On the Relations between Three Methods for Representing 3D Surface Textures under Arbitrary Illumination Directions

Junyu Dong
Department of Computer Science
Ocean University of China, China
dongjunyu@ouc.edu.cn

Mike Chantler
Texture Lab
Heriot-Watt University, UK
M.J.Chantler@hw.ac.uk

Abstract

Representing the appearances of surfaces illuminated from different directions has long been an active research topic. While many representation methods have been proposed, the relationships between the different representations have been less well researched. These relationships are important, as they provide (a) an insight into the different capabilities of the surface representations, and (b) a means by which they may be converted to common computer graphics application formats. In this paper, we introduce a single mathematical framework and use it to express three commonly used surface texture relighting representations: Surface Gradients, Polynomial Texture Maps (PTM) and Eigen base images. The framework explicitly reveals the relations between the three methods, and from this we propose a set of conversion methods.

1. Introduction

Real-world surface textures commonly consist of a spatial variation of reflectance properties (for example variation in colour albedo) combined with rough surface geometry. Variation of illumination can therefore produce dramatic changes their appearance. For example, Figure 1 shows two example images of a 3D surface texture illuminated from two directions. The difference due to varying illumination presents challenges in both computer vision and computer graphics [1,2,3,4]. It is therefore important to extract surface representations of the sample texture under arbitrary illumination directions.

Representing the appearances of an object illuminated from different directions has long been an active research topic. In the past two decades, many methods have been proposed [5,11,12,15,16,17]. Some were specially developed for extracting surface texture representations from a set of pre-captured images [2,3,8,9]. Meanwhile, the relations between different methods or representations also attracted much attention [8,9,10,11,12]. These relations can reveal the connections between different representations and provide optional formats for the use in computer graphics applications. However, no previous work has been published on the explicit connections in mathematics between surface gradient maps [4], Polynomial Texture Maps [8] and eigen images [13, 17], which is the main objective of this paper.

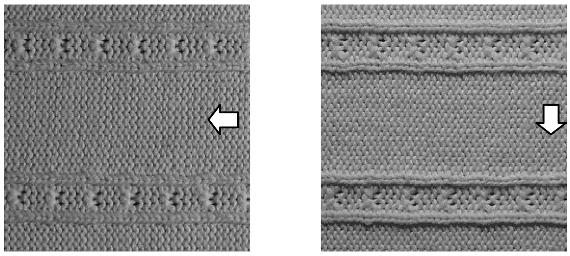

Figure 1. Two images of a 3D surface texture imaged under differing illumination. The texture is from an Aran jumper.

In this paper, we investigate the relations between three commonly used surface representations: *Gradient*[4], *Polynomial Texture Maps(PTM)*[8] and *Eigen*[13,17]. Based on previous work in [14,5], we introduce a single mathematical framework that can summarize the three surface representations. We further propose conversion methods between different representations. The framework and conversion methods present a novel view on the three representations. We believe that no other work has been published on this subject.

The rest of this paper is organized as follows. Section 2 presents the mathematical framework that can describe the relations between the three representations. In section 3, we propose conversion methods between different representations. Experimental results are shown in section 4. Finally, we conclude our work in section 5.

2. The mathematical framework and three surface representation methods

We first present a mathematical framework that summarizes the common properties of the three methods. We then introduce the *Gradient*, *PTM* and *Eigen* methods under the framework.

2.1. The mathematical framework

The framework uses Singular Value Decomposition (SVD) to analyze the image intensity matrix. SVD is based on the following theorem of linear algebra:

Any $m' \times n'$ matrix whose number of rows m' is greater than or equal to its number of columns n', can be written as the product of an $m' \times n'$ column-orthogonal matrix $\mathbf{U}$, an $n' \times n'$ diagonal matrix $\mathbf{W}$ with positive or zero elements, and the transpose of an $n' \times n'$ orthogonal matrix $\mathbf{V}$. That is

$$\mathbf{M} = \mathbf{UWV^T} \qquad (1)$$

where $\mathbf{U^T U} = \mathbf{V^T V} = \mathbf{E}$ and $\mathbf{E}$ is the unit matrix. The elements on the diagonal of $\mathbf{W}$ are called singular values. The *pseudoinverse* of $\mathbf{M}$ is expressed as

$$\mathbf{M^{-1}} = \mathbf{VW^{-1}U^T} \qquad (2).$$

For a group of linear equations $\mathbf{M} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{x} = (x_1, x_2, \ldots, x_{n'})^T$ and $\mathbf{b} = (b_1, b_2, \ldots, b_{n'})^T$ are two vectors, we can solve $\mathbf{x}$ according to equation (2)

$$\mathbf{x} = \mathbf{M^{-1}b} = \mathbf{VW^{-1}U^T b}$$

The mathematical framework is based on the analysis of the image data matrix, which contains all images captured under multiple illumination directions. Assume each image has *m* pixels and we have total of *n* images per sample texture. To simplify notations, let i_{jk} denote the intensity value of pixel *j* in the k^{th} image, where $1 \le j \le m$ and $1 \le k \le n$. Then we write all image intensity data i_{jk} into an $m \times n$ matrix

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} & \ldots & i_{1n} \\ i_{21} & i_{22} & \ldots & i_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ i_{m1} & i_{m2} & \ldots & i_{mn} \end{bmatrix}$$

where each column represents an image captured under a certain illumination direction and each row represents the intensity values of a pixel location under different illumination directions.

The framework expresses the image data matrix as a product:

$$\mathbf{I} = \mathbf{M_1 M_2} \qquad (3)$$

where $\mathbf{M_1}$ and $\mathbf{M_2}$ are two matrices. $\mathbf{M_1}$ is the surface relighting representation matrix that we want to extract. Thus, if we know $\mathbf{M_2}$ and assume a certain reflectance/lighting model, we can solve $\mathbf{M_1}$ by using SVD. The *Gradient* and *PTM* methods fall into this category. If we do not know $\mathbf{M_2}$ or do not want to assume any reflectance/lighting model, we can directly use SVD to analyze the image data matrix $\mathbf{I}$ and obtain $\mathbf{M_1}$ and $\mathbf{M_2}$, as will be shown in the *Eigen* method.

Thus, the relighting process can be expressed as a product of the surface representation matrix $\mathbf{M_1}$ and a vector $\mathbf{c}$ related to the required illumination direction:

$$\mathbf{i} = \mathbf{M_1 c} \qquad (4)$$

where $\mathbf{i} = (i_1, i_2, \ldots, i_m)^T$ is the image data vector and $i_1, i_2, \ldots, i_m$ are pixel values.

2.2. The *Gradient* method

According to Lambert's law, surface gradient and albedo maps can be used to represent 3D surface textures for relighting. We call this method *Gradient*.

At a pixel location *(x,y)*, the Lambertian reflectance function is expressed as

$$i(x, y) = \lambda\alpha\mathbf{n} \cdot \mathbf{l} \qquad (5)$$

where:

$i(x, y)$ is the intensity of an image pixel

λ is the incident intensity to the surface

α is the albedo value of the Lambertian reflection

$\mathbf{l}$ is the unit illumination vector at position *(x, y)* and can be expressed as

$$\mathbf{l} = (l_x, l_y, l_z)^T = (\cos\tau\sin\sigma,\ \sin\tau\sin\sigma,\ \cos\sigma)^T$$

τ and σ are the illumination tilt and slant angles

$\mathbf{n}$ is the normalized surface normal at position *(x, y)* and can be expressed as

$$\mathbf{n} = (n_x, n_y, n_z)^T = \left(\frac{-p}{\sqrt{p^2+q^2+1}}, \frac{-q}{\sqrt{p^2+q^2+1}}, \frac{1}{\sqrt{p^2+q^2+1}}\right)^T$$

p and q are the partial derivatives of the surface height function in the *x* and *y* directions respectively.

In this paper, we assume λ as a constant and merge it with albedo α. We simply use ρ to represent $\lambda\ \alpha$. Thus, the data matrix **I** can be expressed as:

$$\mathbf{I}=\mathbf{ANL} \qquad (6)$$

where:

$$\mathbf{A} = \begin{bmatrix} \rho_1 & & & 0 \\ & \rho_2 & & \\ & & \ddots & \\ 0 & & & \rho_m \end{bmatrix}$$ is the surface albedo matrix;

$\mathbf{N} = (\mathbf{n_1}, \mathbf{n_2}, \ldots, \mathbf{n_m})^{\mathbf{T}}$ is the surface normal matrix;

$\mathbf{L} = (\mathbf{l_1}, \mathbf{l_2}, \ldots, \mathbf{l_m})$ is the lighting matrix.

We further define a new matrix $\mathbf{N_a}$ which is the product of the surface normal matrix $\mathbf{N}$ and the albedo matrix **A**:

$$\mathbf{N_a} = \mathbf{AN}.$$

Thus we can simply express the image data matrix as

$$\mathbf{I} = \mathbf{N_a L} \qquad (7).$$

Comparing equation (7) with equation (3), we have:

$$\mathbf{M_1} = \mathbf{N_a} \text{ and } \mathbf{M_2} = \mathbf{L}.$$

The matrix $\mathbf{N_a}$, which contains surface gradient and albedo information, is the unknown.

It is trivial to obtain $\mathbf{N_a}$ by using SVD. We first decompose the lighting matrix as $\mathbf{L} = \mathbf{U_L W_L V_L}^{\mathbf{T}}$.

Then we have

$$\mathbf{N_a} = \mathbf{IL^{-1}} = \mathbf{IV_L W_L}^{-1}\mathbf{U_L}^{\mathrm{T}}.$$

By relighting $\mathbf{N_a}$, which contains surface gradient maps scaled by albedo, we can generate new images under arbitrary illumination. The Lambertian model is used again for relighting:

$$\mathbf{i} = \mathbf{N_a l}$$

where $\mathbf{i} = (i_1, i_2, \ldots, i_m)^T$ is the image data vector and $\mathbf{l} = (\cos\tau\sin\sigma,\ \sin\tau\sin\sigma,\ \cos\sigma)^T$ is the lighting vector.

The advantage of the *Gradient* method is that the surface gradient and albedo maps are compatible with computer graphics programming or packages for rendering [18, 7].

2.3. The *PTM* method

The PTM method uses Polynomial Texture Maps as surface representations for relighting [8]. Malzbender et. al. proposed a luminance model that employs a quadratic function of the lighting vector to capture variations due to self-shadowing and interreflections. It is based on the Lambertian assumption and uses the first two elements of the unit lighting vector to form a new six-dimensional lighting vector

$$\mathbf{l_{ptm}} = (l_x^2, l_y^2, l_z l_y, l_x, l_y, 1)^T = (\cos^2\tau\sin^2\sigma,\ \sin^2\tau\sin^2\sigma,\ \cos\tau\sin\tau\sin^2\sigma, \cos\tau\sin\sigma, \sin\tau\sin\sigma, 1)^T$$

The image data matrix is expressed as

$$\mathbf{I} = \mathbf{A_{ptm} L_{ptm}} \qquad (8)$$

where
$$\mathbf{A_{ptm}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{m5} & a_{m6} \end{bmatrix}$$

$$\mathbf{L_{ptm}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ l_{y1} & l_{y2} & \cdots & l_{yn} \\ l_{x1} & l_{x2} & \cdots & l_{xn} \\ l_{x1}l_{y1} & l_{x2}l_{y2} & \cdots & l_{xn}l_{yn} \\ l_{y1}^2 & l_{y2}^2 & \cdots & l_{yn}^2 \\ l_{x1}^2 & l_{x2}^2 & \cdots & l_{xn}^2 \end{bmatrix}$$

Each row of matrix $\mathbf{A_{ptm}}$ represents six coefficients of the luminance model at each pixel location. These coefficients are stored as spatial maps and called Polynomial Texture Maps (PTMs). We call $\mathbf{A_{ptm}}$ the PTM matrix and $\mathbf{L_{ptm}}$ is the lighting matrix. Although the lighting matrix contains quadratic terms, it can be pre-calculated offline. In accordance with equation (3) in the mathematical framework, $\mathbf{A_{ptm}}$ and $\mathbf{L_{ptm}}$ are equivalent to $\mathbf{M_1}$ and $\mathbf{M_2}$ respectively.

Since the image data matrix **I** and the lighting matrix $\mathbf{L_{ptm}}$ are known, we can use SVD to solve the over-determined system (8) and obtain the PTM matrix $\mathbf{A_{ptm}}$. This is similar to solving for surface gradient representations described in section 2.2. Given an illumination direction and recalling equation (4), the relit image can be expressed as

$$\mathbf{i} = \mathbf{M_1 c} = \mathbf{A_{ptm} l_{ptm}},$$

where $\mathbf{i} = (i_1, i_2, \ldots, i_m)^T$ is the image data vector and $\mathbf{l_{ptm}}$ is the PTM lighting vector. Thus, the relighting is achieved by linear combinations of PTMs.

Since relighting is implemented using a linear combination of pre-computed quadratic terms, the PTM method is suitable for real-time rendering applications in graphics hardware [8].

2.4. The *Eigen* methods

Eigen-based methods are widely used by many researchers to model the effect due to varying illumination e.g. [13, 17]. These methods have the advantage that an assumption concerning surface reflectance is not required. We apply SVD to generate base images in eigen-space. The image data matrix is expressed as

$$\mathbf{I} = \mathbf{U_I W_I V_I}^{\mathbf{T}}$$

Each column in $\mathbf{U_I}$ therefore is an eigen vector of $\mathbf{II^T}$ corresponding to the singular value in $\mathbf{W_I}$. $\mathbf{U_I}$ is used to construct eigen base images and $\mathbf{V_I}^{\mathbf{T}}$ contains coefficients for linear combinations. We can write $\mathbf{W_I} = diag(w_1, w_2, ..., w_n)$, where w_i is the singular value of the image data matrix $\mathbf{I}$ and $w_i \geq w_{i+1}$.

Since singular values decrease rapidly and the first few eigenvectors account for most of the information, we approximate the original $\mathbf{W_I}$ by

$$\widehat{\mathbf{W}}_\mathbf{I} = diag(w_1, w_2, \ldots, w_k, 0, \ldots, 0),$$

where k is the number of singular values that we want to keep. We then obtain an approximation of the image data matrix $\mathbf{I}$ that can be expressed as

$$\widehat{\mathbf{I}} = \mathbf{U_I} \widehat{\mathbf{W}}_\mathbf{I} \mathbf{V_I}^\mathbf{T} \qquad (9)$$

Recalling equation (3) in the mathematical framework we can write $\mathbf{M_1} = \mathbf{U_I} \widehat{\mathbf{W}}_\mathbf{I}$. We let $\mathbf{M_1}$ be an $m \times k$ matrix, since the last $n-k$ columns of $\mathbf{U_I} \widehat{\mathbf{W}}_\mathbf{I}$ are zeroes. Similarly, we create a $k \times n$ matrix $\mathbf{M_2}$, which only contains the first k rows of $\mathbf{V_I}^\mathbf{T}$, because the last $n-k$ rows of $\mathbf{V_I}^\mathbf{T}$ can be assigned zeroes due to the fact that the last $n-k$ diagonal elements of $\widehat{\mathbf{W}}_\mathbf{I}$ are equal to zeroes. Thus, we obtain a set of k base images in eigen-space which are the k columns of $\mathbf{M_1}$. These base images are called eigen base images. Matrix $\mathbf{M_2}$ provides the coefficients for the linear combination of eigen base images to produce those original images in $\mathbf{I}$. We write

$$\mathbf{I} = (\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n) = \mathbf{M_1 M_2} \qquad (10)$$

where $\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n$ are image data vectors that represent those original images captured under different illumination directions.

If we use coefficients that differ from those in $\mathbf{M_2}$, the linear combinations of these base images allow us to generate new images under new illumination directions. Thus, we can use these eigen base images as representations of 3D surface textures for relighting.

3. Conversions between surface representations

The mathematical framework provides deep sight into the three surface representation methods. Since there are mathematical connections between these three methods, we can convert one surface representation to another one by using SVD method. The conversion can provide alternative ways for relighting/rendering in computer graphics applications. However, these conversions are also based on the least squares sense due to the assumptions and the radical properties of the SVD method. We use previously introduced denotations to show how these representation formats can be converted between each other.

3.1. The *Gradient* and *PTM* methods

In [8], it has been shown that surface normals can be converted to PTMs in the lease squares sense based on Approximation Theory. It is however, more obvious if we use matrix expressions and the SVD method. According to equation (7) and equation (8), we obtain

$$\mathbf{I} = \mathbf{N_a L} = \mathbf{A_{ptm} L_{ptm}}.$$

Thus we have:

$$\mathbf{N_a} = \mathbf{A_{ptm} L_{ptm} L^{-1}} \text{ and } \mathbf{A_{ptm}} = \mathbf{N_a L L_{ptm}}^{-1}.$$

We may use $\mathbf{C_1} = \mathbf{L_{ptm} L^{-1}}$ and $\mathbf{C_2} = \mathbf{L L_{ptm}}^{-1}$ to denote the coefficient matrices for the conversions between the *Gradient* and *PTM* representations. Obviously, the coefficient matrices can be pre-calculated. Thus, we have:

$$\mathbf{N_a} = \mathbf{A_{ptm} C_1} \text{ and } \mathbf{A_{ptm}} = \mathbf{N_a C_2}.$$

3.2. The *Gradient* and *Eigen* methods

By equation (7) and equation (10), we obtain

$$\mathbf{I} = \mathbf{N_a L} = \mathbf{M_1 M_2}.$$

Hence, $\mathbf{N_a} = \mathbf{M_1 M_2 L^{-1}}$ and $\mathbf{M_1} = \mathbf{N_a L M_2}^{-1}$, where $\mathbf{M_1}$ can contains eigen base images

We use $\mathbf{C_3} = \mathbf{M_2 L^{-1}}$ and $\mathbf{C_4} = \mathbf{L M_2}^{-1}$ to denote the coefficient matrices for the conversions between the *Gradient* and *Eigen* representations. Then we obtain

$$\mathbf{N_a} = \mathbf{M_1 C_3} \text{ and } \mathbf{M_1} = \mathbf{N_a C_4}$$

3.3 The *PTM* and *Eigen* method

By equation (8) and equation (10), we obtain

$$\mathbf{I} = \mathbf{A_{ptm} L_{ptm}} = \mathbf{M_1 M_2}.$$

Hence, we have $\mathbf{A_{ptm}} = \mathbf{M_1 M_2 L_{ptm}}^{-1}$ and $\mathbf{M_1} = \mathbf{A_{ptm} L_{ptm} M_2}^{-1}$. We use $\mathbf{C}_5 = \mathbf{M_2 L_{ptm}}^{-1}$ and $\mathbf{C_6} = \mathbf{L_{ptm} M_2}^{-1}$ to denote the coefficient matrices for the conversions between the *PTM* and *Eigen* representations. Finally, we obtain

$$\mathbf{A_{ptm}} = \mathbf{M_1 C_5} \text{ and } \mathbf{M_1} = \mathbf{A_{ptm} C_6}.$$

4. Results

We used texture samples from the PhoTex texture database in our experiments [6]. For each texture, we use 36 images captured under different illumination directions to extract the surface representations. We then apply the conversion methods to obtain new surface representations according to section 3. For the comparison purpose, in Figure 2 to Figure 4 we show the relit images with two illumination directions produced by different surface representations, including both original and converted. Each figure contains two texture samples (a plaster ripple surface and a rock surface). It can be seen that the proposed conversion methods produced reasonable results.

Relighting original *Gradient*	**Relighting converted *Gradient* from *PTM***	**Relighting converted *Gradient* from *Eigen***

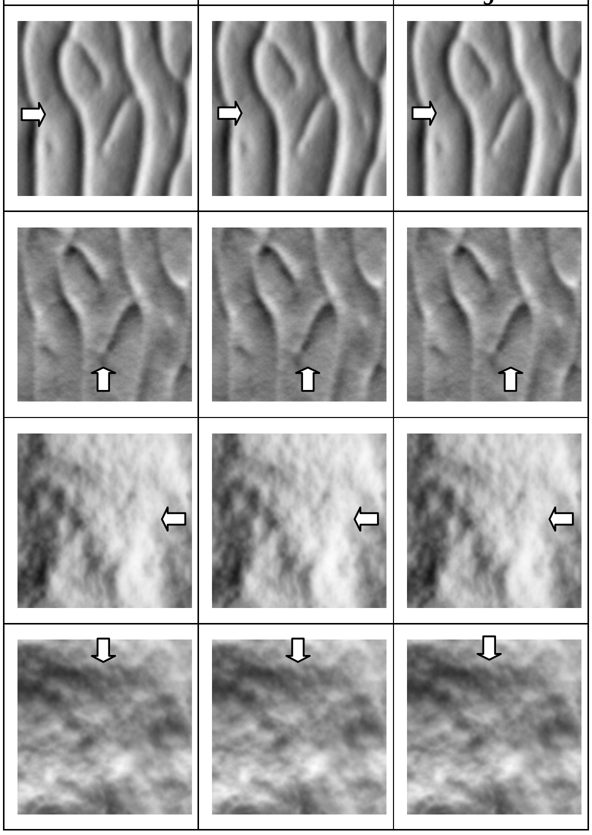

Figure 2 Relighting results produced by the *Gradient* representations. Images in the first column are produced by relighting original surface gradient and albedo maps extracted from texture samples images. Images in the second and third column are produced by relighting the converted gradient and albedo maps from Polynomial Texture Maps and Eigen base images respectively. The block arrows show the illumination directions.

5. Conclusion

In this paper we have defined the relationships between three commonly used surface representations in a single mathematical framework. This provides significant insight as to the properties of the different methods. Furthermore, we have proposed a set of methods that can be used to convert between these different representations – providing improved flexibility and economies in terms of storage and bandwidth.

Relighting original *PTM*	**Relighting converted *PTM* from *Gradient***	**Relighting converted *PTM* from *Eigen***

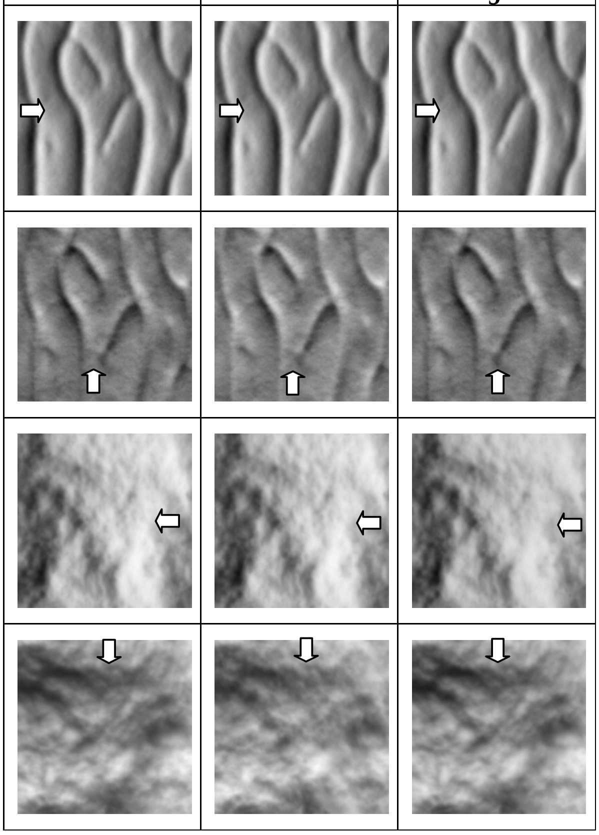

Figure 3 Relighting results produced by the *PTM* representations. Images in the first column are produced by relighting original PTMs extracted from texture samples images. Images in the second and third column are produced by relighting the converted PTMs from *Gradient* and *Eigen* base images respectively. The block arrows show the illumination directions.

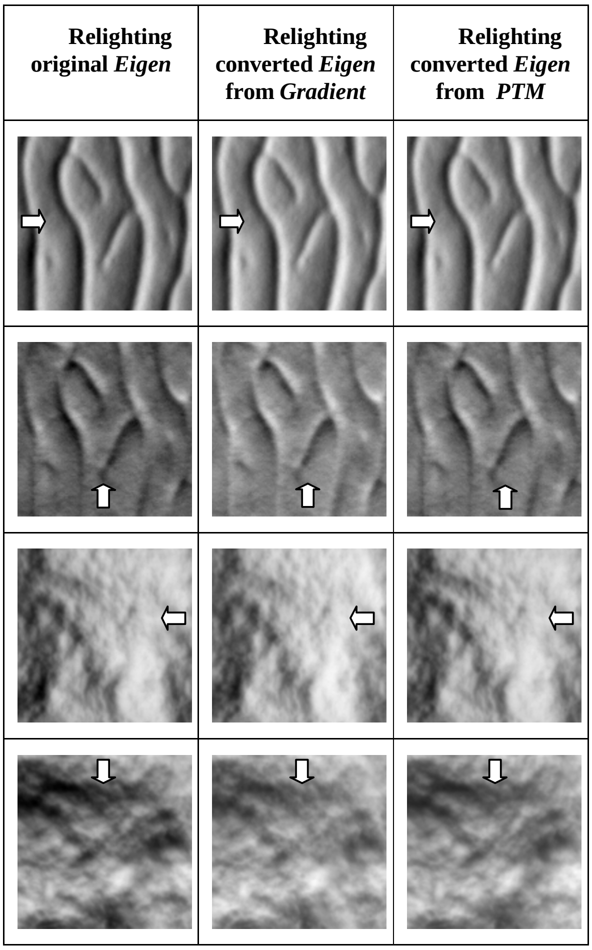

Figure 4 Relighting results produced by the Eigen base images. Images in the first column are produced by relighting original eigen base images extracted from texture samples images. Images in the second and third column are produced by relighting the converted eigen images from Gradient and PTMs respectively. The block arrows show the illumination directions.

References

[1] M. J. Chantler. Why illuminant direction is fundamental to texture analysis, IEE Proceedings: Vision, Image and Signal Processing, Vol. 142, No. 4, August 1995, pp199-206.

[2] K. J. Dana, B. Van Ginneken, S. K. Nayar and J. J. Koenderink. Reflectance and Texture of Real-World Surfaces. ACM Transactions on Graphics, 1999 Vol. 18, No. 1, pp. 1-34.

[3] T. Leung and J. Malik. Representing and Recognizing the Visual Appearance of Materials Using Three-dimensional Textons. International Journal of Computer Vision, 2001, vol 43, Number 1, pp 7-27.

[4] R. Woodham. Analysing images of curved surfaces. Artificial Intelligence, 1981. 17:117-140.

[5] J. Dong. Three-dimensional surface texture synthesis. PhD thesis. School of Mathematical and Computer Sciences, Heriot-Watt University, UK. 2003.

[6] Http://www.macs.hw.ac.uk/texturelab/database/PhoTex

[7] D. Burschka, D. Cobzas, Z. Dodds, G. Hager, M. Jagersand and K. Yerex. IEEE Virtual Reality 2003 tutorial 1: Recent Methods for Image-based Modelling and Rendering. March 2003.

[8] T. Malzbender, D. Gelb and H. Wolters. Polynomial Texture Maps. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001, pp. 519-528.

[9] M. Ashikhmin and P. Shirley. Steerable illumination textures. ACM Transactions on Graphics, v.21, No. 1, 2002, pp. 1-19.

[10] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p.p. 117 – 128, 2001.

[11] R. Ramamoorthi. Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 24, Issue 10, 2002, pp. 1322-1333.

[12] R. Basri, and D. Jacobs. Lambertian reflectance and linear subspaces. In Proceedings 8th IEEE International Conference on Computer Vision. Vol. 2, 2001, pp. 383-390.

[13] R. Epstein, P.W. Hallinan and A.L. Yuille. 5±2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In Proceedings of the Workshop on Physics-Based Modeling in Computer Vision, 1995, pp. 108-116.

[14] J. Dong, and M. Chantler. Capture and synthesis of 3D surface texture. Accepted for publication in 2004/5. International Journal of Computer Vision: special issue on texture analysis and synthesis.

[15] F. E. Nicodemus, J.C. Richmond and J. J. Hsai. Geometrical considerations and nomenclature for reflectance. U.S. Dept. of Commerce, National Bureau of Standards Monograph 160, October 1977.

[16] P. Lalonde and A. Fournier. A wavelet representation of reflectance functions. Visualization and Computer Graphics, IEEE Transactions on, Vol. 3, Issue: 4, October-December, 1997, pp. 329-336.

[17] Z. Zhang. Modeling geometric structure and illumination variation of a scene from real images", In Proceedings of International Conference on Computer Vision, 1998, pp. 1041-1046.

[18] M. Robb, A. D. Spence, M.J. Chantler and M. Timmins. Real-time per-pixel rendering of bump-mapped textures captured using photometric stereo. In Proceedings of Vision, Video, and Graphics 2003, Bath, UK, 10-11 July, pp. 79-87.