T. Yu and N. Ahuja: Simultaneous estimation of texture map and pseudo illumination from multiple views. In *Texture 2005: Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pp. 19–24, 2005.

Simultaneous Estimation of Texture Map and Pseudo Illumination from Multiple Views

Tianli Yu and Narendra Ahuja
{tianli, ahuja}@vision.ai.uiuc.edu
Beckman Institute and ECE Department
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

We consider the problem of simultaneously estimating surface texture map and the pseudo illumination, which is the combined effect of illumination and surface BRDF, given the object geometry and its calibrated images taken from multiple viewpoints. We derive the bilinear reflection equation that relates the texture albedos and the pseudo illumination to the surface reflectance. The reflection equation takes into account the diffuse/specular reflection, and casting shadows. The ambiguities in estimations are resolved by grouping surface points of the same albedo into a single cluster and performing the estimation for the cluster jointly. A modified mean shift color clustering algorithm is used for this purpose. We estimate the pseudo illumination, the diffuse and specular albedos of each cluster by solving iteratively a system of bilinear equations. Experimental results on synthetic and real objects are provided.

1. Introduction

Reconstructing the illumination and/or surface reflectance properties from images is also known as the inverse rendering problem. In this paper, we propose a multi-view based method to the inverse rendering problem. We show that given the images of the object from multiple viewpoints and the object geometry, we can reconstruct the pseudo illumination, which is the joint effect of illumination and surface Bidirectional Reflectance Distribution Function (BRDF), as well as the surface diffuse and specular texture map.

We assume that the object is under a fixed distant illumination and its spatially varying Bidirectional Texture Function (BTF) can be decomposed as a base BRDF multiplied by a diffuse and specular texture map. Due to the ill-poseness of the problem, we only estimate the pseudo illumination instead of the actual illumination. Our proposed algorithm does not assume uniform surface albedo, and can handle complex illumination and self shadows. Our major contributions include:

a. We derive a bilinear reflection equation that relates the surface reflectance to the diffuse/specular texture albedos and the pseudo illumination.

b. We develop a texture map and pseudo illumination estimation algorithm by solving iteratively the reflection equations. Color space clustering is used to identify surface patches that have the same albedo to remove ambiguities.

Over the years many researches have been proposed to solve the inverse rendering problems. One of the major categories is based on the manipulation of illumination [1, 2, 3]. By controlling different known illumination settings, reliable surface reflectance properties can be extracted. However, these methods are often restricted to a highly controlled laboratory environment.

Instead of varying illumination, one can also varying the camera positions to solve the same problem [4, 5]. The advantages of using such a multi-view based method is that it does not require the control of lighting. Therefore it can be applied in less restricted situations, such as outdoors. In [5], Ramamoorthi and Hanrahan propose a signal processing framework and discuss the conditioning of different inverse rendering problems in a multi-view setup. They mostly focuse on objects with homogeneous BRDFs. Due to the ambiguity in estimation, they also assume the knowledge of illumination when solving for the spatially varying surface texture map. Zhang et al. [6] assume a directional light model, and estimate the illumination and diffuse texture map as well as the surface shape by moving the object with respect to camera and light source. Their approach actually combines the advantages of the photometric stereo methods and the multi-view methods.

Estimating illumination itself is an important problem. An object with uniform or piecewise uniform albedo is usually used to assist the estimation of a complex illuminations [5]. Intensity variations due to shadows of a known object on a uniform or known texture surface can also be used to estimate illumination [7, 8].

Estimating illumination and/or surface texture map given only the geometry and the observed images is ambiguous.

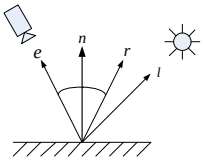


Figure 1: The viewing direction $\vec{e}$, surface normal $\vec{n}$, light direction $\vec{l}$ and the reflected viewing direction $\vec{r}$

This has been observed in several previous researches [2, 3, 5, 9]. To make the problem solvable, different type of clustering methods are suggested. In this paper, we use a modified mean shift clustering algorithm, which is specially designed for a fixed unknown illumination, to group surface patches that has the same reflectance property.

This paper is organized as follows: Section 2 derives our simplified reflection equation for texture map and pseudo illumination on a surface with both diffuse and specular textures. Section 3 presents our texture map and pseudo illumination estimation algorithm. Section 4 describes two experiments and presents estimation results. Section 5 presents conclusions and future work.

2. Reflection Equation for Texture Albedo and Pseudo Illumination

We adopt the distant illumination model and assume that the surface BTF is the product of a texture map and a single BRDF. By merging the effects of BRDF and illumination into the pseudo illumination, we can derive the simplified bilinear reflection equation.

2.1. The Diffuse + Mirror Reflection Equation

Under the distant illumination assumption, the incoming illumination is a 2D spherical function $L(\Theta)$, where $\Theta = (\theta, \phi)$ are the spherical coordinates. Given the illumination L, the reflection of a surface point depends on the surface normal $\vec{n}$ and viewing direction $\vec{e}$. It can be decomposed into the diffuse part R_d and the specular part R_s:

$$R(\vec{n}, \vec{e}) = R_d(\vec{n}, \vec{e}) + R_s(\vec{n}, \vec{e}) \tag{1}$$

R_d can be computed using the lambertian law:

$$R_d(\vec{n}, \vec{e}) = \frac{t_d}{\pi} \int_\Gamma L(\Theta)(\vec{l}(\Theta) \cdot \vec{n}) d\Theta \tag{2}$$

where t_d is the diffuse albedo, $\vec{n}$ is the surface normal, Γ is the upper hemisphere around the normal and $\vec{l}(\Theta)$ is the unit vector of illumination direction.

For R_s, parametric models such as Phong, Cook-Torrance, or isotropic Ward model can be used. However, Ramamoorthi and Hanrahan [5] have shown that it is inherently ambiguous to recover illumination and the Phong exponent (or the equivalent surface roughness parameter in Cook-Torrance model and isotropic Ward model), which controls the BRDF's "blurring" to the illumination. In applications where the separation of BRDF and illumination is not necessary, we can replace them with the pseudo illumination, which is the joint effect of illumination and the BRDF roughness, and assume specular reflection is a perfect mirror. The pseudo illumination can be viewed as a low pass filtered version of the original illumination. This technique is frequently used in the environment map based rendering. We can then write the specular reflectance as:

$$R_s(\vec{n}, \vec{e}) = t_s \tilde{L}(\Theta_r) \tag{3}$$

where $\tilde{L}$ is the pseudo illumination, t_s is the specular albedo and Θ_r are the spherical coordinates of the reflected viewing direction $\vec{r}$ (mirror image of the viewing direction) in Fig. 1. In this paper, we will only discuss the estimation of this pseudo illumination. The actual illumination can be estimated using a separate post-processing step if needed.

As shown in [10], most illumination energy that can affect the lambertian component of the reflectance is in the first 9 spherical harmonics. Therefore, we can replace the illumination $L(\Theta)$ when computing the diffuse component (Eq. 2) with the pseudo illumination $\tilde{L}(\Theta)$, as long as it is has higher frequencies than the first 9 spherical harmonics. After substitute Eq. (2) and (3) into (1), we get:

$$R(\vec{n}, \vec{e}) = \frac{t_d}{\pi} \int_\Gamma \tilde{L}(\Theta)(\vec{l} \cdot \vec{n}) d\Theta + t_s \tilde{L}(\Theta_r) \tag{4}$$

To model the local illumination variations caused by self-shadows, we use a shadow function $s(\Theta)$:

$$s(\Theta) = \begin{cases} 0, p\, is\, in\, shadow\, cast\, by\, light\, \Theta \\ 1, p\, is\, not\, in\, shadow\, cast\, by\, light\, \Theta \end{cases} \tag{5}$$

The shadow function can be integrated into Eq. (4) as

$$R(\vec{n}, \vec{e}) = \frac{t_d}{\pi} \int_\Gamma \tilde{L}(\Theta) s(\Theta)(\vec{l} \cdot \vec{n}) d\Theta + t_s \tilde{L}(\Theta_r) s(\Theta_r) \tag{6}$$

In this formulation, we do not consider any inter-reflection effect in this paper. Also, due to the use of pseudo illumination instead of the actual illumination, we might lose sharp shadow boundaries when using Eq. (6).

2.2. Parameterizing the Texture Map and Pseudo Illumination

The shadow function in Eq. (6) is known given the object geometry. The observed surface reflection color in Eq. (6) can also be computed by projecting a point on the surface to the calibrated images while taking into account the self-occlusion. To represent the pseudo illumination, we decompose it using spherical harmonics:

$$\tilde{L}(\Theta) = \sum_{l=0}^{N_L} \sum_{m=-l}^{l} h_{lm} Y_{lm}(\Theta) \tag{7}$$

where N_L is the maximum order of harmonic series we use, h_{lm} are the spherical harmonic coefficients and Y_{lm} are the real basis functions.

We also tesselate the object surface using a triangular mesh and associate uniform diffuse and specular texture albedos within each triangle. Substitute Eq. (7) into (6) and exchange the summation and integration, we get:

$$R^{(i)}(\vec{e}) = t_d^{(i)} \sum_{m,l} h_{ml} C_{ml}^{(i)} + t_s^{(i)} \sum_{m,l} h_{ml} D_{ml}^{(i)}(\vec{r}) \quad (8)$$

where $C_{ml}^{(i)}$ are the diffuse coefficients and $D_{ml}^{(i)}(\vec{r})$ are the specular coefficients:

$$C_{ml}^{(i)} = \frac{1}{\pi} \int_{\Gamma} s^{(i)}(\Theta) Y_{ml}(\Theta)(\vec{l} \cdot \vec{n})\, d\Theta \quad (9)$$

$$D_{ml}^{(i)}(\vec{r}) = Y_{ml}(\Theta_r) s^{(i)}(\Theta_r) \quad (10)$$

Equation (8) can also be written in a matrix product form:

$$R^{(i)}(\vec{e}) = (t^{(i)})^T * \mathbf{A}^{(i)}(\vec{r}) * h \quad (11)$$

Where $t^{(i)} = [t_d^{(i)} t_s^{(i)}]^T$ is the texture vector, $h = [...h_{ml}...]^T$ is the illumination vector, and

$$\mathbf{A}^{(i)}(\vec{r}) = \begin{bmatrix} \cdots & C_{ml}^{(i)} & \cdots \\ \cdots & D_{ml}^{(i)}(\vec{r}) & \cdots \end{bmatrix} \quad (12)$$

is the reflection coefficient matrix. Eq. (11) becomes a bilinear equation in $t^{(i)}$ and h. Each surface triangle observed from a viewpoint gives one such bilinear equation. Estimating the texture map and the pseudo illumination amounts to solving a system of such bilinear equations.

It is known that for diffuse objects estimating both illumination and texture using only images as input is not well-posed. There are more unknowns than independent equations, which results in ambiguities in the estimation. Similar ambiguity exists in the case of a surface with both diffuse and specular component. To overcome these ambiguities, we will group the under-constrained triangles to solve a single system of bilinear equations.

3. Texture Map and Pseudo Illumination Estimation Algorithm

A schematic of our algorithm is shown in Fig. 2. Our input is a set of calibrated images of the object and its 3D shape. Based on the object geometry, we are able to compute the reflection matrix in Eq. (11) for each surface triangle. A modified mean shift color clustering algorithm is used to group triangles into clusters to reduce the estimation ambiguity. Based on the clustering information, our bilinear system solver iteratively solve for the texture map and the pseudo illumination. The estimated pseudo illumination is sent back to the clustering algorithm to refine the partition of triangles. The result is used to start another iteration of the bilinear system solver. The following subsections will discuss the three major blocks in the diagram.

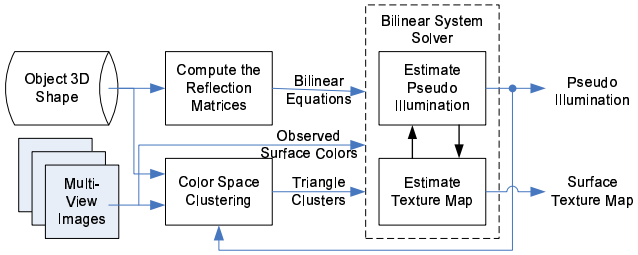


Figure 2: Flow diagram of our texture map and pseudo illumination estimation algorithm

3.1. Compute the Reflection Matrix

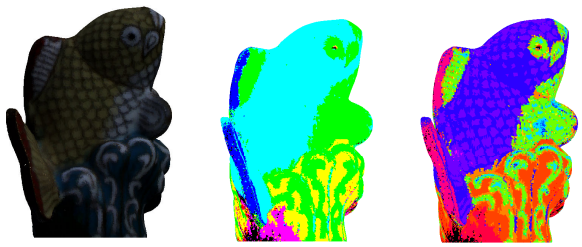
The diffuse and specular coefficients of the reflection matrix in (9) and (10) encode the shading and shadowing effects of different spherical harmonic components to each triangle. However, to compute these coefficients exactly based on the scene geometry is difficult. One central problem is to compute the shadow function $s^{(i)}(\Theta)$. We evaluate this function only on a set of sample directions uniformly distributed on the unit shpere. For each sample direction, we use OpenGL to render the surface mesh orthographically onto a plane perpendicular to that direction. The shadow function is set to 1 for all the visible triangles in that direction, and set to 0 for the rest of the triangles. In computing the specular coefficient $D_{ml}^{(i)}(\vec{r})$, we use the value of the closest sample direction to Θ_r to approximate $s^{(i)}(\Theta_r)$. For the diffuse coefficient $C_{ml}^{(i)}$, we build another triangular mesh from these sample directions on the unit sphere and compute the approximate integration on the mesh surface using:

$$C_{ml}^{(i)} = \sum_f a_f \cdot \frac{1}{3} \sum_{k=1}^{3} s^{(i)}(\Theta_{fk}) Y_{ml}(\Theta_{fk})(\vec{l}_{fk} \cdot \vec{n}^{(i)}) \quad (13)$$

where a_f is the area of the f_{th} triangle on the unit sphere mesh and Θ_{fk} is one of its three vertices. A spherical mesh with 2324 triangles and 1164 sample directions are used in our experiments. We use spherical harmonics up to the 10_{th} order to represent the illumination, which result in an 121×1 illumination vector h and the reflection matrix $\mathbf{A}^{(i)}(\vec{r})$ is of size 2×121.

3.2. Mean Shift Based Color Space Clustering

We use a modified mean shift algorithm [11] to identify clusters of similar diffuse color. As noted in [9], the diffuse colors of a single material have only 1-D intensity variations under a fixed single-color illumination. These colors



(a) (b) (c)

Figure 3: Color space clustering results. (a) Mesh triangles shown using their median colors extracted from multiple views. (b) Mean shift based color clustering result using the angular distance (6 clusters). (c) Clustering result using the euclidean distance after illumination information is estimated (23 clusters). In (b) and (c) each cluster is coded with a unique color.

lie along a line in RGB space that passes through the origin. The angle between two RGB color vectors v_1 and v_2 can be used as the illumination invariant distance measure.

$$d_1(v_a, v_b) = cos^{-1}(v_a \cdot v_b) \tag{14}$$

According to the mean shift algorithm, an initial color sample v_0 can be iteratively updated and will converge to one of the peaks of the underlying kernel density function. We adopt the gaussian kernel and use this update equation:

$$v_{j+1} = \frac{\sum_{i=1}^{N} c_i \cdot exp\left[-\left(d_1(v_j, c_i)/\sigma\right)^2\right]}{\sum_{i=1}^{N} exp\left[-\left(d_1(v_j, ci)/\sigma\right)^2\right]} \tag{15}$$

Where c_i is the diffuse color of i_{th} triangle and σ is the bandwidth parameter. Note that our distance measure is invariant to the norm of the color vector v_j and c_i, and we should normalize v_j and c_i to unit length in each iteration. The diffuse color of each triangle (c_i) is estimated by choosing the median value of its color observed from all the visible viewpoints (Fig. 3(a)).

We build a 3D histogram of the diffuse colors and the center of each nonzero bin is used as an initial sample (v_0) to compute the converging density peak. Each triangle is then assigned to the nearest peak based on the angular distance d_1 in (14). Fig. 3 (b) shows a color clustering result from one of our experiments with a fish model.Each cluster is coded with a different color in the figure.

The above method does not distinguish those triangles with the same diffuse color but different albedo values. This can be seen by comparing Fig. 3 (a) and (b) where the black lines are merged with the yellow areas because their angular distances are very small according to (14). To deal with these cases, we start a new round of clustering after we get an estimation of the illumination from the bilinear system solver. The diffuse colors are divided by their "diffuse illumination" ($\sum_{m,l} h_{ml} C_{ml}^{(i)}$ in Eq. 8) to get the corrected diffuse albedos. We then use the same clustering algorithm on these corrected albedos with the exception that the angular distance is replaced by the euclidean distance in RGB space:

$$d_2(v_a, v_b) = ||v_a - v_b|| \tag{16}$$

Actually we only need to perform the new clustering within each cluster obtained from the angular distance, to further partition the clusters. This saves a lot of computation and increases the convergence rate. The new clusters are send to the bilinear system solver for another iteration of estimation until the clustering result converges. Fig. 3(c) shows the final clustering result with 23 clusters.

3.3. The Bilinear System Solver

Given the reflection matrix, the observed color value $R^{(i)}(\vec{e})$ of the i_{th} triangle from a viewpoint gives a bilinear equation (Eq. 11) with unknown $t^{(i)}$ and h. Equations from the same triangle imaged from different viewpoints and triangles that belong to the same cluster can be merged together, which gives a system of bilinear equations:

$$\begin{cases} \cdots & & \\ R^{(i)}(\vec{e_j}) & = & (t^{(k)})^T * \mathbf{A}^{(i)}(\vec{r_j}) * h \\ \cdots & & \end{cases} \tag{17}$$

$$i \in C_k, j = 1, 2, 3 \cdots$$

where C_k is the set of triangles belong to the k_{th} cluster, and j is the camera view index.

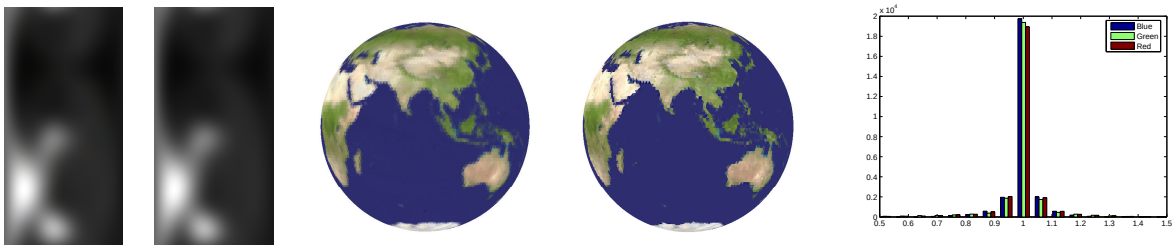
There is no closed form solution to the least squares problem of such an over-determined bilinear system, but we can derive the normal equations of this system, and solved them iteratively [12]. This is equivalent to iterating between fixing one set of parameters (illumination vector h or texture vector t^{C_k}) and solving a linear least squares problem for the other.

4. Experimental Results

We performed two experiments to test our algorithm. The first experiment shows that there is no ambiguity in estimating an arbitrary pseudo illumination and a specular texture map. The second experiment shows the advantage of clustering the surface triangles into distinct albedo clusters in the joint estimation.

4.1. Texture Map and Pseudo Illumination Estimation for Specular Only Objects

In this experiment, we try to estimate the specular texture map (diffuse component assumed to be 0) and the pseudo illumination from a set of multiple view images of a sphere.



(a) (b) (c) (d) (e)

Figure 5: Estimation results for the earth data set. (a) Estimated pseudo illumination as latitude$[0, \pi]$-longitude$[0, 2\pi]$ plot (b) Ground truth filtered illumination after DC normalized to 1 (c) Estimated specular texture map (d) Ground truth specular texture map (e) Histogram of the ratio between estimated specular albeo to the ground truth albedo for all the triangles.

Figure 4: Light probe data used for synthesis (left) and 2 of the 24 synthesized images of the earth data set.

Figure 6: 3 of the 25 input images of the fish data set.

We render 24 different images of the sphere with a pure specular BRDF modulated by a texture map of the earth using the Phong model. The Phong exponent used is 10 for the entire sphere. The illumination is a set of 40 directional sources obtained by clustering the light probe data (http://www.debevec.org/Probes/). These light sources are converted to white color during rendering. Fig. 4 shows the light probe image and two of the synthesized images.

Fig. 5 shows the estimated texture map and pseudo illumination after 20 iterations of our algorithm. The ground truth illumination, after it is filtered by the same Phong BRDF and normalized to have unit DC component, is shown in Fig. 5(b). The ground truth specular texture map is in Fig. 5(d). We also plot the histogram of the ratios between estimated and ground truth specular albedo of each triangle in Fig. 5(e).

The histogram of the albedo ratios shows a distribution with a sharp peak around 1, and the estimated pseudo illumination closely match the filtered ground truth illumination. The estimated texture map appears a little blurred, mostly due to the truncation errors in the spherical harmonic expansion of the illumination, and the approximation in the reflection equation. This experiment shows that Eq. (11) can be used to extract the arbitrary specular texture map and pseudo illumination without ambiguity.

4.2. Texture Map and Pseudo Illumination Estimation for Sparsely Colored Objects

We apply our estimation algorithm to a set of real images of a ceramic fish captured from multiple views. This data set is also used in [13] for surface light field extraction and compression. We have the camera calibration information and the scene geometry was obtained by structured light range scan. The original data set has more than 600 images and we select 25 of them that are distributed around the object. Fig. 6 shows three of the input images we used.

The median color rendered fish model and the clustering results have already been shown in Fig. 3. The diffuse and specular albedos of the clusters as well as the pseudo illumination are estimated by solving iteratively the bilinear system for each cluster. Fig. 7 shows the result of the estimated pseudo illumination and the diffuse and specular albedo maps after 15 bilinear solver iterations and 2 clustering iterations. The actual illumination is composed of two point light sources, which is clearly shown in the estimated illumination distribution. The recovered texture maps where each triangle is estimated independently is shown in Fig. 7(d,e) for comparison. We can see clearly the effect of estimation ambiguity in the results. The specular texture map shows strong specular component only at those places where highlights are present in the observed images, and the tail of the fish has very bright diffuse albedos, which is caused by the specular highlight.

(a) (b) (c) (d) (e)

Figure 7: Estimation results for the fish data set. (a) Estimated pseudo illumination (b) Estimated diffuse texture map (c) Estimated specular texture map (d) Estimated diffuse texture map without color clustering (e) Estimated specular texture map without color clustering.

Figure 8: Ground truth image in a novel view (left), synthesized image using the pseudo illumination and texture map (middle), and synthesized image from the same view but with illumination rotated by 40 degrees vertically (right).

The estimated pseudo illumination and texture map allow us to render novel views, and modify the illumination or geometry to create animations. Fig. 8 shows the synthesized image of the fish model from a novel view along with the ground truth image. One problem of the image is the blurring of the specular highlights. Again this might be due to the approximation in our reflection equation, the 3D shape error, the truncation error of the spherical harmonics, and the incorrect clustering. We also render an image with the illumination rotated around the z-axis. Our work is different from [13], where only specular components are adjusted to match the new illumination. We separate the pseudo illumination and the diffuse/specular texture map and can synthesize various shading and shadowing effects.

5. Conclusion and future work

We have considered the problem of simultaneously estimating the surface texture map and the pseudo illumination using a bilinear reflection equation. The reflection equation takes into account diffuse and specular reflection, and casting shadows. We use a modified mean shift color clustering algorithm to remove estimation ambiguities. We are going to investigate how to use the estimated texture map and pseudo illumination to improve the shape estimation.

References

[1] Y. Sato, M. D. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," *Proc. SIGGRAPH 97*, pp. 379-388, 1997.

[2] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich and H. P. Seidel "Image-based reconstruction of spatial appearance and geometric detail," *ACM Trans. Graphics*, vol. 22, no. 2, pp. 234-257, 2003.

[3] A. Hertzmann and S. M. Seitz, "Shape and Materials by Example: A Photometric Stereo Approach," *Proc. CVPR 2003*, pp. I:533-540, June 2003

[4] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, "Inverse global illumination: Recovering reflectance models of real scenes from photographs," *Proc. SIGGRAPH 99*, pp. 215-224, 1999.

[5] R. Ramamoorthi, and P. Hanrahan, "A Signal-Processing Framework for Inverse Rendering," *Proc. SIGGRAPH 2001*, pp 117-128, 2001.

[6] L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz, "Shape and Motion under Varying Illumination: Unifying Structure from Motion, Photometric Stereo, and Multi-view Stereo," *Proc. ICCV 2003*, pp. 618-625, 2003.

[7] I. Sato, Y. Sato, and K. Ikeuchi, "Illumination from Shadows," *IEEE Trans. PAMI*, vol. 25, no. 3, Mar. 2003.

[8] Y. Li, S. Lin, H. Lu, and H. Y. Shum, "Multiple-cue illumination estimation in textured scenes," *Proc. Intl. Conf. of Computer Vision 2003*, pp. 1366-1373, 2003.

[9] H. Unten, and K. Ikeuchi, "Color Alignment in Texture Mapping of Images under Point Light Source and General Lighting Condition," *Proc. CVPR 2004*, pp. I:234-239, 2004.

[10] R. Basri, and D. W. Jacobs "Lambertian Reflectance and Linear Subspaces," *IEEE Trans. PAMI*, vol. 25, no. 2, pp. 218-233, Feb 2003.

[11] D. Comaniciu, and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. PAMI*, vol. 24, no. 5, pp. 603-619, May 2002.

[12] S. Cohen, and C. Tomasi, "Systems of Bilinear Equations," *Technical Report STAN-CS-TR-97-1588*, Computer Science Department, Stanford University, Apr. 1997.

[13] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface Light Fields for 3D Photography," *Proc. SIGGRAPH 2000*, pp. 287-296, 2000.