A Signal-processing Framework for Image Relighting

Ha Q. Nguyen

***Abstract*—This report reviews, in light of signal processing, the problem of relighting a *Lambertian* convex object with distant light source, whose crucial task is the decomposition of reflectance function into *albedos* (reflection coefficients) and lighting, based on a set of images and the 3-D geometry of the object from which the images were taken. A reflectance function is the result of filtering a lighting with the *half-cosine* kernel through a *spherical convolution*, that defines a linear *rotation-invariant* system, an extension of LTI systems in classical signal processing. This important observation maps the decomposition of reflectance function to a deconvolution, which can be facilitated in frequency-domain using *spherical harmonics*, an analogue of Fourier basis. As the half-cosine kernel is highly compacted in frequency-domain, reflectance functions are well approximated by a low-dimensional linear subspace spanned by the first few spherical harmonics. Therefore, the deconvolution problem can be matricized into a simple-looking matrix factorization problem. The formulation of relighting problem as a matrix factorization is carefully rederived along with discussions about spherical convolutions and spherical harmonics. Early theoretical results from the author's previous work on solving the matrix factorization problem are also briefly reviewed. Experiments are done on synthetic data to demonstrate the use of these results.**

***Index Terms*—relighting, Lambertian surfaces, forward rendering, inverse rendering, reflectance function, spherical convolution, linear rotation-invariant systems, spherical harmonics, matrix factorization,**

I. Introduction

A. Motivation and Literature Review

Image relighting is one of the core problems in *photorealism*, the field of creating, or technically, *rendering* images that look indistinguishably from realistic ones. In relighting problems, we are given a set of images of the same scene under various lighting conditions, and the geometry of the scene as well. Our task is to synthesize another image as if it was obtained by shining the scene with a novel lighting.

Images of an object can be viewed as a *reflectance function* that quantifies the amount of light reflected to the camera from each point on the surface of the object. In general, reflectance function is the result of the interaction between lighting, reflection characteristic of the object (often described by the BRDF - *Bidirectional Reflectance Distribution Function*) and texture of the object. Consequently, the relighting problem actually consists of two sub-problems: *inverse rendering* and *forward rendering*. In the inverse rendering phase, the reflectance function is decomposed into lighting, BRDF and texture; and in forward rendering phase, the recovered BRDF and texture, together with a novel lighting are combined into a novel reflectance function, resulting in a novel image. Fig. 1 illustrates the inputs and outputs of forward and inverse rendering algorithms. Both forward and inverse problems have been researched extensively in the past few decades with a large body of work (see [1], [2] for detailed discussions). Most previous techniques (see, for example, [3]) can deal only with highly *controlled lighting* conditions in which a single point source is usually actively positioned. These methods certainly cannot work in outdoor conditions when the lighting can be arbitrarily complex, coming from various sources of continuous distributions such as the skylight.

The difficulties of rendering under general, or *uncontrolled lighting* are due to the lack of a discretized framework that can efficiently describe reflectance function which has been previously interpreted as an integral. Ramamoorthi and Hanrahan in their series of work [4]–[6] introduced a breakthrough signal-processing framework for both forward and inverse rendering. In this framework, reflectance function is treated simply as a *spherical convolution* (sphere counterpart of circular convolution) of a lighting with the *reflection kernel* (BRDF multiplied by a *half-cosine*). This allows us to relate reflectance function to lighting and reflection kernel in terms of their *spherical harmonic* expansions (sphere counterpart of Fourier series). Transforming from space-domain to frequency-domain yields two great advantages: (1) integrals are mapped to products of coefficients; and (2) reflectance function can be well-approximated by a few low-frequency terms (because reflection kernel often varies slowly.) Based on this framework, Ramamoorthi and Hanrahan developed efficient algorithms for both forward and inverse rendering. However, they only considered homogeneous objects with no texture, and thus ignored the local scalings of the reflection.

Basri and Jacobs in an independent work [7] discovered a similar result for the special case of Lambertian surfaces whose BRDF is a constant. Here, reflectance function is the spherical convolution of a lighting with the half-cosine kernel, then scaled by *albedos* (reflection coefficients that characterize the surface texture). As a sequel, it has been shown in [7] that any reflectance function (of a Lambertian convex object with distant light source[1]) can be well-approximated by its first nine spherical harmonic coefficients. In other words, reflectance functions live close to a 9-dimensional linear subspace spanned by first 9 spherical harmonics. Basri *et. al.* in the subsequent work [8] matricized this important observation and mapped the forward rendering to a matrix multiplication, and inverse rendering to a matrix factorization. In this formulation, the main task of relighting problem is to factorize the image matrix formed by known images into a product of a diagonal albedo

[1]We will elaborate on these assumptions later on.

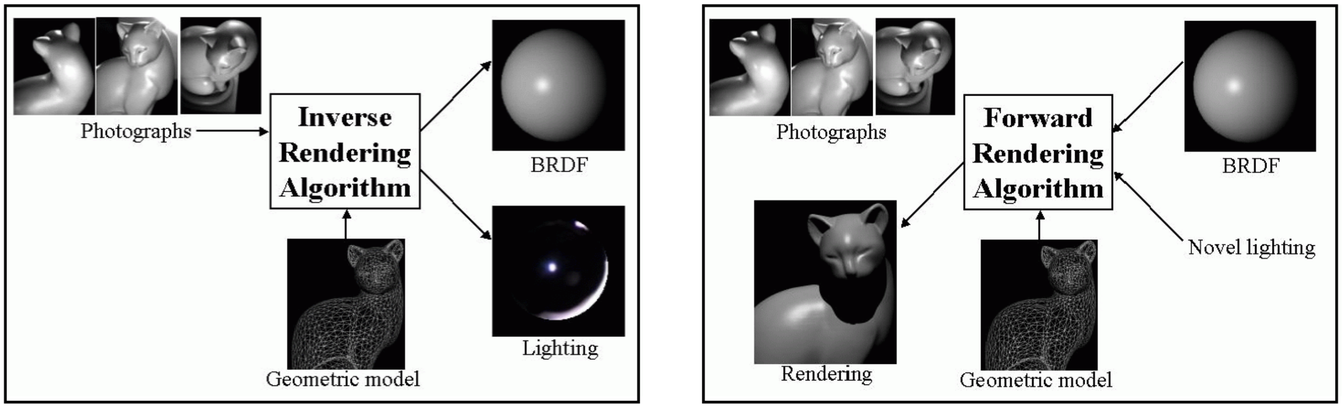


Fig. 1. (Reproduced from [1, Fig. 1.1]) Illustration of the two phases in image relighting.

matrix, the known spherical harmonic matrix, and a lighting matrix. A heuristic SVD-based algorithm was proposed in [8] to solve this factorization problem by simply picking the matrix of right-singular vectors scaled by square roots of singular values of the image matrix as a lighting matrix. However, no rigorous justification for the method has been done and the well-posedness of the factorization has not been addressed.

The forthcoming paper [9] is an effort to fix the shortcomings of [8] for rigorous solving of the matrix factorization problem. By means of subspace methods, it can give answers to the two main questions: (1) when does the factorization have unique solution; and (2) if unique, what is the exact solution of the factorization.

B. Organization and Contributions

The remainder of this report is organized as follows. Sec. II reviews some basic background of reflection equation and spherical harmonic expansion. Sec. III formulates the inverse rendering as a matrix factorization and discusses the well-posedness of the problem as well as factorization algorithms based on vector space methods. Sec. IV presents some numerical experiments on the proposed algorithms. Sec. V draws some concluding remarks and suggests a list of potential future work.

The contributions of the report are

- review of [5], [7] and [9];
- providing algebraically abstract view of spherical convolutions;
- proof for Proposition 1 (stated without proof in [5]);
- implementations of the methods in [9].

II. Background

This section provides a framework to relate reflectance function to lighting and reflection kernel in space-domain as a spherical convolution. The relation is then transformed into frequency-domain via spherical harmonic expansions. The material is adapted from [1] and [7].

A. Reflection as Convolution

Throughout this report, we impose the following commonly used assumptions in interactive graphics and computer vision.

A1 *Curved Surfaces:* characterized by surface normals.
A2 *Lambertian:* same reflection for every viewing angle.
A3 *Convex Objects:* no shadowing or inter-reflection.
A4 *Distant Illumination:* same lighting function everywhere.

We start with the 2-D case to illustrate the key concepts and ideas and then generalize the formula for the 3-D case without taking care of technical details.

1) 2-D Case: Under assumptions A2 and A3, the reflected radiance depends only on a single location on the surface. In particular, consider a point $\boldsymbol{p}$ on the surface with corresponding surface normal $\boldsymbol{n_p} = [1, \alpha]$ in some *global* polar coordinates system ($\boldsymbol{n_p}$ always exists by assumption A1). If there is only one distant light source whose direction is given by the unit vector $[1, \theta]$, the reflection equation is given by the Lambert's cosine law

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) L(\theta) \cos\theta', \qquad (1)$$

where $B(\boldsymbol{p})$ is the *reflectance function* (reflected radiance at location $\boldsymbol{p}$); θ' is the *incident angle*; $\rho(\boldsymbol{p})$ is the *albedo* at point $\boldsymbol{p}$ (a constant independent of θ'); and $L(\theta)$ is the *lighting function* (illumination at angle θ that is independent of location $\boldsymbol{p}$ by assumption A4). It is important to emphasize that θ' is the *local* angular coordinate, i.e., the angle between the lighting vector and the (changing) surface normal $\boldsymbol{n_p}$, whereas θ is the global angular coordinate (see Fig. 2 for an illustration.)

Now consider the general case when the distant lighting comes with continuous incident angles θ' varying from $-\pi/2$ to $\pi/2$. The illumination L also varies with θ, and so the reflection equation must be written in integral form

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) \int_{-\pi/2}^{\pi/2} L(\theta) \cos\theta' \, d\theta' \qquad (2)$$

$$= \rho(\boldsymbol{p}) \int_{-\pi}^{\pi} L(\theta) K(\theta') \, d\theta', \qquad (3)$$

where the *Lambertian kernel* $K(\theta')$ is defined as the *half-cosine* function: $K(\theta') = \max(\cos\theta', 0)$. Since the global and

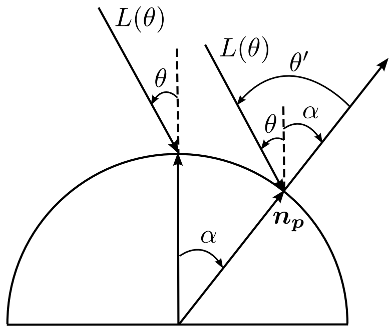


Fig. 2. (Modified from [1, Fig. 2.1]) Global and local coordinates for 2-D case. By nature, lightings are given in global coordinates, whereas incident angles are in local coordinates with respect to the surface normal $\boldsymbol{n_p}$.

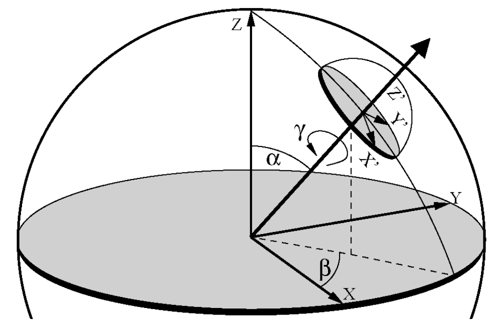


Fig. 3. (Reproduced from [1, Fig. 2.2]) Global and local coordinates for 3-D case. The angular γ is used for anisotropic case and is irrelevant in the scope of this report.

local angular coordinates are related by a circular shift: $\theta = \alpha + \theta'$, (3) can be expressed as a *circular convolution*

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) \int_{-\pi}^{\pi} L(\theta) K(\theta - \alpha)\, d\theta \tag{4}$$

$$= \rho(\boldsymbol{p}) (L \circledast K)(\alpha). \tag{5}$$

The circular convolution is one of the main topics in classical signal processing [10], as it characterizes a linear time-invariant (LTI) system with finite (or periodic) impulse response. This kind of filtering can be mapped to a multiplication of Fourier series coefficients in frequency-domain. The underlying reason for this efficient mapping is that sinusoids form an orthonormal basis of eigensignals for LTI systems.

2) 3-D Case: Suppose now the surface normal is given by $\boldsymbol{n_p} = [1, \alpha, \beta]$ in some *global spherical coordinates* (see Fig. 3). Similarly to (3), the reflection equation in 3-D is given by

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) \int_{0}^{\pi} \int_{0}^{2\pi} L(\theta, \varphi) K(\theta') \sin\theta'\, d\theta'\, d\varphi', \tag{6}$$

where the integral is now taken over the surface of the unit sphere. The global and local angular coordinates are now related by a 3-D rotation (or we can say, a *spherical shift*). This relation is specified by the following result, that was stated in [5] without a proof.

Proposition 1: Let $R_{\alpha,\beta} = R_z(\beta) R_y(\alpha)$, where $R_y(\alpha)$ and $R_z(\beta)$ are respectively the matrices of rotations about y-axis by α, and about z-axis by β. Then we have

$$[1, \theta, \varphi]_c^T = R_{\alpha,\beta} \cdot [1, \theta', \varphi']_c^T, \tag{7}$$

where the subscript c is added to a vector in spherical coordinates to denote its corresponding Cartesian coordinates.

Proof: See Appendix A. ■

With the aid of Proposition 1, we obtain a 3-D analogue of (4)

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) \int_{0}^{\pi} \int_{0}^{2\pi} L(\theta, \varphi) K(R_{\alpha,\beta}^{-1}(\theta, \varphi)) \sin\theta\, d\theta\, d\varphi, \tag{8}$$

with the understanding that $R_{\alpha,\beta}^{-1}(\theta, \varphi)$ are angular coordinates of the rotation $R_{\alpha,\beta}^{-1}$ acted on the unit vector $[1, \theta, \varphi]$. The double integral of (8), in the same manner as 2-D, can be viewed as a *spherical convolution* in which we integrate with respect to 3-D rotations (or spherical shifts). Letting $\circledast_s$ denote the spherical convolution, we can write (8) in a more compact form

$$B(\boldsymbol{p}) = \rho(\boldsymbol{p}) (L \circledast_s K)(\alpha, \beta). \tag{9}$$

This spherical convolution naturally defines a *linear rotation-invariant* (LRI) system with impulse response $K(\theta)$. From an abstract viewpoint, this is a generalized notion in which linear systems are invariant to actions of some algebraic group. Particularly, it is the groups of time-shifts for LTI and 3-D rotations for LRI systems.

B. Spherical Harmonics as Basis

Now we want to look at the convolution (9) in frequency-domain because the energy of the Lambertian kernel K is compressed in low frequencies. As a sphere counterpart of Fourier basis, the *spherical harmonics* $\{S_{m,n}\}_{m \geq 0, |n| \leq m}$ form an orthonormal basis for functions on the unit sphere

$$S_{m,n}(\theta, \varphi) = N_{m,n} \cdot P_{m,n}(\cos\theta)\, e^{jn\varphi}, \tag{10}$$

where $j = \sqrt{-1}$, $N_{m,n}$ is the normalized term, and $P_{m,n}(\cdot)$ is the *associated Legendre function*.[2] The first nine spherical harmonics are plotted in Fig. 4 for illustration.

Every function $F(\theta, \varphi)$ on the unit sphere has a spherical harmonic expansion

$$F(\theta, \varphi) = \sum_{m=0}^{\infty} \sum_{n=-m}^{m} f_{m,n} S_{m,n}(\theta, \varphi), \tag{11}$$

where the harmonic coefficients $f_{m,n}$ are given by

$$f_{m,n} = \int_{0}^{\pi} \int_{0}^{2\pi} F(\theta, \varphi) S_{m,n}^*(\theta, \varphi) \sin\theta\, d\theta\, d\varphi. \tag{12}$$

The feature that really makes spherical harmonics similar to Fourier basis on the circle is that they are eigensignals of LRI systems. This is stated by a special case of the celebrated Funk-Hecke theorem [7].

[2] See [7] for specific formulae of $P_{m,n}(\cdot)$ and $N_{m,n}$.

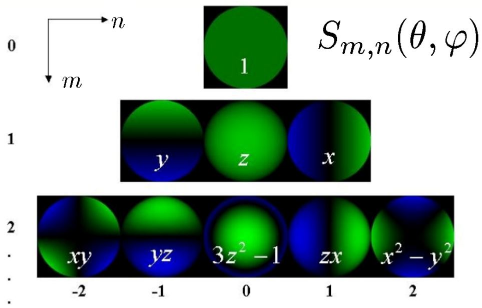


Fig. 4. (Reproduced from [1, Fig. 2.3]) Plot of the first nine spherical harmonics. On top are the corresponding formulae (ignoring normalized terms) of the harmonics in terms of Cartesian coordinates, i.e. $[x, y, z] = [\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta]$. Green color denotes positive values whereas blue color denotes negative ones.

Theorem 1 (Funk-Hecke): Let $K(\theta)$ be a bounded integrable function on $[0, \pi]$, and $S_{m,n}(\theta, \varphi)$ is a spherical harmonic, then

$$S_{m,n} \circledast_s K = \sqrt{\frac{4\pi}{2m+1}} k_m S_{m,n}, \tag{13}$$

where $\{k_m\}_{m\geq 0}$ are spherical harmonic coefficients of $K(\theta)$.

Back to our goal of transforming (9) into frequency-domain, let the harmonic coefficients of $L(\theta, \varphi)$, $K(\theta)$ and $L \circledast_s K$ be $\{\ell_{m,n}\}_{m\geq 0, |n|\leq m}$, $\{k_m\}_{m\geq 0}$, and $\{b_{m,n}\}_{m\geq 0, |n|\leq m}$, respectively. It is an immediate consequence of Thm. 1 that

$$b_{m,n} = \sqrt{\frac{4\pi}{2m+1}} k_m \ell_{m,n}. \tag{14}$$

It was also shown in [7] that the kernel $K(\theta) = \max(\cos\theta, 0)$ can be well-approximated using first M' terms in the spherical harmonic expansion, for small M' (for instance, using $M' = 3$ preserves roughly 99% the energy of $K(\theta)$.) As a result, the reflectance function can be approximated by

$$\begin{aligned} B(\boldsymbol{p}) &\approx \rho(\boldsymbol{p}) \sum_{m=0}^{M'-1} \sum_{n=-m}^{m} b_{m,n} S_{m,n}(\alpha, \beta) \\ &= \rho(\boldsymbol{p}) \sum_{m=1}^{M} b_m S_m(\alpha, \beta), \end{aligned} \tag{15}$$

where $M = M'^2$, and in (15) double index was converted to single index using the relation $(m, n) \leftrightarrow m^2 + m + n + 1$.

III. Algorithms and Derivations

A. Problem Formulation

Suppose we are given K images (this is a little abuse of notation since K was before used for the kernel, but from now on there is no need to worry about it.) These images are taken from the same viewpoint of the same object under different lighting conditions. Let N be the number of pixels in each image. From (15), the intensity at pixel i of image j is

$$B_j(\boldsymbol{p}_i) \approx \rho(\boldsymbol{p}_i) \sum_{m=1}^{M} b_m^j S_m(\alpha_i, \beta_i), \tag{16}$$

for $1 \leq i \leq N$, $1 \leq j \leq K$, and $\boldsymbol{n}_{\boldsymbol{p}_i} = [1, \alpha_i, \beta_i]$.

We can put all equations given in (16) in a single matrix form by defining matrices $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$; $\boldsymbol{\Phi} \in \mathbb{D}^N$(diagonal matrix of size N); $\boldsymbol{S} \in \mathbb{R}^{N\times M}$; and $\boldsymbol{X} \in \mathbb{R}^{M\times K}$ as follows

$$\begin{aligned} \boldsymbol{Y}_{ij} &= B_j(\boldsymbol{p}_i); \qquad & \boldsymbol{\Phi}_{ii} &= \rho(\boldsymbol{p}_i); \\ \boldsymbol{S}_{im} &= S_m(\alpha_i, \beta_i); \qquad & \boldsymbol{X}_{mj} &= b_m^j, \end{aligned} \tag{17}$$

for $1 \leq i \leq N, 1 \leq j \leq K, 1 \leq m \leq M$, and $\boldsymbol{M}_{ij}$ denotes the entry of matrix $\boldsymbol{M}$ at row i and column j. With these notations, (16) becomes

$$\boldsymbol{Y} \approx \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{X}, \tag{18}$$

where $\boldsymbol{Y}, \boldsymbol{\Phi}, \boldsymbol{S}$ and $\boldsymbol{X}$ are called image, albedo, spherical harmonic and lighting matrices, respectively.

For analytical purpose, we first assume the noiseless case when the approximation in (18) is replaced with exact equation

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{X}, \tag{19}$$

In light of Eq. (19), the inverse rendering problem becomes a matrix factorization in which the image matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$ and spherical harmonic matrix $\boldsymbol{S} \in \mathbb{R}^{N\times M}$ are known (can be computed from given images and 3-D model of the object); the albedo matrix $\boldsymbol{\Phi} \in \mathbb{D}^N$ and lighting matrix $\boldsymbol{X} \in \mathbb{R}^{M\times K}$ are to be recovered. Once the albedo and lighting matrices are reconstructed, say $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{X}}$, the forward rendering becomes an obvious matrix multiplication

$$\boldsymbol{Y}_{\text{new}} = \hat{\boldsymbol{\Phi}}\boldsymbol{S}\boldsymbol{X}_{\text{new}},$$

where $\boldsymbol{Y}_{\text{new}}$ corresponds to novel images under novel lightings associated with matrix $\boldsymbol{X}_{\text{new}}$. As a sequel, the rest of this report only focuses on the inverse rendering, or matrix factorization problem. Before tackling the problem, we need a few further assumptions.

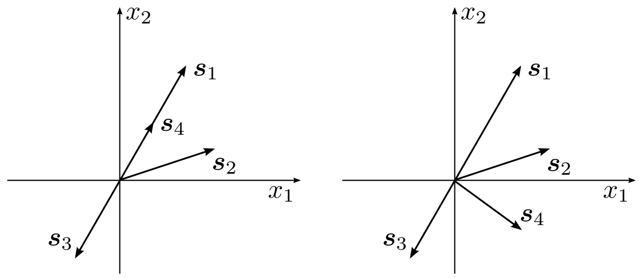
- A5 $K \geq M$: the number of images is greater than the number of spherical harmonics used in approximation.
- A6 Eq. (19) always has a solution $(\boldsymbol{\Phi}, \boldsymbol{X}) \in \mathbb{D}_*^N \times \mathbb{R}_{\text{full}}^{M\times K}$.

In assumption A6, $\mathbb{D}_*^N$ denotes the set of all $N\times N$ diagonal matrices with nonzero entries on the diagonal; and $R_{\text{full}}^{M\times K}$ denotes the set of all full-rank $M \times K$ matrices. Assuming that $\boldsymbol{\Phi} \in \mathbb{D}_*^N$, or all albedos are nonzero, does not restrict ourselves because if there are pixels corresponding to zero albedo, we can mask them out of the equations. Also, the assumption that $\boldsymbol{X}$ has full rank is reasonable and often made in array signal processing [11, Chapter 3]. It essentially says that the images are taken under diversified lighting conditions.

B. Well-posedness of the Factorization

The very first question one always asks when dealing with an inverse problem is when the recovery is unique. This subsection provides a necessary and sufficient condition on the spherical harmonic matrix $\boldsymbol{S}$ such that the matrix factorization (19) is unique up to some scaling factor. It is certainly the best we can hope for since if $(\boldsymbol{\Phi}, \boldsymbol{X})$ is a solution to (19) then so is $\left(\alpha\boldsymbol{\Phi}, \frac{1}{\alpha}\boldsymbol{X}\right)$, for any scalar $\alpha \neq 0$. The condition can be stated neatly by Thm. 2, with an introduction to a new notion of matrix full rank.

Definition 1: A tall matrix $\boldsymbol{S} \in \mathbb{R}_{\text{full}}^{N\times M} (N > M)$ with no zero rows is said to have *nonseparable full rank* if there do not



(a) Regular full rank (b) Nonseparable full rank

Fig. 5. Illustration of the two different notions of full rank for $M = 2, N = 4$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4]^T$.

exist nonempty disjoint sets $\mathcal{N}_1$ and $\mathcal{N}_2$ such that $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$ and $\text{rank}(\boldsymbol{S}_{\mathcal{N}_1}) + \text{rank}(\boldsymbol{S}_{\mathcal{N}_2}) = M$.

Here, $\boldsymbol{S}_{\mathcal{J}}$ denotes the submatrix of $\boldsymbol{S}$ with rows indexed by a subset $\mathcal{J}$ of $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$. In words, a matrix has nonseparable full rank if it has full column rank and we can not separate its rows into two groups with ranks add up to the rank of the matrix. That justifies the term "nonseparable full rank." Fig. 5 illustrates the notion of nonseparable full rank in comparison with regular full rank.

Theorem 2: Equation (19) has no solutions in $\mathbb{D}^N \times \mathbb{R}^{M \times K}$ other than $\left(\alpha\boldsymbol{\Phi}, \frac{1}{\alpha}\boldsymbol{X}\right)$, for some scalar $\alpha \neq 0$, if and only if $\boldsymbol{S}$ has nonseparable full rank.

Proof: See [9]. ∎

C. Factorization Algorithms

This subsection is to answer the second question of how to solve the matrix factorization problem given that it has unique solution, or $\boldsymbol{S}$ has nonseparable full rank by Thm 2. Note that since $\boldsymbol{S}$ has full column rank, once $\boldsymbol{\Phi}$ is known, $\boldsymbol{X}$ can be uniquely recovered by

$$\boldsymbol{X} = \boldsymbol{S}^\dagger(\boldsymbol{\Phi}^{-1}\boldsymbol{Y}), \tag{20}$$

where $\boldsymbol{S}^\dagger = (\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T$ is the psedo-inverse of $\boldsymbol{S}$. Therefore, we can focus on finding $\boldsymbol{\Phi}$ which has a diagonal structure. In the noiseless case, an SVD-based algorithm is proposed to exactly recover $\boldsymbol{\Phi}$. In the noisy case, it is modified into an optimization-based algorithm to find an estimate of $\boldsymbol{\Phi}$.

1) SVD-based Algorithm: The algorithm is based on the following result.

Theorem 3: $\boldsymbol{\Phi} = \text{diag}(\boldsymbol{\varphi})$ is a solution to Equation (19) if and only if $\boldsymbol{z} = \boldsymbol{\varphi}\odot^{-1}$ is a nontrivial solution to

$$\left((\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot (\boldsymbol{Y}\boldsymbol{Y}^T)\right)\boldsymbol{z} = \boldsymbol{0}, \tag{21}$$

where $\odot$ denotes element-wise operators.

Proof: See [9]. ∎

Interestingly, the matrix $(\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger)$ is a reminiscence of the projected gradient descent method used in Frost beamformer [12]. It is nothing but a projection on to the null space of $\boldsymbol{S}^T$ [11], and is used to force a vector to be in the range space of $\boldsymbol{S}$. Thm. 3 naturally gives rise to Algorithm 1 in which we solve (21) for $\boldsymbol{z}$ by picking the eigenvector associated with the smallest eigenvalue of the positive definite matrix $(\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot (\boldsymbol{Y}\boldsymbol{Y}^T)$.

Algorithm 1 SVD-based Recovery of $\boldsymbol{\Phi}$

Inputs: $\boldsymbol{Y}, \boldsymbol{S}$
Output: Estimate $\hat{\boldsymbol{\Phi}}$
1. Let $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot (\boldsymbol{Y}\boldsymbol{Y}^T)$.
2. Compute the eigenvalue decomposition of $\boldsymbol{M}$.
3. Let $\boldsymbol{z}^*$ be the eigenvector associated with the smallest eigenvalue of $\boldsymbol{M}$.
4. Return $\hat{\boldsymbol{\Phi}} = \text{diag}((z_1^*)^{-1}, \ldots, (z_N^*)^{-1})$.

The following corollary is very useful in checking whether a matrix $\boldsymbol{S}$ has nonseparable full rank. It can be deduced directly from Thm. 2 and Thm. 3 by setting $\boldsymbol{\Phi} = \boldsymbol{I}_N$ and $\boldsymbol{X} = \boldsymbol{I}_M$.

Corollary 1: A matrix $\boldsymbol{S} \in \mathbb{R}^{N \times M}$ has nonseparable full rank if and only if

$$\text{rank}\left((\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot (\boldsymbol{S}\boldsymbol{S}^T)\right) = N - 1.$$

We conclude this subsection by giving a few comments on this result. Without identifying the nonseparable full rank with the uniqueness of the corresponding matrix factorization, we can hardly see the connection between Definition 1 and Corollary 1. When fixing M and growing N, checking if an $N \times M$ matrix has nonseparable full rank using the brute-force approach (i.e., computing the rank of every submatrix) would be exponentially complex. However, Corollary 1 provides a much more efficient indirect way to do so, in which only the rank of an $N \times N$ matrix needs to be computed, resulting in a polynomial complexity.

2) Optimization-based Algorithm: Since $\boldsymbol{\Phi}$ is an albedo matrix, its diagonal entries must be all positive. It follows that the vector $\boldsymbol{z} = (\varphi_1^{-1}, \ldots, \varphi_N^{-1})$ must be positive as well. If the proposed model is perfect then $\boldsymbol{\Phi}$ (and therefore $\boldsymbol{z}$) is unique (provided that $\boldsymbol{S}$ has nonseparable full rank), and we do not need to worry about the positivity of $\boldsymbol{z}$. However, it can never be the case due to approximation error, model mismatch, measurement error, etc. Equation (19) should therefore be modified as

$$\boldsymbol{Y} + \boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{X}, \tag{22}$$

where $\boldsymbol{W}$ is a white noise with standard deviation σ_{noise}. Consequently, the matrix $\boldsymbol{M}$ in Algorithm 1 is modified as

$$\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot ((\boldsymbol{Y} + \boldsymbol{W})(\boldsymbol{Y} + \boldsymbol{W})^T). \tag{23}$$

Now the smallest eigenvalue of $\boldsymbol{M}$ may be different from zero, and we can not guarantee that its corresponding eigenvector has all positive entries. Therefore the positivity of $\boldsymbol{z}$ should be incorporated into the recovery.

We first note that, finding the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{M}$ is nothing but solving the optimization problem

$$\min \|\boldsymbol{M}\boldsymbol{z}\|_2 \quad \text{s.t.} \quad \|\boldsymbol{z}\|_2 = 1. \tag{24}$$

Now we can adjust (24) by adding the positivity constraint

$$\min \|\boldsymbol{M}\boldsymbol{z}\|_2 \quad \text{s.t.} \quad \|\boldsymbol{z}\|_2 = 1 \text{ and } \boldsymbol{z} \geq 0. \tag{25}$$

Algorithm 2 Optimization-based Recovery of $\mathbf{\Phi}$

Inputs: $\boldsymbol{Y}, \boldsymbol{S}$

Output: Estimate $\hat{\mathbf{\Phi}}$

1. Let $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{S}\boldsymbol{S}^\dagger) \odot (\boldsymbol{Y}\boldsymbol{Y}^T)$.
2. Let $\boldsymbol{c} = 1_{N\times 1}$.
3. Use a convex programming method to find $\boldsymbol{z}^*$ that minimizes $\|\boldsymbol{M}\boldsymbol{z}\|_2$ s.t. $\boldsymbol{c}^T\boldsymbol{z} = 1$, $\boldsymbol{z} \geq 0$.
4. Return $\hat{\mathbf{\Phi}} = \text{diag}((z_1^*)^{-1}, \ldots, (z_N^*)^{-1})$.

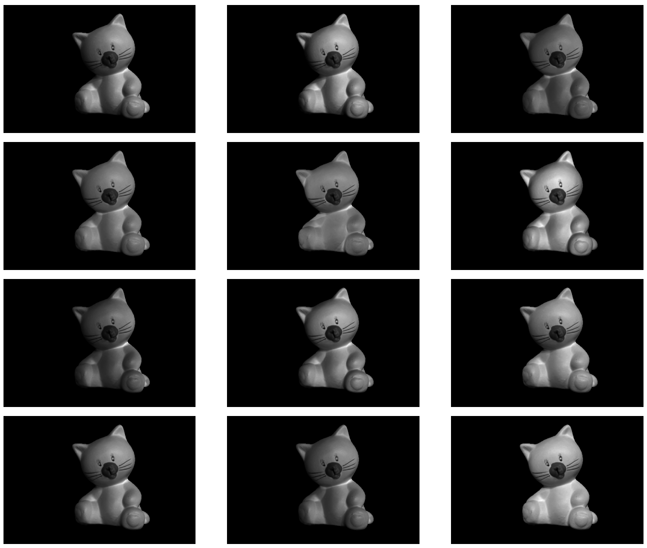


Fig. 6. Synthetic images of the same object under different lighting condtions.

Solving the optimization problem (25) is hard due to the nonconvexity of the feasible set. To convexify the problem, one can relax the constraint as

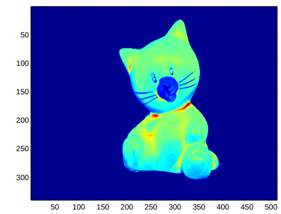
$$\min \|\boldsymbol{M}\boldsymbol{z}\|_2 \quad \text{s.t.} \quad \boldsymbol{c}^T\boldsymbol{z} = 1 \text{ and } \boldsymbol{z} \geq 0, \tag{26}$$

where $\boldsymbol{c} = [1, 1, \ldots, 1]^T \in \mathbb{R}^N$. Solving (26) is now easy using some well-developed convex programming method. This optimization-based algorithm to recover $\mathbf{\Phi}$ is described in Algorithm 2.

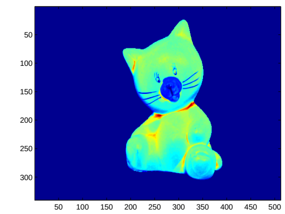
IV. Experimental Methods and Results

Simulations are performed on a data set of $K = 12$ images of the same object[3] shown in Fig. 6 using MATLAB, with the convex programming `cvx` provided by [13]. These images were synthesized by first randomly generating the lighting matrix $\boldsymbol{X}$ and then forming the product $\mathbf{\Phi}\boldsymbol{S}\boldsymbol{X}$, where $\mathbf{\Phi}$ is the ground truth for the albedos. The spherical harmonic matrix $\boldsymbol{S}$ was computed from the 3-D model of the object using (17), with $M = 9$. Using Corollary 1, we can easily verify that $\boldsymbol{S}$ has nonseparable full rank. The size of each image is 340×512; excluding zero-albedo pixels results in $N = 35983$. The albedo matrix $\mathbf{\Phi}$ was recovered using Algorithm 2 under various levels of noise (white Gaussian noise of different variances was added to the images.) The reconstructions of the albedos are visually shown in Fig. 7 in comparison with

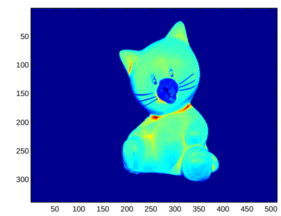
[3]The 3-D model of the object is available at: http://pages.cs.wisc.edu/ lizhang/courses/cs766-2008f/projects/phs/index.htm.



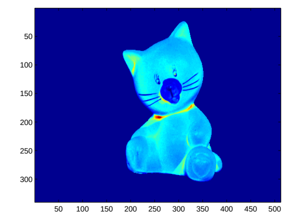
(a) Ground truth



(b) $\sigma_{\text{noise}} = 1, \text{SNR} = 34.7667$



(c) $\sigma_{\text{noise}} = 3, \text{SNR} = 25.3320$



(d) $\sigma_{\text{noise}} = 5, \text{SNR} = 17.9326$

Fig. 7. Recovered albedos with corresponding SNRs under various levels of noise.

the ground truth. All of them were normalized to the same range for comparison.

V. Discussions and Conclusions

We have studied the relighting problem of a Lambertian convex object with distant light source. Under these assumptions, the reflectance functions live close to a low-dimensional linear subspace spanned by the first few spherical harmonics. The inverse rendering phase of relighting thus becomes a factorization of the image matrix into albedo and lighting matrices, given the spherical harmonic matrix. This factorization problem is well-posed if and only if the spherical harmonic matrix has nonseparable full rank, a stronger notion of full rank. In the noiseless case, exact factorization (up to some scale) can be done via an SVD-based algorithm. When the noise is present, an optimization-based algorithm is proposed with slight modifications. Simulations are performed only on synthetic data but already suggest that the proposed algorithm is quite sensitive to noise. Below is a list of potential work to be done in the future research

- experiment on real data;
- analyze the effect of noise on the solution;
- design new algorithms that are more robust to noise;
- investigate how the lighting diversity affects the factorization;
- map the nonseparable full rank of spherical harmonic matrix to the geometry of the object.

VI. Acknowledgements

The author thanks Siying Liu for providing him with the data set and the pre-computed spherical harmonic matrix.

Appendix A
Proof of Proposition 1

We only need to prove that the rotation $R_{\alpha,\beta}$ brings the north-pole vector $[1, 0, 0]^T$ to the surface normal $\boldsymbol{n_p} =$

$[1, \alpha, \beta]^T$ (in spherical coordinates). Indeed, by definition of rotation matrices $R_z(\beta)$ and $R_y(\alpha)$, we have

$$R_y(\alpha) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_c = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin\alpha \\ 0 \\ \cos\alpha \end{bmatrix},$$

and so

$$\begin{aligned} R_{\alpha,\beta} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_c &= R_z(\beta) \cdot R_y(\alpha) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_c \\ &= \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sin\alpha \\ 0 \\ \cos\alpha \end{bmatrix} \\ &= \begin{bmatrix} \sin\alpha\cos\beta \\ \sin\alpha\sin\beta \\ \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha \\ \beta \end{bmatrix}_c . \end{aligned}$$

This coordinate transformation directly implies the proposition.

References

[1] R. Ramamoorthi, "A signal-processing framework for forward and inverse rendering," Ph.D. dissertation, Stanford University, Stanford, CA, Aug. 2002.

[2] J. Lehtinen, "A framework for precomputed and captured light transport," *ACM Trans. Graphics*, vol. 26, no. 4, pp. 13:1–13:22, Oct. 2007.

[3] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Aquiring the reflectance field of a human face," in *Proc. 27th Ann. Conf. Comput. Graphics Interactive Techniques (SIGGRAPH'00)*, New Orleans, LA, Jul. 2000, pp. 145–156.

[4] R. Ramamoorthi and P. Hanrahan, "An efficient representation of irradiance environment maps," in *Proc. 28th Ann. Conf. Comput. Graphics Interactive Techniques (SIGGRAPH'01)*, Los Angeles, CA, Aug. 2001, pp. 497–500.

[5] ——, "A signal-processing framework for inverse rendering," in *Proc. 28th Ann. Conf. Comput. Graphics Interactive Techniques (SIGGRAPH'01)*, Los Angeles, CA, Aug. 2001, pp. 117–128.

[6] ——, "A signal-processing framework for reflection," *ACM Trans. Graphics*, vol. 23, no. 4, pp. 1004–1042, Oct. 2004.

[7] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, Feb. 2003.

[8] R. Basri, D. W. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *Int. J. Comput. Vis.*, vol. 72, no. 3, pp. 239–257, May 2007.

[9] H. Q. Nguyen, S. Liu, and M. N. Do, "Subspace methods for computational relighting," in *Proc. IS&T/SPIE Comput. Imaging XI*, Feb. 2013, to appear.

[10] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Upper Saddle River, N.J.: Prentice-Hall, 1997.

[11] Y. Bresler, S. Basu, and C. Couvreur, *Hilbert Spaces and Least Squares Methods for Signal Processing*, 2012, University of Illinois at Urbana-Champaign ECE 513 Lecture Notes.

[12] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.

[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," Aug. 2012. [Online]. Available: http://cvxr.com/cvx/