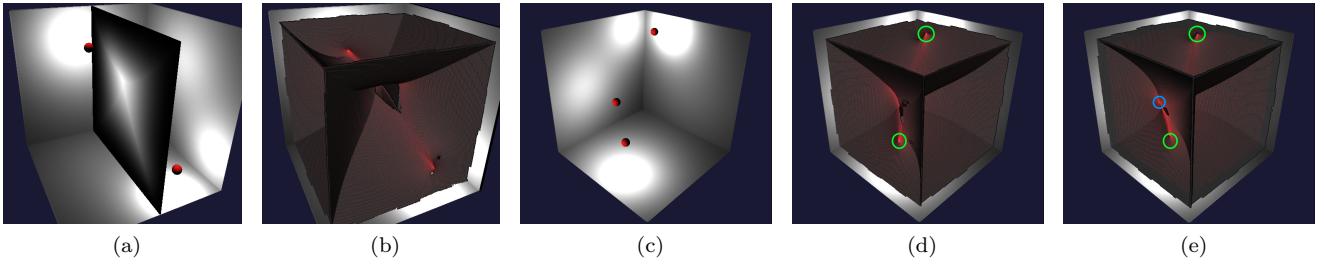


Figure 5: Results on synthetic 3D scenes. In 5a-5b, the maxima of $\mathcal{V}(p)$ directly give the light locations in a two-light scene. We visualize $\mathcal{D}(p)$ on a planar slice in 5a. The medial axis is shown in 5b, colored with $\mathcal{V}(p)$. In a three-light case (5c), the initial $\mathcal{V}(p)$ maxima give the locations of two of the lights (5d, circled in green); after a few iterations the third is revealed (5e, circled in blue).

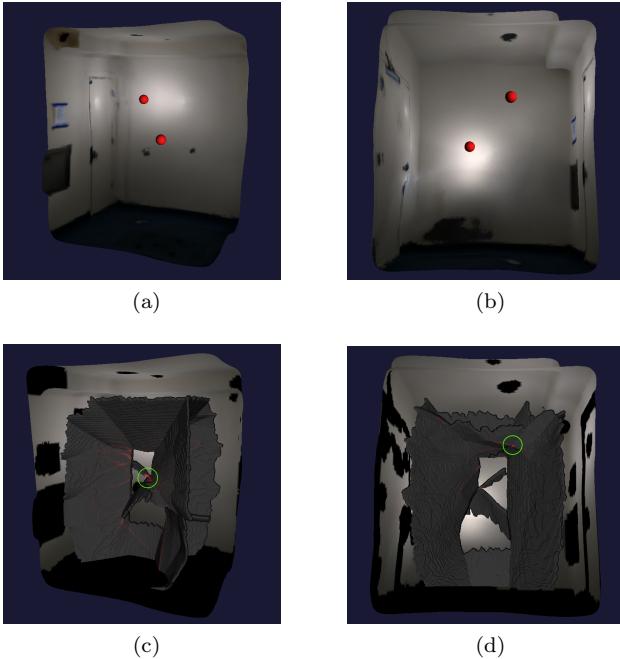


Figure 6: Results of running our method on a scanned 3D dataset. In 6a-6b, we show the scanned scene colored with the averaged observations $B(q)$. The recovered light locations are rendered as red spheres. In the bottom two images, we only show the $B(q)$ used in our computation. In 6c, the initial medial axis and $\mathcal{V}(p)$ are shown, with a clear maximum (circled in green) at the location of one of the light sources. After a few iterations (6d), the second light position is also revealed.

Examining $\mathcal{V}(p)$ shows a clear maximum at one of the true light positions (Figure 6c), and after a few iterations of our refinement algorithm, the other light also appears (Figure 6d), at which point our optimization succeeds. The error in the estimated positions, relative to hand-measured ground truth positions, were 0.05m and 0.08m (in a $2\text{m} \times 2.6\text{m} \times 1.6\text{m}$ room). As the captured images did not give absolute intensities, we compared relative intensities by fixing one of the

light intensities to 1. The estimated intensity for the second light was 1.43, while the measured relative intensity (based on raw images captured with a DSLR) was 1.39.

6.4. Limitations and Future Work

Known Materials The main practical issue with using our method in real-world scenes is the assumption of known materials. It is difficult to separately use a BRDF estimation pipeline and then apply our method because BRDF estimation in scenes is highly dependent on lighting models. While it is straightforward to incorporate material estimation into the optimization part of our pipeline, we would like to investigate ways to adapt our candidate position proposal method to handle unknown materials.

Directly Imaged Lights Our method is designed for cases in which light emitters are not directly imaged, for example due to physical constraints during the scanning process. However, if emitting surfaces are imaged, they must be removed before computing the IDF. Ideally, we would incorporate information about known light emitters, rather than discarding it.

Beyond Isotropic Point Lights Several works [19, 25, 30] note that real-world emitters are much more complex than isotropic point sources; instead, they frequently have radially-varying intensity distributions and different geometries (e.g. lines or areas). While our refinement algorithm can apply to any lighting model, examining how directional and geometric variations affect our candidate light proposal method is an important area for further investigation.

Acknowledgements

This work was supported by the NSF/Intel Visual and Experiential Computing Award #1538618, with additional support from Google and the University of Washington Reality Lab.

References

- [1] J. Ackermann, S. Fuhrmann, and M. Goesele. Geometric point light source calibration. In *Symposium on Vision, Modelling, and Visualization*, 2013. [2](#)
- [2] S. Agarwal, K. Mierle, and Others. Ceres Solver. [5](#)
- [3] B. Boom, S. Orts-Escalano, X. Ning, S. McDonagh, P. Sandilands, and R. B. Fisher. Point light source estimation based on scenes recorded by a rgb-d camera. In *BMVC*, 2013. [2](#)
- [4] P. Debevec. Rendering synthetic objects into real scenes. In *SIGGRAPH*, 1998. [1](#)
- [5] F. Einabadi and O. Grau. Discrete light source estimation from light probes for photorealistic rendering. In *BMVC*, 2015. [2](#)
- [6] J.-M. Frahm, K. Koeser, D. Grest, and R. Koch. Markerless augmented reality with light source estimation for direct illumination. In *CVMP*, 2005. [2](#)
- [7] K. Hara, K. Nishino, and K. Ikeuchi. Light source position and reflectance estimation from a single view without the distant illumination assumption. *TPAMI*, 27(4), 2005. [2](#)
- [8] B. K. Horn. Obtaining shape from shading information. *The Psychology of Computer Vision*, pages 115–155, 1975. [1](#)
- [9] X. Huang, M. Walton, G. Bearman, and O. Cossairt. Near light correction for image relighting and 3d shape recovery. In *Digital Heritage*, volume 1. IEEE, 2015. [2](#)
- [10] S. Jiddi, P. Robert, and E. Marchand. Reflectance and illumination estimation for realistic augmentations of real scenes. In *ISMAR*, 2016. [2](#)
- [11] T. Kim and K.-S. Hong. A practical approach for estimating illumination distribution from shadows using a single image. *International Journal of Imaging Systems and Technology*, 15(2), 2005. [2](#)
- [12] P. Lagger and P. Fua. Using specularities to recover multiple light sources in the presence of texture. In *ICPR*, 2006. [2](#)
- [13] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *ECCV*, 2016. [1](#)
- [14] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22(2), 2003. [2](#)
- [15] Y. Li, S. Lin, H. Lu, and H.-Y. Shum. Multiple-cue illumination estimation in textured scenes. In *ICCV*, 2003. [2](#)
- [16] J. Lopez-Moreno, E. Garces, S. Hadap, E. Reinhard, and D. Gutierrez. Multiple light source estimation in a single image. In *Computer Graphics Forum*, volume 32, 2013. [1](#)
- [17] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Illumination estimation and cast shadow detection through a higher-order graphical model. In *CVPR*, 2011. [2](#)
- [18] T. Papadimitri and P. Favaro. Uncalibrated near-light photometric stereo. In *BMVC*, 2014. [2](#)
- [19] J. Park, S. N. Sinha, Y. Matsushita, Y.-W. Tai, and I. So Kweon. Calibrating a non-isotropic near point light source using a plane. In *CVPR*, 2014. [2, 8](#)
- [20] M. W. Powell, S. Sarkar, and D. Goldgof. A simple strategy for calibrating the geometry of light sources. *TPAMI*, 23(9), 2001. [2](#)
- [21] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *TPAMI*, 25(3), 2003. [2](#)
- [22] D. Schnieders, K.-Y. K. Wong, and Z. Dai. Polygonal light source estimation. In *ACCV*, 2009. [2](#)
- [23] T. Takai, A. Maki, and T. Matsuyama. Self shadows and cast shadows in estimating illumination distribution. In *CVMP*, 2007. [2](#)
- [24] T. Takai, A. Maki, K. Niinuma, and T. Matsuyama. Difference sphere: an approach to near light source estimation. *Computer Vision and Image Understanding*, 113(9), 2009. [2](#)
- [25] M. Visentini-Scarzanella and H. Kawasaki. Simultaneous camera, light position and radiant intensity distribution calibration. In *Pacific-Rim Symposium on Image and Video Technology*, 2015. [2, 8](#)
- [26] Y. Wang and D. Samaras. Estimation of multiple illuminants from a single image of arbitrary known geometry. In *ECCV*, 2002. [1](#)
- [27] Y. Wang and D. Samaras. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models*, 65(4), 2003. [1](#)
- [28] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *SIGGRAPH*, 2000. [2](#)
- [29] Y. Yang and A. Yuille. Sources from shading. In *CVPR*, 1991. [1](#)
- [30] E. Zhang, M. F. Cohen, and B. Curless. Emptying, refurbishing, and relighting indoor spaces. In *SIGGRAPH Asia*, 2016. [1, 2, 7, 8](#)
- [31] Y. Zhang and Y.-H. Yang. Multiple illuminant direction detection with application to image synthesis. *TPAMI*, 23(8), 2001. [1](#)
- [32] W. Zhou and C. Kambhamettu. Estimation of the size and location of multiple area light sources. In *ICPR*, 2004. [2](#)
- [33] M. Zollhöfer, A. Dai, M. Innmann, C. Wu, M. Stammerger, C. Theobalt, and M. Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Trans. Graph.*, 34(4), 2015. [1](#)