Final Report

Spring Semester, 2012

Chi-Wei Tseng

June 27, 2012

Reference

Pacific Graphics 2009
S. Lee, D. Lischinski, and Y. Yu
(Guest Editors)

Volume 28 (2009), Number 7

Lighting by Guides

paper 1035

Abstract

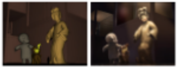
This paper introduces and presents a solution to the lighting-by-guide problem: given the description of a scene and an image as the lighting guide, automatically find a lighting configuration so that the resulting rendering best matches the lighting guide. Solving the general lighting-by-guide problem is difficult. To make progress along this direction, this paper instead focuses on an easier problem by assuming point lights and a simple lighting model. Our solution framework consists of four main modules: initial guess, optimization, relighting and difference metric evaluation. Since the problem is a high-dimensional nonlinear optimization problem, it is crucial to have a good initial guess. We propose an efficient and effective algorithm for this task. Experiments show that our system can produce good quality lighting setups. The result is an automatic lighting algorithm which infers the lighting configuration from the lighting guide, greatly alleviating lighting artists' workload. Even though such a system is not yet ready for serious animation production, it could be very useful as a lighting tool for hobbyist animators or for mass production of low-quality animations.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Lighting, placing and configuring lights, is fundamental and crucial to computer animation as it establishes the moods of shots and enhances storytelling [PBSMF07]. In a typical production pipeline, an artist paints a *lighting guide* for a shot to set up the mood he wants the shot to express. Figure 1 gives an example of lighting guide. The scene is rendered with simple default lighting (a directional light or a point light at camera's position). The artist then paints on the default-lighting rendering to create lighting guide. Given the lighting guide, a *lighting artist* (also called a *lighter*) uses a relighting tool to adjust the configurations of lights so that the rendering of the scene visually matches the lighting guide as much as possible. Such an adjustment procedure involves a large number of trials to match the visual guide, making this process time-consuming, labor-intensive and counter-intuitive. Thus, often a large crew of lighters are needed.

Motivated by the process, this paper introduces the lighting-by-guide problem: find a proper lighting setup for a given scene so that the rendering matches a given lighting guide. Unfortunately, solving the general problem is very difficult. Thus, to make progress along this line, this paper solves an easier problem by assuming point lights and a simple lighting model. With these assumptions, similar to Pellacini *et al.*, we formulate the lighting-by-guide problem as a high-dimensional nonlinear optimization problem [PBSMF07] and propose a framework to solve it. The framework decomposes the lighting-by-guide problem into four components: optimization, initial guess, relighting and difference measurement. As with most nonlinear optimiza-



(a) default-lighting rendering (b) lighting guide

Figure 1: *An example of lighting guide. Given the rendering with the default lighting configuration (a), artists paints on it to create a lighting guide (b) to illustrate the atmosphere and mood the shot should express. Lighters then manually adjust lighting parameters by referring the guide so that the resulting rendering visually matches the lighting guide.*

Lighting by Guide

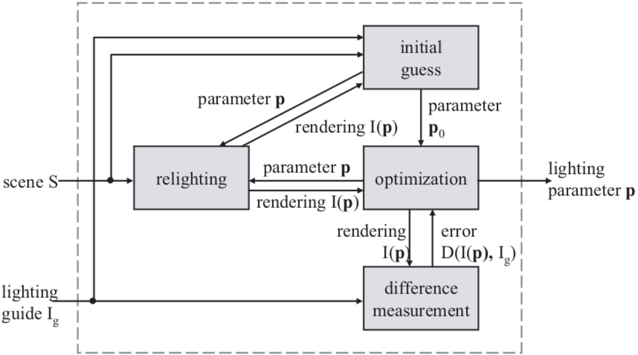
Problem Definition

(a) default-lighting rendering

(b) lighting guide

- Input: 3D models and an artistic painting that depicts lighting anticipation.
- Output: positions and properties of lights in the scene.

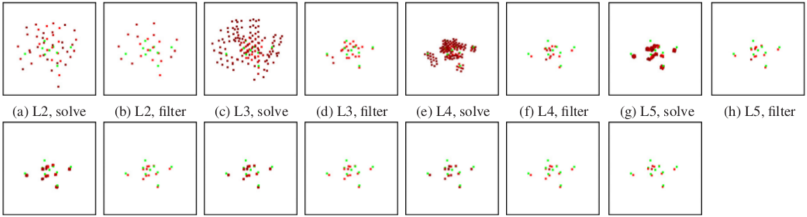
Components of the System



- Relighting module: PBRT.
- Optimization module: simplex algorithm in GSL.

Making an Initial Guess

Overview



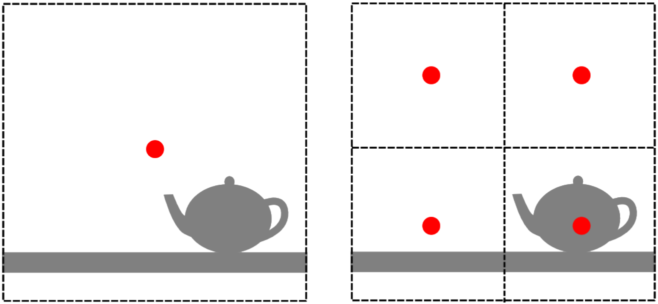
(a) L2, solve (b) L2, filter (c) L3, solve (d) L3, filter (e) L4, solve (f) L4, filter (g) L5, solve (h) L5, filter

(i) L6, solve (j) L6, filter (k) L7, solve (l) L7, filter (m) L8, solve (n) L8, filter (o) final solution

1. Place a point light at the center of the specified bounding box.
2. Run the following routine several times: *expand*, *render*, *solve*, *merge* and *filter*.
3. Solve for intensities of each light that is not removed.

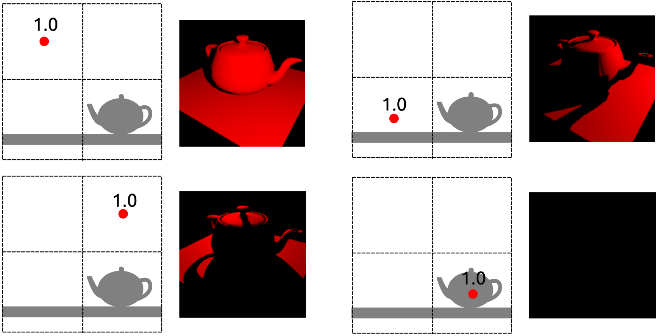
Making an Initial Guess

Expand: Divide A Light into Many



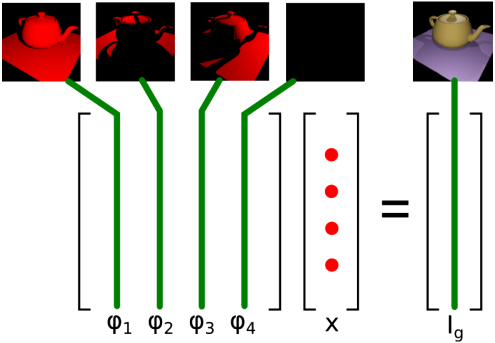
Making an Initial Guess

Render: Calculate the Contribution Vector of Each Light



Making an Initial Guess

Solve: Calculate Intensities of Each Light

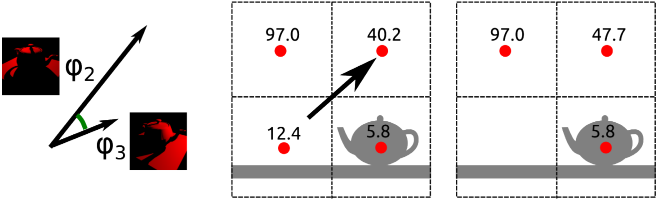


With SVD, solve the overdetermined system for:

$$\arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{I_g}\|$$

Making an Initial Guess

Merge: Combine Lights with Similar Contributions

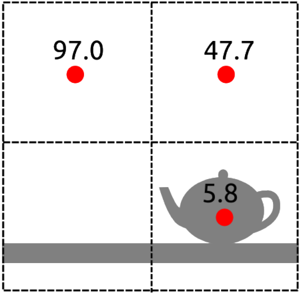


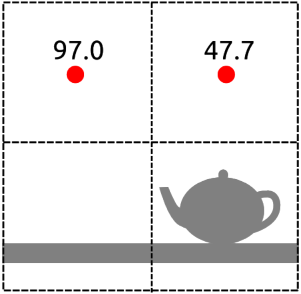
$$\text{light similarity: } \tau(\phi_\mathbf{i}, \phi_\mathbf{j}) = \frac{\phi_\mathbf{i} \cdot \phi_\mathbf{j}}{\|\phi_\mathbf{i}\| \, \|\phi_\mathbf{j}\|}$$

- Merge dimmer light into brighter one by projecting contribution vector.
- Merge until
  1. maximum light similarity is less than a specified threshold.
  2. no negative light intensity exists.

Making an Initial Guess

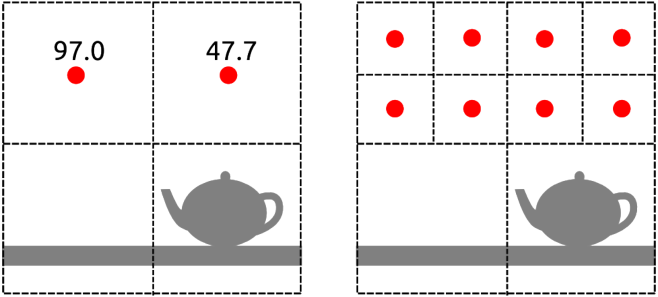
Filter: Remove Dim Lights.





Making an Initial Guess

Expand Again...



Optimization and Difference Measurement

Use simplex algorithm. Determine $6n$ parameters for n lights.

$$\mathbf{p} = \{(x, y, z, r, g, b)\}$$

Ways of measuring difference between images have been proposed.

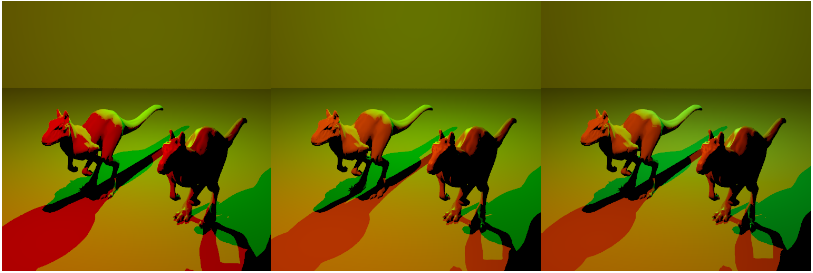
- L_2-norm or importance-weighted L_2-norm.
- Perceptual based metrics.

They give visually similar results.

Results

A Simple Killeroo Scene

2 lights. A red one and a green one.



(a) Lighting guide. (b) Initial guess. (c) Optimized parameters.

Results

A More Complex Killeroo Scene

4 lights, each with different radiance and color.

(a) Lighting guide.

(b) Optimized parameters.

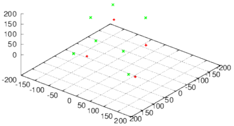
Preliminary Analysis

Is there possibility for improvement?

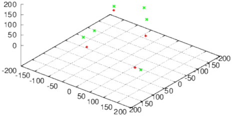
- Observation: slow convergence in optimization.
- Resaon: too many discontinuities in goal function?

Preliminary Analysis

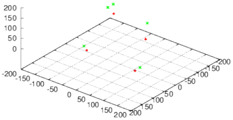
Reasons for Slow Convergence



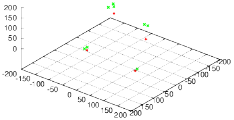
(a) Iteration 2.



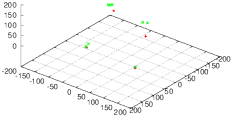
(b) Iteration 3.



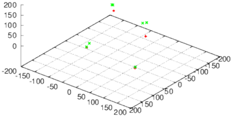
(c) Iteration 4.



(d) Iteration 5.

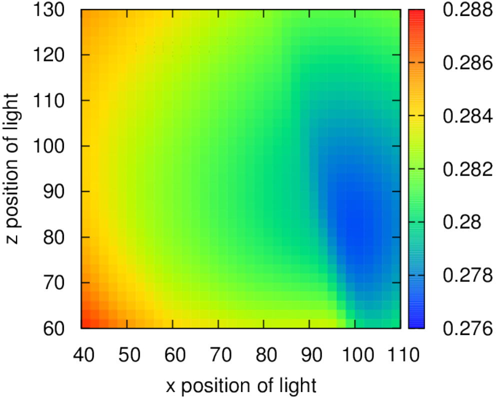


(e) Iteration 6



(f) Iteration 7.

Interestingly, the energy surface looks quite well-behaved.



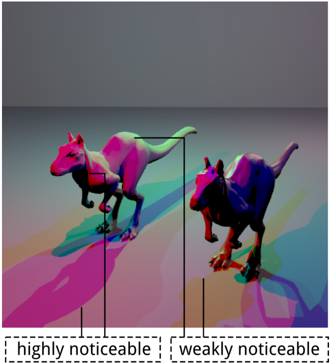
Other possible reasons for slow convergence?

Preliminary Analysis

Reasons for Slow Convergence

- $6n$ parameters may be too many for simplex algorithm to converge fast?
- Solution: Reduce the number of parameters.
- Issue: This will limit the search domain for the parameters.

A "Most-Noticeable-Light-First" Optimization Strategy

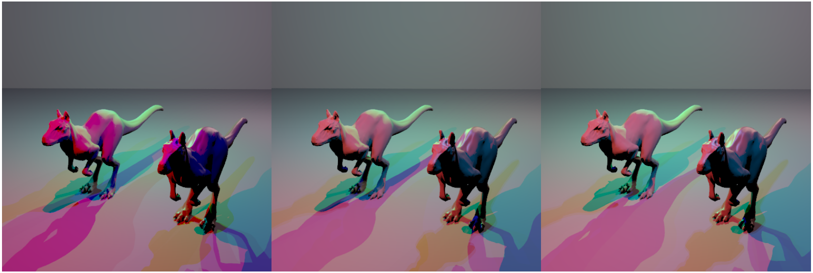


- Decrease iterations with as little quality degradation as possible.
- Try to mimic real photographers setting up lights in a scene.

$$\text{Noticeability: } N(I_i) = \sum_{x,y} \|\nabla I_i(x,y)\|$$

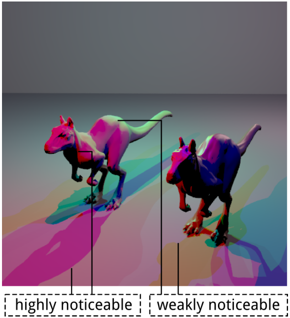
Results

Improvement in the More Complex Killeroo Scene



(g) Lighting guide. (h) Global Optimization. (i) MNLF.

A "Most-Noticeable-Light-First" Optimization Strategy



$$\text{Noticeability: } N(I_i) \stackrel{?}{=} \sum_{x,y} \|\nabla I_i(x,y)\|$$

Proposition: Noticeability should be the difference in sum-of-gradient between output image *with* and *without* the light.

Conclusion

- A lighting-by-guide system is implemented as a tool to guess light parameters from an illustration of a 3D scene.
- Slow convergence and not very satisfying guessing quality currently obstruct the system from being further developed.
- The most-noticeable-light-first optimization approach may be a potential solution to the mentioned issues.

Thanks for your listening.