

Light Source Estimation in Synthetic Images

Mike Kasper^(✉), Nima Keivan, Gabe Sibley, and Christoffer Heckman

University of Colorado, Boulder, USA

{michael.kasper,nima.keivan,gsibley,christoffer.heckman}@colorado.edu

Abstract. We evaluate a novel light source estimation algorithm with synthetic image data generated using a custom path-tracer. We model light as an environment map as light sources at infinity for its benefits in estimation. However the synthetic image data are rendered using spherical area lights as to better represent the physical world as well as challenge our algorithm. In total, we generate 55 random illumination scenarios, consisting of either one or two spherical area lights with different intensities and positioned at different distances from the observed scene. Using this data we are able to tune our optimization parameters and determine under which conditions this algorithm and model representation is best suited.

Keywords: Light source estimation · Path-tracing · Synthetic data

1 Introduction

Computer vision is often referred to as the “inverse graphics” problem. This is because many of the equations and relations used in computer vision find their roots in the understanding of image formation and light interaction. However, the complete process of image formation is mainly ignored in most applications of computer vision. For instance, simulating light as it propagates through a scene is traditionally accomplished by means of ray-casting. This method ignores subsequent interactions a ray of light may have with the scene, instead terminating the ray on the first collision with a surface.

As a result of such simplifications, many visual quantities such as light position, sensor characterization and in-scene surface properties are irrecoverable. Ideally these quantities could be estimated as properties of the scene through simulating the propagation of light and employing optimization, requiring a light-weight but powerful simulation stack for graphics modeling.

In this work, we build on [7] to develop a light source estimation algorithm that includes light source location correction based on photometric differences and has the capability to extend to in-scene surface property estimation, a critical step toward semantic scene understanding. This algorithm relies heavily on leveraging synthetic data in order to calculate cost functions and to isolate individual aspects of the problem such as capturing realistic shadow diffusion and light reflection from in-scene surfaces. A visual sample of the results from our method are shown in Fig. 1.

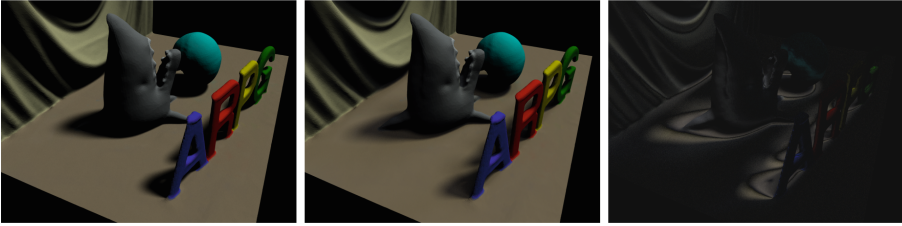


Fig. 1. *Left:* synthetic reference image. *Middle:* our rendered result after light source estimation. *Right:* photometric error image.

2 Method

Our approach relies on a custom image rendering system to compare synthetic data with our generative model’s output for scene reconstruction. The synthetic data generation includes the encoding of scene geometry, albedo and 3D light positions. Results from our light source estimation system are compared with these synthetic results and guide an optimization over light position.

2.1 Image Rendering

To render our scene we have developed a path-tracer using NVIDIA’s OptiX ray-tracing library. We employ a custom path-tracer due to our need to calculate analytical derivatives of the *light transport equation* (LTE) in order to guide later optimization. The LTE describes how radiance emitted from a light source interacts with the scene. Formally, we compute the exitant radiance L_o leaving a point p in direction ω_o as:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\mathcal{H}^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i \quad (1)$$

where L_e is the radiance emitted at point p in direction ω_o . The integral term evaluates all the incident radiance L_i arriving at point p over the unit hemisphere \mathcal{H}^2 , oriented with the surface normal found at p , and subsequently reflected in the direction ω_o . The function f evaluates the *bidirectional reflectance distribution function* (BRDF) found at point p . The BRDF defines the amount of radiance leaving in direction ω_o as a results of incident radiance arriving along the direction ω_i . Finally, θ_i is the angle between the surface normal found at p and ω_i . Using Monte Carlo integration we can rewrite Eq. (1) as the finite sum:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \frac{1}{N} \sum_{i=1}^N \frac{f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i|}{p(\omega_i)} \quad (2)$$

where $i = 1, \dots, N$ is the number of samples drawn from the distribution described by the *probability density function* (PDF) p .

To compute the final pixel intensity I , we integrate the intensity of all rays $i = 1, \dots, M$ arriving at our synthetic sensor (of the form of Eq. (2)). Using Monte Carlo integration we can evaluate this with the finite sum:

$$I = \frac{1}{M} \sum_{i=1}^M L_o(\mathbf{p}_i, \omega_o) \quad (3)$$

where \mathbf{p}_i refers to the point where a ray originating from our sensor and traveling along ω_o first intersects with the scene. For more information on path-tracing and the LTE see [12].

2.2 Synthetic Data Generation

Scene Geometry. In this work we operate on a static 3D scene representing a tabletop with several items placed on its surface as seen in Fig. 1. This scene is constructed to afford interesting illumination conditions without addressing pathological factors such as the presence of mirrors, etc. The 3D geometry was captured from a real scene using KinectFusion [11] with an Asus Xtion Pro 3D sensor. While any scene constructed with 3D modeling software would suffice, we use a captured real-life scene so that we may compare results between real and synthetic data in future research.

Albedos. To render scenes under different illumination conditions it is necessary to associate surface albedos (i.e. color devoid of any shading information) with the 3D geometry. The problem of separating albedos and shading information found in images, often referred to as *intrinsic image decomposition*, is the subject of a rich field of ongoing research [2–4]; to obviate this challenge we assume albedo associations are known, although this knowledge need not be perfect accurate. Utilizing synthetic data allows us to both modulate the accuracy of the albedo map as well as address correcting it within our framework, as a topic of future work.

Area Lights. We employ spherical area lights to provide an arbitrary source of illumination in our synthetic reference images. Crucially, this representation of light used in rendering reference images is distinct from the environment map light we are estimating as described in Sect. 2.3. This enables us to assess how well our environment map-based light model can represent more complex illumination scenarios.

2.3 Light Source Estimation

Environment Light. As mentioned in Sect. 2.2 we model light using an environment map [6, 9]. Instead of sampling points in 3D space as used with area lights, with environment map lighting we sample directions. This representation works well for approximating lights located further from the observed scene.

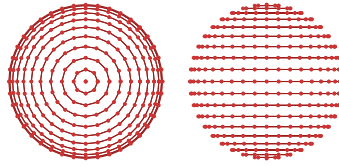


Fig. 2. Visualization of environment light discretization with a top-down view on the left, and a side view on the right. The light depicted here consists of 21 rings and 522 total points.

While many works have considered in-scene lighting examples [8, 10], we instead focus on out-of-scene sources [1, 5, 13, 14]. To compute the incident radiance L_i arriving at a point p we trace a ray with origin p in some direction ω . If the ray is unobstructed by the scene geometry, point p will receive the full radiance traveling along ω as determined by the environment map.

To compute the radiance emitted by the environment map along a given direction, we first discretize a unit sphere into a finite number of uniformly spaced points. We perform the same discretization as described in [6] for the entire sphere. The resolution of this discretization is indicated only by the number of desired rings. The spacing of points around each ring is computed to be as close to the inter-ring spacing as possible, as seen in Fig. 2. When tracing a ray along a given direction we determine the nearest-neighbor direction from the discretized environment map and return its associated RGB value λ as the emitted radiance.

Direction Sampling. To render a scene illuminated by an environment map we must sample a direction each time a ray intersects the scene. We perform importance sampling by sampling environment map directions that are more likely to contribute a larger amount of light. To achieve this we constructed a 2D probability distribution function that reflects the current environment light parameters as described in [12], however with the small modification to handle the unique discretized structure of the our environment map. It is from this 2D PDF that we can compute the probability of the sample $p(\omega)$.

Light Transport Derivatives. To estimate the environment light parameters we need to compute the Jacobian of partial derivatives of color channel α of each pixel I with respect to each environment lighting parameter λ . We first drop the $L_e(p, \omega)$ term from Eq. (2) as there is no point p on the surface of environment map that emits light. We then define a visibility function $V(p, \omega)$ which equates to 1 if the ray leaving from point p in direction ω is not obstructed by the scene geometry, and 0 otherwise. We now define the partial derivative of the intensity of color channel α at p with respect to the light source color channel α as:

$$\left. \frac{dI_\alpha}{d\lambda_\alpha} \right|_p = \frac{1}{N} \sum_{i=1}^N \frac{f(p, \omega_o, \omega_i) V(p, \omega_i) |\cos \theta_i|}{p(\omega_i)}, \tag{4}$$

for which we then sum over the incident rays $1, \dots, M$ to obtain the derivative of per-channel pixel intensity.

Optimization. We employ sequential Monte Carlo (sMQ) to estimate the parameters of our environment light. For each iteration we sample the scene according to the currently-estimated lighting parameters and compute the Jacobian of the LTE. We then perform gradient descent with backtracking until we have converged on a new set of lighting parameters. We continue this process until the optimization has converged, as indicated by the Wolfe conditions on gradient magnitude.

3 Results

We evaluated the proposed light source estimation algorithm to determine what environment map resolution can best represent a wide-variety of lighting conditions. For this we constructed 55 scenes illuminated by either one or two randomly placed, spherical area lights and rendered two synthetic reference images for each scene. We then replaced the area light with an environment light, uniformly initialized all environment map intensities to be near zero, and computed the LTE derivatives sampling the scene 512 times per pixel. For each scene we ran our light source estimation algorithm using 13 different environment map resolutions for a total of 715 different trials. While rendering the synthetic reference images only took 1–2s, an entire optimization typically took 4–5 min to

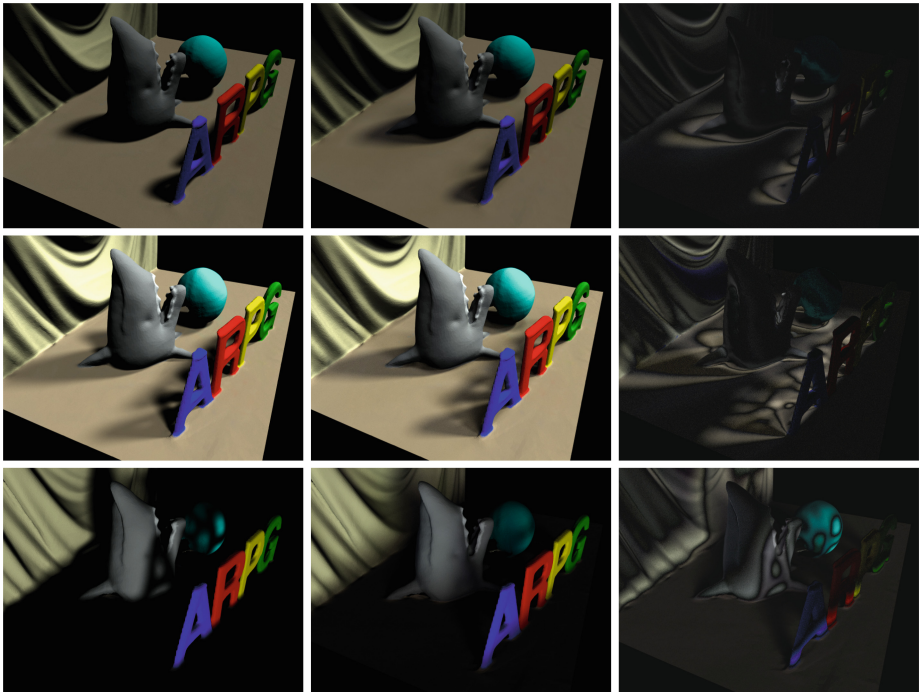


Fig. 3. Visual comparison of three different illumination scenarios. The left column shows the synthetic reference images, the middle column our estimation, and the right column their photometric error.

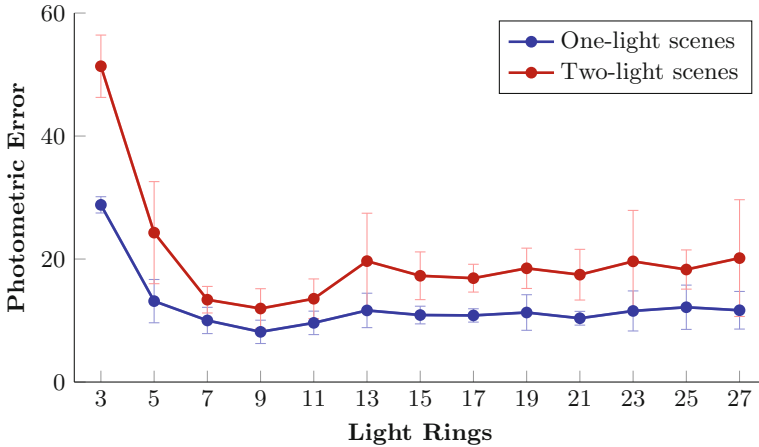


Fig. 4. Photometric error between reference image and rendered estimate for an environment light with the indicated number of rows. All images are rendered at 128×96 . Mean and standard deviation computed for 55 different illumination scenarios.

converge on a consumer-grade laptop. The summarized results of this experiment can be seen in Fig. 4. Surprisingly a relatively coarse resolution of 9 light rings achieved the best results. However we suspect that for the higher-resolution models, 512 samples per pixel was insufficient and the resulting variance hindered their optimization.

4 Conclusions and Future Work

We have presented an algorithm that generates synthetic visual data in a 3D environment and developed a generative model with output that is refined through an optimization procedure over light position. We have also demonstrated a robust and efficient method of generating high-quality synthetic visual datasets which may be used to guide semantic scene understanding through optimization. Our results suggest that in-scene property estimation tasks may be successfully executed in an efficient optimization framework. In future work we will demonstrate full path-tracing and shadow detection within the postulated environment map to improve the accuracy of our estimation.

References

1. Boom, B., Orts-Escolano, S., Ning, X., McDonagh, S., Sandilands, P., Fisher, R.B.: Point light source estimation based on scenes recorded by a RGB-D camera. In: British Machine Vision Conference, BMVC, Bristol, UK (2013)
2. Chen, Q., Koltun, V.: A simple model for intrinsic image decomposition with depth cues. In: International Conference on Computer Vision, pp. 241–248. IEEE (2013)

3. Duchêne, S., Riant, C., Chaurasia, G., Moreno, J.L., Laffont, P.Y., Popov, S., Bousseau, A., Drettakis, G.: Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Trans. Graph.* **34**, 1–16 (2015)
4. Hachama, M., Ghanem, B., Wonka, P.: Intrinsic scene decomposition from RGB-D images. In: *International Conference on Computer Vision*, pp. 810–818. IEEE (2015)
5. Hara, K., Nishino, K., Ikeuchi, K.: Multiple light sources and reflectance property estimation based on a mixture of spherical distributions. In: *10th IEEE International Conference on Computer Vision (ICCV 2005)*, 17–20 October 2005, Beijing, China, pp. 1627–1634 (2005). <http://doi.ieeecomputersociety.org/10.1109/ICCV.2005.162>
6. Jachnik, J., Newcombe, R.A., Davison, A.J.: Real-time surface light-field capture for augmentation of planar specular surfaces. In: *International Symposium on Mixed and Augmented Reality*, pp. 91–97. IEEE (2012)
7. Keivan, N., Sibley, G.: Generative scene models with analytical path-tracing. In: *Robotics Science and Systems (RSS) Workshop on Realistic, Repeatable and Robust Simulation* (2015)
8. Knorr, S.B., Kurz, D.: Real-time illumination estimation from faces for coherent rendering. In: *Proceedings IEEE International Symposium on Mixed and Augmented Reality (ISMAR2014)*, pp. 113–122 (2014)
9. Lalonde, J.F., Matthews, I.: Lighting estimation in outdoor image collections. In: *International Conference on 3D Vision*, pp. 131–138. IEEE (2014)
10. Meilland, M., Barat, C., Comport, A.: 3D high dynamic range dense visual slam and its application to real-time object re-lighting. In: *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 143–152. IEEE (2013)
11. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: real-time dense surface mapping and tracking. In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136. ISMAR 2011. IEEE Computer Society, Washington, D.C. (2011). <http://dx.doi.org/10.1109/ISMAR.2011.6092378>
12. Pharr, M., Humphreys, G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2010)
13. Takai, T., Maki, A., Matsuyama, T.: Self shadows and cast shadows in estimating illumination distribution. In: *4th European Conference on Visual Media Production, 2007, IETCVMP*, pp. 1–10, November 2007
14. Zhou, W., Kambhampettu, C.: Estimation of illuminant direction and intensity of multiple light sources. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2353, pp. 206–220. Springer, Heidelberg (2002). doi:[10.1007/3-540-47979-1_14](https://doi.org/10.1007/3-540-47979-1_14)