Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints⋆

S. Kevin Zhou, Rama Chellappa, and David W. Jacobs

Center for Automation Research
University of Maryland, College Park MD 20742, USA

Abstract. Photometric stereo algorithms use a Lambertian reflectance model with a varying albedo field and involve the appearances of only one object. This paper extends photometric stereo algorithms to handle all the appearances of all the objects in a class, in particular the class of human faces. Similarity among all facial appearances motivates a rank constraint on the albedos and surface normals in the class. This leads to a factorization of an observation matrix that consists of exemplar images of different objects under different illuminations, which is beyond what can be analyzed using bilinear analysis. Bilinear analysis requires exemplar images of different objects under same illuminations. To fully recover the *class-specific* albedos and surface normals, integrability and face symmetry constraints are employed. The proposed linear algorithm takes into account the effects of the varying albedo field by approximating the integrability terms using only the surface normals. As an application, face recognition under illumination variation is presented. The rank constraint enables an algorithm to separate the illumination source from the observed appearance and keep the illuminant-invariant information that is appropriate for recognition. Good recognition results have been obtained using the PIE dataset.

1 Introduction

Recovery of albedos and surface normals has been studied in the computer vision community for a long time. Usually a Lambertian reflectance model, ignoring both attached and cast shadows, is employed. Early works from the shape from shading (SFS) literature assume a constant albedo field: this assumption is not valid for many real objects and thus limits the practical applicability of the SFS algorithms. Early photometric stereo approaches require knowledge of lighting conditions, but full control of the lighting sources is also constraining. Recent research efforts [1,2,4,5,6,7,8] attempt to go beyond these restrictions by (i) using a varying albedo field, a more accurate model of the real world, and (ii) assuming no prior knowledge or requiring no control of the lighting sources. As a consequence, the complexity of the problem has also increased significantly.

If we fix the imaging geometry and only move the lighting source to illuminate one object, the observed images (ignoring the cast and attached shadows) lie in a subspace completely determined by three images illuminated by three independent lighting sources [4]. If an ambient component is added [6], this subspace becomes 4-D. If attached shadows are considered as in [1,2], the subspace dimension grows to infinity but most of its energy is packed in a limited number of harmonic components, thereby leading to a low-dimensional subspace approximation. However, all the photometric-stereo-type approaches (except [5]) commonly restrict themselves to using *object-specific* samples and cannot handle the appearances not belonging to the object. In this paper, we extend the photometric stereo algorithms to handle all the appearances of all the objects in a class, in particular the human face class.

To this end, we impose a rank constraint (i.e. a linear generalization) on the albedos and surface normals of all human faces. This rank constraint enables us to accomplish a factorization of the observation matrix that decomposes a *class-specific* ensemble into a product of two matrices: one encoding the albedos and surfaces normals for a class of objects and the other encoding blending linear coefficients and lighting conditions. A class-specific ensemble consists of exemplar images of different objects under different illuminations, which can not be analyzed using bilinear analysis [14]. Bilinear analysis requires exemplar images of different objects under the same illuminations. Because a factorization is always up to an invertible matrix, a full recovery of the albedos and surface normals is not a trivial task and requires additional constraints. The surface integrability constraint [9,10] has been used in several approaches [6,8] to successfully perform the recovery task. The symmetry constraint has also been employed in [7,11] for face images. In Section 3, we present an approach which fuses these constraints to recover the albedos and surface normals for the face class, even in the presence of shadows. More importantly, this approach takes into account the effects of the varying albedo field by approximating the integrability terms using only the surface normals instead of the product of the albedos and the surface normals. Due to the nonlinearity embedded in the integrability terms, regular algorithms such as steepest descent are inefficient. We derive a linearized algorithm to find the solution.

In addition, the blending linear coefficients offer an illuminant-invariant identity signature, which is appropriate for face recognition under illumination variation. In Section 4, we first present a method for computing such coefficients and then report face recognition results using the PIE database [12].

1.1   Notations

In general, we denote a scalar by a, a vector by a, and a matrix with r rows and c columns by $\mathsf{A}_{r \times c}$. The matrix transpose is denoted by A^{T}, the pseudo-inverse by $\mathsf{A}^{\dagger}$. The matrix L_2-norm is denoted by $||.||_2$.

The following notations are introduced for the sake of notational conciseness and emphasis of special structure.

– Concatenation notations: $\Rightarrow$ and $\Downarrow$.
  $\Rightarrow$ and $\Downarrow$ mean horizontal and vertical concatenations, respectively. For example, we can represent a $n \times 1$ vector $\mathsf{a}_{n \times 1}$ by $\mathsf{a} = [a_1, a_2, ..., a_n]^\mathsf{T} = [\Downarrow_{i=1}^n a_i]$ and its transpose by $\mathsf{a}^\mathsf{T} = [a_1, a_2, ..., a_n] = [\Rightarrow_{i=1}^n a_i]$. We can use $\Rightarrow$ and $\Downarrow$ to concatenate matrices to form a new matrix. For instance, given a collection of matrices $\{\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n\}$ of size $r \times c$, we construct a $r \times cn$ matrix[1] $[\Rightarrow_{i=1}^n \mathsf{A}_i] = [\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n]$ and a $rn \times c$ matrix $[\Downarrow_{i=1}^n \mathsf{A}_i] = [\mathsf{A}_1^\mathsf{T}, \mathsf{A}_2^\mathsf{T}, \ldots, \mathsf{A}_n^\mathsf{T}]^\mathsf{T}$. In addition, we can combine $\Rightarrow$ and $\Downarrow$ to achieve a concise notation. Rather than representing a matrix $\mathsf{A}_{r \times c}$ as $[a_{ij}]$, we represent it as $\mathsf{A}_{r \times c} = [\Downarrow_{i=1}^r [\Rightarrow_{j=1}^c a_{ij}]] = [\Rightarrow_{j=1}^c [\Downarrow_{i=1}^r a_{ij}]]$. Also we can easily construct 'big' matrices using 'small' matrices $\{\mathsf{A}_{11}, \mathsf{A}_{12}, \ldots, \mathsf{A}_{1n}, \ldots, \mathsf{A}_{mn}\}$ of size $r \times c$. The matrix $[\Downarrow_{i=1}^m [\Rightarrow_{j=1}^n \mathsf{A}_{ij}]]$ is of size $rm \times cn$, the matrix $[\Rightarrow_{i=1}^m [\Rightarrow_{j=1}^n \mathsf{A}_{ij}]]$ of size $r \times cmn$.
– Kronecker (tensor) product: $\otimes$.
  It is defined as $\mathsf{A}_{m \times n} \otimes \mathsf{B}_{r \times c} = [\Downarrow_{i=1}^m [\Rightarrow_{j=1}^n a_{ij}\mathsf{B}]]_{mr \times nc}$.

2   Setting and Constraints

We assume a Lambertian imaging model with a varying albedo field and no shadows. A pixel h is represented as

$$h = p\, \mathsf{n}^\mathsf{T}\mathsf{s} = \mathsf{t}^\mathsf{T}\mathsf{s}, \tag{1}$$

where p is the albedo at the pixel, $\mathsf{n} \doteq [\hat{a}, \hat{b}, \hat{c}]^\mathsf{T}$ is the unit surface normal vector at the pixel, $\mathsf{t}_{3 \times 1} \doteq [a \doteq p\hat{a}, b \doteq p\hat{b}, c \doteq p\hat{c}]^\mathsf{T}$ is the product of albedo and surface normal, and s specifies a distant illuminant (a 3×1 unit vector multiplied by the light's intensity).

For an image h, a collection of d pixels $\{h_i, i = 1, ..., d\}$ [2], by stacking all the pixels into a column vector h, we have

$$\mathsf{h}_{d \times 1} \doteq [\Downarrow_i h_i] = [\Downarrow_i \mathsf{t}_i^\mathsf{T}]\mathsf{s} = \mathsf{T}_{d \times 3}\, \mathsf{s}_{3 \times 1}, \tag{2}$$

where $\mathsf{T} \doteq [\Downarrow_i \mathsf{t}_i^\mathsf{T}]$ contains all albedo and surface normal information about the object. We call the T matrix the *object-specific albedo-shape* matrix.

In the case of photometric stereo, we have n images of the *same* object, say $\{\mathsf{h}_1, \mathsf{h}_2, \ldots, \mathsf{h}_n\}$, observed at a fixed pose illuminated by n different lighting sources, forming an *object-specific* ensemble. Simple algebraic manipulation gives:

$$\mathsf{H}_{d \times n} \doteq [\Rightarrow_i \mathsf{h}_i] = \mathsf{T}[\Rightarrow_i \mathsf{s}_i] = \mathsf{T}_{d \times 3}\, \mathsf{S}_{3 \times n}, \tag{3}$$

---

[1] We do not need the size of $\{\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n\}$ to be exactly same. We use matrices of the same size in this example for simplicity. For example, for $[\Rightarrow_{i=1}^n \mathsf{A}_i]$, we only need the number of rows of these matrices to be same.
[2] The index i corresponds to a spatial position $\mathsf{x} \doteq (x, y)$. If no confusion, we will interchange both notations. For instance, we might also use $\mathsf{x} = 1, ..., d$.

where H is the *observation matrix* and $S \doteq [\Rightarrow_i s_i]$ encodes the information about the illuminants. Hence photometric stereo is rank-3 constrained. Therefore, given at least three exemplar images for one object under three different independent illuminations, we can determine the identity of a new probe image by checking if it lies in the linear span of the three exemplar images [4]. This requires obtaining at least three images for each object in the gallery set, which may not be possible in many applications. Note that in this recognition setting, there is no need for the training set that is defined below; in other words, the training set is equivalent to the gallery set.

We follow [13] in defining a typical recognition protocol for face recognition algorithms. Three sets are needed: Gallery, probe, and training sets. The gallery set consists of images with known identities. The probe set consists of images whose identities are to be determined by matching with the gallery set. In addition, the training set is provided for the recognition algorithm to learn characteristic features of the face images. In general, we assume no identity overlap between the gallery set and the training set and often store only one exemplar image for each object in the gallery set. However, the training set can have more than one image for each object. In order to generalize from the training set to the gallery and probe sets, we note that all images in the training, gallery, and probe sets belong to the same face class, which naturally leads to a rank constraint.

2.1 The Rank Constraint

We impose the rank constraint on the T matrix by assuming that any T matrix is a linear combination of some basis matrices $\{T_1, T_2, \ldots, T_m\}$, i.e., there exist coefficients $\{f_j; \ j = 1, \ldots, m\}$ such that

$$T_{d \times 3} = \sum_{j=1}^{m} f_j T_j = [\Rightarrow_j T_j](f \otimes I_3) = W_{d \times 3m}(f_{m \times 1} \otimes I_3), \tag{4}$$

where $f \doteq [\Downarrow_j f_j]$, $W \doteq [\Rightarrow_j T_j]$, and I_n denotes an identity matrix of dimension $n \times n$. Since the W matrix encodes all albedos and surface normals for a class of objects, we call it a *class-specific albedo-shape* matrix. Similar rank constraints are widely found in the literature; see for example [20,21].

Substitution of (4) into (2) yields

$$h_{d \times 1} = Ts = W(f \otimes I_3)s = W(f \otimes s) = W_{d \times 3m} \ k_{3m \times 1}, \tag{5}$$

where $k \doteq f \otimes s$. This leads to a two-factor bilinear analysis [14]. Recently, a multilinear analysis has been proposed in [20,22].

With the availability of n images $\{h_1, h_2, \ldots, h_n\}$ for *different* objects, observed at a fixed pose illuminated by n different lighting sources, forming a *class-specific* ensemble, we have

$$H_{d \times n} = [\Rightarrow_i h_i] = W[\Rightarrow_i (f_i \otimes s_i)] = W[\Rightarrow_i k_i] = W_{d \times 3m} \ K_{3m \times n}, \tag{6}$$

where $K \doteq [\Rightarrow_i (f_i \otimes s_i)] = [\Rightarrow_i k_i]$. It is a rank-$3m$ problem. Notice that K takes a special form.

The rank constraint generalizes many approaches in the literature and is quite easily satisfied. If $m = 1$, this reduces to the case of photometric stereo; if the surface normal is fixed and the albedo field lies in a rank-m linear subspace, we have (4) satisfied too. Interestingly, the 'Eigenface' approach [15] is just a special case of our approach, but this is only for a fixed illumination source. Suppose that the illuminant vector is $\tilde{\mathsf{s}}$. (5) and (6) reduce to equations:

$$\mathsf{h}_{d\times 1} = \mathsf{W}(\mathsf{f}\otimes\tilde{\mathsf{s}}) = \tilde{\mathsf{W}}_{d\times m}\mathsf{f}_{m\times 1}; \ \mathsf{H}_{d\times n} = [\Rightarrow_i \mathsf{h}_i] = \tilde{\mathsf{W}}[\Rightarrow_i \mathsf{f}_i] = \tilde{\mathsf{W}}_{d\times m}\mathsf{F}_{m\times n}, \quad (7)$$

where $\tilde{\mathsf{W}} \doteq [\Rightarrow_i \mathsf{T}_i\tilde{\mathsf{s}}]$. Therefore, our approach can be regarded as a generalized 'Eigenface' analysis able to handle illumination variation.

Our immediate goal is to estimate W and K from the observation matrix H. The first step is to invoke an SVD factorization, say $\mathsf{H} = \mathsf{U}\Lambda\mathsf{V}^\mathsf{T}$, and retain the top $3m$ components as $\mathsf{H} = \mathsf{U}_{3m}\Lambda_{3m}\mathsf{V}_{3m}^\mathsf{T} = \hat{\mathsf{W}}\ \hat{\mathsf{K}}$, where $\hat{\mathsf{W}} = \mathsf{U}_{3m}$ and $\hat{\mathsf{K}} = \Lambda_{3m}\mathsf{V}_{3m}^\mathsf{T}$. Thus, we can recover W and K up to a $3m \times 3m$ invertible matrix R with $\mathsf{W} = \hat{\mathsf{W}}\mathsf{R}$, $\mathsf{K} = \mathsf{R}^{-1}\hat{\mathsf{K}}$. Additional constraints are required to determine the R matrix. We use the integrability and symmetry constraints, both related to W. Moreover, $\mathsf{K} = [\Rightarrow_i (\mathsf{f}_i \otimes \mathsf{s}_i)]$.

2.2   The Integrability Constraint

One common constraint used in SFS research is the integrability of the surface [9,10,6,8]. Suppose that the surface function is $z = z_{(\mathsf{x})}$ with $\mathsf{x} \doteq (x,y)$, we must have $\frac{\partial}{\partial x}\frac{\partial z}{\partial y} = \frac{\partial}{\partial y}\frac{\partial z}{\partial x}$. If given the unit surface normal vector $\mathsf{n}_{(\mathsf{x})} \doteq [\hat{a}_{(\mathsf{x})}, \hat{b}_{(\mathsf{x})}, \hat{c}_{(\mathsf{x})}]^\mathsf{T}$ at pixel x, we have $\frac{\partial}{\partial x}\frac{\hat{b}_{(\mathsf{x})}}{\hat{c}_{(\mathsf{x})}} = \frac{\partial}{\partial y}\frac{\hat{a}_{(\mathsf{x})}}{\hat{c}_{(\mathsf{x})}}$. In other words, with $\alpha_{(\mathsf{x})}$ defined as an integrability constraint term,

$$\alpha_{(\mathsf{x})} \doteq \hat{c}_{(\mathsf{x})}\frac{\partial\hat{b}_{(\mathsf{x})}}{\partial x} - \hat{b}_{(\mathsf{x})}\frac{\partial\hat{c}_{(\mathsf{x})}}{\partial x} + \hat{a}_{(\mathsf{x})}\frac{\partial\hat{c}_{(\mathsf{x})}}{\partial y} - \hat{c}_{(\mathsf{x})}\frac{\partial\hat{a}_{(\mathsf{x})}}{\partial y} = 0. \quad (8)$$

If instead are given the product of albedo and surface normal $\mathsf{t}_{(\mathsf{x})} \doteq [a_{(\mathsf{x})}, b_{(\mathsf{x})}, c_{(\mathsf{x})}]^\mathsf{T}$ with $a_{(\mathsf{x})} \doteq p_{(\mathsf{x})}\hat{a}_{(\mathsf{x})}$, $b_{(\mathsf{x})} \doteq p_{(\mathsf{x})}\hat{b}_{(\mathsf{x})}$, and $c_{(\mathsf{x})} \doteq p_{(\mathsf{x})}\hat{c}_{(\mathsf{x})}$, (8) still holds with $\hat{a}$, $\hat{b}$, and $\hat{c}$ replaced by a, b, and c, respectively. Practical algorithms approximate the partial derivatives by forward or backward differences or other differences that use an inherent smoothness assumption. Hence, the approximations based on $\mathsf{t}_{(\mathsf{x})}$ are very rough especially at places where abrupt albedo variation exists (e.g. the boundaries of eyes, iris, eyebrow, etc) since the smoothness assumption is seriously violated. We should by all means use $\mathsf{n}_{(\mathsf{x})}$ in order to remove this effect.

2.3   The Symmetry Constraint

For a face image in a frontal view, one natural constraint is its symmetry about the central y-axis as proposed in [7,11]:

$$p_{(x,y)} = p_{(-x,y)}; \hat{a}_{(x,y)} = -\hat{a}_{(-x,y)}; \hat{b}_{(x,y)} = \hat{b}_{(-x,y)}; \hat{c}_{(x,y)} = \hat{c}_{(-x,y)}, \quad (9)$$

which is equivalent to, using $\mathsf{x} \doteq (x, y)$ and its symmetric point $\bar{\mathsf{x}} \doteq (-x, y)$,

$$a_{(\mathsf{x})} = -a_{(\bar{\mathsf{x}})};\ b_{(\mathsf{x})} = b_{(\bar{\mathsf{x}})}; c_{(\mathsf{x})} = c_{(\bar{\mathsf{x}})}. \tag{10}$$

If a face image is in a non-frontal view, such a symmetry still exists but the coordinate system should be modified to take into account the view change.

3 The Recovery of Albedos and Surface Normals

The recovery task is to find from the observation matrix H the *class-specific albedo-shape* matrix W (or equivalently R), which satisfies both the integrability and symmetry constraints, as well as the matrices F and S. Denote $\mathsf{R} \doteq [\Rightarrow_{j=1}^{m} [r_{aj}, r_{bj}, r_{cj}]]$ and $\hat{\mathsf{W}} \doteq [\Downarrow_{\mathsf{x}=1}^{d} \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}]$. As $\mathsf{W} \doteq [\Downarrow_{\mathsf{x}=1}^{d} [\Rightarrow_{j=1}^{m} [a_{j(\mathsf{x})}, b_{j(\mathsf{x})}, c_{j(\mathsf{x})}]]] = \hat{\mathsf{W}}\mathsf{R}$, we have

$$a_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} r_{aj},\ b_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} r_{bj},\ c_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} r_{cj};\ j = 1, \ldots, m. \tag{11}$$

Practical systems must take into account attached and cast shadows as well as sensor noise. The existence of shadows in principle increases the rank to infinity. But, if we exclude shadowed pixels or set them as missing values, we still have rank 3. Performing a SVD with missing values is discussed in [3,16].

In view of the above circumstances, we formulate the following optimization problem: minimizing over R, F, and S the cost function $\mathcal{E}$ defined as

$$\mathcal{E}(\mathsf{R}, \mathsf{F}, \mathsf{S}) = \frac{1}{2} \sum_{i=1}^{n} \sum_{\mathsf{x}=1}^{d} \mathtt{i}_i(\mathsf{x}) \{h_{i(\mathsf{x})} - \hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}} \mathsf{R}(\mathsf{f}_i \otimes \mathsf{s}_i)\}^2$$

$$+ \frac{\lambda_1}{2} \sum_{j=1}^{m} \sum_{\mathsf{x}=1}^{d} \{\alpha_{j(\mathsf{x})}\}^2 + \frac{\lambda_2}{2} \sum_{j=1}^{m} \sum_{\mathsf{x}=1}^{d} \{\beta_{j(\mathsf{x})}\}^2,$$

$$= \mathcal{E}_0(\mathsf{R}, \mathsf{F}, \mathsf{S}) + \lambda_1 \mathcal{E}_1(\mathsf{R}) + \lambda_2 \mathcal{E}_2(\mathsf{R}), \tag{12}$$

where $\mathtt{i}_i(\mathsf{x})$ is an indicator function describing whether the pixel x of the image h_i is in shadow, $\alpha_{j(\mathsf{x})}$ is the integrability constraint term based only on surface normals as defined in (8), and $\beta_{j(\mathsf{x})}$ is the symmetry constraint term given as

$$\beta_{j(\mathsf{x})}^2 = \{a_{j(\mathsf{x})} + a_{j(\bar{\mathsf{x}})}\}^2 + \{b_{j(\mathsf{x})} - b_{j(\bar{\mathsf{x}})}\}^2 + \{c_{j(\mathsf{x})} - c_{j(\bar{\mathsf{x}})}\}^2;\ j = 1, \ldots, m. \tag{13}$$

One approach could be to directly minimize the cost function over W, F, and S. This is in principle possible but numerically difficult as the number of unknowns depends on the image size, which can be quite large in practice.

As shown in [17], the surface normals can be recovered up to a generalized bas-relief (GBR) ambiguity. To resolve the GBR ambiguity, we normalize the matrix R by keeping $||\mathsf{R}||_2 = 1$. Another ambiguity between f_j and s_j is a nonzero scale, which can be removed by normalizing f: $\mathsf{f}_j^{\mathsf{T}} \mathbf{1} = 1$, where $\mathbf{1}_{m \times 1}$ is a vector of 1's.

To summarize, we perform the following task:

$$min_{\mathsf{R,F,S}} \quad \mathcal{E}(\mathsf{R,F,S}) \quad subject\ to\ ||\mathsf{R}||_2 = 1, \mathsf{F^T}1 = 1. \tag{14}$$

An iterative algorithm can be designed to solve (14). While solving for F and S with R fixed is quite easy, solving for R given F and S is very difficult because the integrability constraint terms require partial derivatives of the surface normals that are nonlinear in R. Regular algorithms such as steepest descent are inefficient. One main contribution in this paper is that we propose a linearized algorithm to solve for R. Appendix-I presents the details of the complete algorithm.

To demonstrate how the algorithm works, we design the following scenario with $m = 2$ so that the rank of interest is 2x3=6. To defeat the photometric stereo algorithm, which requires one object illuminated by at least three sources, and the bilinear analysis which requires two fixed objects illuminated by at least the same three lighting sources, we synthesize eight images by taking random linear combinations of two basis objects illuminated by eight different lighting sources. Fig. 1 displays the synthesized images and the recovered class-specific albedo-shape matrix, which clearly shows the two basis objects. Our algorithm converges within 100 iterations.

One notes that the special case $m = 1$ of our algorithm can be readily applied to photometric stereo (with the symmetry constraint removed) to robustly recover the albedos and surface normals for one object.
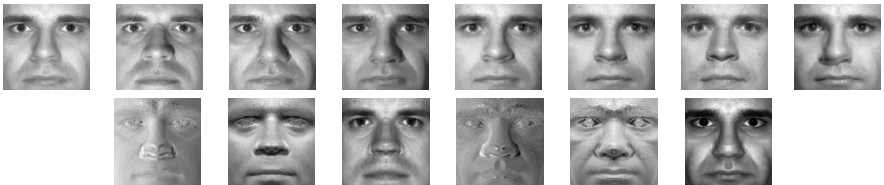
Fig. 1. Row 1: Eight Synthesized images that are random linear combinations of two basis objects illuminated by eight different lighting sources. Row 2: Recovered class-specific albedo-shape matrix showing the two basis objects (i.e. the three columns of T_1 and T_2).

4   Recognition Experiments

We study an extreme recognition setting with the following features: there is no identity overlap between the training set and the gallery and probe sets; only one image for one object is stored in the gallery set; the lighting conditions for the training, gallery and probe sets are completely unknown. The only known fact is that the face is in frontal view. Our strategy is to: (i) Learn W from the

training set using the recovery algorithm described in Section 3; (ii) With W given, learn the identity signature f's for both the gallery and probe sets using the recovery algorithm described in Section 4.1 and no knowledge of illumination directions; and (iii) Perform recognition using the nearest correlation coefficient. Suppose that one gallery image g has its signature f_g and one probe image p has its signature f_p, their correlation coefficient is $cc(p,g) = (f_p, f_g)/\sqrt{(f_p, f_p)(f_g, f_g)}$, where (x, y) is an inner-product such as $(x, y) = x^T \Sigma y$ with Σ learned or given.

4.1   Separating Illumination

With the class-specific albedo-shape matrix W available, we proceed to solve the problem of separating illumination, *v.i.z.*, for an arbitrary image h, find the illuminant vector s and the identity signature f. For convenience of recognition, we normalize f to the same range: $f^T 1 = 1$. Appendix-II presents the recovery algorithm which infers the shadow pixels as well.

4.2   PIE Dataset

We use the Pose and Illumination and Expression (PIE) dataset [12] in our experiment. Fig. 2 shows the distribution of all 21 flashes used in PIE and their estimated positions using our algorithm. Since the flashes are almost symmetrically distributed about the head position, we only use 12 of them distributed on the right half of the unit sphere in Fig. 2. In total, we used 68x12=816 images in a fixed view as there are 68 subjects in the PIE database. Fig. 2 also displays one PIE object under the selected 12 illuminants. Registration is performed by aligning the eyes and mouth to canonical positions. No flow computation is carried out for further alignment. We use cropped face regions. After the pre-processing step, the actual image size is 50×50. Also, we only study gray images by taking the average of the red, green, and blue channels. We use all 68 images under one illumination to form a gallery set and under another illumination to form a probe set. The training set is taken from sources other than the PIE dataset. Thus, we have 132 tests, with each test giving rise to a recognition score.

4.3   Recognition Performance

The training set is first taken from the Yale illumination database [8]. There are only 10 subjects (i.e. $m = 10$) in this database and each subject has 64 images in frontal view illuminated by 64 different lights. We only use 9 lights and Fig. 2 shows one Yale object under 9 lights.

   Table 1 lists the recognition rate for all test protocols for the PIE database using the Yale database as the training set. Even with $m = 10$, we obtain quite good results. One observation is that when the flashes become separated, the recognition rate decreases. Also, using images under frontal or near-frontal illuminants as galleries produces good results. For comparison, we also implemented the 'Eigenface' approach [15] by training the projection directions from the same training set. Its average recognition rate is only 35% while ours is 67%.
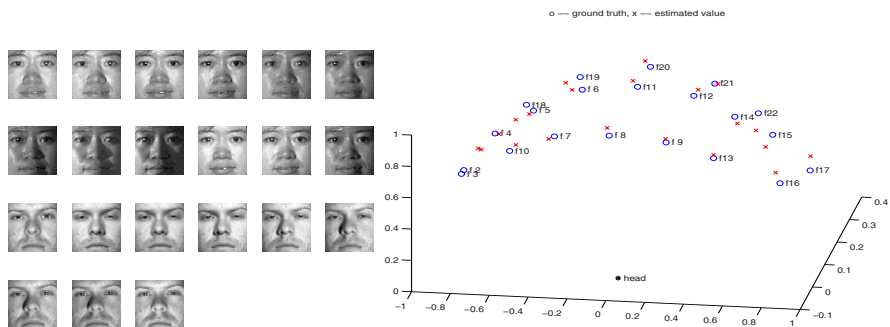
Fig. 2. Left: Rows 1-2 display one PIE object under the selected 12 illuminants and Rows 3-4 one Yale object under 9 lights used in the training set. Right: Flash distribution in the PIE database. For illustrative purposes, we move their positions on a unit sphere as only the illuminant directions matter. 'o' means the ground truth and 'x' the estimated values. It is quite accurate for estimation of directions of flashes near frontal pose. But when the flashes are very off-frontal, accuracy goes down slightly.

Table 1. Recognition rate obtained by our rank constrained approach using the Yale database (the left number in each cell) and Vetter database (the right number in each cell) as the training set. 'F' means 100 and 'fnn' flash number nn.

Gallery Probe	f08	f09	f11	f12	f13	f14	f15	f16	f17	f20	f21	f22	Average
f08	-	96/F	96/99	87/99	66/97	60/97	46/79	29/72	22/43	85/99	78/97	53/93	65/88
f09	94/F	-	96/99	96/99	90/99	87/99	56/97	40/91	24/60	84/97	96/97	68/97	75/94
f11	94/99	91/99	-	97/F	72/F	72/F	38/90	28/76	16/65	F/F	94/F	51/99	69/93
f12	88/99	94/99	97/F	-	88/F	93/F	57/F	41/93	28/76	94/F	F/F	76/F	78/97
f13	56/99	87/99	59/F	85/F	-	F/F	90/F	71/F	50/88	54/99	87/F	F/F	76/99
f14	51/99	85/99	63/F	93/F	F/F	-	90/F	66/F	49/96	59/99	91/F	99/F	77/99
f15	33/84	40/94	37/93	49/F	85/F	88/F	-	93/F	78/F	32/88	49/F	97/F	62/96
f16	19/69	26/87	26/78	32/90	59/F	44/F	84/F	-	93/F	26/69	31/F	63/F	46/89
f17	14/44	28/60	19/51	26/71	50/84	41/91	68/99	94/F	-	19/56	26/75	44/94	39/75
f20	90/97	85/97	99/F	97/F	65/F	69/F	38/90	26/74	21/68	-	93/F	53/F	67/93
f21	79/97	94/97	93/F	F/F	88/F	94/F	62/F	49/97	28/82	91/F	-	76/F	78/98
f22	43/90	65/97	46/96	75/F	99/F	99/F	97/F	76/F	59/99	43/97	74/F	-	70/98
Average	60/89	72/93	66/92	76/96	78/98	77/99	66/96	56/91	42/80	63/91	74/96	71/98	67/93

Generalization capacity with $m = 10$ is rather restrictive. We now increase m from 10 to 100 by using the Vetter's 3D face database [18]. As this is a 3D database, we actually have W available. However, we believe that using a training set of $m = 100$ from other sources can yield similar performance. Table 1 presents the recognition rates. Significant improvements have been achieved due to the increase in m. The average rate is 93%. This seems to suggest that a moderate rank of 100 is enough to span the entire face space under a fixed view.

As a comparison, Romdhani et. al. [18] reported the recognition rates only with 'f12' being the gallery set and their average is 98% while ours is 96%. Our approach is very different from [18]. In [18] depths and texture maps of explicit 3D face models are used, while our approach is image-based and recovers albedos and surface normals. 3D models can be then be reconstructed. In the

experiments, (i) we use the 'illum' part of the PIE database that is close to the Lambertian model and they use the 'light' part that includes an ambient light; (ii) we use gray-valued images and they use color images; (iii) we assume known pose but unknown illumination but they assume unknown pose but known illumination; and (iv) compared to [18], our alignment is rather crude and can be improved using flow computations. We believe that our recognition rates can be boosted using the color images and finer alignment.

5   Conclusions

We presented an approach that naturally combines the rank-constraint for identity with illumination modeling. By using the integrability and symmetry constraints, we then achieved a linear algorithm that recovers the albedos and surface normals for a class of face images under the most general setting, i.e., the observation matrix consists of different objects under different illuminations. Further, after separating the illuminations, we obtained illumination-invariant identity signatures which produced good recognition performances under illumination variations. We still need to investigate pose variations and extreme lighting conditions that cause more shadows.

References

1. R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218 - 233, 2003.
2. R. Basri and D. Jacobs, "Photometric stereo with general, unknown lighting," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 374-381, 2001.
3. D. Jacobs, "Linear fitting with missing data for structure-from-motion," *Computer Vision and Image Understanding*, vol. 82, pp. 57 - 81, 2001.
4. A Shashua, "On photometric issues in 3D visual recognition from a single 2D image," *International Journal of Computer Vision*, vol. 21, no. 1, pp. 99–122, 1997.
5. A. Shashua and T. R. Raviv, "The quotient image: Class based re-rendering and recognition with varying illuminations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 129–139, 2001.
6. A. L. Yuille, D. Snow, Epstein R., and P. N. Belhumeur, "Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 203–222, 1999.
7. W. Zhao and R. Chellappa, "Symmetric shape from shading using self-ratio image," *International Journal of Computer Vision*, vol. 45, pp. 55–752, 2001.
8. A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 643 –660, 2001.
9. R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shading problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-10, pp. 439–451, 1987.

10. D. Forsyth, "Shape from texture and integrability," *International Conference on Computer Vision*, pp. 447-453, 2001.
11. I. Shimshoni, Y. Moses, and M. Lindenbaum., "Shape reconstruction of 3D bilaterally symmetric surfaces," *International Journal of Computer Vision*, vol. 39, pp. 97–100, 2000.
12. T. Sim, S. Baker, and M. Bsat, "The CMU pose, illuminatin, and expression (PIE) database," *Prof. Face and Gesture Recognition*, pp. 53-58, 2002.
13. P. J. Philipps, H. Moon, P. Rauss, and S. Rivzi, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.
14. W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," *IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.
15. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 72–86, 1991.
16. M.E. Brand, "Incremental singular value decomposition of uncertain data with missing values," *European Conference on Computer Vision*, pp. 707-720, 2002.
17. P. Belhumeur, D. Kriegman, and A. Yuille, "The bas-relief ambiguity," *International Journal of Computer Vision*, vol. 35, pp. 33-44, 1999.
18. V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1063-1074, 2003.
19. Q. F. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo and shape from shading," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-13, pp. 680–702, 1991.
20. A. Shashua and A. Levin. "Linear Image Coding for Regression and Classification using the Tensor-rank Principle." *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
21. C. Bregler, A. Hertzmann, and H. Biermann "Recovering Non-Rigid 3D Shape from Image Streams." *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.
22. M.A.O. Vasilescu and D. Terzopoulos, "Multilinear Analysis of Image Ensembles: TensorFaces," *European Conference on Computer Vision*, 2002.

Appendix-I: Recovering R, F, and S from H

This appendix presents an iterative algorithm that recovers R, F and S from the observation matrix H. In fact, we also infer $\mathsf{I} = [\Downarrow_\mathsf{x} [\Rightarrow_i \mathtt{i}_i(\mathsf{x})]]$, which is an indication matrix for H.

We first concentrate on the most difficult part of updating R with F, S, and I fixed. We take vector derivatives of $\mathcal{E}$ with respective to $\{\mathsf{r}_{ij};\ i = a, b, c;\ j = 1, \ldots, m\}$ and treat the three terms in $\mathcal{E}$ separately.

[*About $\mathcal{E}_0$.*] With $\mathsf{f}_{j'} \doteq [\Downarrow_{j=1}^m f_{j'j}]$ and $\mathsf{s}_{j'} \doteq [s_{j'a}, s_{j'b}, s_{j'c}]^\mathsf{T}$,

$$\frac{\partial \mathcal{E}_0}{\partial \mathsf{r}_{ij}} = \sum_{j'=1}^n \sum_{\mathsf{x}=1}^d \mathtt{i}_{j'(\mathsf{x})} \{\hat{\mathsf{w}}(\mathsf{x})^\mathsf{T} \mathsf{R}(\mathsf{f}_{j'} \otimes \mathsf{s}_{j'}) - h_{j'(\mathsf{x})}\} \hat{\mathsf{w}}(\mathsf{x}) f_{j'j} s_{j'i}$$

$$= \sum_{j'=1}^{n} \sum_{\mathsf{x}=1}^{d} \mathtt{i}_{j'(\mathsf{x})} \{ \sum_{l=a,b,c} \sum_{k=1}^{m} \hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}} \mathsf{r}_{lk} f_{j'k} s_{j'l} - h_{j'(\mathsf{x})} \} \hat{\mathsf{w}}(\mathsf{x}) f_{j'j} s_{j'i}$$

$$= \sum_{l=a,b,c} \sum_{k=1}^{m} \{ \sum_{j'=1}^{n} \sum_{\mathsf{x}=1}^{d} \mathtt{i}_{j'(\mathsf{x})} f_{j'k} s_{j'l} f_{j'j} s_{j'i} \hat{\mathsf{w}}(\mathsf{x}) \hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}} \} \mathsf{r}_{lk}$$

$$- \sum_{j'=1}^{n} \sum_{\mathsf{x}=1}^{d} \mathtt{i}_{j'(\mathsf{x})} h_{j'(\mathsf{x})} f_{j'j} s_{j'i} \hat{\mathsf{w}}(\mathsf{x})$$

$$= \sum_{l=a,b,c} \sum_{k=1}^{m} \mathsf{O}_{ij}^{lk} \mathsf{r}_{lk} - \gamma_{ij}, \tag{15}$$

where $\{\mathsf{O}_{ij}^{lk}; l = a, b, c; k = 1, \dots, m\}$ are properly defined $3m \times 3m$ matrices, and γ_{ij} is a properly defined $3m \times 1$ vector.

[*About $\mathcal{E}_1$.*] Using forward differences to approximate partial derivatives [3],

$$\frac{\partial \hat{a}_{j(x,y)}}{\partial y} \simeq \hat{a}_{j(x,y+1)} - \hat{a}_{j(x,y)}; \quad \frac{\partial \hat{b}_{j(x,y)}}{\partial x} \simeq \hat{b}_{j(x+1,y)} - \hat{b}_{j(x,y)};$$
$$\frac{\partial \hat{c}_{j(x,y)}}{\partial x} \simeq \hat{c}_{j(x+1,y)} - \hat{c}_{j(x,y)}; \quad \frac{\partial \hat{c}_{j(x,y)}}{\partial y} \simeq \hat{c}_{j(x,y+1)} - \hat{c}_{j(x,y)}, \tag{16}$$

we have

$$\alpha_{j(x,y)} \approx \hat{b}_{j(x+1,y)} \hat{c}_{j(x,y)} - \hat{b}_{j(x,y)} \hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)} \hat{c}_{j(x,y+1)} - \hat{a}_{j(x,y+1)} \hat{c}_{j(x,y)}. \tag{17}$$

Suppose we are given the product of albedo and surface normal as in (11), we can derive the albedo $p_j(\mathsf{x})$ and surface normals $\hat{a}_{j(\mathsf{x})}$, $\hat{b}_{j(\mathsf{x})}$, and $\hat{c}_{j(\mathsf{x})}$ as follows:

$$p_{j(\mathsf{x})} = \sqrt{(\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj})^2 + (\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{bj})^2 + (\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{cj})^2}, \tag{18}$$

$$\hat{a}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj}}{p_{j(\mathsf{x})}}, \quad \hat{b}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{bj}}{p_{j(\mathsf{x})}}, \quad \hat{c}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{cj}}{p_{j(\mathsf{x})}}. \tag{19}$$

So, their partial derivatives with respect to r_{aj} are

$$\frac{\partial \hat{a}_{j(\mathsf{x})}}{\partial \mathsf{r}_{aj}} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}}{p_{j(\mathsf{x})}} - \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj} \frac{\hat{\mathsf{w}}_{(\mathsf{x})} \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj}}{p_{j(\mathsf{x})}^3} = \frac{1 - \hat{a}_{j(\mathsf{x})}^2}{p_{j(\mathsf{x})}} \hat{\mathsf{w}}_{(\mathsf{x})}, \tag{20}$$

$$\frac{\partial \hat{a}_{j(\mathsf{x})}}{\partial \mathsf{r}_{bj}} = -\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj} \frac{\hat{\mathsf{w}}_{(\mathsf{x})} \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{bj}}{p_{j(\mathsf{x})}^3} = \frac{-\hat{a}_{j(\mathsf{x})} \hat{b}_{j(\mathsf{x})}}{p_{j(\mathsf{x})}} \hat{\mathsf{w}}_{(\mathsf{x})}, \quad \frac{\partial \hat{a}_{j(\mathsf{x})}}{\partial \mathsf{r}_{cj}} = \frac{-\hat{a}_{j(\mathsf{x})} \hat{c}_{j(\mathsf{x})}}{p_{j(\mathsf{x})}} \hat{\mathsf{w}}_{(\mathsf{x})}. \tag{21}$$

Similarly, we can derive their partial derivatives with respect to r_{bj} and r_{cj}, which are summarized as follows:

$$\frac{\partial \hat{k}_{j(\mathsf{x})}}{\partial \mathsf{r}_{lj}} = \frac{-\hat{k}_{j(\mathsf{x})} \hat{l}_{j(\mathsf{x})}}{p_{j(\mathsf{x})}} \hat{\mathsf{w}}_{(\mathsf{x})}, \quad \frac{\partial \hat{k}_{j(\mathsf{x})}}{\partial \mathsf{r}_{kj}} = \frac{1 - \hat{k}_{j(\mathsf{x})}^2}{p_{j(\mathsf{x})}} \hat{\mathsf{w}}_{(\mathsf{x})}, \quad k, l \in \{a, b, c\}, \ k \neq l. \tag{22}$$

---
[3] Partial derivatives of boundary pixels require different approximations. But, similar derivations (skipped here due to space limitation) can be derived.

Notice that $\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial r_{bj}} = \frac{\partial \hat{b}_{j(\mathsf{X})}}{\partial r_{aj}}$, $\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial r_{cj}} = \frac{\partial \hat{c}_{j(\mathsf{X})}}{\partial r_{aj}}$, and $\frac{\partial \hat{b}_{j(\mathsf{X})}}{\partial r_{cj}} = \frac{\partial \hat{c}_{j(\mathsf{X})}}{\partial r_{bj}}$, which implies saving in computations.

We now compute the partial derivative of $\alpha_{j(x,y)}$ with respect to r_{aj}:

$$
\frac{\partial \alpha_{j(x,y)}}{\partial r_{aj}} = \frac{\partial}{\partial r_{aj}} \{ \hat{b}_{j(x+1,y)} \hat{c}_{j(x,y)} - \hat{b}_{j(x,y)} \hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)} \hat{c}_{j(x,y+1)}
$$

$$
- \hat{a}_{j(x,y+1)} \hat{c}_{j(x,y)} \}
$$

$$
= \{ \frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{w}_{(x,y)} \hat{w}_{(x,y+1)}^{\mathsf{T}} - \frac{\hat{a}_{j(x,y+1)} \hat{c}_{j(x,y+1)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{w}_{(x,y+1)} \hat{w}_{(x,y)}^{\mathsf{T}} \} r_{aj}
$$

$$
+ \{ \frac{\hat{a}_{j(x+1,y)} \hat{c}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{w}_{(x+1,y)} \hat{w}_{(x,y)}^{\mathsf{T}} - \frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{w}_{(x,y)} \hat{w}_{(x+1,y)}^{\mathsf{T}} \} r_{bj}
$$

$$
+ \{ \frac{\hat{a}_{j(x,y)} \hat{b}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{w}_{(x,y)} \hat{w}_{(x+1,y)}^{\mathsf{T}} - \frac{\hat{a}_{j(x+1,y)} \hat{b}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{w}_{(x+1,y)} \hat{w}_{(x,y)}^{\mathsf{T}}
$$

$$
+ \frac{1 - \hat{a}_{j(x,y)}^2}{p_{j(x,y)} p_{j(x,y+1)}} \hat{w}_{(x,y)} \hat{w}_{(x,y+1)}^{\mathsf{T}} - \frac{1 - \hat{a}_{j(x,y+1)}^2}{p_{j(x,y)} p_{j(x+1,y)}} \hat{w}_{(x,y+1)} \hat{w}_{(x,y)}^{\mathsf{T}} \} r_{cj}
$$

$$
= \mathsf{P}^a_{aj(x,y)} r_{aj} + \mathsf{P}^b_{aj(x,y)} r_{bj} + \mathsf{P}^c_{aj(x,y)} r_{cj} = \sum_{l=a,b,c} \mathsf{P}^l_{aj(x,y)} r_{lj}, \tag{23}
$$

where $\mathsf{P}^a_{aj(x,y)}$, $\mathsf{P}^b_{aj(x,y)}$, and $\mathsf{P}^c_{aj(x,y)}$ are properly defined matrices of dimension $3m \times 3m$. By the same token, using properly defined $\mathsf{P}^a_{bj(x,y)}$, $\mathsf{P}^b_{bj(x,y)}$, $\mathsf{P}^c_{bj(x,y)}$, $\mathsf{P}^a_{cj(x,y)}$, $\mathsf{P}^b_{cj(x,y)}$, and $\mathsf{P}^c_{cj(x,y)}$, we can calculate $\frac{\partial \alpha_{j(x,y)}}{\partial r_{ij}} = \sum_{l=a,b,c} \mathsf{P}^l_{ij(x,y)} r_{lj}$ for $i = a, b, c$, and, finally,

$$
\frac{\partial \mathcal{E}_1}{\partial r_{ij}} = \sum_{\mathsf{X}=1}^{d} \alpha_{j(\mathsf{X})} \sum_{l=a,b,c} \mathsf{P}^l_{ij(\mathsf{X})} r_{lj} = \sum_{l=a,b,c} \mathsf{P}^l_{ij} r_{lj}; \quad \mathsf{P}^l_{ij} \doteq \sum_{\mathsf{X}=1}^{d} \alpha_{j(\mathsf{X})} \mathsf{P}^l_{ij(\mathsf{X})}. \tag{24}
$$

[*About* $\mathcal{E}_2$.] The symmetry constraint term $\beta_{j(\mathsf{X})}$ defined as in (13) can be expressed as

$$
\beta_{j(\mathsf{X})}^2 = r_{aj}^{\mathsf{T}} \mathsf{Q}^a_{(\mathsf{X})} r_{aj} + r_{bj}^{\mathsf{T}} \mathsf{Q}^b_{(\mathsf{X})} r_{bj} + r_{cj}^{\mathsf{T}} \mathsf{Q}^c_{(\mathsf{X})} r_{cj}, \tag{25}
$$

where $\mathsf{Q}^a_{(\mathsf{X})}$, $\mathsf{Q}^b_{(\mathsf{X})}$, and $\mathsf{Q}^c_{(\mathsf{X})}$ are symmetric matrices with size $3m \times 3m$:

$$
\mathsf{Q}^a_{(\mathsf{X})} = (\hat{w}_{(\mathsf{X})} + \hat{w}_{(\bar{\mathsf{X}})})(\hat{w}_{(\mathsf{X})} + \hat{w}_{(\mathsf{X})})^{\mathsf{T}}, \mathsf{Q}^b_{(\mathsf{X})} = (\hat{w}_{(\mathsf{X})} - \hat{w}_{(\bar{\mathsf{X}})})(\hat{w}_{(\mathsf{X})} - \hat{w}_{(\mathsf{X})})^{\mathsf{T}}, \mathsf{Q}^c_{(\mathsf{X})} = \mathsf{Q}^b_{(\mathsf{X})}. \tag{26}
$$

The derivatives of $\beta_{j(\mathsf{X})}^2/2$ and $\mathcal{E}_2$ with respect to r_{aj}, r_{bj}, and r_{cj} are

$$
\frac{\partial \{ \beta_{j(\mathsf{X})}^2/2 \}}{\partial r_{ij}} = \mathsf{Q}^i_{(\mathsf{X})} r_{ij}; \quad \frac{\partial \mathcal{E}_2}{\partial r_{ij}} = \sum_{\mathsf{X}=1}^{d} \mathsf{Q}^i_{(\mathsf{X})} r_{ij} = \mathsf{Q}^i r_{ij}; \quad \mathsf{Q}^i = \sum_{\mathsf{X}=1}^{d} \mathsf{Q}^i_{(\mathsf{X})}. \tag{27}
$$

Putting the above derivations together and using $\frac{\partial \mathcal{E}}{\partial r_{ij}} = 0$, we have

$$
\sum_{l=a,b,c} \sum_{k=1}^{m} \mathsf{O}^{lk}_{ij} r_{lk} + \lambda_1 \sum_{l=a,b,c} \mathsf{P}^l_{ij} r_{lj} + \lambda_2 \mathsf{Q}^i r_{ij} = \gamma_{ij}; i = a, b, c; \; j = 1, \dots, m. \tag{28}
$$

We therefore arrive at a set of equations linear in $\{r_{ij};\ i = a, b, c;\ j = 1, ..., m\}$ that can be solved easily. After finding the new R, we normalize it using $\mathsf{R} = \mathsf{R}/||\mathsf{R}||_2$.

We now illustrate how to update $\mathsf{F} = [\Rightarrow_i \mathsf{f}_i]$, $\mathsf{S} = [\Rightarrow_i \mathsf{s}_i]$, and $\mathtt{I} = [\Rightarrow_i \mathtt{I}_i]$ with R fixed (or W fixed). First notice that they are only involved in $\mathcal{E}_0$. Moreover, f_i, s_i and $\mathtt{I}_i$ are related with only the image h_i. This becomes the same as the illumination separation problem defined in Section 4 and Appendix-II presents such a recovery algorithm, which also is iterative in nature. After running one iterative step to obtain the updated F, S, and $\mathtt{I}$, we proceed to update R again and this process carries on until convergence.

Appendix-II: Recovering f and s from h given W

This recovery task is equivalent to minimizing the cost function defined as

$$\mathcal{E}_{\mathsf{h}}(\mathsf{f}, \mathsf{s}) \doteq \|\mathtt{I} \circ (\mathsf{h} - \mathsf{W}\,(\mathsf{f} \otimes \mathsf{s}))\|^2 + (\mathbf{1}^{\mathsf{T}}\mathsf{f} - 1)^2, \tag{29}$$

where $\mathtt{I}_{d \times 1}$ indicates the inclusion or exclusion of the pixels of the image h and $\circ$ denotes the Hadamard (element-wise) product. Notice that (29) actually can be easily generalized as a cost function for robust estimation if the L_2 norm $||.||$ is replaced by a robust function, and $\mathtt{I}$ by an appropriate weight function.

The following algorithm is an extension of bilinear analysis, with occlusion embedded. Firstly, we solve the least square (LS) solution f, given s and $\mathtt{I}$.

$$\mathsf{f} = \begin{bmatrix} \mathsf{W}_{\mathsf{f}} \\ \mathbf{1}^{\mathsf{T}} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathtt{I} \circ \mathsf{h} \\ 1 \end{bmatrix}; \quad \mathsf{W}_{\mathsf{f}} \doteq [\Rightarrow_i (\mathsf{T}_i \mathsf{s})]_{d \times m}. \tag{30}$$

where $[.]^{\dagger}$ denotes the pseudo-inverse. Secondly, we solve the LS solution s, given f and $\mathtt{I}$:

$$\mathsf{s} = \mathsf{W}_{\mathsf{s}}^{\dagger}(\mathtt{I} \circ \mathsf{h}); \quad \mathsf{W}_{\mathsf{s}} \doteq [\ [\Rightarrow_i \mathsf{a}_i]\mathsf{f},\ [\Rightarrow_i \mathsf{b}_i]\mathsf{f},\ [\Rightarrow_i \mathsf{c}_i]\mathsf{f}\ ]_{d \times 3} \doteq [\mathsf{Af}, \mathsf{Bf}, \mathsf{Cf}], \tag{31}$$

where $\mathsf{A}_{d \times m} \doteq [\Rightarrow_i \mathsf{a}_i]$, $\mathsf{B}_{d \times m} \doteq [\Rightarrow_i \mathsf{b}_i]$, and $\mathsf{C}_{d \times m} \doteq [\Rightarrow_i \mathsf{c}_i]$ contain the information on the product of albedos and x, y, and z directions of the surface normals, respectively. In the third step, given f and s we update $\mathtt{I}$ as follows[4]:

$$\mathtt{I} = [\ |\mathsf{h} - \mathsf{W}\,(\mathsf{f} \otimes \mathsf{s})| < \eta\ ], \tag{32}$$

where η is a pre-defined threshold.

Note that in Eqs. (30) and (31), additional saving in computation is possible. We can form matrices W_{f}' and W_{s}' and vector h', with a reduced dimension, from W_{f}, W_{s}, and h, respectively, by discarding those rows corresponding to the excluded pixels and applying the primed version in (30) and (31).

For fast convergence, we use the following initial values in our implementation. We estimate s using the algorithm presented in [19] and set $\mathtt{I}$ to exclude those pixels whose intensities are smaller than a certain threshold.

---

[4] This is a Matlab operation which performs an element-wise comparison.