Light Transport Analysis by Krylov Subspace Illumination

Matthew O'Toole*
University of Toronto

Abstract

We propose a recursive illumination framework for light transport analysis of an unknown scene. In this framework, we repeatedly illuminate a scene by capturing a photo and projecting the photo back onto the scene. This recursive illumination procedure produces a set of photos that spans the Krylov subspace of the light transport matrix. Many iterative methods in numerical linear algebra rely on this Krylov subspace for solving the generalized eigenvalue and minimal residual problems of large systems. Our framework expresses this recursive illumination procedure as a means to solve novel problems in computational illumination. In particular, we discuss some applications in light transport segmentation and light transport compensation. Light transport segmentation identifies pixels that mutually illuminate each other by segmenting the image according to low-level global illumination cues. Light transport compensation improves image projection quality by constructing an environment-aware projector that iteratively corrects for defocus, distortions, and global lighting. We also discuss the numerical stability of Krylov subspace illumination by considering the effects of sensor noise and signal quantization.

Keywords: coded illumination, numerical linear algebra, global illumination, image segmentation, image manipulation.

1 Introduction

The light transport matrix models light transport between projector and camera pixels. In addition to image-based relighting, measuring the full matrix of a real-world scene has several applications. In this paper, we study computational illumination with two applications in mind: light transport segmentation and light transport compensation. First, combining light transport with image segmentation provides the means to aggregate the low-level cues of global transport into an image that describes how light scatters in a scene. Second, environment-aware projectors preserve image content by inverting light transport to compensate for defocused or distorted projections. These problems are formulated as linear systems that draw on light transport matrices, although measuring, storing, and solving these large systems directly proves impractical. Our task is to efficiently approximate their solutions.

Image segmentation algorithms operate on low-level affinity cues to identify groups of pixels as objects in a scene. These algorithms often require, as input, a similarity matrix that tabulates the affinities between pixels. Most similarity matrices are a function of color and proximity, in which two adjacent pixels with similar color values share a large affinity value. We introduce the novel concept of light transport segmentation, a method that groups pixels according to how light scatters from one pixel to the next by reinterpreting a symmetric light transport matrix as a similarity matrix. Our image segmentation results provide an intuitive visualization of light scattering in real-world scenes without explicitly reconstructing the light transport matrix.

Image compensation studies how to project image content onto arbitrary surfaces without deteriorating the image quality. Prior works partially solve this problem for scenes with non-planar surfaces and multiple depth planes [Raskar et al. 2003; Zhang and Nayar 2006], though their solutions cannot handle scenes with global illumination. Our light transport compensation algorithm betters projection quality by applying a minimal residual method to the light transport matrix itself.

The solutions to these applications share a common recursive lighting procedure that, in short, illuminates a scene, takes a photo, and projects the photo back onto the scene. By repeating the latter steps, the photos eventually converge to the principal eigenvector of the light transport matrix. Moreover, the photographs span the Krylov subspace of the light transport matrix. Many iterative methods in numerical linear algebra rely on the Krylov subspace to efficiently approximate the largest eigenvectors or solve large systems of equations. In the context of computational illumination, Krylov subspace illumination requires no explicit knowledge of the light transport matrix in order to efficiently solve the light transport segmentation and compensation problems.

The main contribution of this paper is the introduction of Krylov subspace methods to computational illumination. We build upon this framework to introduce a novel concept in light transport segmentation to visualize global transport. We also introduce a generalized formulation of the image compensation problem, and create an environment-aware projector that accounts for any type of light transport. We also discuss the numerical stability of the Krylov subspace illumination procedure in terms of sensor noise and signal quantization.

2 Related Work

The light transport matrix models outgoing light as a function of incident light. Many works study these matrices in its various forms. A matrix that models incident light fields to outgoing light fields describes an 8D light transport function [Levoy and Hanrahan 1996]. Because the cost of capturing high fidelity 8D light transport is high, most applications reduce the dimensionality of the matrix to simplify the acquisition process. Light stages are built to acquire 4D and 6D reflectance fields for image-based relighting of stationary scenes to facial animations [Chabert et al. 2006]. A discretized BRDF is itself a light transport matrix parameterized over incident and outgoing light directions [Ghosh et al. 2007]. In addition to image-based relighting and BRDF sampling, reconstructing the full light transport matrix allows one to compute the interreflection operator for a Lambertian scene, and find the contribution of light after n bounces [Seitz et al. 2005].

Brute force methods reconstruct these matrices directly column by column with many photographs. Other methods attempt to lower the number of photographs while maintaining high fidelity in the reconstruction results. Adaptively point sampling a reflectance field allows an algorithm to capture important features faster than us-

*e-mail: motoole@dgp.toronto.edu

ing a fixed sampling pattern [Fuchs et al. e 07]. Dual photography adaptively samples the matrix by hierarchically searching for radiometrically independent blocks of pixels [Sen et al. 2005], though the worst case occurs for scenes with significant amounts of interreflections. Symmetric photography extends this work by approximating the blocks with rank-1 factorizations [Garg et al. 2006]. Compressive light transport sensing approximates the signal with a sparse set of basis functions using non-adaptive random measurements patterns as a constraint [Peers et al. 2009; Sen and Darabi 2009]. The generalized kernel Nyström method reconstructs low rank matrices by exploiting nonlinear coherence [Wang et al. 2009]. Unlike these methods, our approach avoids reconstructing the matrix entirely.

Prior work uses coded illumination patterns to perform 3D reconstruction of a scene without reconstructing the full matrix [Salvi et al. 2004; Narasimhan et al. 2008; Zhang et al. 2002]. Projecting a small number of polarized sinusoidal patterns onto a scene produces measurements to robustly estimate a disparity map even in the presence of global illumination [Chen et al. 2007]. A few high-frequency illumination patterns also provides the means to approximate direct and global separation efficiently [Nayar et al. 2006]. Flash and no-flash image pairs allow for low-light photography [Petschnigg et al. 2004; Krishnan and Fergus 2009]. Unlike these methods, our recursive illumination procedure assumes nothing about the structure of the matrix.

Light transport segmentation. We are not aware of any prior work in light transport segmentation. However, there is an abundance of research in image segmentation. Spectral clustering relaxes the problem of partitioning a graph into distinct sets according to some metric function. Specifically, minimizing cuts according to the normalized cut, average association, and average cut metrics relate to finding the eigenvectors of certain numerical systems [Shi and Malik 1997; Weiss 1999]. Soft segmentation for separating foreground and background objects in images are useful for matting applications [Levin et al. 2008; Lischinski et al. 2007]. Most of these image segmentation algorithms operate on the color and proximity of pixels to determine the appropriate grouping.

Light transport compensation. The light transport compensation, also known as image compensation, asks what image to project onto a scene to get a specific image back. Projecting an image onto anything other than a flat surface incurs image distortions, among other effects. A projector can minimize these distortions by becoming aware of the geometry in the scene itself [Raskar et al. 2003]. Another issue with projectors is the defocus that blurs the incident image. By performing defocus analysis and computing the per-pixel point spread function that blurs the image, it is possible to project an image that compensates for the defocus [Zhang and Nayar 2006]. These prior works in light transport compensation explicitly design their approach to handle these select cases, and thus their approach breaks down for general scenes. For example, projecting a pattern onto a concave surface produces global illumination and washes out the projection. We are, again, not aware of any work in light transport compensation for this general case.

3 Light Transport

In this paper, we restrict our setup to that of a single camera and projector. Suppose that a projector illuminates a scene and a camera measures the response of light scattering through the scene. In general, the projected light reflects and refracts multiple times within the scene before reaching the camera. The camera measures irradiance as a function of the choice of projected light pattern and the scene itself. We model this process to describe the appearance of a scene under any given illumination.

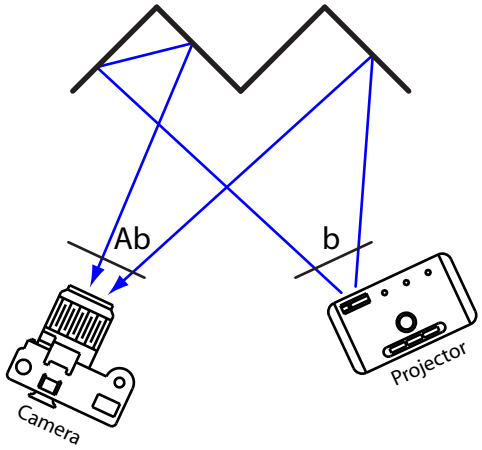

Figure 1: *Computational illumination setups often consist of a camera and light source (i.e. a projector, display, or laser) that illuminates a scene. The figure illustrates how a nonsymmetric light transport matrix models the scattering of light from projector to camera. When light scatters through a scene, light might reflect or refract with multiple surfaces before reaching the sensor. Global illumination refers to the contribution of light due to multiple scattering, and direct illumination refers to that of single scattering.*

For this paper, all images are represented by a single vector $\mathbf{x} \in \mathbb{R}^k$ for some k, where the component i of $\mathbf{x}$ stores the value of the i^{th} pixel of an image. Light transport, as discussed in this paper, is a function $f : \mathbb{R}^n \to \mathbb{R}^m$ that takes as input an n-pixel projected pattern and outputs an m-pixel photograph. If we are capable of defining this function, we are able to reconstruct the photograph taken of a scene under any given illumination pattern. Function f has two important properties. First, f is a homogeneous function of degree 1, where $f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$ for any scalar α. To explain, imagine projecting some pattern onto a scene and taking its corresponding photograph. If the projected image is made twice as bright, the resulting photograph is also twice as bright. Second, the function f is additive as given by $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$. Taking two photos under different illumination and adding these photos together is equal to taking a single photo of the scene under the two illumination patterns added together. The additivity and homogeneity properties imply that the function $f(\mathbf{x})$ is linear. Evaluating $f(\mathbf{x})$ is therefore equivalent to the matrix product operation $\mathbf{Ax}$ for a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, commonly known in practice as the light transport matrix. Each matrix component $\mathbf{A}_{j,i}$ describes the irradiance for light emitted from projector pixel i and measured by camera pixel j.

If we had the resources to build the entire light transport matrix, one option is to construct the matrix column by column as follows. Suppose that we illuminate the scene with a single pixel, and this pattern is given by a unit vector $\mathbf{e}_k$ with a 1 at the k^{th} position and 0 elsewhere. The matrix product $\mathbf{Ae}_k$ extracts the k^{th} column of the matrix $\mathbf{A}$. By iteratively taking photographs under illumination patterns $\mathbf{e}_k$ for all k, we can build up the transport matrix one column at a time. Once the light transport matrix is found, we can reconstruct the scene under any illumination $\mathbf{x}$ by evaluating the product $\mathbf{Ax}$ through a linear combination of the photographs taken.

The columns have a clear physical meaning, and so do the rows. Reading a row from the transport matrix is like taking a photo with the camera and projector swapped. This stems from the idea that

light flow reverses as per Helmholtz reciprocity, and the same irradiance value applies for light traveling in the opposite direction. An image produced by a linear combination of rows is a dual image of this system, and a linear combination of columns form a primal image.

Light transport has three properties that are of interest to our discussions. First, light transport is linear, and modeled by some matrix. The matrix product $\mathbf{Ax}$ produces an image of the scene as though illuminated by $\mathbf{x}$, as used in image-based relighting. Second, given Helmholtz reciprocity, aligning the camera and projector produces equivalent primal and dual images. In this situation, the light transport matrix becomes symmetric, or $\mathbf{A} = \mathbf{A}^T$. A symmetric matrix is also known to have orthogonal eigenvectors and real eigenvalues, a useful constraint for numerical analysis. Third, light transport is nonnegative. In other words, the matrix itself has nonnegative components. Combining the nonnegativity and symmetry properties allows light transport to be reinterpreted as a Laplacian matrix, a matrix often used in image segmentation.

4 Transport Eigenvectors by Recursive Illumination

At the heart of our approach is a simple recursive procedure that illuminates a scene with an initial pattern, takes a photo of the scene, lights the scene with the photo, takes another photo of the scene, and so on and so forth. The set of images produced by following this approach spans a basis considered to be among the most important for solving large linear systems. Below, we show why this recursive procedure is a powerful tool for light transport matrix analysis.

The initial pattern of this recursive procedure is some n-pixel image $\mathbf{b} \in \mathbb{R}^n$ where the component $\mathbf{b}_i$ of this vector is the intensity value of a pixel i. A n-pixel projector takes the image $\mathbf{b}$ as input and projects the image onto some scene. A photo is taken by an n-pixel camera while the scene is being lit. Remember that, as described in section 3, this process is equivalent to a matrix-vector multiplication operation that involves the light transport matrix.

The light transport matrix $\mathbf{A}$ is an $n \times n$ square matrix, given that both the projector and camera have n pixels. If an image $\mathbf{b}$ is projected onto the scene, the photo captured by the camera is given by the matrix-vector multiplication $\mathbf{Ab}$. What happens when we now illuminate the scene using the previously captured image $\mathbf{Ab}$?

$$\begin{aligned}
\mathbf{x}_0 &= \mathbf{b} \\
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k \\
(\mathbf{x}_m) &= (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m-1}) \\
&= (\mathbf{x}_0, \mathbf{A}\mathbf{x}_0, \mathbf{A}(\mathbf{A}\mathbf{x}_0), \ldots, \mathbf{A}(\cdots(\mathbf{A}\mathbf{x}_0)\cdots)) \\
&= (\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{m-1}\mathbf{b})
\end{aligned}$$

By recursively projecting and capturing photos, a sequence $(\mathbf{x}_m)$ of m images is produced, where the k^{th} image is given by the vector $\mathbf{A}^{k-1}\mathbf{b}$. This sequence is related to one of the simplest iterative methods in numerical linear algebra: *power iteration*.

Power iteration is an algorithm that finds the largest eigenvalue and its corresponding eigenvector. If the largest absolute eigenvalue of matrix $\mathbf{A}$ is distinct in magnitude and the initial image $\mathbf{b}$ is not orthogonal to the eigenvector corresponding to the largest eigenvalue, the iteration converges as the number of photos m taken becomes large. After performing the recursive illumination procedure, the last image in the sequence $(\mathbf{x}_m)$, namely $\mathbf{v}_m = \mathbf{A}^{m-1}\mathbf{b}$, is an approximation to the principal eigenvector of the light transport matrix $\mathbf{A}$. This algorithm is efficiently implemented by only storing

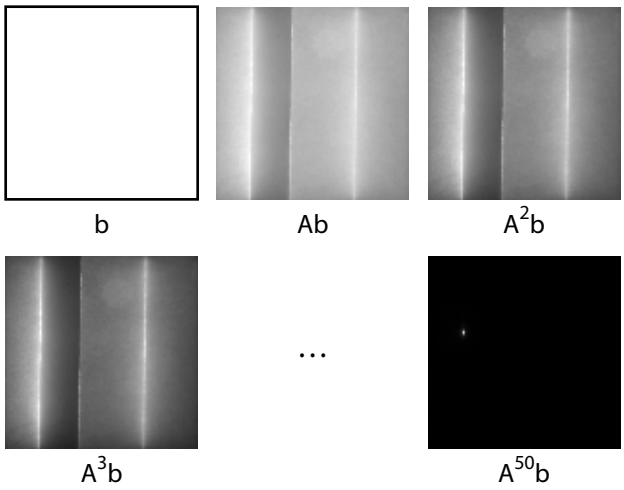

Figure 2: *An optical realization of the power iteration method for an initial white image $\mathbf{b} = \mathbf{1}$. The scene is a folded sheet of paper, as illustrated in figure 1. While a uniformly white image lights the scene, a photo of the scene is taken. By recursively illuminating the scene with the previous photo taken, the light from the photos converge to a single point in this particular example. The bottom-right image is the principal eigenvector corresponding to the largest eigenvalue of the light transport matrix. In this example, the eigenvector appears to indicate the location of the most specular point in the scene, located on one of the concave folds. Moreover, because the matrix is nonnegative, the principal eigenvector is nonnegative according to the Perron-Frobenius theorem from appendix A.*

Algorithm 1 Power Iteration

for $k = 1$ to m **do**
  $\mathbf{w} = \mathbf{A}\mathbf{v}_{k-1}$
  $\mathbf{v}_k = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$
  $\lambda_k = \mathbf{v}_k^T\mathbf{A}\mathbf{v}_k$
end for

the current image $\mathbf{v}_k$ in memory at any given step k. Unfortunately, there are several disadvantages associated with the power iteration. First, this method only approximates the largest eigenvalue λ_m and corresponding eigenvector $\mathbf{v}_m$. Second, the power iteration converges linearly at a rate given by the ratio of the two largest absolute eigenvalues of $\mathbf{A}$. If the largest two eigenvalues are close in magnitude, the convergence will slow down as well. Third, the largest eigenvalue is not always distinct in magnitude.

A better approach is to store and use all the images in the sequence $(\mathbf{x}_m)$, not just the last image captured. In fact, the power iteration is part of a larger class of numerical techniques called *Krylov subspace* methods.

$$\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \langle \mathbf{b}, \mathbf{A}\mathbf{b}, \cdots, \mathbf{A}^{m-1}\mathbf{b} \rangle$$

The Krylov subspace, $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, of a light transport matrix $\mathbf{A}$ and initial image $\mathbf{b}$ is the basis spanned by all of the photos captured in our recursive procedure, and so any element within this subspace is a linear combination of the captured images. Importantly, building this subspace requires no explicit knowledge of the matrix $\mathbf{A}$ itself. This subspace is particularly useful for light transport analysis because the subspace is constructed by treating the transport matrix as a black box; computing the product of an unknown transport matrix

with a known vector amounts to projecting the vector onto the scene and capturing the resulting image. These Krylov subspaces allow for approximations to multiple eigenvalues and eigenvectors, and a way to solve $\mathbf{Ax} = \mathbf{b}$. The idea behind Krylov subspace methods is that solving large systems involving a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ requires finding a solution in an equally large space $\mathbb{R}^n$. By limiting the search to a much smaller subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b}) \subset \mathbb{R}^n$, we are able to find approximations to these large systems much more efficiently.

Because an element of the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ is a linear combination of the images $\mathbf{A}^k\mathbf{b}$, the Krylov subspace can be expressed as the set of $(m-1)$ degree polynomials in $\mathbf{A}$, multiplied by the vector $\mathbf{b}$.

$$\begin{aligned} p(\mathbf{A})\mathbf{b} &= \left(\alpha_0 + \alpha_1\mathbf{A} + \cdots \alpha_{m-1}\mathbf{A}^{m-1}\right)\mathbf{b} \\ &= \left(\sum_{k=0}^{m-1} \alpha_k \mathbf{A}^k\right)\mathbf{b} \in \mathcal{K}_m(\mathbf{A}, \mathbf{b}) \end{aligned}$$

Suppose for the moment that $\mathbf{b}$ is not orthogonal to any eigenvector of $\mathbf{A}$. What does it mean to find the nonzero polynomial of least degree such that $p(\mathbf{A})\mathbf{b} = \mathbf{0}$? The answer is given by the *matrix minimal polynomial* denoted as $q(\mathbf{A})$, and the roots of this special polynomial are eigenvalues of the matrix $\mathbf{A}$.

$$\begin{aligned} \mathbf{0} = q(\mathbf{A})\mathbf{b} &= \left(\sum_{k=0}^{m-1} \alpha_k \mathbf{A}^k\right)\mathbf{b} \\ &= \left(\prod_{k=0}^{m-1} (\mathbf{A} - \lambda_k \mathbf{I})\right)\mathbf{b} \end{aligned} \tag{1}$$

A few slight modifications to the same minimal polynomial allow us to solve for $\mathbf{Ax} = \mathbf{b}$ by finding the element in the Krylov subspace that equals to $\mathbf{A}^{-1}\mathbf{b}$ for nonsingular matrix $\mathbf{A}$.

$$\begin{aligned} \mathbf{0} = q(\mathbf{A})\mathbf{b} &= \left(\left(\sum_{k=1}^{m-1} \alpha_k \mathbf{A}^k\right) + \alpha_0 \mathbf{I}\right)\mathbf{b} \\ &= \frac{1}{\alpha_0}\left(\sum_{k=1}^{m-1} \alpha_k \mathbf{A}^{k-1}\right)\mathbf{b} + \mathbf{A}^{-1}\mathbf{b} \end{aligned} \tag{2}$$

For large enough Krylov subspace, equations 1 and 2 provide solutions to $\mathbf{Ax} = \lambda\mathbf{x}$ and $\mathbf{Ax} = \mathbf{b}$ respectively. The Krylov subspace must equal to $\mathbb{R}^n$ in general to provide an exact solution, and this becomes impractical when n is even remotely large. If the Krylov subspace is any smaller, there is no nonzero polynomial that satisfies $p(\mathbf{A})\mathbf{b} = \mathbf{0}$.

What if, instead, the algorithm searches for a nonzero polynomial by minimizing the vector $p(\mathbf{A})\mathbf{b}$ with respect to the Euclidean norm?

$$\min_{\alpha_k} \|p(\mathbf{A})\mathbf{b}\|_2 \quad \text{subject to one of} \quad \begin{cases} \alpha_{m-1} = 1 \\ \alpha_0 = 1 \end{cases}$$

This polynomial approximation problem in $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ is a compact description of the *Arnoldi approximation problem* for the normalization constraint $\alpha_{m-1} = 1$. The polynomial's roots are known as Arnoldi eigenvalue estimates of the matrix $\mathbf{A}$, and converge to the exact eigenvalues as m approaches n. Similarly, the *general minimal residual approximation problem* (GMRES) estimates $\mathbf{A}^{-1}\mathbf{b}$ by minimizing the norm according to the normalization constraint $\alpha_0 = 1$.

There are numerous iterative methods that use the Krylov subspace. Some of these alternative algorithms are optimized for symmetric matrices, and as such can be applied to light transport matrices when the camera and projector are aligned. Other iterative methods rely on the transpose of $\mathbf{A}$ to compute $\mathbf{A}^T\mathbf{b}$. The transpose of the light transport matrix is not generally known, though it's possible to compute optically by swapping the locations of the projector and camera. Otherwise, the algorithms in light transport analysis are known as *transpose-free* iterative methods. We restrict our attention to the MATLAB functions *eigs*, *gmres*, and *minres* for computing eigenvectors and solving $\mathbf{Ax} = \mathbf{b}$ respectively.[1]

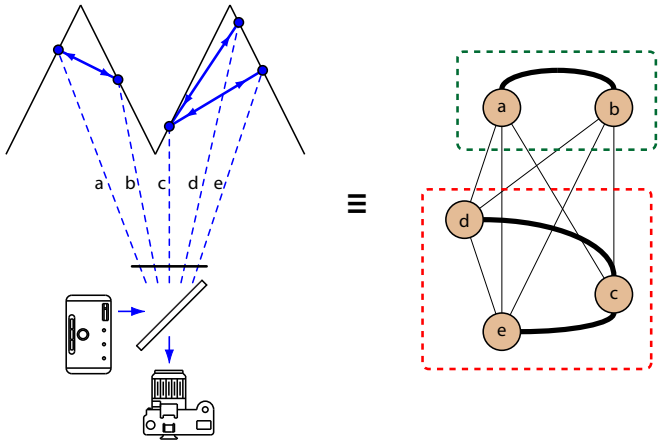

Figure 3: *This figure illustrates the light transport for a set of pixels $V = \{a, b, c, d, e\}$ in terms of a complete graph $G = (V, E)$. The edges of the graph are given by the throughput of light from some pixel i to another pixel j. In the left diagram, light scatters between pixels a and b, and therefore the corresponding edge (a, b) has a nonzero value describing the irradiance contribution. Similarly, no light scatters between pixel a and c, and therefore the weight of edge (a, c) is 0. The ideal bipartioning of this graph is $A = \{a, b\}$ and $B = \{c, d, e\}$, since this cuts the edges with weight 0. In other words, the sets A and B are radiometrically independent.*

5 Light Transport Segmentation

5.1 Light Transport Graph

As discussed in the previous section, recursive illumination allows eigenvectors and eigenvalues to be approximated without explicitly knowing the light transport matrix. It is not initially clear how to interpret these often complex eigenvectors and relate them to light transport for general matrices. By definition, the photograph of a scene illuminated by an eigenvector of the light transport matrix is the eigenvector scaled by its corresponding eigenvalue. By itself, this definition provides little insight on the inner workings of the light transport matrix. Ideally, we want to have a more intuitive physical interpretation of the eigenvectors to better analyze light transport.

For this section, suppose that the light transport matrix is symmetric by aligning the camera and projector with each other. An equivalent representation of this matrix comes in the form of a graph, one where the pixels are given by vertices and the edges describe the light transport. Specifically, we provide a visualization of light transport by drawing a complete graph according to an adjacency-matrix representation given by the light transport matrix. In short, this adjacency-matrix is a robust representation of a

[1] The details behind these algorithms are omitted - we refer the reader to [Lehoucq et al. 1997; Saad 1992; Bau, D. and Trefethen, L. N. 1996]. For more insight about the Krylov subspace, see appendix B.

weighted graph $G = (V, E)$. Each element a_{ij} of the adjacency-matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ has a value that describes the weight of an edge $(i, j) \in E$ going from vertex $i \in V$ to vertex $j \in V$. If the matrix is symmetric as given by $a_{ij} = a_{ji}$, the weight of edges (i, j) and (j, i) become equivalent and thus the graph itself becomes undirected. Our light transport graph weighs an edge (i, j) by the irradiance measured at some pixel i while the scene is illuminated by another pixel j.

If the amount of irradiance measured according to the weight of some edge (i, j) is high, we might say that pixels i and j are similar. The light transport graph has a number of properties that makes it a similarity graph. First, similarity graphs are defined by a non-negative weight function that produces small values if two vertices are dissimilar, and large otherwise. Second, as the concept of similarity between two pixels is commutative, similarity graphs are undirected. For light transport, the notion of similarity is related to the amount of light transport between two pixels.

Provided this mapping from matrices to similarity graphs, a variety of graph-based analysis methods can be applied to light transport. Graph-based segmentation is one such method that attempts to bipartition a graph into two disjoint sets A and B satisfying $A \cup B = V$ and $A \cap B = \emptyset$ by a function of similarity. In essence, we want for the partitions to have a high degree of similarity within each set and low across sets, according to some measure. There is no precise criterion for a "good" segmentation method, though there has been a number of measures suggested for image-based segmentation. For example, a popular choice among these measures is the normalized cut.

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{3}$$

By finding the two sets that minimizes this measure, the combined weight is low for the edges across sets and high for edges within each set. What does this mean to light transport graph segmentation? Performing this segmentation algorithm will group the pixels of an image into the two disjoint sets A and B, as a function of light transport. If we illuminate the scene with pixels from one of the partitions, we expect that the irradiance measured by pixels from the other partition to be small, because the weights of the edges across the disjoint sets are also small. Moreover, as the edge weight is high within each set, this suggests that light scatters across pixels within the set, or that the pixels are related to each other by global illumination.

Let's break this concept down a little further. Suppose that a scene has two concave regions, like in figure 3. If we project light into one region, the light will bounce around within the same region before reaching the camera. Similarly, any light in the second region stays within that region. There is no light that travels between the regions for this particular scene. Therefore, a good segmentation of this scene is expected to bipartition pixels according to these two regions.

Unfortunately, there are two problems associated with the discrete normalized cut algorithm of a light transport graph. As we have already discussed, the light transport matrix is often too large to measure directly, and therefore the light transport graph is unknown. Even if we have the graph, finding the optimal solution to the normalized cut problem is an *NP*-complete problem. However, a key feature of this problem is that it can be approximated by a generalized eigenvalue problem and, hence, the methods of the previous sections apply.

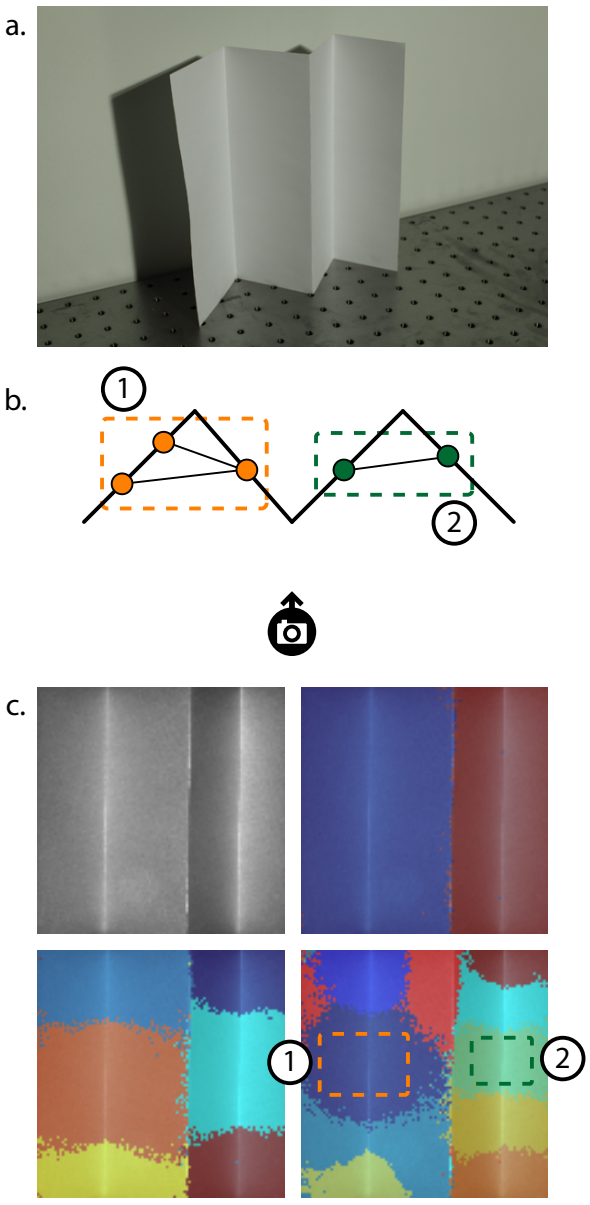

Figure 4: *Applying normalized cuts to light transport allows one to decipher how light scatters through a scene. For paper folded in a zig-zag shape, there are two concave regions within the scene that should be radiometrically independent from each other. After* 50 *iterations of the recursive procedure, a bipartition of the graph distinguishes the left and right concave regions of the image (top-right image from section c). Partitioning the graph into* 6 *groups (bottom-left image from section c) and* 10 *groups (bottom-right image from section c) begins cutting the concave regions horizontally. Region* 1 *and region* 2 *do not contribute light to one another, and therefore are grouped separately.*

5.2 Normalized Cuts by Recursive Illumination

Minimizing the normalized cut measure from equation 3 is equivalent to the following statement [Shi and Malik 1997].

$$\min_{\mathbf{x}} \frac{\mathbf{x}^T(\mathbf{D} - \mathbf{A})\mathbf{x}}{\mathbf{x}^T\mathbf{D}\mathbf{x}} \quad \text{subject to} \quad x_i \in \{1, -b\}, \mathbf{x}^T\mathbf{D}\mathbf{1} = 0 \tag{4}$$

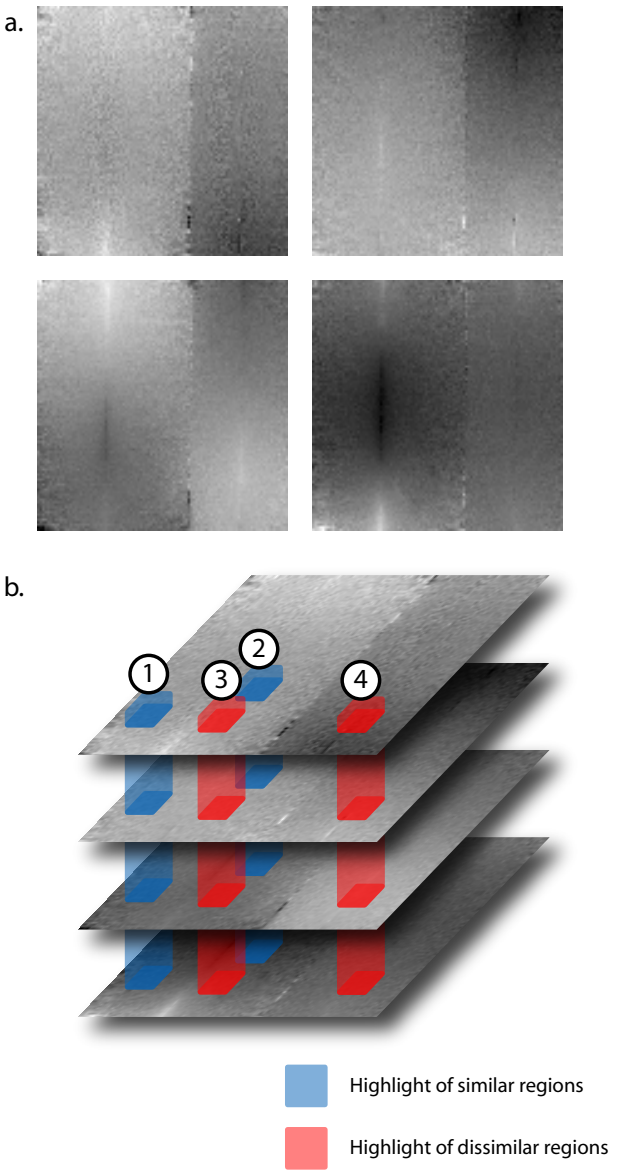

Figure 5: *Figure 5a illustrates the smallest generalized eigenvectors of the light transport matrix for the scene in figure 4. Stacking the k smallest eigenvectors on top of each other produces an image where each pixel has a feature vector of size k, as illustrated by figure 5b. The Euclidean distance of two feature vectors determines the similarity between pixels. If the distance is small as in columns 1 and 2, the pixels are similar and are likely to group together. Otherwise, the pixels are dissimilar as in columns 3 and 4.*

The degree matrix $\mathbf{D}$ is a diagonal matrix with components along the diagonal given by the vector $\mathbf{A1}$ for the symmetric light transport matrix $\mathbf{A}$. In other words, the degree of a vertex is the sum of its adjacent edge weights. Without going into too much detail, the vector $\mathbf{x}$ bipartitions the graph according to the constraint $x_i \in \{1, -b\}$ for some b. Removing this constraint allows the problem to be relaxed at the cost of the solution vector $\mathbf{x}$ having real components.

$$\min_{\mathbf{x}} \frac{\mathbf{x}^T(\mathbf{D}-\mathbf{A})\mathbf{x}}{\mathbf{x}^T\mathbf{D}\mathbf{x}} \quad \text{subject to} \quad \mathbf{x}^T\mathbf{D1} = 0 \tag{5}$$

Equation 5 has two distinct advantages over equation 4. First, the discrete *NP*-complete problem is reduced to solving a continuous problem in polynomial time. Second, the second smallest generalized eigenvector of $(\mathbf{D}-\mathbf{A})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$ solves the continuous problem exactly, and we can apply the recursive illumination methods discussed in section 4 to finding this eigenvector. Although there is no correlation between the solutions to the discrete and continuous problems, grouping pixels according to the real-valued components of the second smallest generalized eigenvector provides a good segmentation of the image with respect to the weights of light transport matrix $\mathbf{A}$. Specifically, if the difference between two real-valued components is small, the two corresponding pixels are likely to be grouped within the same partition. Similarly, if the difference is large, the pixels are more likely to be grouped with distinct partitions.

What if the second smallest generalized eigenvector $\mathbf{z}_2$ is not good enough? Perhaps we like to find another sufficiently different solution that also minimizes the continuous problem. We say that two solutions are different if they are orthogonal to each other with respect to the weight function $\mathbf{D}$, or that $\mathbf{z}_i^T\mathbf{D}\mathbf{z}_j = 0$. Applying $\mathbf{x}^T\mathbf{D}\mathbf{z}_2 = 0$ as a constraint to equation 5 produces a different solution to the segmentation problem, and the third smallest generalized eigenvector $\mathbf{z}_3$ solves this new problem exactly. By applying this argument recursively, this statement generalizes to finding the m smallest generalized eigenvectors, and suggests a segmentation algorithm based on these m vectors. Effectively, if the difference between two real-valued components for each of the m eigenvectors is small, the corresponding pixels are, again, likely to be grouped together. Otherwise, the pixels might not be grouped together.

Algorithm 2 Normalized Cuts

Compute the degree matrix $\mathbf{D} = diag(\mathbf{A1})$.
Find the m smallest generalized eigenvectors $(\mathbf{D}-\mathbf{A})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$.
Apply k-means to the n pixels using the m eigenvectors as feature vectors.

How do we find these generalized eigenvectors using recursive illumination? First, the diagonal degree matrix requires a single photo to compute. Specifically, the diagonal elements of the matrix are equal to the irradiance values of a scene illuminated by a plain white illumination pattern. The degree matrix can also be efficiently stored in memory because it is sparse. Second, we identify that the generalized eigenvalue problem is equivalent to the standard eigenvalue problem studied in section 4.

$$(\mathbf{D}-\mathbf{A})\mathbf{x} = \lambda\mathbf{D}\mathbf{x} \iff \mathbf{D}^{-0.5}(\mathbf{D}-\mathbf{A})\mathbf{D}^{-0.5}\mathbf{y} = \lambda\mathbf{y} \tag{6}$$

The two eigenvalue problems are related to each other by a transformation of their eigenvectors $\mathbf{y} = \mathbf{D}^{0.5}\mathbf{x}$. Therefore, the same Krylov-based methods are applicable to the generalized eigenvalue problem. Third, computing the matrix product of this standard eigenvalue problem requires a single photo, provided that we precompute the degree matrix. For example, if we define the light transport matrix product by a function $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$, the Krylov subspace for the generalized eigenvalue problem is built according to the following recursive illumination procedure.

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{D}^{-0.5}(\mathbf{D}-\mathbf{A})\mathbf{D}^{-0.5}\mathbf{x}_k \\ &= \mathbf{x}_k - \mathbf{D}^{-0.5}f(\mathbf{D}^{-0.5}\mathbf{x}_k)\end{aligned}$$

Thus, combining the normalized cuts algorithm with recursive illumination provides an efficient method for image segmentation as a function of the light transport matrix.

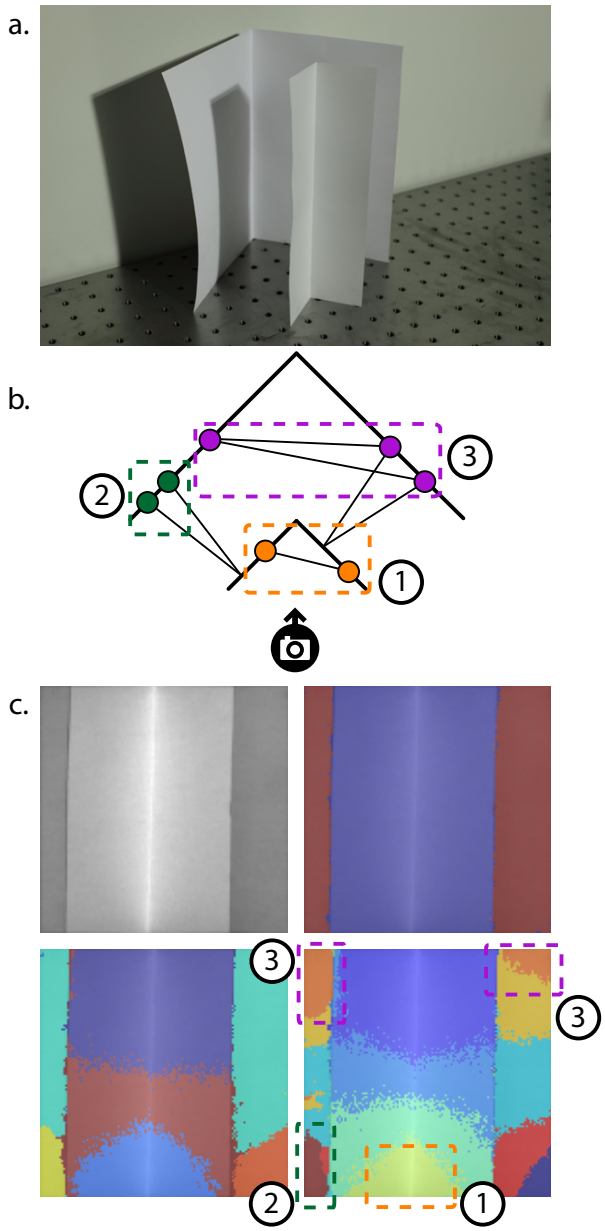

Figure 6: *This scene consists of two radiometrically independent regions, though one occludes the other. This example demonstrates that the regions are grouped together even though they are not connected in image space. Region* 1 *demonstrates similar cuts to that from figure 4 when raising the number of groups. Region* 2 *groups pixels due to reflections off the backside of the front sheet of paper. Region* 3 *transmits light behind the front paper to connect two patches that are spatially independent from each other in image space.*

5.3 Discussion of Light Transport Segmentation

There are some differences between light transport segmentation and standard image-based segmentation methods. In section 5.1, we associated light transport to a non-negatively weighted and undirected similarity graph, and its corresponding non-negative and symmetric adjacency-matrix representation. To keep our discussion simple, our description of similarity graphs omitted two key properties required by normalized cuts.

One issue has to do with the diagonal of the light transport matrix. A non-zero diagonal component is represented in a graph by drawing an edge from a vertex to itself. Similarity graphs usually assume that these self-loops are weighted by zero, as the notion of similarity only applies to different vertices. Why might non-zero diagonals be a problem? Segmentation has everything to do with cutting the edges between different vertices, and nothing to do with self-loops. In other words, modifying the self-loop weights should not change the segmentation solution. The normalized cut algorithm is, however, dependent on the diagonal.

How can we classify the diagonal and off-diagonal contributions in light transport terms? The off-diagonal components are due to the illumination of a pixel by other pixels in the scene, and constitutes as the global component. Conversely, each diagonal component is due to the direct illumination of a pixel. The light transport matrix satisfies $\mathbf{A} = \mathbf{A}_{direct} + \mathbf{A}_{global}$, where the diagonal matrix $\mathbf{A}_{direct}$ is extracted from the light transport matrix $\mathbf{A}$. Figure 7 demonstrates how a matrix is separated into its two matrix components. Ideally, any segmentation algorithm is uniquely a function of $\mathbf{A}_{global}$, and modifying $\mathbf{A}_{direct}$ should not affect its segmentation results.

$$\underset{\mathbf{A}}{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}} = \underset{\mathbf{A}_{direct}}{\begin{pmatrix} a_{11} & & \\ & a_{22} & \\ & & a_{33} \end{pmatrix}} + \underset{\mathbf{A}_{global}}{\begin{pmatrix} & a_{12} & a_{13} \\ a_{21} & & a_{23} \\ a_{31} & a_{32} & \end{pmatrix}}$$

Figure 7: *The decomposition of a* 3×3 *symmetric light transport matrix into its direct and global matrix components.*

We can approximately separate the direct and global vector components using high frequency illumiation [Nayar et al. 2006]. With the assumption that global illumination is a low frequency phenomenon, a high frequency pattern is swept over a scene and the variation in response becomes a function of the direct illumination. This method produces two direct and global images $\mathbf{A}_{direct}\mathbf{1}$ and $\mathbf{A}_{global}\mathbf{1}$, and their sum is given by $\mathbf{A1}$. Although we know very little about the global matrix component, it is trivial to construct the direct matrix component from the direct vector component for symmetric scenes. Effectively, the direct matrix component is a diagonal matrix, and its diagonal is the direct vector component. How does this relate back to segmentation? If we precompute the direct matrix component, any segmentation algorithm becomes a function of the global matrix component by defining the following light transport function.

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{A}_{global}\mathbf{x} \\ &= \mathbf{A}\mathbf{x} - \mathbf{A}_{direct}\mathbf{x} \end{aligned}$$

This light transport function takes a photo of a scene under some illumination and subtracts the direct contribution of the illumination pattern. Thus, the first discrepancy is remedied by precomputing the direct and global separation.

The second discrepancy occurs when there is no global contribution. If all the edges adjacent to some vertex are weighted by zero, how do we classify this vertex? The sum of these edge weights

is zero, and so is the corresponding component within the degree matrix. The generalized eigenvector problem requires the diagonal of the degree matrix to be strictly positive. Otherwise, the problem becomes ill-conditioned and is pointed out by equation 6. Although this is not likely to happen for synthetic image-based segmentation algorithms, there might be many pixels in a scene having no global contribution. We can ignore these pixels by creating a mask that only projects light onto the pixels with nonzero global components. Alternatively, we can add a $\mu\mathbf{1}$ term to the light transport function f to increase the weight of all edges in the graph by μ.

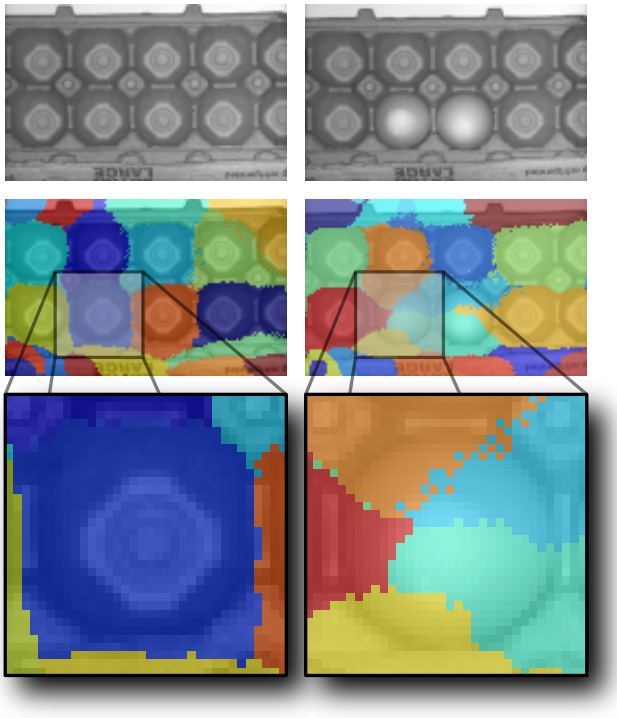

Figure 8: *We apply our recursive illumination method to partition a scene consisting of eggs sitting on their cardboard container. One hundred photos approximate the eigenvectors for this scene, and we partition the pixels into ten groups by applying the k-means algorithm to the twenty smallest generalized eigenvectors. Because of the scene is complex, we find that blurring the light transport matrix produces better results, as it introduces spatial coherency in the segmentation results.*

Not all segmentation algorithms suffer from these inconsistencies. Average cut is another measure applicable to image-based segmentation algorithms. The average cut measure is to the standard eigenvalue problem $(\mathbf{D}-\mathbf{A})\mathbf{x} = \lambda\mathbf{x}$, as normalized cut measure is to the generalized eigenvalue problem. This eigenvalue problem is by definition stable for any light transport matrix, and is already invariant with respect to the direct matrix component. However, the type of segmentations produced by average cut is different from that of normalized cut, though whether one is better than the other is up for debate.

6 Light Transport Compensation

6.1 Inverting Light Transport

A projector transforms any surface into a display. It is important for the surface to preserve the projector's image content, and so images are generally projected onto white diffuse planar surfaces. Otherwise, the quality of the projection deteriorates due to the scene distorting the image. The cause for these distortions falls into one of three categories: non-uniform albedos, non-planar surfaces, and indirect transport. The objective for an image compensation algorithm is to project an image that corrects for these distortions.

To illustrate the problem associated with non-uniform albedos, consider projecting an image onto a textured diffuse planar surface. The observed image becomes the projection image modulated with the textured surface. For example, consider projecting some image onto surface with either white or gray patches. The pixels that fall into the gray patches become darker than those in the white patches. By making the pixels in the gray regions brighter and those in the white areas darker, the projected image compensates for the variations in surface brightness.

Projections onto non-planar surfaces alters the observed image in two ways: disparities and defocus. First, if the observer is not colinearly aligned with the projector, image disparities or warping appears. To illustrate this point, suppose an image is projected onto a spherical surface. Depending on the position of the viewer, the image appears distorted on parts of the sphere. Some work on geometry-aware projectors studies this case in detail [Raskar et al. 2003]. Second, any depth variation within the scene incurs some amount of projector defocus. Projectors often have large apertures to produce bright images, at the cost of having a shallow depth of field. Most projectors cannot focus their image on scenes with large depth variations. Work on projector defocus analysis suggests an image compensation algorithm to correct for such a case [Zhang and Nayar 2006].

Correcting for non-uniform albedos and non-planar surfaces is sufficient for scenes with direct illumination only. Indirect illumination decreases projection quality with a wide range of other effects from caustics to low-frequency global illumination. In this section, we study a generalized image compensation algorithm that corrects for all three types of distortions by inverting the light transport matrix itself.

We formalize the image compensation problem as follows. Given a target image $\mathbf{b}$, the objective is to minimize the residual $\|\mathbf{b}-\mathbf{Ax}\|$ with a compensation image $\mathbf{x}$. If the difference between the observed image $\mathbf{Ax}$ and the target image $\mathbf{b}$ is small, the projection successfully preserves the image content. Suppose that the light transport matrix is nonsingular and square. Inverting the light transport matrix gives the exact solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, and minimizes the residual to 0. There are two problems with this approach. First, computing the inverse of the light transport matrix is computationally expensive. Second, this approach requires reconstructing the full matrix.

We relax the problem in the following way. Illuminating the scene with an image $\mathbf{u}_k$ produces a photo $\mathbf{v}_k = \mathbf{Au}_k$. Suppose now that a linear combination of the photographs produces the target image $\mathbf{b}$. It follows that the same linear combination of illumination patterns produces the compensation image solution, since $\mathbf{b} = \sum \alpha_k \mathbf{v}_k = \mathbf{A}\sum \alpha_k \mathbf{u}_k$. Projecting any random illumination pattern $\mathbf{u}_k$ converges slowly to the actual solution. Another approach is to build the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ by recursively illuminating the scene, starting with the target image $\mathbf{b}$. The generalized minimal residual method (GMRES) from section 4 is an

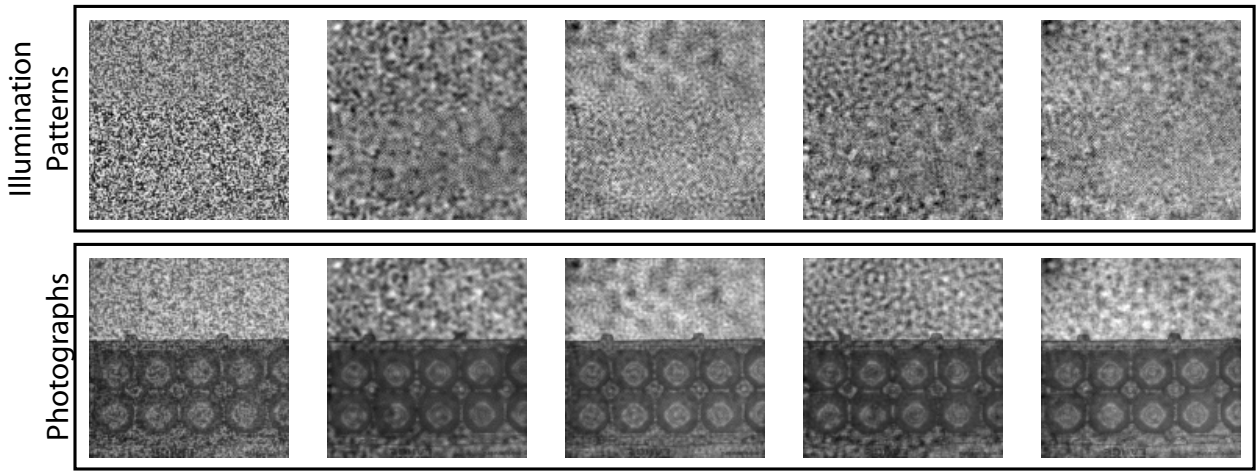

Figure 9: *The top row illustrates the first five illumination patterns for the light transport segmentation algorithm, from left to right. The images in the bottom row are the corresponding photographs of the scene. This scene is the same carton of eggs from figure 8, without the eggs.*

example of such an approach.

6.2 Residual Minimization Algorithms

The GMRES algorithm is the most general method for residual minimization. The algorithm converges in no more than n steps for a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. Unfortunately, the amount of work and storage increases linearly with each iteration. One workaround is to restart the method with a better initial solution after every m steps, at the cost of no longer guaranteeing convergence.

For symmetric systems, GMRES does more work than necessary. There are faster algorithms available, such as the minimal residual method (MINRES) based on a three-term recurrence relation and, for positive definite system, the conjugate gradient method. The amount of work and storage at each step remains constant, and the algorithms guarantee convergence after n steps. The convergence of these algorithms depends on the eigenvalue distribution. Moreover, the conjugate gradient method converges to a specified tolerance in $O(\sqrt{\kappa})$ iterations, where the condition number $\kappa = |\frac{\lambda_{\max}}{\lambda_{\min}}|$ of the positive definite matrix $\mathbf{A}$ is high for ill-conditioned systems and low for well-conditioned systems.

There are many more iterative residual minimization algorithms that exist, mostly for nonsymmetric systems. Some of these methods require both matrix multiplications $\mathbf{Ax}$ and matrix transpose multiplications $\mathbf{A}^T\mathbf{x}$.

6.3 Nonnegativity Constraint

Minimizing the residual does not always produce a valid solution. Even though the light transport matrix and target image are nonnegative, the solution to $\mathbf{Ax} = \mathbf{b}$ is not necessarily nonnegative and thus infeasible since one cannot project negative light.

Clamping the solution's negative components to zero produces projection artifacts. Another workaround is to modify the target image itself by linearly mapping the pixel values from within $[0, 1]$ to $[\alpha, 1]$, reducing the number of negative components in the solution at the cost of decreasing image contrast. The most sensible option is to modify the residual minimization algorithms to include a nonnegativity constraint.

6.4 Matrix Preconditioning

The convergence of these Krylov-based methods are a function of the distribution of eigenvalues, and this distribution depends on the scene. Even though the GMRES algorithm, and Arnoldi for that matter, converge monotonically to the optimal solution, the rate of convergence slows down for badly conditioned matrices. One uses preconditioners to improve the convergence of these iterative methods.

$$(\mathbf{M}_{left}^{-1}\mathbf{A}\mathbf{M}_{right}^{-1})\mathbf{M}_{right}\mathbf{x} = \mathbf{M}_{left}^{-1}\mathbf{b}$$

The solution to the equation above is the same as $\mathbf{Ax} = \mathbf{b}$. With the inclusion of left and right preconditioners $\mathbf{M}_{left}$ and $\mathbf{M}_{right}$, the convergence of Krylov-based methods now depends on $\mathbf{M}_{left}^{-1}\mathbf{A}\mathbf{M}_{right}^{-1}$ instead of $\mathbf{A}$. A good preconditioner drastically improves the convergence of these iterative methods. The best preconditioner matrices satisfy $\mathbf{A} = \mathbf{M}_{left}\mathbf{M}_{right}$, though this assumes the light transport matrix and its inverse are already known. Instead, the objective is to find preconditioners that approximate the matrix.

In practice, these preconditioner matrices approximate the effects of non-uniform albedos and non-planar surfaces on the light transport matrix. Although modeling indirect illumination is nontrivial, several nonlinear methods exist that efficiently approximate pixel disparities, projector defocus, and variations in direct illumination. By preconditioning the matrix for these effects, the performance of the image compensation algorithm increases.

7 Imaging and Illumination

7.1 Numerical Stability

As the Krylov subspace grows through recursive illumination, the photos become almost linearly dependent on each other. For instance, these photos converge to the same principal eigenvector when the largest eigenvalue of the matrix is distinct. Therefore, these photos span an exceedingly ill-conditioned Krylov subspace. Any error, in terms of numerical precision or sensor noise, greatly affects the subspace spanned by these photos.

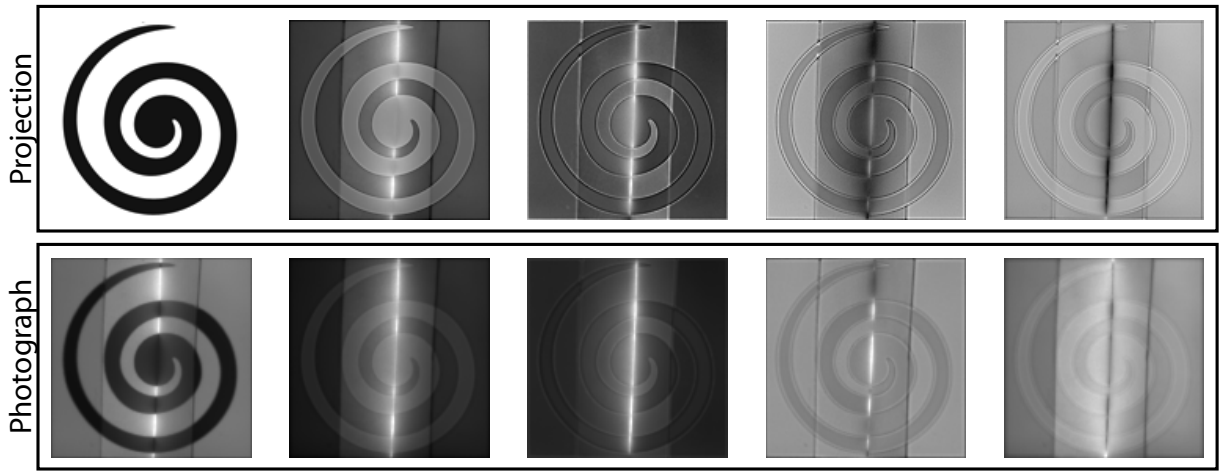

Figure 10: *Illumination patterns and the corresponding photographs for the first five iterations of the image compensation procedure, increasing from left to right. This is the same scene as in figure 12, and demonstrates the effects of global illumination and projector defocus.*

Krylov subspace methods minimize numerical issues through an orthogonalization process named Gram-Schmidt. An orthogonal projector transforms each photo to be orthogonal to all previous photos, without affecting the subspace spanned by the images. An orthogonal set of vectors is well-conditioned, and the vectors span a subspace less sensitive to errors. This stabilizes the recursive illumination procedure by reducing the influence of error on the Krylov subspace.

An image is no longer necessarily nonnegative after the orthogonal projection step, and illuminating a scene with negative light is physically impossible. To compensate, an affine transformation maps the illumination $\mathbf{x}$ to a nonnegative space, and another function maps the corresponding photo back to $\mathbf{Ax}$. For example, $g(\mathbf{x}; a, b) = \frac{\mathbf{x}-a}{b-a}$ maps an image $\mathbf{x} \in [a, b]^n$ to $g(\mathbf{x}) \in [0, 1]^n$. After illuminating the scene with $g(\mathbf{x})$ to get a photo $\mathbf{A}g(\mathbf{x})$, the next step is to evaluate $\mathbf{Ax} = (b - a)(\mathbf{A}g(\mathbf{x})) + a(\mathbf{A1})$. Note that this requires taking an additional photo $\mathbf{A1}$ of the scene under uniform illumination, though this step is only necessary once for a scene.

7.2 Quantization

Another source of error comes from signal quantization, due to rounding or truncation. The colour depth for modern projectors and displays is generally 8-bit per channel.[2] If the illumination is not an 8-bit signal, some information is lost because of rounding. To clarify, imagine a 1-bit projector that illuminates a scene with either white or black pixels. Any input image with gray intensity values needs to either round its pixel values up to white or down to black. The round-off error in the illumination in turn affects the photographs.

A workaround is to decompose the n-bit image into n 1-bit images. Although the decomposition is not unique, one approach is to recursively project the quantization error of a signal, as in figure 13. A weighted sum of these n 1-bit images reconstructs the original n-bit image. After illuminating the scene with the 1-bit images and taking the corresponding photographs, the same weighted sum of the photos synthesizes a photo of the scene as though illuminated by the n-bit image.

[2]Some recent HDR imaging solutions provide higher bit depths.

Photographs are also subject to quantization error. Many modern cameras produce photos with high bit-depths, reducing the influence of rounding errors. However, truncation error occurs when light undersaturates or oversaturates camera pixels. To overcome this limitation, a high dynamic range (HDR) photo combines multiple photos with different exposure settings such that each pixel is appropriately saturated in at least one photo [Debevec and Malik 1997].

8 Results

8.1 System Setup

A prototype camera-projector system is built for the light transport segmentation and compensation algorithms. The system consists of a Mitsubishi PK20 DLP projector and a Canon EOS-1Ds Mark III camera, both aligned with a beam splitter. The light from the projector intersects the beam splitter, and half of the light enter the scene. For the other half, we position and tilt an inactive LCD monitor to minimize the amount of light that reflects back towards the camera. The Canon camera captures photographs in a 14-bit sRAW format with a linear response. The projector is also calibrated to have a linear gamma function.

When assuming symmetry in light transport, the camera and projector alignment must be accurate. One approach to verify symmetry is to place several objects within the scene and start the projector. The light from the projector produces shadows within the scene. If the system is calibrated correctly, the camera does not observe any direct shadows.

For most camera and projector combinations, there are more camera pixels than projector pixels. The discussion so far assumes that the light transport matrix is square, or that a linear mapping exists from camera pixels to projector pixels. One approach is to project a checkerboard pattern, and pair each projector pixel to one camera pixel at the center of the corresponding checkerboard square. The calibration toolbox from `www.vision.caltech.edu/bouguetj/calib_doc/` provides some useful functionality for this approach. Another alternative is to sample multiple camera pixels that fall within the footprint of a single projector pixel and averaging their values. Gray patterns are useful for this approach, though requires several photographs.

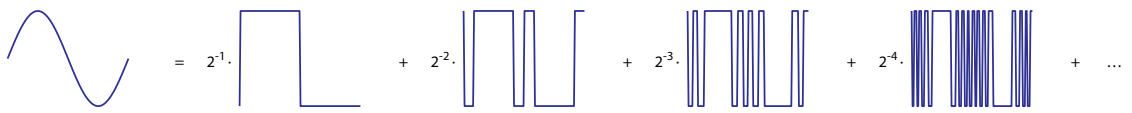

Figure 13: *A projector illuminates the scene with a sinusoidal pattern, and a camera takes the corresponding photo. Multiple photos at various exposures capture high dynamic range images to produce photos with high bit-depths. If the projector pixels have binary values, the sinusoidal pattern is subject to severe quantization errors, and same goes for the photographs. By decomposing the signal into a series of binary illumination patterns and collecting their corresponding photos, a weighted sum of the photos produces the correct response. Note that photos with small weights become dominated by the sensor noise of the photos with large weights.*

A short MATLAB program implements the light transport segmentation and compensation algorithms. The program defines a function that takes an input vector **x**, projects the unvectorized image onto the scene, photographs the response **Ax**, and returns the result. The function also implements the workaround for images with negative pixel values. The function handle is passed as a parameter to the MATLAB function *eigs* for computing eigenvalues and eigenvectors, or to *gmres*/*minres* for residual minimization.

Several flags are available to the *eigs* function. Enabling the symmetry flag produces eigenvectors that are numerical orthogonal. Another flag controls the number of Lanczos vectors that restarts the algorithm once the Krylov subspace becomes a certain size. To avoid memory and performance issues when working with large systems, the number of Lanczos vectors is often set to a small value, at the cost of accuracy. In practice, we determine its best to not restart the Krylov subspace, unless memory becomes an issue. Therefore, the number of Lanczos vectors is set to the quota for photographs, and the maximum iterations (restarts) flag is set to one.

8.2 Image Segmentation

We apply light transport segmentation to three scenes. The two scenes from figures 4 and 6 are simple examples to illustrate the results. The last example from figure 8 applies light transport segmentation to a more general scenes. After calibration, a uniform black pattern and a uniform white pattern illuminate the scene. The black pattern measures the ambient light of the scene to remove this component from all the photographs. The algorithm uses the white pattern for computing the degree matrix, and projecting negative illumination patterns. The Krylov subspace method begins after projecting these two images.

The first two scenes involves folded paper. The paper is a diffuse material with a fair amount of subsurface scattering properties. Apart from the initial images, the Krylov subspace grows to a maximum size of $n \times 50$, at which point the algorithm terminates. This subspace contains approximations to the 49 smallest eigenvector, though only the smallest of these eigenvectors are accurate. The algorithm uses k-means on smallest 10 eigenvectors to produce the segmentation results. The same eigenvectors allow pixels to be clustered into any number of groups. For the scene from figure 4, we ask the algorithm to cluster the image into two, six, and ten groups. The segmentation into two groups is clean because the two concave regions are radiometrically independent. Increasing the number of groups to six, the segmentation algorithm begins to cut the two concave regions horizontally, describing how light scatters from one face to another. Increasing the number of groups to ten continues the same trend, though some vertical cuts appear. Figure 6 demonstrates how the segmentation is spatially independent. Although there is a foreground object that occludes the background object, the segmentation algorithm disambiguates between the two objects. Region 3 highlights how two spatially disconnected patches relate to each other by light transport. For the six and ten groupings, horizontal cuts appear in both the foreground and background objects, as in the previous scene. Although the light transport matrix has dimension $100^2 \times 100^2$ and $128^2 \times 128^2$ for these two scenes, 50 photographs appear sufficient for approximating the segmentation of these scenes.

The third scene from figure 8 produces segmentation results of an egg carton from 50 photos. To improve the rate of convergence, the algorithm blurs each photograph and illumination pattern slightly, which biases the solution to be spatial coherent. The same procedure is performed twice: once for a scene with eggs in the carton and once without. When no eggs are within the carton, each cell becomes highlighted as a cluster. Once an egg enters the cell, the clustering changes accordingly because of the convex geometry of the egg.

8.3 Image Compensation

The image compensation results in this paper are generated from the minimum residual method *minres*, available in MATLAB. The minimum residual method solves the image compensation problem for symmetric systems by iterating through a three-term recurrence relation. This method only ever has to store a few images in memory at any given step. Solving the image compensation problem for nonsymmetric light transport matrices requires using methods that handle nonsymmetric systems. The conjugate gradient squared (*cgs*) and generalized minimal residual (*gmres*) methods are examples of algorithms that handle these cases.

In figure 11, an image is projected onto a piece of cardboard paper folded to make a 90 degree angle. Any light received on one plane also illuminates the second plane due to indirect illumination. This effect is highlighted by region 1 in the figure. Because of the scene geometry, the images appear brighter and washed out near the fold of the paper. In as few as five iterations, the image compensation algorithm finds a compensation image that minimizes the residual error in the observed image. Note that region 1 appears to match the target image relatively well.

The second example from figure 12 illustrates the effects of both indirect illumination and defocus. Region 1 highlights the area with indirect illumination, and region 2 highlights the effects of defocus. The camera and projector are not focused on the backplane of the scene. Although the residual error is not minimized to zero, the algorithm reaches convergence after about eight iterations. Note that the algorithm does not necessarily converge to zero, as one would expect for singular systems. Moreover, the algorithm does not specify any nonnegativity constraints, and instead clamps negative values to zero. Although the results are not seamless, the solution approximately matches the target image. Region 2 demonstrates how the edges are enhanced to preserve the sharp detail from the target image.

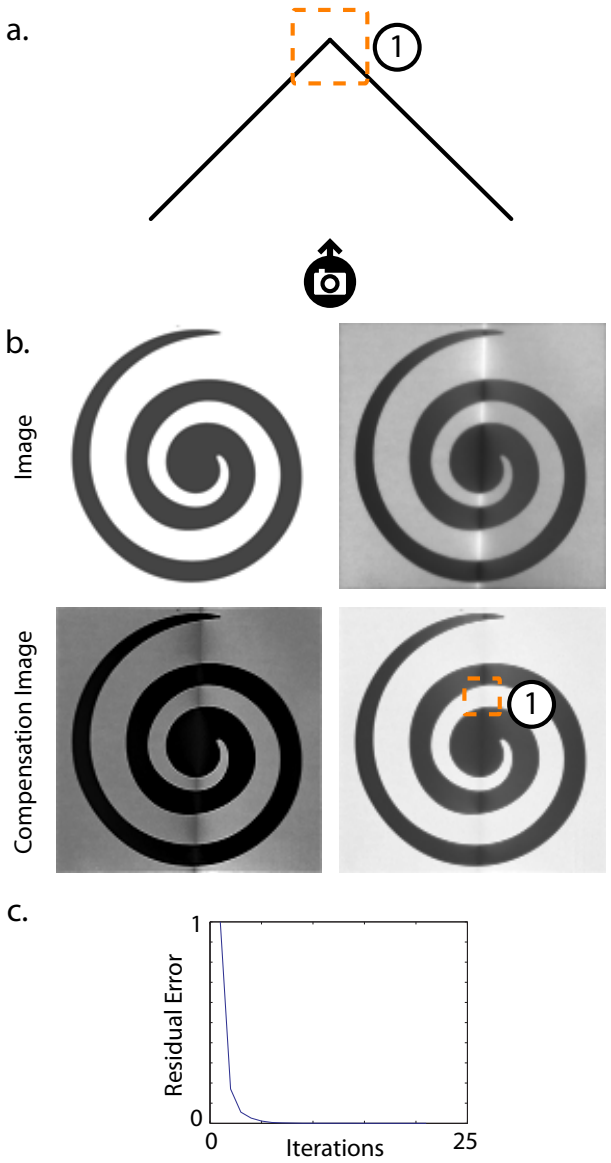

Figure 11: *A simple scene demonstrating the effects of projecting into the corner of a room. In figure 11b, the top-left image is the target image, and the top-right is the result of projecting the image onto the scene. By compensating for the scene, the compensation image (bottom-left) produces a projection much closer to the target image (bottom-right). The residual error from figure 11c also decreases very quickly, and converges after about* 5 *images.*

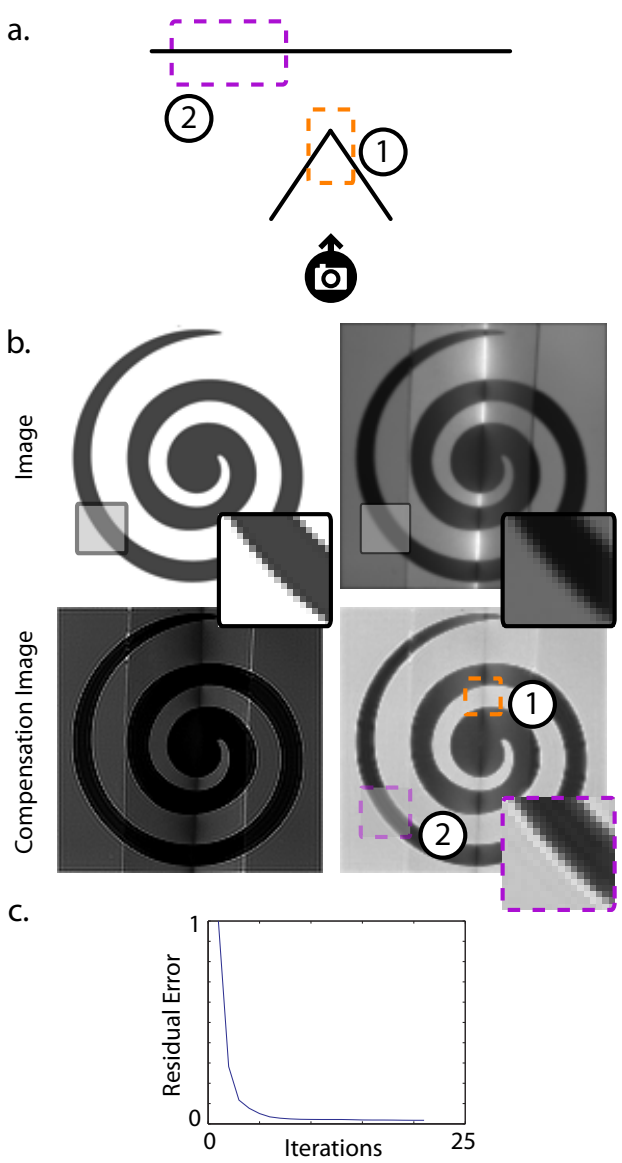

Figure 12: *The image compensation algorithm also handles defocus, and large amount of indirect illumination. The back wall, highlighted by region* 2*, produces a blurry image. When projecting the target image (top-left in figure 12b) directly onto the scene, the parts of the image projected onto the back wall become blurry, as demonstrated by the zoomed-in region of the projected result (top-right in figure 12b). By finding a compensation image that corrects for both defocus and indirect illumination (bottom-left in figure 12b), the final result is a good approximation of the target image.*

9 Conclusion

In this paper, we describe a computational illumination method that takes a photo of a scene and lights the scene with the photo recursively. We relate our adaptive computation illumination approach to the power iteration, Arnoldi, and GMRES algorithms. These algorithms provide a means to segment images according to sets of pixels that mutually illuminate each other, by performing spectral clustering using the light transport matrix. A general light transport compensation algorithm preserves projected image content by accounting for blur, disparity, and global illumination effects from light transport.

These algorithms differ from other adaptive methods in the following ways. First, the adaptive method is a 1-term recurrence relation, where the photo from iteration $k-1$ is the illumination pattern for iteration k. Second, unlike methods that work hierarchically, there is no notion of spatial coherency. For example, the image segmentation algorithm groups pixels independently from the distance between pixels, and any spatial coherency is a result of light transport. Third, these adaptive methods output unique properties from

Figure 14: *A top-down view of the setup for capturing the eigenvectors of symmetric light transport matrices.*

the light transport matrix, namely its eigenvalues, eigenvectors, and the solution to the minimal residual problem.

One of the limitations in the approach is that the light transport matrix must be square. Otherwise, there is no one-to-one mapping from camera pixels to projector pixels. The convergence rate of the algorithms is also dependent on the light transport matrix itself, though they converge exactly after n iterations. A light transport matrix with complex effects slows the convergence of these Krylov subspace methods.

There are a number of areas for future work. We discuss finding the eigenvectors and eigenvalues of nonsymmetric systems, but it is not yet clear how to use this information. This paper does not discuss the application of these Krylov subspace methods to anything other than the dual photography setup. Another extension is to remove the transpose-free restriction by swapping the camera and projector, similar to the approach from Wang et al. [Wang et al. 2009]. Although we briefly discuss preconditioners, there are many variants of preconditioners for both *eigs* and *minres*, including some multiscale approaches that exploit coherency. Lastly, there are perhaps ways to approximate the full light transport matrix, direct/global separation, or even disparity maps more efficiently with these Krylov subspace methods.

This paper uses Krylov subspace illumination towards some novel applications in light transport segmentation and compensation, though the work only covers a few optical setups, applications, and algorithms. Projecting the high-dimensional light transport matrix into a much smaller Krylov subspace reduces the dimensionality of the eigenvalue and minimal residual problems, allowing for an efficient adaptive approach to light transport analysis.

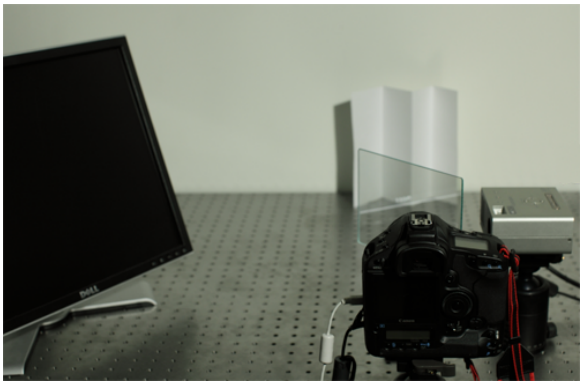

Figure 15: *Another view of the setup for our recursive illumination procedure. The LCD monitor on the left is a light trap, minimizing the amount of light reflecting back towards the camera. We found that a diffuse black cloth reflects back more light than the specular black screen of an LCD.*

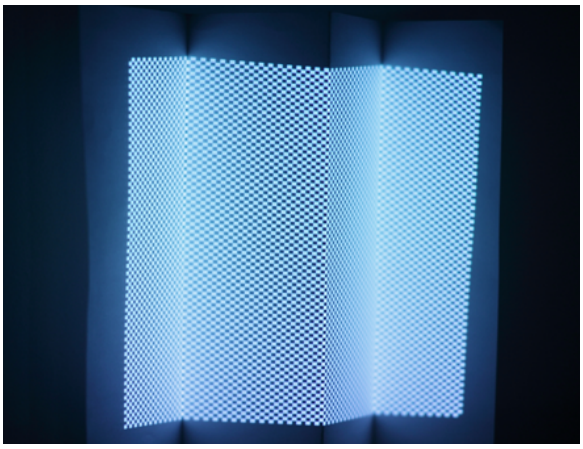

Figure 16: *The checkerboard pattern for mapping projector pixels to camera pixel. We apply a corner detection algorithm to this image to find the center of each checkerboard square. This mapping defines the square light transport matrix.*

Acknowledgements

I am indebted to my supervisor Kyros Kutulakos and the insightful conversations we shared over the past couple of years. I would also like to thank Allan Jepson for his helpful comments on this work. Finally, thanks to Christian Lessig and Nigel Morris for all the interesting conversations, Jack Wang and Koji Yatani for the dinners, and all the other DGP'ers for making Toronto feel like home.

References

BAU, D., AND TREFETHEN, L. N. 1996. *Numerical Linear Algebra.* SIAM Publications. (To be published).

BERMAN, A., AND PLEMMONS, R. J. 1979. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York. Reprinted by SIAM, Philadelphia, 1994.

CHABERT, C.-F., EINARSSON, P., JONES, A., LAMOND, B., MA, W.-C., SYLWAN, S., HAWKINS, T., AND DEBEVEC, P. 2006. Relighting human locomotion with flowed reflectance fields. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, ACM, New York, NY, USA, 76.

CHEN, T. B., LENSCH, H. P. A., FUCHS, C., AND SEIDEL, H. P. 2007. Polarization and phase-shifting for 3D scanning of translucent objects. In *CVPR*, 1–8.

CHEN, T. B., SEIDEL, H. P., AND LENSCH, H. P. A. 2008. Modulated phase-shifting for 3D scanning. In *CVPR*, 1–8.

DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *SIGGraph-97*, 369–378.

FUCHS, M., BLANZ, V., LENSCH, H. P., AND SEIDEL, H.-P. e 07. Adaptive sampling of reflectance fields. *ACM Trans. Graph. 26*, 2, 10.

GARG, G., TALVALA, E.-V., LEVOY, M., AND LENSCH, H. P. A. 2006. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Eurographics Workshop/ Symposium on Rendering*, Eurographics Association, Nicosia, Cyprus, T. Akenine-M, Ed., 251–262.

GHOSH, A., ACHUTHA, S., HEIDRICH, W., AND O'TOOLE, M. 2007. BRDF acquisition with basis illumination. In *ICCV*, 1–8.

KRISHNAN, D., AND FERGUS, R. 2009. Dark flash photography. *ACM Transactions on Graphics, SIGGRAPH 2009 Conference Proceedings*, To appear.

LEHOUCQ, R. B., SORENSEN, D. C., AND YANG, C. 1997. ARPACK USERS GUIDE: Solution of large scale eigenvalue problems by implicitly restarted arnoldi methods.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2008. A closed-form solution to natural image matting. *IEEE Trans. Pattern Analysis and Machine Intelligence 30*, 2 (feb), 228–242.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 31–42.

LISCHINSKI, D., ACHA, A. R., AND LEVIN, A. 2007. Spectral matting. In *CVPR*, 1–8.

NARASIMHAN, S. G., KOPPAL, S. J., AND YAMAZAKI, S. 2008. Temporal dithering of illumination for fast active vision. In *ECCV*, IV: 830–844.

NAYAR, S. K., KRISHNAN, G., GROSSBERG, M. D., AND RASKAR, R. 2006. Fast separation of direct and global compnents of a scene using high frequency illumination. *ACM Transactions on Graphics 25*, 3 (jul), 935–944.

PEERS, P., MAHAJAN, D. K., LAMOND, B., GHOSH, A., MATUSIK, W., RAMAMOORTHI, R., AND DEBEVEC, P. 2009. Compressive light transport sensing. *ACM Transactions on Graphics 28*, 1 (jan), 3:1–3:??

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3, 664–672.

RASKAR, R., VAN BAAR, J., BEARDSLEY, P., WILLWACHER, T., RAO, S., AND FORLINES, C. 2003. iLamps: geometrically aware and self-configuring projectors. *ACM Transactions on Graphics 22*, 3 (jul), 809–818.

SAAD, Y. 1992. *Numerical Methods for Large Eigenvalue Problems*. Halsted Press, New York.

SALVI, J., PAGES, J., AND BATLLE, J. 2004. Pattern codification strategies in structured light systems. *Pattern Recognition 37*, 4 (apr), 827–849.

SEITZ, S. M., MATSUSHITA, Y., AND KUTULAKOS, K. N. 2005. A theory of inverse light transport. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 1440–1447.

SEN, P., AND DARABI, S. 2009. Compressive Dual Photography. *Computer Graphics Forum 28*, 2, 609 – 618.

SEN, P., CHEN, B., GARG, G., MARSCHNER, S. R., HOROWITZ, M., LEVOY, M., AND LENSCH, H. P. A. 2005. Dual photography. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 24*, 3 (aug), 745–755.

SHI, J., AND MALIK, J. 1997. Normalized cuts and image segmentation. *IEEE Conf. Computer Vision and Pattern Recognition* (jun).

WANG, J., DONG, Y., TONG, X., LIN, Z., AND GUO, B. 2009. Kernel nyström method for light transport. *ACM Trans. Graph. 28*, 3, 1–10.

WEISS, Y. 1999. Segmentation using eigenvectors: A unifying view. In *ICCV*, 975–982.

ZHANG, L., AND NAYAR, S. K. 2006. Projection defocus analysis for scene capture and image display. *ToG 25* (jul), xx–yy.

ZHANG, L., CURLESS, B., AND SEITZ, S. M. 2002. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *3DPVT02*, 24–36.

A Perron-Frobenius Theorem

Define $\rho(\mathbf{A})$ to be the spectral radius of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where the spectral radius returns the absolute value of the largest eigenvalue in magnitude. The Perron-Frobenius theorem for non-negative matrices states the following [Berman and Plemmons 1979]:

1. If $\mathbf{A}$ is positive, then $\rho(\mathbf{A})$ is a simple eigenvalue, greater than the magnitude of any other eigenvalue.
2. If $\mathbf{A} \geq \mathbf{0}$ is irreducible then $\rho(\mathbf{A})$ is a simple eigenvalue, any eigenvalue of $\mathbf{A}$ of the same modulus is also simple, $\mathbf{A}$ has a positive eigenvector $\mathbf{x}$ corresponding to $\rho(\mathbf{A})$, and any non-negative eigenvector of $\mathbf{A}$ is a multiple of $\mathbf{x}$.

It follows that, for a reducible matrix $\mathbf{A} \geq \mathbf{0}$, $\mathbf{A}$ has a set of n orthogonal positive eigenvectors $\{\mathbf{x}_k\}$ corresponding to $\rho(\mathbf{A})$ with multiplicity n.

B Analysis of Krylov Subspace

The Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ of a matrix $\mathbf{A}$ and random vector $\mathbf{b}$ contains information about the largest eigenvalues of the matrix. To explain this concept a little further, note first that every matrix with linearly independent eigenvectors has an eigenvalue decomposition, or eigendecomposition.

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$$

The $n \times n$ matrix $\mathbf{\Lambda}$ is a diagonal matrix whose entries are the eigenvalues of matrix $\mathbf{A}$, and their corresponding eigenvectors are the columns of matrix $\mathbf{Q}$.

In the power iteration algorithm, the vector $\mathbf{A}^m\mathbf{b}$ converges to the principal eigenvector as m increases, assuming that the largest eigenvalue is distinct in magnitude.

$$\begin{aligned}\mathbf{A}^m\mathbf{b} &= (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1})^m\mathbf{b} \\ &= \mathbf{Q}\mathbf{\Lambda}^m\mathbf{Q}^{-1}\mathbf{b}\end{aligned}$$

As the exponent m gets large, the eigenvectors of matrix $\mathbf{A}^m$ do not change, but the eigenvalues become smaller in magnitude relative to the largest eigenvalue of the matrix. Once m is large enough, the matrix $\mathbf{\Lambda}^m$ becomes approximatly rank-1. It follows that $\mathbf{A}^m$ also becomes a rank-1 matrix. Therefore, so long as $\mathbf{b}$ is not orthogonal to the principal eigenvector, $\mathbf{A}^m\mathbf{b}$ reads off the largest eigenvector of the matrix.

When the largest eigenvalue is not distinct, power iteration does not necessarily converge. However, the latter images in the recursive sequence do become linearly dependent. Suppose that the matrix has k eigenvalues that equal in magnitude. The matrix $\mathbf{\Lambda}^m$ and thus $\mathbf{A}^m$ converge to rank-k matrices. Therefore, $\mathbf{A}^m\mathbf{b}$ produces a vector that is a linear combination of the principal eigenvectors of the matrix $\mathbf{A}$.