

Optical Computing for Fast Light Transport Analysis

Matthew O’Toole*

Kiriakos N. Kutulakos*

University of Toronto

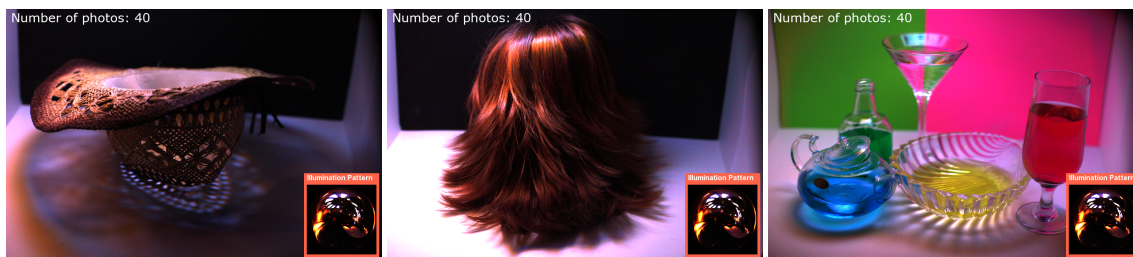


Figure 1: Our approach enables very efficient acquisition and analysis of light transport: to create the relighting results shown above, just forty low dynamic range photos were used to acquire $700\text{Kpixel} \times 100\text{Kpixel}$ transport matrices. Note the complex shadows cast by the hat (both sharp and soft), the complex highlights on the hair and the shadows it casts, and the many shadows, caustics and indirect lighting effects in the glass scene. We used an optical implementation of the Arnoldi algorithm to do both photo acquisition and low-rank matrix approximation; the entire process (photo capture, matrix reconstruction, relighting) took four minutes on a standard PC for each scene.

Abstract

We present a general framework for analyzing the transport matrix of a real-world scene at full resolution, without capturing many photos. The key idea is to use projectors and cameras to directly acquire eigenvectors and the Krylov subspace of the unknown transport matrix. To do this, we implement Krylov subspace methods partially in optics, by treating the scene as a “black box subroutine” that enables optical computation of arbitrary matrix-vector products. We describe two methods—*optical Arnoldi* to acquire a low-rank approximation of the transport matrix for relighting; and *optical GMRES* to invert light transport. Our experiments suggest that good quality relighting and transport inversion are possible from a few dozen low-dynamic range photos, even for scenes with complex shadows, caustics, and other challenging lighting effects.

1 Introduction

More than forty years ago, analog optical computing—processing data with light sources, modulators, lenses and detectors—sought to revolutionize the field of information processing [Leith 2000; Goodman 2005]. Its key tenet was that the speed and parallelism of light enables very high data rates, especially for signal processing and pattern recognition tasks [Ams 2009]. Although the versatility and rapid advance of microprocessors ultimately overshadowed optical computing’s early ambitions, the designs studied were very diverse; examples range from general-purpose optical computers [Guilfoyle and Stone 1991] to highly specialized ones for matrix calculations [Athale and Collins 1982; Kumar and Casasent 1981].

*e-mail: {motoole, kyros}@cs.toronto.edu

In this work we present a first attempt to apply optical computing principles to the analysis of light transport in a real-world scene. Specifically, we show how to efficiently perform numerical computations on a scene’s unknown transport matrix (eigenvector analysis, low-rank approximation, inversion) by doing part of those computations in the optical domain with projectors and cameras.

The light transport matrix has been gaining attention in computer graphics as an effective tool for relighting real-world scenes [Debevec et al. 2000; Wang et al. 2009]. This matrix allows the photo of any scene, no matter how complex, to be expressed as a matrix-vector product [Ng et al. 2003; Sen et al. 2005]:

$$\mathbf{p} = \mathbf{T} \mathbf{l} \quad (1)$$

where \mathbf{p} is the photo, represented as a column vector of camera pixels, \mathbf{T} is the transport matrix, and \mathbf{l} is the scene’s illumination condition (e.g., pixels in a pattern that is projected onto the scene). By construction, \mathbf{T} takes into account all transport paths from light sources to camera pixels. Therefore, if we know \mathbf{T} , we can render the scene from the camera’s viewpoint under any illumination condition \mathbf{l} —with shadows, caustics and interreflections all included.

Real-world transport matrices pose unique challenges. They can be extremely large, with over 10^{11} elements for typical camera and projector resolutions. Storage and analysis of the full matrix is therefore infeasible. Moreover, the only way to get information about \mathbf{T} is by multiplying it with a vector, i.e., by illuminating the scene with vector \mathbf{l} and capturing the product, $\mathbf{T}\mathbf{l}$, in a photo.

To deal with these issues, a common methodology emerged for designing an acquisition pipeline [Sen et al. 2005]: (1) choose a specific arrangement for cameras and light sources, (2) choose a scene-independent ensemble of vectors for illumination, (3) design an algorithm that draws samples from the ensemble, adaptively or non-adaptively, to illuminate the scene during acquisition, and (4) choose an algorithm to reconstruct \mathbf{T} from captured photos. Unfortunately, despite several recent efforts [Fuchs et al. 2007; Wang et al. 2009; Peers et al. 2009; Sen and Darabi 2009], existing systems require hundreds or thousands of high dynamic range photos for complex scenes, can be computationally demanding, and their performance is hard to characterize in a scene-independent way.

Although interest in the transport matrix is relatively recent, very large matrices have long been a subject of study in numerical linear algebra. In particular, the family of *Krylov subspace methods* [Saad

2003] is designed for matrices just like \mathbf{T} , *i.e.*, very large and unobservable matrices that can only be accessed by computing their product with a vector. These iterative algorithms are well understood and come with explicit accuracy and convergence guarantees.

Here we leverage this body of work for light transport by implementing Krylov subspace methods partially in optics. Our approach is based on a simple principle: treat the scene as a “black-box subroutine” that accepts any non-negative vector \mathbf{l} as “input” and returns as “output” the vector’s product, $\mathbf{T}\mathbf{l}$, with the unknown transport matrix. Thus, any efficient numerical method that relies exclusively on matrix-vector products can be readily implemented in optics and used to analyze \mathbf{T} . To do the conversion, we just replace all matrix-vector products with calls to a function that computes them *optically*, with illuminate-and-capture operations (Figure 2). This turns Krylov subspace methods into complete pipelines for analyzing \mathbf{T} —as they pursue their numerical objective, they fully specify how to illuminate the scene and how to process its photos.

Implementing Krylov subspace methods directly in the optical domain has several advantages. First, the convergence rate of these methods depends only on the distribution of \mathbf{T} ’s singular values, not its absolute size. This means that \mathbf{T} can be analyzed at full resolution by capturing very few photos. Second, computations are efficient because the only computationally-expensive step is multiplying the full-resolution \mathbf{T} with a vector—which we do optically. Third, optical implementations are straightforward because they differ from widely-available numerical software in just one step, *i.e.*, multiplication with \mathbf{T} . Last but not least, by moving this multiplication to the optical domain we make other computations feasible on the full-resolution \mathbf{T} , beyond mere acquisition—computing eigenvectors of \mathbf{T} , computing products with \mathbf{T} ’s inverse—*without having to acquire the transport matrix first*.

We focus on optical versions of two Krylov subspace methods in this paper: *Arnoldi iteration* to acquire a low-rank approximation of \mathbf{T} for relighting (Section 3); and *generalized minimal residual (GMRES)* to invert light transport (Section 4). In the following we assume that illumination vectors in Equation 1 have m elements and photos have n pixels, *i.e.*, \mathbf{T} is an $n \times m$ matrix.

2 Computing with Light

2.1 A Simple Example: Optical Power Iteration

We begin by showing how to implement *power iteration* in optics. Power iteration is a simple numerical algorithm for estimating the principal eigenvector of a square matrix with distinct eigenvalues [Trefethen and Bau 1997]. When implemented optically, it estimates the principal eigenvector of \mathbf{T} without advance knowledge of the matrix and without directly capturing any of its elements.

Power iteration uses the fact that the sequence $\mathbf{l}, \mathbf{T}\mathbf{l}, \mathbf{T}^2\mathbf{l}, \mathbf{T}^3\mathbf{l}, \dots$ converges to \mathbf{T} ’s principal eigenvector for almost any initial vector \mathbf{l} . The algorithm simply generates this sequence for a fixed number of iterations using the boxed matrix-vector product on the left:

Algorithm 1 The power iteration algorithm.

Numerical Implementation:	Optical Implementation:
In: matrix \mathbf{T} , iterations K	In: iterations K
Out: principal eigenvector of \mathbf{T}	Out: principal eigenvector of \mathbf{T}
1: $\mathbf{l}_1 =$ random vector	$\mathbf{l}_1 =$ positive vector
2: for $k = 1$ to K	for $k = 1$ to K
3: $\mathbf{p}_k = \mathbf{T}\mathbf{l}_k$	illuminate scene with vector \mathbf{l}_k capture photo & store in \mathbf{p}_k
4: $\mathbf{l}_{k+1} = \mathbf{p}_k / \ \mathbf{p}_k\ _2$	$\mathbf{l}_{k+1} = \mathbf{p}_k / \ \mathbf{p}_k\ _2$
5: return \mathbf{l}_{k+1}	return \mathbf{l}_{k+1}

Implementing power iteration in optics amounts to replacing this product with the illuminate-and-capture operation shown on the right. This is possible when the transport matrix is square, *i.e.*, when illumination vectors and captured photos are the same size.

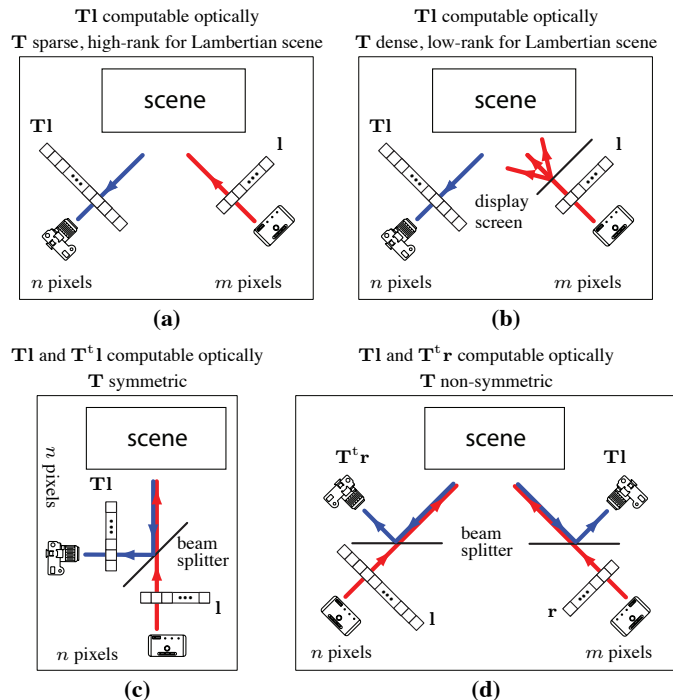


Figure 2: Principle of optical computing for light transport. To analyze an unknown transport matrix \mathbf{T} , we use standard numerical methods with just one modification. As they execute, whenever they must compute \mathbf{T} ’s product with a vector \mathbf{l} , we call a function that computes this product optically by illuminating the scene with \mathbf{l} and capturing photos. **Camera-projector arrangements.** We use three degrees of freedom in our arrangements: number of distinct viewpoints (one or two); ability to compute products with \mathbf{T}^t (yes or no); and type of illumination, *i.e.*, projecting patterns directly onto the scene as in (a), or displaying them on a screen, as in (b). Four of the eight possible arrangements are shown above. **How scenes and light sources affect \mathbf{T} ’s rank and sparsity.** Suppose the scene in (a) and (b) is the same convex Lambertian object. Displaying a pattern directly onto the object, as in (a), will transfer it to the photo in the form of a distorted texture; this means that \mathbf{T} has maximal rank and sparsity, and can be acquired by structured-light scanning [Salvi *et al.* 2004]. Displaying the same pattern on a screen, as in (b), transfers only low frequencies to the camera in the form of shading; \mathbf{T} is now a dense low-rank matrix whose acquisition is equivalent to photometric stereo [Basri and Jacobs 2003]. General scenes make acquisition much more challenging by bridging the gap between these two extremes, *i.e.*, by reducing rank and sparsity in (a) and increasing them in (b). Note that \mathbf{T} ’s properties in (a) and (b) would be reversed if the object was a perfect mirror.

Optical implementation turns the power iteration algorithm into an illumination procedure with a feedback loop. The procedure repeatedly captures a photo, converts it to a unit vector, and uses it to illuminate the scene. See Figure 3 for a diagram of this procedure. Figure 4 shows an example of using it to compute the principal eigenvector of the full-resolution transport matrix for a real scene.

Although very simple algorithmically, optical power iteration highlights an important point about our general approach: the efficiency of image acquisition is directly related to the convergence proper-

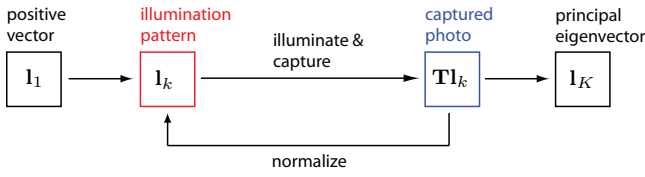


Figure 3: Power iteration with a projector and a camera.

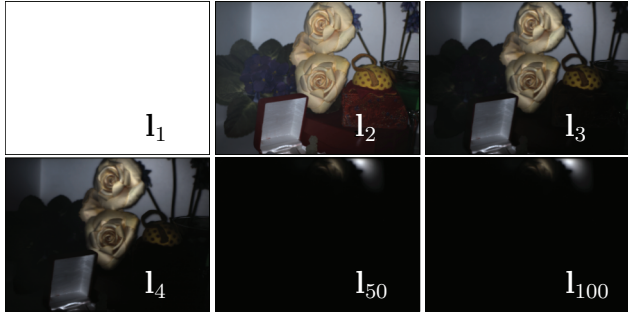


Figure 4: Optical power iteration in action. We used the coaxial arrangement of Figure 2(c) for this example, where a camera and a projector share the same viewpoint and \mathbf{T} is symmetric. We started with a constant illumination vector \mathbf{l}_1 , shown above, so the first photo of the scene was captured under constant illumination. That photo became the next illumination vector, \mathbf{l}_2 , also shown above. The illumination vectors change very little after about 50 captured photos, indicating that a good approximation of \mathbf{T} 's principal eigenvector has already been found.

ties of the underlying numerical algorithm—the faster it converges, the fewer photos its optical implementation needs to capture.

From a numerical standpoint, power iteration is not an efficient algorithm for computing eigenvectors. It computes just one eigenvector, albeit the principal one, and the approximation error decreases by a factor of $|\lambda_2|/|\lambda_1|$ at each iteration, where λ_1, λ_2 are the top two eigenvalues of \mathbf{T} [Trefethen and Bau 1997]. The algorithm may converge very slowly when \mathbf{T} 's top two eigenvalues are similar, and may not converge at all if they are identical. Naturally, these limitations are shared by its optical counterpart.

To analyze light transport efficiently, we focus on much more efficient numerical algorithms from the class of Krylov subspace methods, discussed below.

2.2 Optical Krylov Subspace Methods

Krylov subspace methods represent some of the most important iterative algorithms for solving large linear systems [Saad 2003]. Their relevance for light transport comes from the existence of powerful methods for analyzing large sparse matrices, like \mathbf{T} , be it square or rectangular, and symmetric or non-symmetric.

Briefly, the Krylov subspace of dimension k is the span of vectors produced by power iteration after k steps:

$$\begin{array}{ccccccc} \mathbf{l}_1 & \mathbf{l}_2 & \mathbf{l}_3 & \cdots & \mathbf{l}_{k+1} & & \\ & \Downarrow & \Downarrow & & \Downarrow & & \\ & \mathbf{T}\mathbf{l}_1 & \mathbf{T}^2\mathbf{l}_1 & \cdots & \mathbf{T}^k\mathbf{l}_1 & & \end{array} . \quad (2)$$

While individual algorithms differ in their specifics, Krylov subspace methods take an initial vector \mathbf{l}_1 as input and, in their k -th iteration, compute a vector in the Krylov subspace of dimension k .

The important characteristic of these methods is that they do not require direct access to the elements of \mathbf{T} ; all they need is the ability to multiply \mathbf{T} , and potentially its transpose, with a vector. This makes them readily implementable in optics.

Optical matrix-vector products for general vectors Unlike power iteration, general Krylov subspace methods require multiplying \mathbf{T} with vectors that may contain negative elements. Even though we cannot illuminate the scene with negative light, implementing such products optically is straightforward. We follow the approach outlined by Goodman [2005], and express a general vector \mathbf{l} as the difference of two non-negative vectors \mathbf{l}^p and \mathbf{l}^n :

$$\mathbf{l} = \mathbf{l}^p - \mathbf{l}^n \quad (3)$$

$$\mathbf{T}\mathbf{l} = (\mathbf{T}\mathbf{l}^p) - (\mathbf{T}\mathbf{l}^n) . \quad (4)$$

To implement Equation 4 optically, we use two illuminate-and-capture operations: one to compute $\mathbf{T}\mathbf{l}^p$ and one to compute $\mathbf{T}\mathbf{l}^n$. We then subtract the two captured photos to get the product with \mathbf{l} .

Symmetric vs. non-symmetric transport matrices The convergence behavior of Krylov subspace methods like GMRES depends quite significantly on whether or not \mathbf{T} is a symmetric matrix [Liesen and Tichy 2004]. In this paper we restrict ourselves to the symmetric case, where convergence is well understood, by choosing appropriate projector-camera arrangements.

There are two general ways to enforce symmetry when implementing Krylov subspace methods in optics. The first is to make sure that \mathbf{T} itself is symmetric. This can be done with the coaxial arrangement of Figure 2(c). This configuration takes advantage of Helmholtz reciprocity and is quite common [Seitz et al. 2005; Zhang and Nayar 2006]. It is also quite limited because it does not allow any viewpoint variations between the projector and camera.

A second way to enforce symmetry is to apply the methods to a different matrix whose symmetry is guaranteed:

$$\mathbf{T}^* = \mathbf{T}^t \mathbf{T} . \quad (5)$$

Optically multiplying \mathbf{T}^* with a vector, however, involves matrix-vector products with both \mathbf{T} and its transpose:

$$\mathbf{T}^* \mathbf{l} = \mathbf{T}^t (\mathbf{T} \mathbf{l}) . \quad (6)$$

A single camera-projector pair is not enough to compute both products optically. For this, we use the arrangement of Garg *et al.* [2006] shown in Figure 2(d). This arrangement uses two camera-projector pairs and enables two distinct project-and-capture operations: one for computing $\mathbf{T}\mathbf{l}$ (“illuminate from left, capture from right”) and one for computing $\mathbf{T}^t\mathbf{l}$ (“illuminate from right, capture from left”).

Arnoldi and GMRES Krylov subspace methods come in many flavors depending on the numerical objective (eigenvalue estimation, solution of linear systems, *etc.*); type of matrix (symmetric, non-symmetric, positive definite, *etc.*); and error tolerance. We explore two of these methods here: *Arnoldi iteration* for efficiently acquiring a low-rank approximation of \mathbf{T} , and *generalized minimal residual (GMRES)* for inverting light transport.

When implemented in optics, Arnoldi and GMRES follow the same basic loop as optical power iteration. They capture a photo, process it, project the result back onto the scene, and repeat for a fixed number of iterations. Both methods differ from power iteration in just three steps. These differences are summarized in Table 1.

Algorithm	Numerical objective	Step 1	Step 4	Step 5
Power iteration (Section 2.1)	estimate principal eigenvector of \mathbf{T}	$\mathbf{l}_1 =$ positive vector	$\mathbf{l}_{k+1} = \mathbf{p}_k / \ \mathbf{p}_k\ _2$	return \mathbf{l}_{k+1}
Arnoldi (Section 3)	compute rank- K approximation of \mathbf{T}	$\mathbf{l}_1 =$ non-zero vector	$\mathbf{l}_{k+1} = \text{ortho}(\mathbf{l}_1, \dots, \mathbf{l}_k, \mathbf{p}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \ \mathbf{l}_{k+1}\ _2$	return $[\mathbf{p}_1 \cdots \mathbf{p}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^\dagger$
Generalized minimal residual (Section 4)	find vector \mathbf{l} such that $\mathbf{p} = \mathbf{T} \mathbf{l}$	$\mathbf{l}_1 =$ target photo \mathbf{p}	$\mathbf{l}_{k+1} = \text{ortho}(\mathbf{l}_1, \dots, \mathbf{l}_k, \mathbf{p}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \ \mathbf{l}_{k+1}\ _2$	return $[\mathbf{l}_1 \cdots \mathbf{l}_K][\mathbf{p}_1 \cdots \mathbf{p}_K]^+ \mathbf{p}$

Table 1: The two Krylov subspace methods we implement in this paper, along with their differences from power iteration. In the numerical steps listed above, the function $\text{ortho}()$ projects the last vector in its parameter list onto the orthogonal subspace of the rest; $[\]^\dagger$ denotes the matrix pseudoinverse. See the relevant sections for a discussion of the algorithms and their optical implementation.

3 Optical Arnoldi for Transport Acquisition

We now show how to efficiently capture low-rank approximations of full-resolution transport matrices with the Arnoldi method.

The Arnoldi method can be thought of as a much more efficient version of power iteration. At the k -th iteration, the algorithm has enough information to estimate the top k eigenvectors (or singular vectors) of a matrix rather than just one of them. Unlike power iteration, Arnoldi does not estimate these eigenvectors directly. It generates a sequence of orthogonal vectors, $\mathbf{l}_1, \dots, \mathbf{l}_k$, whose span approximates the span of the top k eigenvectors. This approximation, which tightens as k increases, makes it possible to compute a rank- k approximation of the matrix without direct access to its elements. We refer the reader to Saad [2003] for the basic theory behind the Arnoldi method and to [Greenbaum and Trefethen 1994; Simon and Zha 2000] for a convergence analysis and detailed error bounds on the rank- k approximation it enables.¹

Two optical implementations of Arnoldi are shown in Algorithm 2, one for symmetric and one for non-symmetric \mathbf{T} . For a symmetric \mathbf{T} , the implementation amounts to substituting three steps in power iteration’s optical implementation with the corresponding steps from Table 1. Because the vectors generated by Arnoldi contain negative values, we use two illuminate-and-capture operations per matrix-vector product, one for their positive and one for their negative component. The optical implementation for non-symmetric transport matrices uses two sets of illuminate-and-capture operations, applied to \mathbf{T} and \mathbf{T}^\dagger respectively, in order to implement Equation 6. Therefore, we capture four photos per Arnoldi iteration when \mathbf{T} is non-symmetric. Figure 5 shows the schematic diagram for this case and Figure 6 shows an example run.

Algorithm 2 Optical Arnoldi for rank- K transport acquisition.

For symmetric \mathbf{T} :	For non-symmetric \mathbf{T} :
In: iterations K	In: iterations K
Out: rank- K approximation of \mathbf{T}	Out: rank- K approximation of \mathbf{T}
1: $\mathbf{l}_1 =$ non-zero vector	$\mathbf{l}_1 =$ non-zero vector
2: for $k = 1$ to K	for $k = 1$ to K
3: <div style="border: 1px solid black; padding: 5px; display: inline-block;">illuminate with \mathbf{l}_k^p and \mathbf{l}_k^n capture photos \mathbf{p}_k^p and \mathbf{p}_k^n $\mathbf{p}_k = \mathbf{p}_k^p - \mathbf{p}_k^n$</div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">left-illuminate with \mathbf{l}_k^p and \mathbf{l}_k^n right-capture photos \mathbf{d}_k^p and \mathbf{d}_k^n $\mathbf{d}_k = \mathbf{d}_k^p - \mathbf{d}_k^n$ $\mathbf{r}_k = \mathbf{d}_k / \ \mathbf{d}_k\ _2$ <div style="border: 1px solid black; padding: 5px; display: inline-block;">right-illuminate with \mathbf{r}_k^p and \mathbf{r}_k^n left-capture photos \mathbf{s}_k^p and \mathbf{s}_k^n $\mathbf{s}_k = \mathbf{s}_k^p - \mathbf{s}_k^n$</div></div>
4: $\mathbf{l}_{k+1} = \text{ortho}(\mathbf{l}_1, \dots, \mathbf{l}_k, \mathbf{p}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \ \mathbf{l}_{k+1}\ _2$	$\mathbf{l}_{k+1} = \text{ortho}(\mathbf{l}_1, \dots, \mathbf{l}_k, \mathbf{s}_k)$ $\mathbf{l}_{k+1} = \mathbf{l}_{k+1} / \ \mathbf{l}_{k+1}\ _2$
5: return $[\mathbf{p}_1 \cdots \mathbf{p}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^\dagger$	return $[\mathbf{d}_1 \cdots \mathbf{d}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^\dagger$

¹Note that another often-used method, *Lanczos*, is the Arnoldi method optimized for symmetric matrices.

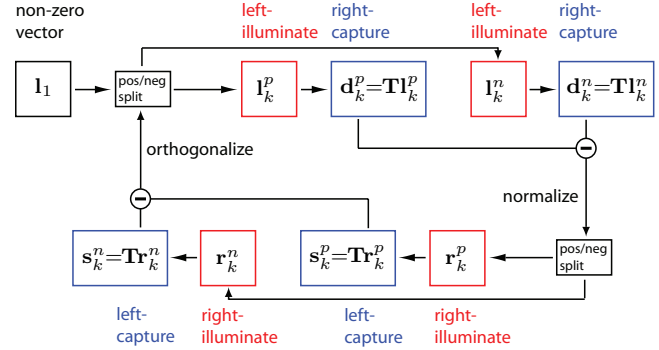


Figure 5: Arnoldi with two projectors and two cameras.

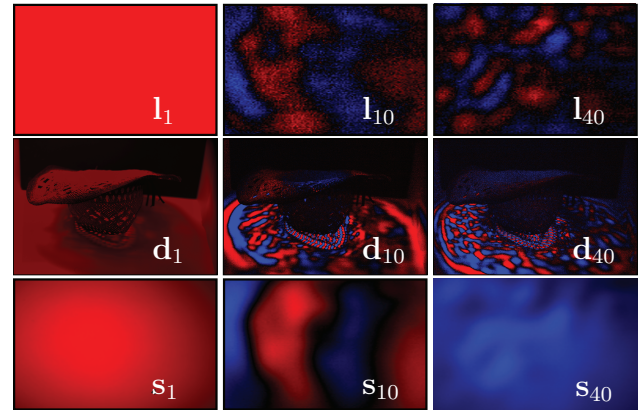


Figure 6: Optical Arnoldi in action for the hat scene rendered in Figure 1. We used the non-symmetric configuration in Figure 11(top), with the left pair behind a display screen. The photos and illumination vectors contain both positive (red) and negative (blue) values. We started with a constant positive left illumination vector (\mathbf{l}_1) so the first photo captured with the right camera (\mathbf{d}_1) was positive as well. This photo was then projected onto the scene with the right projector. Note that since the left camera views the scene from behind the display screen, which is essentially a diffuser, its photos contain mainly low frequencies (\mathbf{s}_1). We produced the relighting result in Figure 1 by estimating a low-rank \mathbf{T} from ten Arnoldi iterations (i.e., 40 low dynamic range photos).

The optical Arnoldi algorithm can be thought of as building two bases simultaneously: an orthogonal basis for the subspace of illumination vectors and a basis for the subspace of photos. For a symmetric \mathbf{T} , the illumination basis contains all vectors the algorithm used to illuminate the scene; the photo basis contains all photos it captured. In the non-symmetric case, the illumination basis

contains the left-illumination vectors (first row of Figure 6) and the photo basis the right-captured photos (second row of Figure 6). The matrix itself, returned in the algorithm’s last step, just multiplies these two bases together.

Scene relighting To render a scene under a novel illumination vector \mathbf{l} , we rewrite Equation 1 in terms of the captured illumination and photo bases. The equation for the non-symmetric case becomes

$$\mathbf{p} = [\mathbf{d}_1 \cdots \mathbf{d}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^t \mathbf{l}. \quad (7)$$

This equation can be thought of as a two-step relighting procedure: first we compute \mathbf{l} ’s coordinates in the left-illumination basis by projecting \mathbf{l} onto it; then we linearly combine the right-captured photos to obtain the relighting result, \mathbf{p} .

3.1 Relation to Prior Work on Transport Acquisition

We discuss related work from a numerical perspective in terms of four properties— \mathbf{T} ’s rank, sparsity, row space, and symmetry.

An important distinction between methods is the rank and sparsity of matrices they acquire. As we illustrate in Figure 2, this distinction is implicit in the choice of a light source and scene. Techniques geared toward *sparse high-rank matrices* [Sen et al. 2005; Garg et al. 2006; Peers and Dutré 2005] rely on \mathbf{T} ’s ability to transfer both high and low frequencies from the illumination domain to the camera domain; techniques acquiring *dense low-rank matrices* [Debevec et al. 2000; Fuchs et al. 2007; Wang et al. 2009] assume that high-frequency illumination does not propagate to the camera domain. Optical Arnoldi is primarily applicable to dense low-rank matrices. These are often representative of natural settings, where illumination comes from point or area sources and where mirror reflection and sharp shadows usually do not dominate light transport.

The choice of illumination ensemble used to acquire the transport matrix is critical because it controls the basis for \mathbf{T} ’s row space. To maximize efficiency, this ensemble should allow accurate reconstruction of \mathbf{T} ’s rows from as few illumination vectors as possible. Many ensembles have been used for this purpose, including Haar wavelets [Peers and Dutré 2003], Hadamard patterns [Schechner et al. 2007] and single-source illuminations [Fuchs et al. 2007]. For instance, Wang *et al.* [2009] use the low-rank configuration in Figure 2(b) and single-source illumination vectors to reconstruct \mathbf{T} ’s rows with the kernel Nyström method. These ensembles have been *scene independent* in all previous work on transport acquisition.² For low-rank matrices, no scene-independent ensemble is optimal. The optimal ensemble under the Frobenius norm is *scene dependent* and consists of \mathbf{T} ’s singular vectors [Trefethen and Bau 1997]. This is precisely the ensemble optical Arnoldi approximates.

Garg *et al.* [2006] and Wang *et al.* [2009] used coaxial camera-projector arrangements to exploit the fact that knowing a subset of both rows and columns of \mathbf{T} makes it easier to reconstruct the rest. Numerically, however, symmetry has much more fundamental effect on a matrix, as it affects its eigenstructure. While we use similar camera-projector arrangements, our choices are guided primarily by numerical convergence considerations.

Sen *et al.* [2009] and Peers *et al.* [2009] recently used compressed sensing techniques to reconstruct individual rows of \mathbf{T} . These methods are complementary to our own, as they apply to a different matrix class—sparse, high-rank matrices—for which a low-rank approximation might lead to rendering artifacts. The scene-independent ensembles of these methods, however, are inefficient

²Although techniques have been proposed for sampling vectors from *within* an ensemble in a scene-dependent way (*e.g.*, [Fuchs et al. 2007]), the ensemble itself is fixed and independent of the scene.

for capturing dense low-rank matrices. They are also very expensive computationally and depend on the size of \mathbf{T} . Here, by seeking to maximize the “information content” of each captured photo, optical Arnoldi makes the number of iterations required for convergence dependent on \mathbf{T} ’s singular value distribution, not its size.

Computing transport eigenvectors Eigenvectors and singular vectors of real-world transport matrices have been used for compression [Matusik et al. 2002] and to accelerate rendering [Mahajan et al. 2007]. In all cases, they were computed after acquiring \mathbf{T} . With optical Arnoldi, we analyze light transport in reverse: we first construct a basis that approximates the span of the top K transport eigenvectors, and then use that basis to reconstruct the matrix.

4 Optical GMRES for Inverse Transport

We now consider an optical solution to the following problem. We are given a target photo \mathbf{p} and seek an illumination vector \mathbf{l} that produces it. Mathematically, this can be expressed as a solution to Equation 1 where the unknown is \mathbf{l} , not \mathbf{p} .

Generalized minimal residual (GMRES) is a Krylov subspace method that iteratively solves this problem for unobservable matrices without inverting them, using just matrix-vector products [Saad 2003]. As shown in Table 1, the method is almost identical to Arnoldi: the only difference is its initial vector (it is always \mathbf{p}) and its return value, which is a solution to the least-squares problem:

$$\mathbf{l} = \arg \min_{\mathbf{x}} \left\| [\mathbf{p}_1 \cdots \mathbf{p}_K][\mathbf{l}_1 \cdots \mathbf{l}_K]^t \mathbf{x} - \mathbf{p} \right\|_2, \quad (8)$$

where \mathbf{p}_k and \mathbf{l}_k are computed in the k^{th} Arnoldi iteration. In essence, GMRES builds a rank- K approximation of \mathbf{T} and then inverts it to compute \mathbf{l} .

Despite its apparent simplicity, GMRES is an extremely powerful algorithm. It applies to any matrix (low-rank, high-rank, dense, sparse, *etc.*) and converges rapidly for arbitrary non-singular symmetric matrices [Liesen and Tichy 2004]. Intuitively, GMRES does this by “exploring” only a portion of \mathbf{T} ’s row space, *i.e.*, the subspace that is precisely suitable for inverting the initial vector \mathbf{p} .

The optical implementation of GMRES is identical to Arnoldi’s. We simply run optical Arnoldi with a photo \mathbf{p} as the initial illumination vector and, after the algorithm terminates, we solve Equation 8 computationally (Step 5 of GMRES in Table 1).³

In principle, it should be possible to use optical GMRES to invert light transport efficiently for any full-rank transport matrix, regardless of sparsity and size. This, for instance, would allow us to infer the illumination that produced a given photo of a scene, even when both the scene and the illumination are very complex. Figures 7, 8 and Section 6.2 show initial demonstrations of such a capability on both high-rank and low-rank matrices, at high resolution.

For singular transport matrices two possibilities exist: there may be many different illuminations that can produce a given photo (Equation 1 has multiple solutions) or none at all (Equation 1 is infeasible). When many solutions exist, optical GMRES can efficiently return one of them, although not necessarily the one used to produce the original photo. When no solutions exist, it will return the best-possible approximation lying within the rank- K subspace it built. Its convergence behavior in this case is unclear, however.

³The solution to Equation 8 may contain negative elements. Although we just clamp them to zero, another approach is to add a non-negativity constraint and solve the equation using MATLAB’s *lsqnonneg()* function).

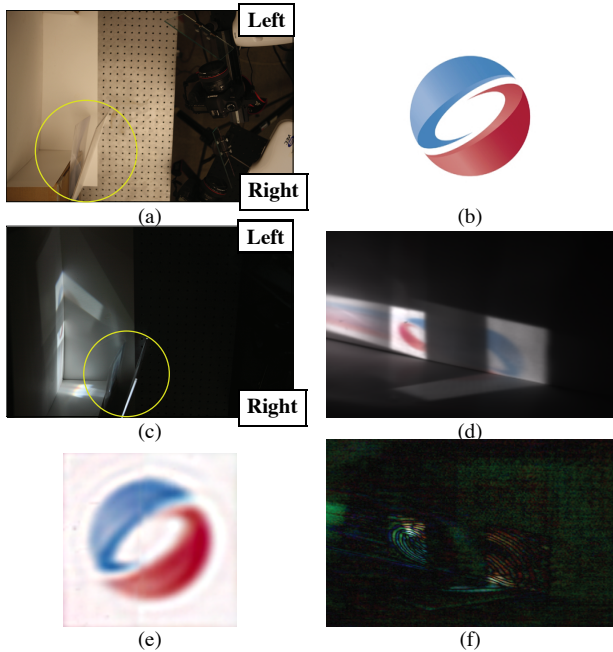


Figure 7: Inverse light transport for a high-rank \mathbf{T} . A top view of our setup is shown in (a). We placed a fresnel lens and an LCD backlight diffuser (both circled) in front of two camera-projector pairs to create the arrangement of Figure 2(d). We then displayed the SIGGRAPH logo onto the scene with the right projector and captured a photo with the left camera. A top view of the scene under this illumination is shown in (c) and the captured photo in (d). Note the evidence of complex low-rank and high-rank transport in this photo (diffused regions and distorted/replicated/broken-up logo, respectively). We then provided the photo in (d)—with no additional information—to optical GMRES to test its ability to reassemble the logo we actually displayed. The result of optical GMRES is shown in (e): the logo is reconstructed successfully although some ringing exists. It took 20 iterations of optical GMRES, i.e., 80 low dynamic range photos, to get this result. Displaying the image in (e) onto the scene with the right projector produces a left-camera photo that is indistinguishable from the one in (d). The difference between those two photos, amplified $50\times$, is shown in (f). This suggests \mathbf{T} is mildly singular, confirming the ringing.

4.1 Relation to Prior Work on Inverse Transport

Implicit in all prior work is the assumption that to invert light transport we must first acquire \mathbf{T} . Grossberg *et al.* [2004] consider this problem under the assumption of no indirect transport. This makes \mathbf{T} maximally sparse and full rank, and therefore easy to capture and invert. To deal with non-negligible (but still small) indirect illumination, Wetzstein and Bimber [2007] capture the full-resolution \mathbf{T} , enforce sparsity by thresholding, and invert it with a sparse-system solver. For more significant indirect lighting effects, Ng *et al.* [2009] and Bai *et al.* [2010] rely either on low-resolution or on sparse transport matrices to make brute-force acquisition tractable. In other applications, Zhang and Nayar [2006] compute \mathbf{T}^{-1} to account for projector defocus and Seitz *et al.* [2005] use it to decompose an image of a Lambertian scene into the contribution of individual light bounces. Both approaches rely on low-resolution transport matrices and brute-force acquisition.

Here we show that acquiring \mathbf{T} is not necessary as long as we pursue a more modest goal—rather than solve inverse light transport for every possible input photo, which is equivalent to computing

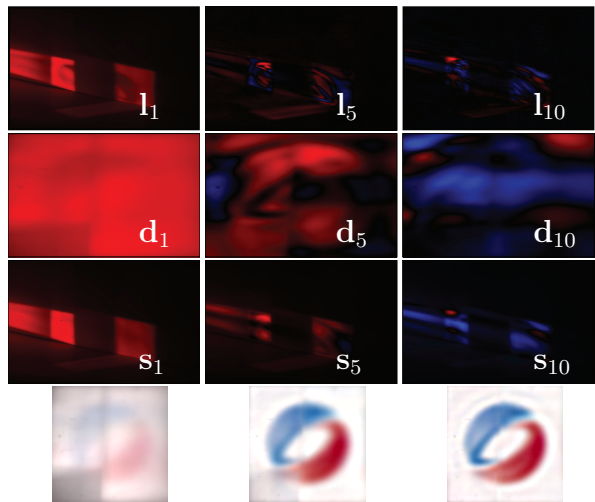


Figure 8: Optical GMRES in action for the inverse transport task of Figure 7. Although GMRES operates in full color, we show only the red channel with its positive (red) and negative (blue) components. Note that the algorithm begins by displaying the input photo in Figure 7(d) with the left projector. The last row shows the solution computable by GMRES after the corresponding iteration.

\mathbf{T}^{-1} and hard, we use optical GMRES to solve it for *specific* ones, which is much easier. Moreover, GMRES can be generalized to handle many input vectors/photos at a time [Gutknecht 2007].

5 Implementation

Cameras and projectors We use either one or two coaxial camera-projector pairs, each composed of an NEC LT30 DLP projector, a beamsplitter, and a Canon EOS 5D MarkII camera with a 50mm f1.2L lens, chosen for its light-gathering abilities. Since we capture RAW photos, we consider their response to be linear. The projectors are calibrated geometrically and radiometrically relative to the cameras. See Figures 7a and 11a for typical arrangements.

We operate each projector at 85Hz and capture photos with a 1/13sec exposure time at ISO100. The only exception is when taking photos of the display screen, which is typically quite dim, where we use ISO200 instead (e.g., for the left camera in Figure 11a).⁴ We capture only low dynamic range photos and use a fixed exposure level for each run of our algorithms. To control it, we adjust the lens aperture and the maximum intensity of projector pixels so that photos taken under maximum illumination have an intensity at most 80% of the maximum possible. Beyond arranging the cameras and projectors; adjusting exposure; choosing a photo resolution; and choosing the number K of Arnoldi iterations, there are no user-tunable parameters in our implementation. To minimize defocus in all experiments, we kept the scene within the camera’s depth of field as much as possible.

Optical Arnoldi Numerical packages for large matrices typically allow the matrix-vector calculation function to be passed as a parameter. To acquire \mathbf{T} , we simply pass a function that implements matrix-vector products with two illuminate-and-capture operations. For example, acquisition of a symmetric \mathbf{T} with the *PROPACK* package [Larsen] for *MATLAB* amounts to one line of code:

$$[\mathbf{V}, \mathbf{D}] = \text{LANEIG}(\text{opt-prod}, n, k, \text{"LM"})$$

⁴ To increase contrast even further for photos of the screen, one can block the aperture of the “inactive” projector (i.e., left projector in Figure 11a).

where $opt\text{-}prod()$ is a function that computes optical vector products with \mathbf{T} and $LANSVD()$ solves the eigenvalue problem⁵ $\mathbf{T}\mathbf{v} = \lambda\mathbf{v}$. The implementation for a non-symmetric \mathbf{T} is similar:

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = LANSVD(opt\text{-}prod, opt\text{-}prod\text{-}t, m, n, k, "L")$$

where $opt\text{-}prod\text{-}t()$ computes optical vector products with \mathbf{T}^t and $LANSVD()$ performs singular value decomposition. We do not actually use the output of these routines in our implementation; we simply store to disk all illumination vectors and all photos captured during the routines' execution. For relighting, we use Equation 7.

Optical matrix-vector products Our implementation essentially amounts to implementing the $opt\text{-}prod()$ and $opt\text{-}prod\text{-}t()$ functions for use in MATLAB. Given an illumination vector supplied by MATLAB, we reshape it into a 2D image and divide it by the maximum absolute pixel value to bring it to the range $[-1, 1]$. We then implement Equation 4 directly, by splitting the vector into its positive and negative components; multiplying each component by the maximum intensity of projector pixels; capturing one photo per displayed component; and taking the difference of the two photos. The differencing operation automatically eliminates contributions from ambient light and the projector's black level.

Since demosaicing large RAW photos is computationally expensive, we capture and store them on disk but do not return them as vectors to MATLAB. Instead, we return the 362×234 thumbnail images contained in the RAW files, which are linear, demosaiced, and small enough to fit in main memory. Steps 3 and 4 of optical Arnoldi and GMRES thus operate on a $(362 \times 234)^2$ version of \mathbf{T} .

High-resolution acquisition and relighting Even though the illumination vectors in Steps 3 and 4 of optical Arnoldi are computed from low-resolution thumbnails, the RAW photos we capture do enable high-resolution acquisition and relighting. For relighting in the non-symmetric case, we simply replace in Equation 7 the low-resolution basis for \mathbf{T} 's photo subspace with a high-resolution one:

$$\mathbf{p} = [\mathbf{d}_1^h \cdots \mathbf{d}_K^h][\mathbf{1}_1 \cdots \mathbf{1}_K]^t \mathbf{1} \quad (9)$$

where \mathbf{d}_k^h is the RAW photo captured in the k -th Arnoldi iteration. To reconstruct a high-resolution \mathbf{T} , we replace the matrix returned in Step 5 of the algorithm with the matrix in Equation 9.⁶

Color handling Mismatches in the spectral response of projectors and cameras can lead to cross-talk between color channels. To avoid such issues in the Arnoldi method, the scene is lit with monochrome illumination by converting each illumination vector to grayscale prior to projection. For GMRES, we capture each color channel with a separate illuminate-and-capture operation.

Timings Our entire relighting pipeline consists of running optical Arnoldi for K iterations, demosaicing the high-resolution photos it captured, and computing the relit images with Equation 9. We use an Apple iMac with a 2.8Ghz Intel Core i7 processor and 16 GB of memory for all capture, processing and display. Optical Arnoldi for a non-symmetric \mathbf{T} runs at a rate of 12 seconds per iteration, including all processing and photo acquisition. Demosaicing runs at a similar rate, about 12 seconds to process the positive and negative components of \mathbf{d}_k captured in each iteration. We use an image resolution of 1080×720 for all relighting results in the paper, with each image taking $0.3 \times K$ seconds to compute.

⁵Strictly speaking, $LANSVD()$ and $LANSVD()$ use the Lanczos method.

⁶While thumbnails make each Arnoldi iteration more efficient, computing illumination vectors directly from demosaiced RAW photos may lead to lower reconstruction error for a given number of iterations.

6 Results

6.1 Acquiring Transport Matrices with Optical Arnoldi

Ground-truth comparisons Although optical Arnoldi computes rank- K matrix approximations at the resolution of cameras and projectors, it is not possible to capture ground-truth matrices of that size for evaluation. To validate our algorithm and compare against other low-rank approximation methods, we acquired low-resolution transport matrices for the four scenes shown in Figure 9. We deliberately chose these scenes to exhibit a variety of challenging lighting effects, including shadows, interreflections, caustics, *etc.*

To capture the ground-truth matrix, we treated the left projector as a low resolution display with 32×32 or 16×16 superpixels. We used the full set of Hadamard illumination patterns because of their good noise properties [Schechner et al. 2007] and captured photos at thumbnail resolution (362×234).

The accuracy and efficiency of a particular acquisition technique can be measured by how well the ground-truth matrix can be reconstructed after a certain number of steps. We compute the relative reconstruction error under the Frobenius norm after k iterations:

$$\epsilon_k = \frac{\|\mathbf{T}_k - \hat{\mathbf{T}}\|_F}{\|\hat{\mathbf{T}}\|_F}, \quad (10)$$

where $\hat{\mathbf{T}}$ is the ground-truth matrix and \mathbf{T}_k is the transport matrix computable after k iterations. We report this error for \mathbf{T}_k computed five different ways:

- running k iterations of optical Arnoldi;
- acquiring k random rows and columns of the transport matrix and then using the Nyström method of Wang *et al.* [2009];
- applying the kernel Nyström method [Wang et al. 2009] to the same input, with its exponent parameter chosen to minimize ϵ_k (this represents the best-case scenario for that method);
- acquiring k photos under the discrete cosine transform (DCT) illumination basis (*i.e.*, a brute-force technique);
- computing the best rank- k approximation of $\hat{\mathbf{T}}$ numerically using singular value decomposition (this is optimal under the Frobenius norm).

To assess the relative performance of these methods in the absence of measurement noise, we simulated photo acquisition for the four scenes shown in Figure 9. Specifically, instead of capturing more photos as needed by each method, we used the ground-truth transport matrix to compute them.

The plots in Figure 9 show that optical Arnoldi's convergence rate under noiseless conditions is close to optimal, and significantly faster than the Nyström methods. Note that it should be possible, in principle, to reconstruct $\hat{\mathbf{T}}$ perfectly once the number of iterations becomes equal to the number of rows in $\hat{\mathbf{T}}$ (*i.e.*, 1024). We found, however, that this was not the case with Nyström because it depends on a pseudoinverse tolerance parameter that does not guarantee perfect reconstruction when $\hat{\mathbf{T}}$ is singular. This is the case in the *Waldorf* and *Bull* scenes, as can be seen by SVD's error profile. Finally, note that brute-force acquisition can outperform other methods in some cases (*e.g.*, *Juice*).

Figure 9 also shows comparisons under noisy conditions for the *Flower* and *Juice* scenes. In this case, all photos were captured rather than simulated. Since the single-source illuminations required by the Nyström methods (*i.e.*, only one superpixel is on) produce very dim images, we doubled the exposure time of the

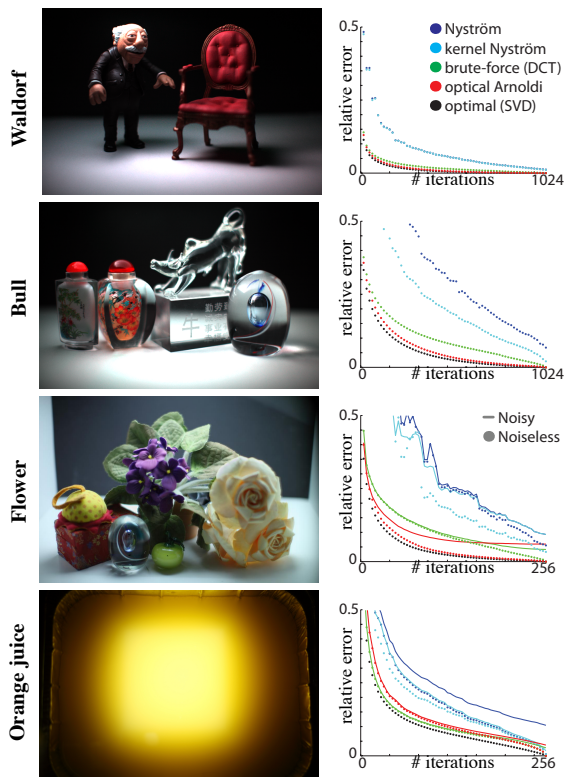


Figure 9: Error comparisons for five methods applied to scenes shown above. We used the non-symmetric arrangement in Figure 11(a) for the Waldorf, Bull and Flower scenes. Since the left projector was behind a diffuser, the transport matrix for these scenes was dense; the SVD error profiles suggest the rank is not very low for Bull and Flower because of mirror reflections from the glass objects in those scenes. For Juice, we used the symmetric arrangement in Figure 2(c), with light projected directly onto the scene. The transport matrix in this case was 256×256 and, as the SVD error profile indicates, has a high rank.

photos provided to Nyström. Thus, the total exposure time in Figure 9 is the same after k iterations for the Nyström methods and for optical Arnoldi (the latter captures twice as many photos per iteration).⁷ Despite the increase in the per-photo exposure time for the Nyström methods, their input images are still very noisy. Arnoldi, on the other hand, uses illumination vectors with much broader spatial support (e.g., see Figure 6) and therefore has some of the advantages of the Hadamard illumination patterns.

Since noise is present in all photos of *Flower* and *Juice*—including those we used for ground-truth capture—perfect reconstruction of $\hat{\mathbf{T}}$ cannot be guaranteed. Indeed, no method other than SVD, which depends only on $\hat{\mathbf{T}}$, achieves perfect reconstruction. As in the noiseless case, optical Arnoldi performs substantially better than the Nyström methods. Its convergence rate resembles that of SVD, although it converges to a non-zero reconstruction error. This is expected because it computes matrix-vector products with noisy 8-bit camera pixels rather than noiseless floating point arithmetic.

Relighting results and discussion

Figure 10 shows relighting

⁷We consider constant total exposure time a more accurate baseline, given the different illumination conditions required by the Nyström methods. See Figure 10 for a comparison with a constant number of photos.

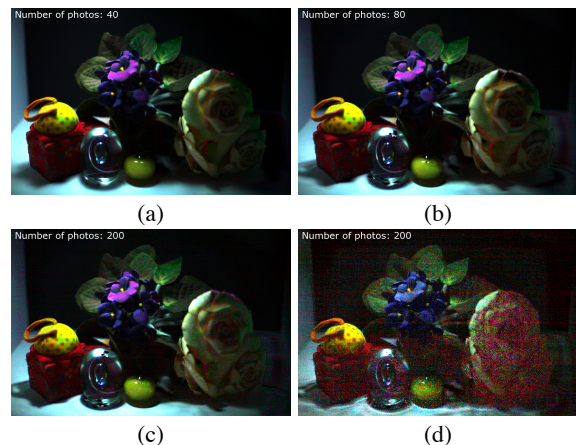


Figure 10: Relighting results for the Flower scene in Figure 9 under point-source illumination. We computed \mathbf{T} after (a) 10, (b) 20, and (c) 50 iterations of optical Arnoldi (i.e., 200 photos, 200/13 sec total exposure time). In (d) we show an image relit with the transport matrix computed by kernel Nyström from 100 row and 100 column samples (200 photos, 400/13 sec total exposure time).

ing results for a point light source, and Figures 1, 12 and the supplementary video show results for several more scenes. Unlike the ground-truth experiments, these scenes are relit with high-resolution, $(362 \times 234) \times (1080 \times 720)$ transport matrices.

These results suggest that it is possible to create good-quality relit images from a small number of captured photos, even in the presence of complex light transport phenomena. Clearly, since we acquire a rank- K approximation of the transport matrix from $4K$ captured photos, high-rank transport components cannot be acquired if K is very small. This is evident in the second row of the *Hat* scene in Figure 12, where a rank-10 approximation cannot reproduce the light passing through holes in the hat’s brim. These are reproduced much better with a rank-50 matrix. The *Crystal* scene in the same figure is more challenging, with numerous mirror reflections and light transported to the backdrop in complex ways. The result is a significant difference between the rank-10 and ground-truth photos.

An interesting property of optical Arnoldi is that even when the relit images are not entirely correct, as in the *Crystal* scene, they are still visually plausible. This is mainly due to the scene-dependent basis. All relit images are linear combinations of photos of the scene itself.

We deliberately did not use high dynamic range photos for experiments in this paper. This makes noise apparent in some of our photos (both real and rendered). Our purpose was to illustrate the performance of our optical-domain approach in a challenging setting. Cameras and projectors have limited dynamic range, are subject to noise, and represent numbers with limited precision. As such, optical matrix-vector products cannot be considered as accurate as matrix-vector products done with floating-point arithmetic. Nevertheless, we observe that despite these unavoidable issues, the presence of noise does not cause additional rendering artifacts. For example, the ground-truth photos in Figure 12, which were captured with the same exposure settings as Arnoldi’s, appear noisier than the rendered ones. The good behavior of optical Arnoldi comes from the overall stability of Krylov subspace methods in the presence of matrix-vector product errors [Simoncini and Szyld 2003].

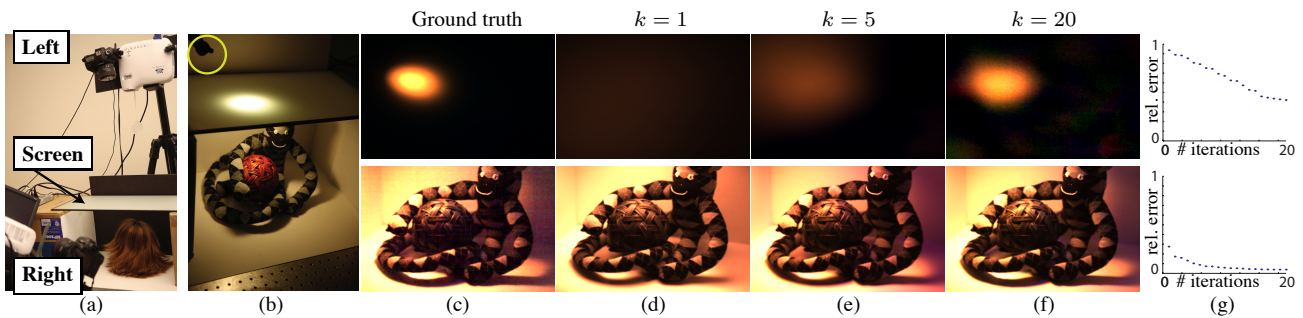


Figure 11: Simulating one light source with another. We used a non-symmetric arrangement with a display screen between the left camera and the scene (a). We then placed a toy on the other side (b) and used a flashlight (circled) to illuminate it. We captured a photo with the right camera (c, bottom) and gave it as input to optical GMRES. The goal was to get the left projector to illuminate the scene exactly like the flashlight. A left-camera photo of the display screen under flashlight illumination is shown in (c, top). This photo was not given to GMRES. It took GMRES 20 iterations to compute it (f, top). Note how the illumination computed by GMRES correctly replicates the shadows in the input photo, both hard and soft. (g) Relative L_2 reconstruction errors of the flashlight illumination (top) and the input photo (bottom).

6.2 Inverting Light Transport with Optical GMRES

The experiment in Figures 7 and 8 showed the ability of GMRES to invert light transport at high resolution for a high-rank \mathbf{T} . To test its ability to do the same for a low-rank \mathbf{T} , we consider the problem of creating an “illumination impostor” *i.e.*, using one light source (a projector) to simulate the complex lighting effects produced by another (a flashlight). This experiment is described in Figure 11.

The experiment shows that GMRES is able to invert light transport at high resolution in a very challenging setting, with complex occlusions, shadows and interreflections. This would not have been possible with existing methods without capturing the full \mathbf{T} first. Note that even though the illumination that GMRES estimates is similar to the one we used, it is not exactly the same. The photo of the scene it produces, however, matches that of the flashlight-illuminated scene very well. This means that \mathbf{T} has a whole subspace of feasible illuminations that produce the same photo—and GMRES just found one of them.

7 Relation to Work in Optical Computing

Numerical linear algebra with optics In discrete analog optical processing, matrices and vectors are encoded by transparency masks or spatial light modulators [Goodman 2005]. Athale and Collins [1982] showed how a sequential arrangement of such elements (along with light sources, lenses and detectors) can be used to implement matrix multiplication optically.

For more complex operations, the idea of using optical feedback loops for processing was well established [Cederquist and Lee 1979]. Kumar and Casasent [1981] describe several optical feedback loops for computing eigenvectors of a matrix, including power iteration. Rajbenbach *et al.* [1987] computed matrix inverses optically by implementing the Neumann series in a feedback loop. Our approach can be thought of as a re-interpretation of this general approach, where the matrix is encoded implicitly by the scene and where all other optical elements are replaced by cameras, projectors and a PC. In this respect, it is a hybrid system [Casasent 1981].

We refer readers interested in exploring this area to the review articles of Leith [2000] and Ambs [2009] as well as the critical analysis of Psaltis and Athale [1986] on the feasibility of using linear optics for numerical calculations.

Context-aware illumination Wang *et al.* [2010] recently used the coaxial arrangement in Figure 2(c) to enhance the appearance

of a real-world scene in real time. Their idea is to illuminate the scene with a processed version of the live video in which certain features, like edges, have been enhanced. Their approach cannot be considered a form of optical computation because it explicitly prevents feedback loops. No photo is captured when the scene is illuminated by their context-aware light source.

Perhaps closer in spirit to our work is the dynamic contrast enhancement technique of Amano and Kato [2008]. To enhance the contrast of a real-world planar scene under room illumination, they capture a photo, contrast-enhance it, and project it back onto the scene in a simple feedback loop. Contrary to our work, their analysis is done per pixel and assumes no indirect light transport.

8 Conclusions

In this paper we introduced the idea of performing numerical computations on the full-resolution transport matrix in the optical domain. We have shown that computations often assumed to require complete acquisition of the matrix—such as computing eigenvectors or inverting light transport—can actually be done efficiently without it. The numerical methods we use are widely available and can be readily moved to the optical domain; importantly, they bring with them a wealth of numerical research that can potentially be transferred to the optical domain as well.

Our efforts in this paper were specifically aimed at low-rank approximation and inverse light transport. Although we believe the low-rank constraint can be a powerful one, clearly many transport phenomena cannot be represented by low-rank matrices. The question of how to bridge the gap between the low-rank constraint we exploit and the sparsity constraints employed recently [Peers *et al.* 2009] is wide open. One advantage of optical GMRES, at least for inverse transport, is that it can operate regardless of matrix rank.

Finally, there is some evidence that individual transport eigenvectors provide information about shadows, interreflections, or even a scene’s 3D shape [Koenderink and van Doorn 1983; Langer 1999; Mahajan *et al.* 2007]. Since direct capture of eigenvectors is possible with our approach, we have begun to explore these connections.

Acknowledgements The authors thank the support of NSERC under the Discovery, GRAND NCE, RGPIN, and CGS-D programs.

References

AMANO, T., AND KATO, H. 2008. Real world dynamic appear-

- ance enhancement with procam feedback. *Proc. IEEE PRO-CAMS*.
- AMBS, P. 2009. A short history of optical computing: rise, decline, and evolution. *Proc. SPIE 7388*.
- ATHALE, R. A., AND COLLINS, W. C., 1982. Optical matrix-matrix multiplier based on outer product decomposition.
- BAI, J., CHANDRAKER, M., NG, T.-T., AND RAMAMOORTHI, R. 2010. A dual theory of inverse and forward light transport. *Proc. ECCV*.
- BASRI, R., AND JACOBS, D. 2003. Lambertian reflectance and linear subspaces. *IEEE T-PAMI 25*, 2, 218–233.
- CASASANT, D. 1981. Hybrid processors. *Optical Information Processing Fundamentals*, 181–233.
- CEDERQUIST, J., AND LEE, S. 1979. The use of feedback in optical information processing. *Applied Physics A: Materials Science & Processing*.
- DEBEVEC, P., HAWKINS, T., TCHOU, C., DUKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. *Proc. SIGGRAPH*.
- FUCHS, M., BLANZ, V., LENSCH, H., AND SEIDEL, H.-P. 2007. Adaptive sampling of reflectance fields. *ACM TOG 26*, 2 (Jun).
- GARG, G., TALVALA, E., LEVOY, M., AND LENSCH, H. 2006. Symmetric photography: Exploiting data-sparseness in reflectance fields. *Proc. Eurographics Symp. Rendering*.
- GOODMAN, J. W. 2005. Introduction to fourier optics, 3rd edition. *Roberts & Company Publishers*.
- GREENBAUM, A., AND TREFETHEN, L. N. 1994. Gmres/cr and arnoldi/lanczos as matrix approximation problems. *SIAM J. Sci. Comput.* 15, 2, 359–368.
- GROSSBERG, M., PERI, H., NAYAR, S., AND BELHUMEUR, P. 2004. Making one object look like another: controlling appearance using a projector-camera system. *Proc. CVPR*, 452–459.
- GUILFOYLE, P., AND STONE, R. 1991. Digital optical computer ii. *Proc. SPIE 1563*, 214–222.
- GUTKNECHT, M. H. 2007. Block krylov space methods for linear systems with multiple right-hand sides: an introduction. In *Modern Mathematical Models, Methods and Algorithms for Real World Systems*. 420–447.
- KOENDERINK, J. J., AND VAN DOORN, A. J. 1983. Geometrical modes as a general method to treat diffuse interreflections in radiometry. *J. Opt. Soc. Am* 73, 6, 843–850.
- KUMAR, B. V. K. V., AND CASASANT, D. 1981. Eigenvector determination by iterative optical methods. *Applied Optics 20*, 21, 3707–3710.
- LANGER, M. 1999. When shadows become interreflections. *Int. J. Computer Vision 34*, 2, 193–204.
- LARSEN, R. M. <http://soi.stanford.edu/rmunk/propack/>.
- LEITH, E. 2000. The evolution of information optics. *IEEE J. Select Topics in Quantum Electronics 6*, 6, 1297–1304.
- LIESEN, J., AND TICHY, P. 2004. Convergence analysis of krylov subspace methods. *GAMM Mitt. Ges. Angew. Math. Mech.* 27, 2, 153–173 (2005).
- MAHAJAN, D., SHLIZERMAN, I., RAMAMOORTHI, R., AND BELHUMEUR, P. 2007. A theory of locally low dimensional light transport. *Proc. SIGGRAPH*.
- MATUSIK, W., PFISTER, H., ZIEGLER, R., AND NGAN, A. 2002. Acquisition and rendering of transparent and refractive objects. *Proc. Eurographics Symp. on Rendering*.
- NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *Proc. SIGGRAPH*.
- NG, T., PAHWA, R., BAI, J., QUEK, T., AND TAN, K. 2009. Radiometric compensation using stratified inverses. *Proc. ICCV*.
- PEERS, P., AND DUTRÉ, P. 2003. Wavelet environment matting. *Proc. Eurographics Symp. on Rendering*.
- PEERS, P., AND DUTRÉ, P. 2005. Inferring reflectance functions from wavelet noise. *Proc. Eurographics Symp. Rendering*.
- PEERS, P., MAHAJAN, D., LAMOND, B., GHOSH, A., MATUSIK, W., RAMAMOORTHI, R., AND DEBEVEC, P. 2009. Compressive light transport sensing. *ACM TOG 28*, 1.
- PSALTIS, D., AND ATHALE, R. A. 1986. High accuracy computation with linear analog optical systems: a critical study. *Applied Optics 25*, 18, 3071–3077.
- RAJBENBACH, H., FAINMAN, Y., AND LEE, S. H. 1987. Optical implementation of an iterative algorithm for matrix-inversion. *Applied Optics 26*, 6, 1024–1031.
- SAAD, Y. 2003. *Iterative methods for sparse linear systems*.
- SALVI, J., PAGES, J., AND BATLLE, J. 2004. Pattern codification strategies in structured light systems. *Pattern Recogn 37*, 4, 827–849.
- SCHECHNER, Y., NAYAR, S., AND BELHUMEUR, P. 2007. Multiplexing for optimal lighting. *IEEE T-PAMI 29*, 8, 1339–1354.
- SEITZ, S., MATSUSHITA, Y., AND KUTULAKOS, K. 2005. A theory of inverse light transport. *Proc. ICCV*, 1440–1447.
- SEN, P., AND DARABI, S. 2009. Compressive dual photography. *Proc. Eurographics*.
- SEN, P., CHEN, B., GARG, G., MARSCHNER, S., HOROWITZ, M., LEVOY, M., AND LENSCH, H. 2005. Dual photography. *Proc. SIGGRAPH*.
- SIMON, H. D., AND ZHA, H. 2000. Low-rank matrix approximation using the lanczos bidiagonalization process with applications. *SIAM J. Sci. Comput.* 21, 6, 2257–2274.
- SIMONCINI, V., AND SZYLD, D. B. 2003. Theory of inexact krylov subspace methods and applications to scientific computing. *SIAM J. Sci. Comput.* 25, 2, 454–477.
- TREFETHEN, L. N., AND BAU, I. 1997. Numerical linear algebra. *SIAM*, xii+361.
- WANG, J., DONG, Y., TONG, X., LIN, Z., AND GUO, B. 2009. Kernel nystrom method for light transport. *Proc. SIGGRAPH*.
- WANG, O., FUCHS, M., FUCHS, C., DAVIS, J., SEIDEL, H.-P., AND LENSCH, H. P. A. 2010. A context-aware light source. *Proc. ICCP*.
- WETZSTEIN, G., AND BIMBER, O. 2007. Radiometric compensation through inverse light transport. *Pacific Graphics*, 391–399.
- ZHANG, L., AND NAYAR, S. 2006. Projection defocus analysis for scene capture and image display. *Proc. SIGGRAPH*.

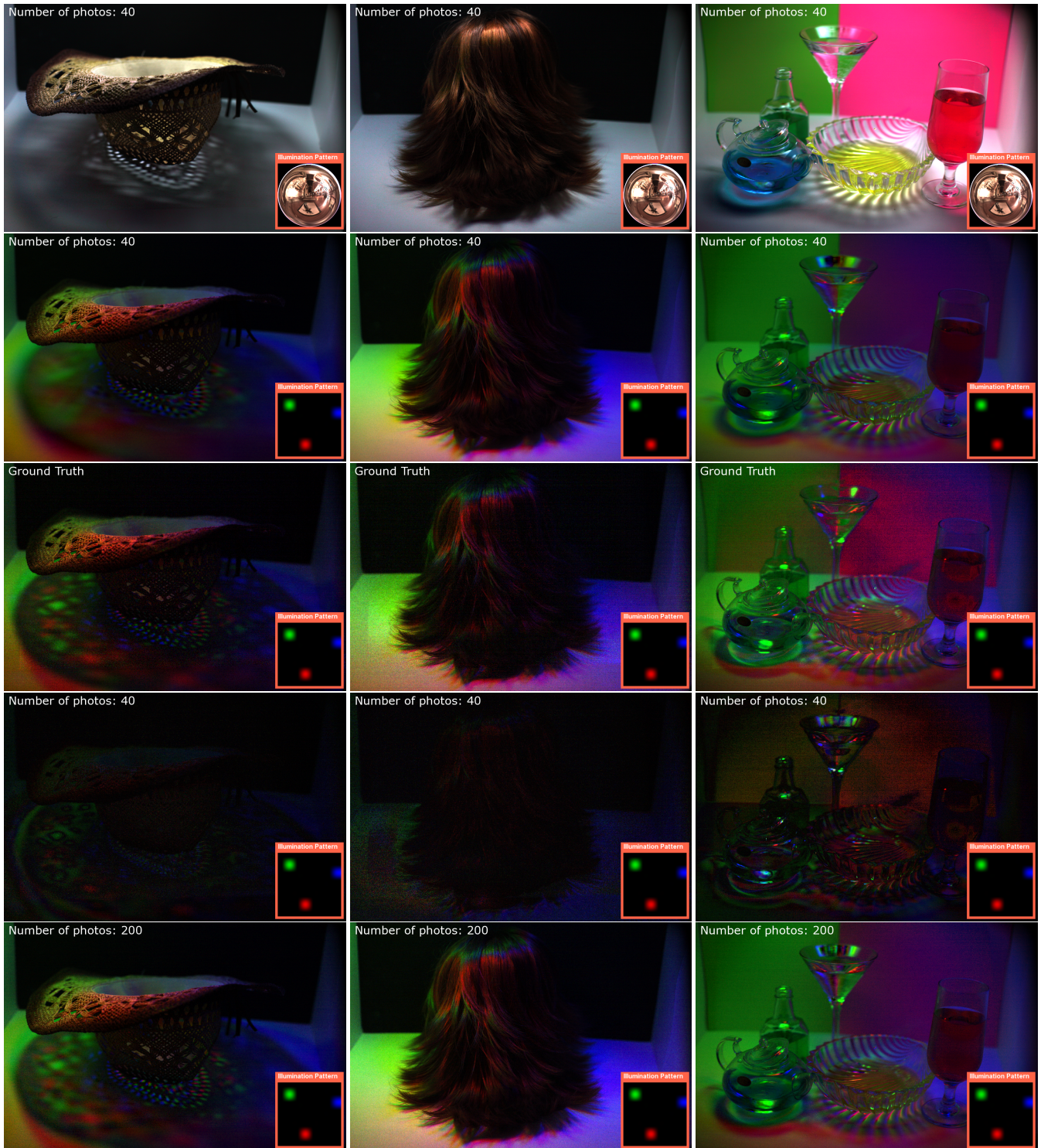


Figure 12: *First row:* Relighting results under complex illumination with a rank-10 transport matrix. *Second row:* We use spatially-localized lighting and a rank-10 matrix to identify inaccuracies in high-rank light transport. *Third row:* Actual photo of the scene captured under the same illumination with identical exposure settings. *Fourth row:* Difference between actual photo and the relit image above it. *Fifth row:* Relighting with a rank-50 matrix. Please refer to the supplementary video for more results and comparisons.