Chapter 4

Surface Texture Representations for Relighting

4.1. Introduction

In chapter 3, we proposed a framework for the synthesis and relighting of 3D surface textures. The framework can combine 2D texture synthesis algorithms and relighting techniques to synthesise new texture images under arbitrary illumination directions.

The first stage of the framework abstracts a 3D surface texture representation from a set of sample images. This normally comprises two phases: (1) converting the set of pre-recorded images into surface relighting representations, and (2) rendering these representations according to desired lighting conditions. It is impractical to discuss the two phases separately. The goal of this chapter is therefore to study a set of candidate methods for extracting representations of the 3D surface texture sample **and** to investigate the relighting of these representations.

We first propose the criteria for selecting the methods. Then we present a detailed review on candidate methods. According to our criteria, we select five low dimensional representations, which can be extracted from a set of images captured by a fixed camera and varied illumination directions. These methods are listed below.

3I: This method uses three images of the sample texture taken at an illumination slant angle of 45° and tilt angles of 0°, 90° and 180° [Shashua1992].

Gradient: The second method uses surface gradient and albedo maps derived using photometric stereo [Woodham1981 and Rushmeier1997].

PTM: This approach uses Polynomial Texture Maps (PTM), due to Malzbender et. al. [Malzbender2001].

Eigen3: The fourth method uses the first three eigen base images [Epstein1995].

Eigen6: This is identical to the previous method except that it uses the first six base images.

Thus, the first half of this chapter selects five techniques for future study. The second half presents the results of a quantitative comparison of these approaches. We use two comparison metrics, namely *Ability-of-reconstruction* and *Ability-of-prediction*, to perform the analysis. Twenty-three real textures are tested for each method. We calculate the normalised *root mean-squared* (*rms*) errors by comparing relit images generated by each method with original real images. Based on the results, we show that *Eigen6* produces the smallest normalised *rms* errors while *3I* produces the largest. Those of *Gradient*, *PTM* and *Eigen3* vary, depending on the texture.

This chapter is organised as follows. Section 4.2 proposes the criteria for selecting 3D surface texture representation and relighting methods. Section 4.3 presents a detailed review on available methods of representing and relighting 3D surface textures. Section 4.4 describes the selected five methods. Section 4.5 presents two approaches to quantitatively assess the five methods. Finally we conclude the work of this chapter in section 4.6.

4.2. Criteria

The choice of surface relighting representations has a significant impact both on the computational requirements and the quality of final results. According to the main objective of this thesis, we set the criteria for selecting the methods as follows:

1. *Practicality of physical data capture*

   We would like the sample data to be captured in an inexpensive way, e.g. using off-the-shelf digital cameras, and the synthesised representations to be capable of being rendered in real-time on current desktop machines.

2. *Low dimensionality of representations*

   The relighting representations of the sample 3D surface texture should consist of as few components as possible.

3. *Compatibility of representations with graphic systems*

   The surface relighting representations should be compatible with computer graphics packages or be able to be programmed into modern graphics hardware. For computer graphics packages, the common input is surface bump or height maps and albedo maps. For graphics hardware, it is preferable to use texture units and register combiners to speed up rendering by linear combining surface representations. Modern graphics hardware and APIs provides a number of texture units and register combiners that can efficiently process the relighting representation maps and perform linear combinations [Burschka2003]. The real-time rendering can be achieved by using these hardware acceleration facilities.

4. *Capability of dealing with complex reflectance including shadows and specularities*

   Most real-world surface textures have complex reflectance properties. We would like the representation to be able to represent these more complex functions.

In addition to the four criteria for selecting surface representation methods, we also need a criterion to assess the performance of different methods. Ideally, the relit images produced by different surface representations should be identical to the original images. This is however, not possible in practice. We therefore set the criterion for the assessment to be a measure of how close the relit results are to the original images. We use the normalised *rms* error as the numerical metric.

4.3. A detailed review and selection of surface representation and relighting methods

The goal of this detailed review is to survey available surface representation methods using the criteria introduced in the previous section. Five methods are selected based on the review.

In 1977, Nicodemus *et. al.* introduced Bidirectional Reflectance Distribution Functions (BRDF) to accurately characterise surface reflectance properties [Nicodemus1977]. The BRDF is the ratio of the reflected intensity in the exitant direction to the incident energy per unit area along the incident direction. With full BRDF data and surface geometry information, images of the sample surface under arbitrary illumination can be produced. Dana *et. al.* further proposed the Bidirectional Texture Function (BTF) by allowing the BRDF to vary spatially across a surface location [Dana1999a]. The CUReT image database is constructed to describe BTFs and has included 61 sample textures with various reflectance properties. However, the measurement of BRDF or BTF is expensive and time-consuming, because the BRDF and BTF depend on both the chemical composition and the roughness condition of the surface. Meanwhile, BTFs imply high dimensionalities due to numerous images required (e.g. the CUReT BTF database contains 205 unregistered images for each sample). Although the 3D textons are introduced to characterise the essential information of BTFs, they still need 960-dimentional vectors to represent the sample surface [Leung2001]. The reconstruction of BTFs from 3D textons is expensive [Tong2002]. Several other techniques approximate BRDFs by projection into basis functions [Lalonde1997 and Lafortune1997].

Estimating surface representations using reflectance models only requires a relatively small number of sample images, which are inexpensive to obtain [Woodham1981, Horn1989, Nayar1990, Kay1995, Rushmier1997, Saito1996, Lin2000, Ikeuchi1991, Lu1995, Sato1997, Ramamoorthi2001 and Nishino2001]. Traditional Photometric Stereo techniques use three or more images to estimate surface gradient and albedo maps based on the Lambertian model [Woodham1981 and Horn1989]. Integration techniques can be further used to obtain the depth information or the height map from surface gradient maps [Coleman1982 and Frankot1988]. In [Shashua1990], Shashua proves that three images captured under linearly independent illumination directions can represent a non-shadowed Lambertian surface. Nayar *et. al.* estimate the surface shape and reflectance of a hybrid model by photometric sampling [Nayar1990]. Saito *et. al.* recovers the parameters of the Phong model by fitting the pixel intensities into a sine curve

[Saito1996]. Based on the experiments, Kay and Caelli conclude that it is more difficult to estimate geometric and material parameters of a specular surface because specularities can only be captured using certain lighting and viewing angles [Kay1995]. Accordingly, many approaches make assumptions concerning the reflectance properties on the sample surface, e.g. uniform surface roughness [Saito1996 and Lin1999].

In general, the above techniques are more practical to implement if the reflectance models are accurate enough to describe the sample. The estimated surface geometric and reflectance representations lie in low-dimensional space and are compatible with graphics systems. For example, a Lambertian surface can be effectively represented in 3-dimensional space (surface gradient and albedo maps) or even 2-dimensional space (surface height and albedo maps) [Woodham1981 and Horn1989], and the Nayar model needs a 7-dimensional representation [Kay1995]. Furthermore, the albedo map and surface normals, which can be obtained from surface gradient maps, are standard inputs for rendering the Lambertian reflectance models or the Lambertian component in reflectance models [Blinn1978, Phong1975 and Cook1982]. However, many reflectance models only characterise certain classes of surfaces. The accuracy of the extracted representations therefore depends on whether the models are capable of accurately describing the reflectance properties of the sample surface [Koudelka2001].

Without using a reflectance model, many mathematically based methods have been developed to represent images of a surface illuminated from different directions. Huang employs Fourier Series to approximate the pixel values of a set of images under different illumination directions [Huang1984]. The number of harmonics, or the dimensionality of the surface representation, depends on the reflectance complexity. Epstein *et. al.* suggest that five eigen basis images (plus or minus two) can be effectively used to represent arbitrary lighting for many different objects, although specular spikes and cast-shadows require more base images [Epstein1995]. The relighting is achieved by a linear combination. Basri and Jacobs use 9-demension spherical harmonics to represent a convex Lambertian surface under distant and isotropic lighting [Basri2001]. The Polynomial Texture Maps proposed in [Malzbender2001] use a 6-dimensional representation to capture the colour variance for a surface exhibiting shadows and interreflections with varied illumination directions. Instead of using a physically based reflectance model, a quadratic function is employed to relight a Lambertian surface. In [Ramamoorhi2001], spherical harmonics are used to estimate isotropic BRDFs based on certain assumptions, including known geometry, distant illumination and curved objects without interreflections. Ashikhmin uses a set of 49 steering basis functions to relight bumpy surfaces, which exhibit shadows and interreflections [Ashikhmin2001]. McAllister *et. al.* use the Lafortune BRDF representations, which is capable of representing Fresnel reflection, off-specular peak and retro-reflection, to perform real-time rendering in graphics hardware [Lafortune1997 and McAllister2002].

In theory, these mathematically based methods can be seen as data approximation functions. Thus, the dimensionality is related to the accuracy required. Normally using more base images achieves more accurate relighting results. The linearly based representations, such as eigen base images, spherical harmonics, Polynomial Texture Maps and steering basis functions, can be effectively programmed into graphics systems, as the relighting is performed in linear space.

More recently, several image-based relighting (rendering) techniques were proposed and showed realistic relighting results for scenes with complex reflectance properties [Matusik2002, Koudelka2001, Wong2002 and Lin2002]. These methods require a great number of sample images for relighting and even complex hardware set-up. Matusik *et. al.* built a system that can acquire and render surface reflectance fields under varying illumination from arbitrary viewpoints [Matusik2002]. They captured 53136 images using an array of cameras and lights, and perform a weighted linear combination to generate new images. Wong *et. al.* propose the plenoptic illumination function that can be also used to support relighting and view interpolation [Wong2002]. They need to employ compression techniques to reduce the storage space. Lin *et. al.* define the reflected irradiance field as the relighting representation [Lin2002]. They show that the method can produce accurate relighting results on surfaces with complex reflectance properties e.g. steel and anisotropic surfaces, but their relighting representation requires 240MB to 320MB storage space. All these methods have the advantage that they do not assume a particular reflectance model. However, they have extremely high dimensionalities due to the number of images required for interpolation. Since common graphics cards designed for desktop PCs can not provide unlimited memory, these techniques are less practical for synthesis and real-time relighting applications on desktop PCs.

To summarise:

We have reviewed typical surface representation and relighting methods based on the criteria introduced in section 4. 2. These methods have different merits and drawbacks under different criteria. In general, the surface geometric and material parameters estimated using reflectance models are the most compact representations and compatible with graphics systems. The drawback is that existing models can not represent complex reflectance. Representations in linear sub-spaces, such as eigen base images, Polynomial Texture Maps (PTM), steering base functions and spherical harmonics, can be used for representing surfaces with complex reflectance, but specularities require more base images. Although the Bidirectional Texture Functions (BTF) and some image-based relighting/rendering techniques are able to produce accurate relighting results, they are too expensive to be used for the purpose of this thesis. Figure 4.3.1 shows the analysis of typical surface representations using different criteria.

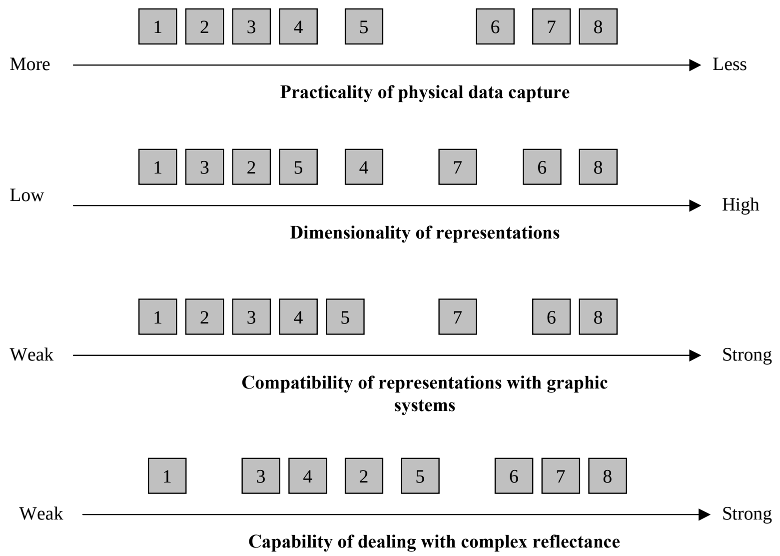


Figure 4.3.1 Different representations v.s. criteria. (1)Estimated surface geometry and reflectance parameters using reflectance models [Woodham1981]; (2) Eigen-based methods [Epstein1995]; (3) Polynomial Texture Maps

(PTM)[Malzbender2001]; (4) Steering basis functions [Ashikhmin2002]; (5) Spherical harmonics [Basri2001]; (6)Opacity hulls[Matusik2002]; (7)3D texons [Leung2001]; (8) BRDF/BTF [Dana1999a].

Since our main concern in this chapter is to select inexpensive surface representation approaches, we need to trade-off the expense and performance between different methods and criteria. We have chosen five methods that can produce efficient relighting representations. The first two methods—**3I** and **Gradient**—are based on the Lambertian reflectance model: the **3I** method uses three images of the sample texture taken at an illumination slant angle of 45° and tilt angles of 0°, 90° and 180° [Shashua1992], while the **Gradient** method uses surface gradient and albedo maps derived from photometric stereo techniques [Woodham1981 and Rushmeier1997]. We also select the **PTM** method that employs Polynomial Texture Maps (PTM) to represent a surface exhibiting shadows and interreflections under different illumination directions [Malzbender2001]. Finally, we select the **Eigen3** and **Eigen6** methods, which use the first three and six eigen base images respectively, to represent a surface with complex reflectance.

We summarise the selected methods in Table 4.3.1 and provide further details in the next section.

Table 4.3.1. Summary of the selected surface representations vs. criteria

	Practical to obtain?	**Compatible of using linear combinations in graphics hardware?**	**Capable of capturing shadows?**	**Capable of dealing with specularity?**	**Dimensionality**
3I	Yes	Yes	**No**	**No**	3
Gradient	Yes	Yes	**No**	**No**	3
PTM	Yes	Yes	Yes	**No**	6
Eigen3	Yes	Yes	Yes	**No**	3
Eigen6	Yes	Yes	Yes	Yes	6

4.4. The selected methods

4.4.1. Mathematical framework

In section 4.3, we selected five methods, which all use a set of images as input in order to extract surface representations for relighting. In this section, we propose a mathematical framework that can be used to describe and compare these methods. This framework summarises the common properties of the five methods—the point of departure is the known image intensity matrix, which contains all images of a sample texture captured under different illumination directions. The lighting matrix, which contains lighting elements, is also analysed when a reflectance or lighting model is assumed.

We first briefly introduce Singular Value Decomposition (SVD), which is commonly used in matrix analysis. It is the appropriate tool for analysing a mapping from one vector space into another vector space, possibly with a different dimension. Most systems of simultaneous linear equations fall into this category. Thus, SVD can be used to for solve most *linear least squares* problems, e.g. an over-constrained linear or well-constrained equation group [Press1988]. 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

$$\mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathbf{T}} \qquad (4.4.1)$$

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

$$\mathbf{M}^{-1} = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^{\mathbf{T}}.$$

For a group of linear equations $\mathbf{M} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{x} = (x_1, x_2, \dots, x_{n'})^T$ and $\mathbf{b} = (b_1, b_2, \dots, b_{n'})^T$ are two vectors, we can solve $\mathbf{x}$ according to equation (4.4.1)

$$\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^{\mathbf{T}}\mathbf{b} \qquad (4.4.2)$$

The mathematical framework is based on the analysis of the image data matrix. The image data matrix contains all images 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 $1 \le k \le n$. If we use two-dimensional co-ordinates *(x, y)* to denote the pixel location, then index *j* can be calculated by using $j = (x-1) * w + y$, where *w* is the image width. Then we write all image intensity data i_{jk} into an $m \times n$ matrix

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} & \cdots & i_{1n} \\ i_{21} & i_{22} & \cdots & i_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ i_{m1} & i_{m2} & \cdots & i_{mn} \end{bmatrix} \qquad (4.4.3)$$

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:

$$\mathbf{I} = \mathbf{M}_1\mathbf{M}_2 \qquad (4.4.4)$$

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 according to (4.4.2). The *Gradient* and *PTM* methods fall into this category. For the *3I* method, $\mathbf{M_1}$ is simply the original image data matrix **I**. If we do not know $\mathbf{M_2}$ or do not want to assume any reflectance/lighting model, we can directly use SVD to analyse the image data matrix **I** and obtain $\mathbf{M_1}$ and $\mathbf{M_2}$, as will be shown in the eigen-based methods (*Eigen3* and *Eigen6*).

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

$$\mathbf{i} = \mathbf{M_1 c} \qquad (4.4.5)$$

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.

4.4.2. Lambertian methods--*3I* and *Gradient*

At a pixel location, the Lambertian reflectance function is expressed as

$$i(x, y) = \lambda \rho \mathbf{n} \cdot \mathbf{l} \qquad (4.4.6)$$

where:

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

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 is the tilt angle of illumination

σ is the slant angle of illumination

$\mathbf{n}$ is the normalised 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 and defined by:

$$p(x,y)=\frac{\partial s(x,y)}{\partial x}, \qquad q(x,y)=\frac{\partial s(x,y)}{\partial y}$$

$s(x,y)$ is the surface height function

If the incident intensity to the texture surface λ is constant—as assumed in this thesis, we can treat λ as a scalar and merge it with albedo α. To simplify notations, we use ρ to represent $\lambda\,\alpha$. Thus, the image data matrix **I** can be expressed as:

$$\mathbf{I}=\mathbf{ANL} \qquad (4.4.7)$$

where:

$$\mathbf{A}=\begin{bmatrix} \rho_1 & & & 0 \\ & \rho_2 & & \\ & & \ddots & \\ 0 & & & \rho_m \end{bmatrix}$$

is the surface albedo matrix;

$$\mathbf{N}=(\mathbf{n}_1,\mathbf{n}_2,\ldots,\mathbf{n}_m)^{\mathrm{T}}=\begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ \vdots & \vdots & \vdots \\ n_{mx} & n_{my} & n_{mz} \end{bmatrix}$$

is the surface normal matrix;

$$\mathbf{L}=(\mathbf{l}_1,\mathbf{l}_2,\ldots,\mathbf{l}_m)=\begin{bmatrix} l_{1x} & l_{2x} & \ldots & l_{mx} \\ l_{1y} & l_{2y} & \ldots & l_{my} \\ l_{1z} & l_{2z} & \ldots & l_{mz} \end{bmatrix}$$

is the lighting matrix.

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

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

This matrix contains the set of “scaled surface normals” [Drbohlav2002]. Thus we can simply express the image data matrix as

$$\mathbf{I}=\mathbf{N_a L} \qquad (4.4.8).$$

It is convenient to use equation (4.4.8) to introduce Lambertian based methods—*3I* and *Gradient*.

The *3I* method—a linear combination of three *photometric images*

Shashua shows that an image of a convex object can be represented as a linear combination of three base images under the assumption of Lambertian reflectance [Shashua1992]. We call this method *3I*. The three base images can be obtained by positioning the light at three linearly independent directions. These three base images are called *photometric images*. Thus, if we recall the equation (4.4.5), we only need to decide the vector **c**, which contains the coefficients used for the linear combination. This can be achieved by using (4.4.8) and calculating the inverse lighting matrix.

Since we have three known linearly independent lighting vectors, and we can express it using the lighting matrix

$$\mathbf{L} = (\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3) = \begin{bmatrix} l_{1x} & l_{2x} & l_{3z} \\ l_{1y} & l_{2y} & l_{3z} \\ l_{1z} & l_{2z} & l_{3z} \end{bmatrix}.$$

Accordingly, we can also write the image data matrix as an $m \times 3$ matrix

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ \vdots & \vdots & \vdots \\ i_{m1} & i_{m2} & i_{m3} \end{bmatrix},$$

which is also the surface representation matrix $\mathbf{M_1}$. Thus, according to (4.4.8), we have $\mathbf{IL}^{-1} = \mathbf{N_a}$, where $\mathbf{L}^{-1}$ can be easily calculated because it is a non-singular square matrix. Note SVD can also be used here to obtain $\mathbf{L}^{-1}$.

Given any illumination direction with the corresponding lighting vector

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

the new image $\mathbf{i}$ can be expressed as

$$\mathbf{i} = \mathbf{IL}^{-1}\mathbf{l} \qquad (4.4.9)$$

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

By equation (4.4.5) in the mathematical framework, we have $\mathbf{c} = \mathbf{L}^{-1}\mathbf{l}$. Then (4.4.9) becomes

$$\mathbf{i} = \mathbf{M_1 c} = \mathbf{Ic} \qquad (4.4.10)$$

which means an image under a given lighting vector can be expressed as a linear combination of three images. The vector **c** is called the coefficient vector.

In our case, we capture three images with illumination tilt angles separated by 90°. Thus, the illumination is provided at a common slant (45° in our case) and at tilt angles of 0°, 90° and 180°. The reason for using these three tilt angles is that they simplify the inversion of **L** for use in photometric stereo [McGunnigle1998] and provide near optimum results [Spence2003]. We calculate the inverse lighting vector $\mathbf{L}^{-1}$ and express the coefficient vector in terms of the illumination tilt angle τ and the illumination slant angle σ of the new image:

$$\mathbf{c} = (c_1, c_2, c_3)^T \quad (*)$$

where $c_1 = \dfrac{\cos\tau\sin\sigma}{2\sin 45^\circ} - \dfrac{\sin\tau\sin\sigma}{2\sin 45^\circ} + \dfrac{\cos\sigma}{2\cos 45^\circ}$

$$c_2 = \frac{\sin\tau\sin\sigma}{\sin 45^\circ}$$

$$c_3 = \frac{\cos\sigma}{2\cos 45^\circ} - \frac{\cos\tau\sin\sigma}{2\sin 45^\circ} - \frac{\sin\tau\sin\sigma}{2\sin 45^\circ}.$$

Thus, the new image is calculated using (*) and (4.4.10).

The *Gradient* method—using surface gradient and albedo maps as the surface representation for relighting

According to Lambert's law (4.4.6), surface gradient and albedo maps can be used to represent 3D surface textures for relighting. We call this method *Gradient*. Traditional photometric stereo techniques [Woodham1981] use three images to estimate the gradient and albedo maps of a Lambertian surface. Additional images lead to an over-constrained system, which may be solved using least squares techniques (e.g. SVD) to provide potentially more accurate solutions. The *Gradient* method uses 36 images under different known illumination angles for each texture in the image database. Thus, in equation (4.4.8)

$$\mathbf{I} = \mathbf{N_a}\mathbf{L},$$

the image data matrix **I** becomes a known $m\times 36$ matrix and the lighting matrix **L** is a known $3\times m$ matrix. Comparing equation (4.4.8) with equation (4.4.4), 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 albedo map and the surface gradient maps, which can be calculated from $\mathbf{N_a}$, or the displacement map, which can be further generated from surface gradient maps, are compatible with computer graphics programming or packages for rendering [Robb2003 and Burschka2003].

To summarise:

Based on the assumption of Lambertian reflectance, the *3I* method uses three *photometric images* to represent 3D surface texture for relighting. A linear combination of the three images can produce new images under arbitrary illuminant directions. This provides the simplest way to represent a 3D surface texture for relighting. However, this method can only achieve accurate results for unshadowed Lambertian surfaces.

The *Gradient* method uses surface gradient and albedo maps to represent a Lambertian surface for relighting. The surface gradient and albedo maps are generated by using SVD to solve an over-determined system. This surface representation method only has three dimensions and provides the most common format used in computer graphics programming or packages.

4.4.3. The *PTM* method

The *PTM* method uses Polynomial Texture Maps [Malzbender2001] as surface representations for relighting. 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 lighing vector $\mathbf{l} = (l_x, l_y, l_z)^T = (\cos\tau\sin\sigma,\ \sin\tau\sin\sigma,\ \cos\sigma)^T$ to form a new six-dimensional lighting vector

$$\begin{aligned}\mathbf{l}_{\mathbf{ptm}} &= (l_x{}^2, l_y{}^2, l_z l_y, l_x, l_y, 1)^T \\ &= (\cos^2\tau\sin^2\sigma,\ \sin^2\tau\sin^2\sigma,\ \cos\tau\sin\tau\sin^2\sigma, \cos\tau\sin\sigma, \sin\tau\sin\sigma, 1)^T\end{aligned}$$

The image data matrix is expressed as

$$\mathbf{I} = \mathbf{A}_{\mathbf{ptm}}\mathbf{L}_{\mathbf{ptm}} \qquad (4.4.11)$$

where $\mathbf{A}_{\mathbf{ptm}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{m5} & a_{m6} \end{bmatrix}$ and $\mathbf{L}_{\mathbf{ptm}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ l_{y1} & l_{y2} & \cdots & l_{yn} \\ l_{x1} & l_{x2} & \cdots & l_{xn} \\ l_{x1}l_{y1} & l_{x2}l_{y2} & \cdots & l_{xn}l_{yn} \\ l_{y1}{}^2 & l_{y2}{}^2 & \cdots & l_{yn}{}^2 \\ l_{x1}{}^2 & l_{x2}{}^2 & \cdots & l_{xn}{}^2 \end{bmatrix}$

Each row of matrix $\mathbf{A}_{\mathbf{ptm}}$ ($a_1 - a_6$) represents six coefficients of the luminance model at each pixel location. These coefficients are stored as spatial maps and called Polynomial Texture Maps (PTM). We call $\mathbf{A}_{\mathbf{ptm}}$ the PTM matrix and $\mathbf{L}_{\mathbf{ptm}}$ is the lighting matrix. Although the lighting matrix contains quadratic terms, it can be pre-calculated offline. In accordance with equation (4.4.4) in the mathematical framework, $\mathbf{A}_{\mathbf{ptm}}$ and $\mathbf{L}_{\mathbf{ptm}}$ are equivalent to $\mathbf{M_1}$ and $\mathbf{M_2}$ respectively.

Since the image data matrix **I** and the lighting matrix $\mathbf{L}_{\mathbf{ptm}}$ are known, we can use SVD to solve the over-determined system (4.4.11) and obtain the PTM matrix $\mathbf{A}_{\mathbf{ptm}}$. This is similar to solving for surface gradient representations described in section 4.4.2. Given an illumination direction and recalling equation (4.4.5), the relit image can be expressed as

$$\mathbf{i} = \mathbf{M_1 c} = \mathbf{A}_{\mathbf{ptm}}\mathbf{l}_{\mathbf{ptm}}$$

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

To summarise:

Polynomial Texture Maps (PTM) can be used to represent 3D surface textures with self-shadowing and interreflection under varied illumination. They are actually the six coefficient maps of a quadratic luminance model based on Lambertian assumption. PTMs can be obtained by solving an over-determined system using SVD.

Since relighting is implemented using a linear combination of pre-computed quadratic terms, they are suitable for real-time rendering applications in graphics hardware.

4.4.4. The eigen-based methods (*Eigen3* and *Eigen6*)

Eigen based methods are widely used by many researchers to model the effect due to varying illumination e.g. [Dana1999, Epstein1995, Nishino2001 and Zhang1998a]. These methods have the advantage that an assumption concerning surface reflectance is not required. Based on experiments, Epstein *et. al.* in [Epstein1995] suggested that five base images (plus or minus two) can be effectively used to represent arbitrary lighting for many different objects. They concluded that this approach could accurately model Lambertian surfaces with specular lobes, while specular spikes, small shadows and occludes can be treated as residuals. Naturally both the specularity and the complexity of surface geometry increases the number of base images required.

We have elected to use 3 base images and 6 base images in eigen-space to represent 3D surface texture for relighting. Three eigen base images can represent 3D surface texture with Lambertian reflectance, while six eigen base images can further capture certain specularities and shadows [Epstein1995]. We apply SVD to generate base images in eigen-space. The image data matrix is expressed as

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

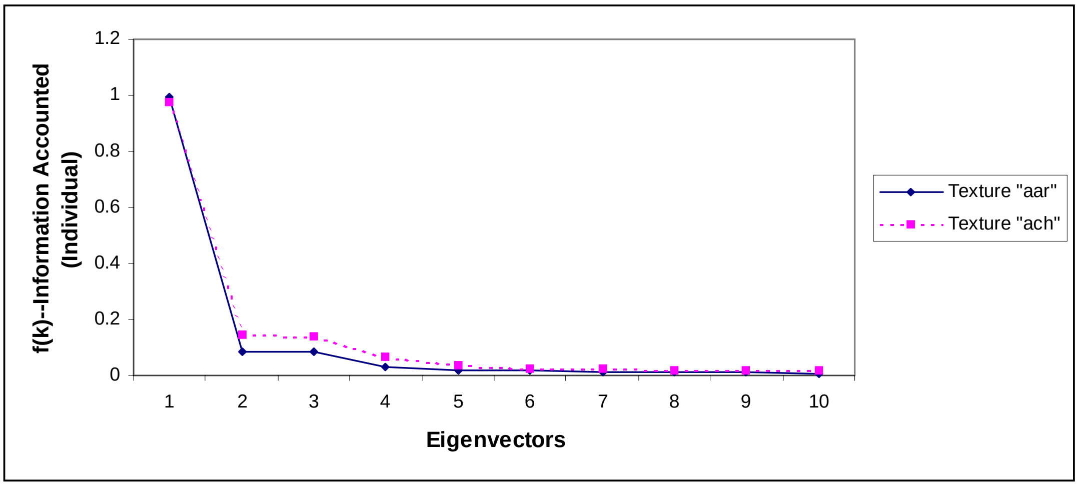
Each column in $\mathbf{U_I}$ therefore is an eigen vector of $\mathbf{II}^{\mathbf{T}}$ corresponding to the singular value in $\mathbf{W_I}$. $\mathbf{U_I}$ is used to construct eigen base images and $\mathbf{V_I}^{\mathbf{T}}$ contains coefficients for linear combinations. We can write $\mathbf{W_I} = diag(w_1, w_2, ..., w_n)$, where w_i is the singular value of the image data matrix $\mathbf{I}$ and $w_i \geq w_{i+1}$. An important property of $\mathbf{W_I}$ is that the singular values decrease dramatically. If we use the following definitions as the measurement for information accounted for by individual eigenvector [Epstein1995]:

$$f_{indi}(k) = \sqrt{w_k{}^2 \Big/ (\sum_{i=1}^{n} w_i{}^2)} \quad (4.4.12)$$

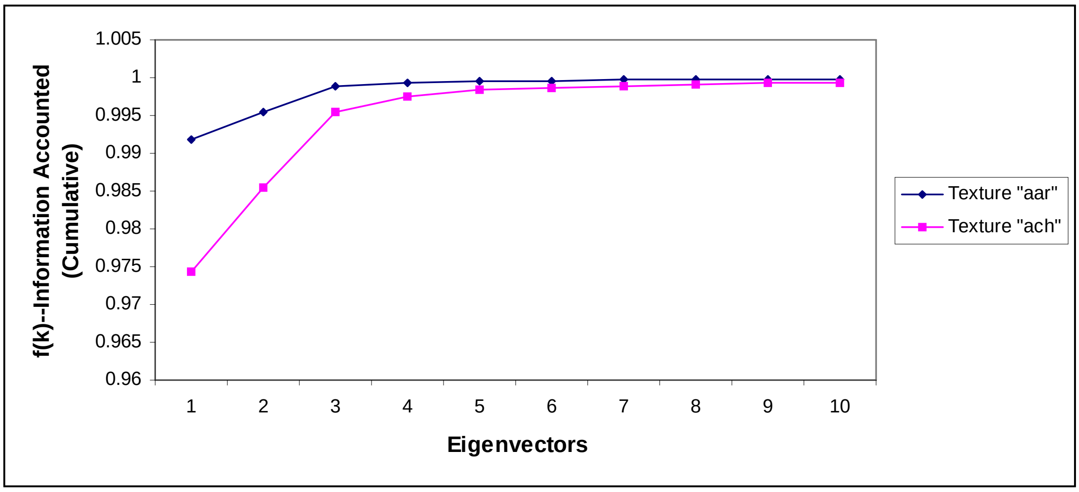
and cumulative eigenvectors [Zhang1998a]:

$$f_{cumu}(k) = \sqrt{(\sum_{i=1}^{k} w_i{}^2) \Big/ (\sum_{i=1}^{n} w_i{}^2)} \quad (4.4.13)$$

we find that the first few eigenvectors account for more than 99% of the total information contained in the image data matrix $\mathbf{I}$. For illustration, we show the plots of information accounted by eigenvectors for two textures "aar" (with near Lambertian reflectance) and "ach" (with specularities) in Figure 4.4.1.



(a)



(b)

Figure 4.4.1 Information accounted for by the first ten eigenvectors. Texture "aar" has a near-Lambertian surface; texture "ach" has a specular surface. In (a), f(k)—Information Accounted(Individual) is calculated using (4.4.12); In (b), f(k)—Information Accounted(Cumulative) is calculated using (4.4.13).

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

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

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

$$\grave{\mathbf{I}} = \mathbf{U_I}\grave{\mathbf{W}}_{\mathbf{I}}\mathbf{V_I}^{\mathbf{T}} \quad (4.4.14)$$

Recalling equation (4.4.4) in the mathematical framework we can write $\mathbf{M_1} = \mathbf{U_I}\grave{\mathbf{W}}_{\mathbf{I}}$. We let $\mathbf{M_1}$ be an $m\times k$ matrix, since the last $n-k$ columns of $\mathbf{U_I}\grave{\mathbf{W}}_{\mathbf{I}}$ 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 $\grave{\mathbf{W}}_{\mathbf{I}}$ 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

$$\mathbf{I} = (\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n) = \mathbf{M_1}\mathbf{M_2} \qquad (4.4.15)$$

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. In our case, we use 36 images and therefore $n = 36$.

If we use coefficients that differ from those in $\mathbf{M_2}$, the linear combinations of these base images allows 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. In our case, we use 3 eigen base images to represent 3D surface textures with Lambertian reflectance and 6 eigen base images to represent 3D surface textures with complex reflectance.

Interpolation

Although the linear combinations of eigen base images can produce novel images under illumination conditions that differ from those of the original, there are no direct links between the coefficients used for the linear combinations and illumination slant and tilt angles. Many researchers naturally employ interpolation techniques to relate the illumination directions with the coefficients because they are inexpensive, practical and able to produce reasonable results (with limitations) [Epstein1995, Zhang1998a, Wong2002]. Therefore, we also apply an interpolation technique to generate new images under given arbitrary illumination directions.

The illumination direction is specified by the slant angle σ and the tilt angle τ. We apply the bilinear interpolation method to generate a novel image with a given tilt angle τ and a slant angle σ. It is obvious that $0 \le \tau \le 2\pi$ and $0 \le \sigma \le \pi/2$. Since images are captured under different illumination slant and tilt angles for each texture, these illumination slant angle and tilt angle pairs form a sampling grid. In order to simplify further explanation, we use an image data vector $\mathbf{i}_{(\tau i,\sigma j)}$ to denote an image obtained under illumination tilt angle τ_i and slant angle σ_j. Thus each $\mathbf{i}_{(\tau i,\sigma j)}$ corresponds to an image vector in $\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n$ of (4.4.15). Firstly, we search for the intervals that contain τ and σ such that $\tau_i \le \tau \le \tau_{i+1}$ and $\sigma_j \le \sigma \le \sigma_{j+1}$. Then we define $t_1 \equiv (\tau - \tau_i)/(\tau_{i+1} - \tau_i)$ and $t_2 \equiv (\sigma - \sigma_j)/(\sigma_{j+1} - \sigma_j)$. Finally we calculate the new image with the illumination direction (τ, σ) using the algorithm from [Press1988]:

$$\mathbf{i}_{(\tau,\sigma)} = (1-t_1)(1-t_2)\mathbf{i}_{(\tau i,\sigma j)} + t_1(1-t_2)\mathbf{i}_{(\tau i+1,\sigma j)} + t_1 t_2 \mathbf{i}_{(\tau i+1,\sigma j+1)} + (1-t_1)t_2\mathbf{i}_{(\tau i,\sigma j+1)} \tag{4.4.16}$$

where $\mathbf{i}_{(\tau i,\sigma j)}$, $\mathbf{i}_{(\tau i+1,\sigma j)}$, $\mathbf{i}_{(\tau i+1,\sigma j+1)}$ and $\mathbf{i}_{(\tau i,\sigma j+1)}$ can be approximated by linear combinations of eigen base images using equation (4.4.15). Thus, $\mathbf{i}_{(\tau,\sigma)}$ is also a linear combination of eigen base images.

To summarise:

The *Eigen3* and *Eigen6* methods use 3 and 6 eigen base images respectively to represent 3D surface textures for relighting. These methods do not assume a particular reflectance model. The eigen base images are generated by using SVD. New images under arbitrary illumination directions can be constructed by a bilinear interpolation. These two methods are compatible with the input requirement of computer graphics hardware because the relighting can be expressed as a sum of products.

4.4.5. Summary

In sections 4.4.1 to 4.4.4, we introduced a mathematical framework and five inexpensive methods to extract 3D surface texture representations for relighting. The mathematical framework expresses the image data matrix as a product of two matrices; one is the surface representation matrix and the other can be either a lighting matrix or a coefficient matrix. With the exception of the *3I* method, the surface representations can therefore all be obtained using SVD. The five methods are:

3I: This method uses only three images of the sample texture taken at an illumination slant angle of 45° and tilt angles of 0°, 90° and 180° [Shashua1992]. It can produce accurate results for Lambertian surfaces with no shadows.

Gradient: The second method uses surface gradient and albedo maps, which are obtained by solving an over-determined linear system, to represent a 3D surface texture for relighting [Woodham1981].

PTM: This approach uses Polynomial Texture Maps (PTM), due to Malzbender et. al. [Malzbender2001]. PTMs are obtained by solving an over-determined linear system. Malzbender *et. al.* report that this method requires the assumption of a Lambertian surface, but it can capture the intensity variations due to surface self-shadows and interreflection.

Eigen3: The fourth method uses the first three eigen base images. Eigen base images are generated using SVD. Three eigen base images can capture the Lambertian component under varied illumination directions [Epstein1995]. New images with different illumination can be constructed by using linear combinations of base images. A bilinear interpolation is used to relate the illuminant slant and tilt angles with the coefficients of linear combinations.

Eigen6: This is identical to the previous method except that it uses the first six base images. This method can be used to represent 3D surface textures with specular components [Epstein1995].

We will further assess and compare these methods in next section.

4.5. Quantitative assessment of 3D surface texture representation methods

In Section 4.4, we introduced five inexpensive methods that can extract 3D surface texture representations. This section evaluates these methods by testing the *ability-of-reconstruction* and *ability-of-prediction*. The *ability-of-reconstruction* indicates the capability of these methods in reconstructing images that have already been used for the extraction of surface representations, whereas the *ability-of-prediction* shows the capability of these methods in predicting new images which are not used for the extraction of surface representations. We perform a quantitative assessment by comparing the relit results with original real images. In order to assess the performances of these methods on textures with different reflectance, we select 23 different textures from the PhoTex database (shown in Appendix A). Some of these textures have near-Lambertian surfaces; some have complex surface reflectance including self-shadowing, interreflectance and/or specularities. The normalised *root mean-squared* (*rms*) errors are used as the metric for the assessment, since large *rms* errors are not as noticeable in high variance textures as in low variance textures.

4.5.1. Normalised root mean-squared errors

The reason we use the normalised *root mean-squared* (*rms*) error as the metric is that we wish to assess the performances of the five methods on different textures. Gullón showed that this metric could produce reasonable assessment results [Gullón2002]. Because we have captured 36 images under different illumination directions for each texture, the normalised *rms* errors are averaged across 36 images per texture. It is expressed as

$$\eta = \frac{1}{36}\sum_{k=1}^{36}\frac{e_k}{Var(k)} \tag{4.5.1}$$

where:

$\frac{e_k}{Var(k)}$ is called the normalised *rms* error

$e_k = \frac{1}{NM}\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(r(x,y) - i(x,y)\right)^2}$ is the *rms* error

$Var(k)$ is the standard deviation of original image k

NM is the size of the images in pixels

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

$r(x,y)$ is the intensity of a relit image pixel at position x,y

The relit image has the same illumination condition as that used in one of the original input images.

Assessment of the ability-of-reconstruction

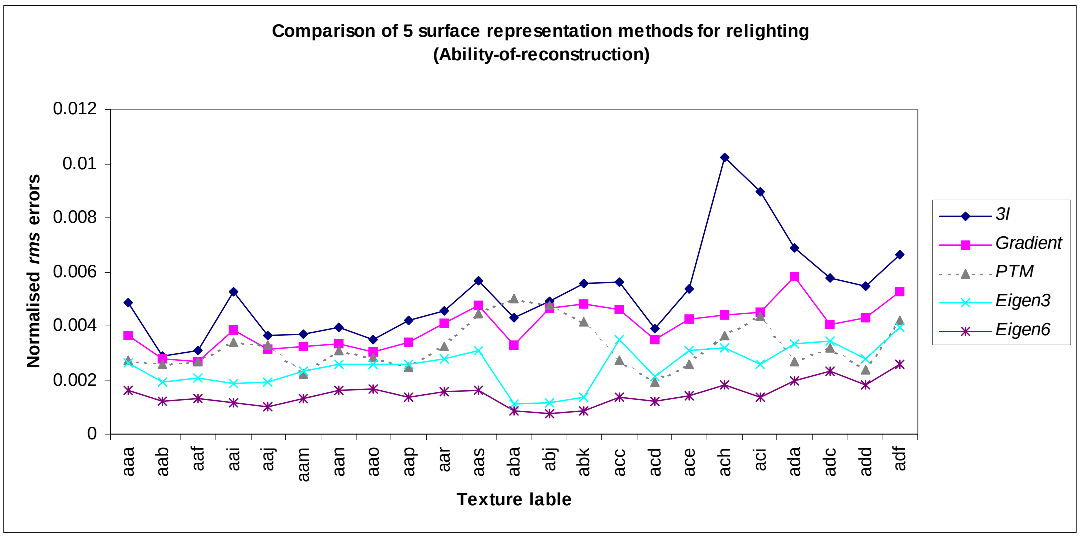
When assessing the *ability-of-reconstruction* of each method, we use all 36 images per texture as input to extract surface representations. Then the surface representations are relit to reconstruct 36 images using the same illumination conditions as those used in original images. The normalised *rms* error is calculated based on the 36 relit images and 36 original input images. It is obvious that for the *3I* method we only use three images, although we produce 36 relit images using the same illumination conditions as those used for the other methods.

Assessment of the ability-of-prediction

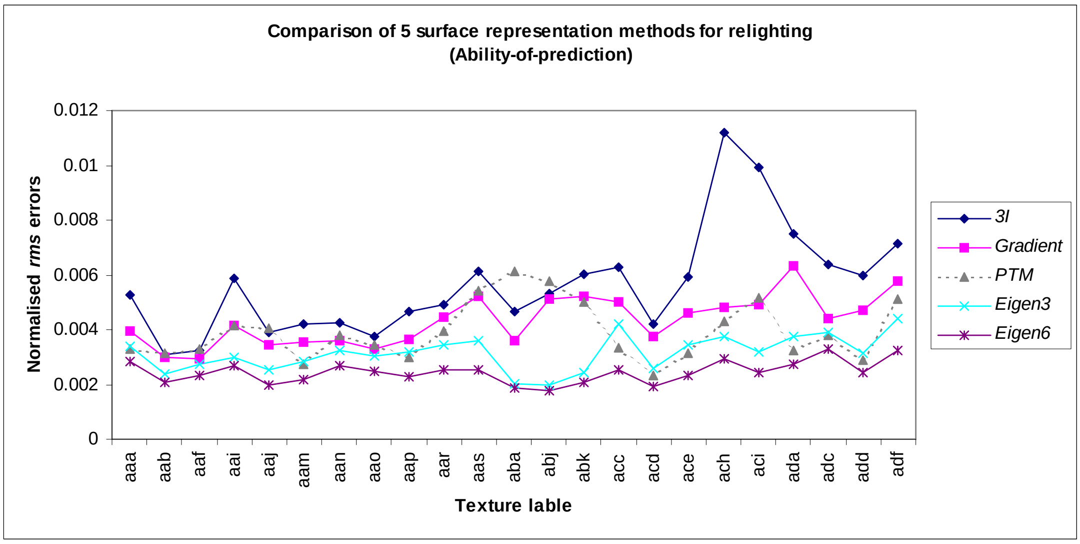
We would like to evaluate the ability of these five methods in predicting new images with illumination conditions that differ from those used for the extraction of surface representations. We employ a *leave-one-out* method, which leaves one image out of the 36 images that we have captured for each texture and tests it as an unknown. Thus, for *Gradient*, *PTM*, *Eigen3* and *Eigen6*, thirty-five images of each texture are used as a training image set to extract surface representations. For the *3I* method, we simply select three images with illumination directions that differ from those in predicted images. The surface representations are then relit using the same illumination condition as that used in the image which is not included in the training set. This process is repeated 36 times for each texture, and each time an image with a different illuminant direction is left out of the training set and then is tested. We therefore still produce 36 relit images in total, which are compared with 36 original images to calculate the normalised *rms* error.

4.5.2. Assessment results

Figure 4.5.1 shows the assessment results of these five methods across 23 textures.



(a)



(b)

Figure 4.5.1 Relighting error vs texture for the five approaches: (a)Ability-of-reconstruction; (b)Ability-of-prediction(Leave-one-out).

From Figure 4.5.1 it can be seen that the *3I* method produces the worst performance. This is not surprising given that it uses three input images whereas the other four methods use 36. Of the remaining methods, two (*Eigen6* & *PTM*) use more expensive $\mathbf{R}^6$ representations while *Gradient* & *Eigen3* use $\mathbf{R}^3$. We would therefore expect the first pair of techniques to outperform the latter, and on aggregate the *Eigen6* method does indeed provide the best figure. However, the performance of the *PTM* approach can not really be separated from that of its cheaper *Eigen3* competitor.

We further subtract the normalised *rms* errors produced by testing *ability-of-prediction* from those produced by testing *ability-of-reconstruction*. The difference is shown in Figure 4.5.2. Since all the difference are positive, it can be concluded that these methods perform better in reconstructing original training images than in predicting new images. Among these five methods, *Eigen6* has the largest difference between its *ability-of-reconstruction* and *ability-of-prediction*, while *Gradient* has the smallest difference in general.

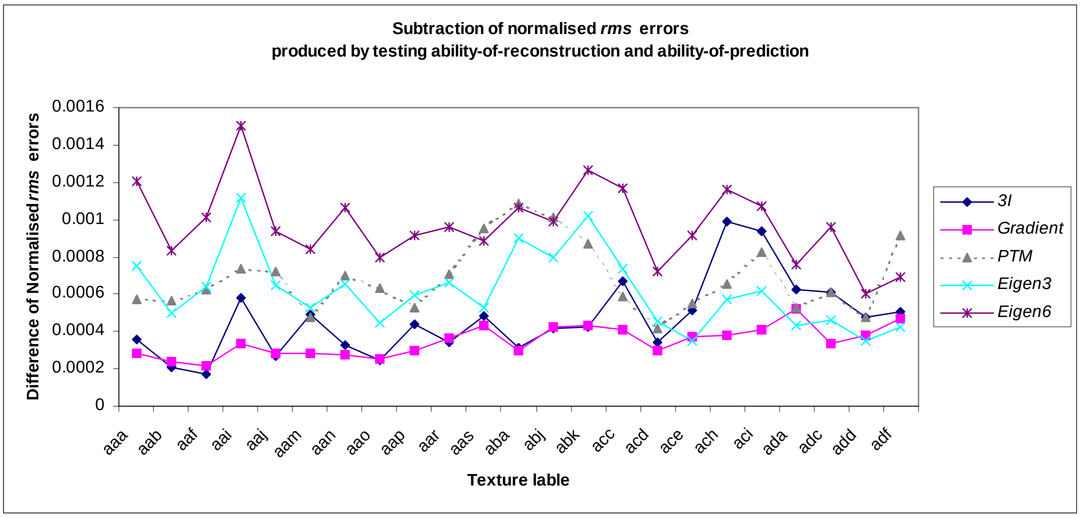
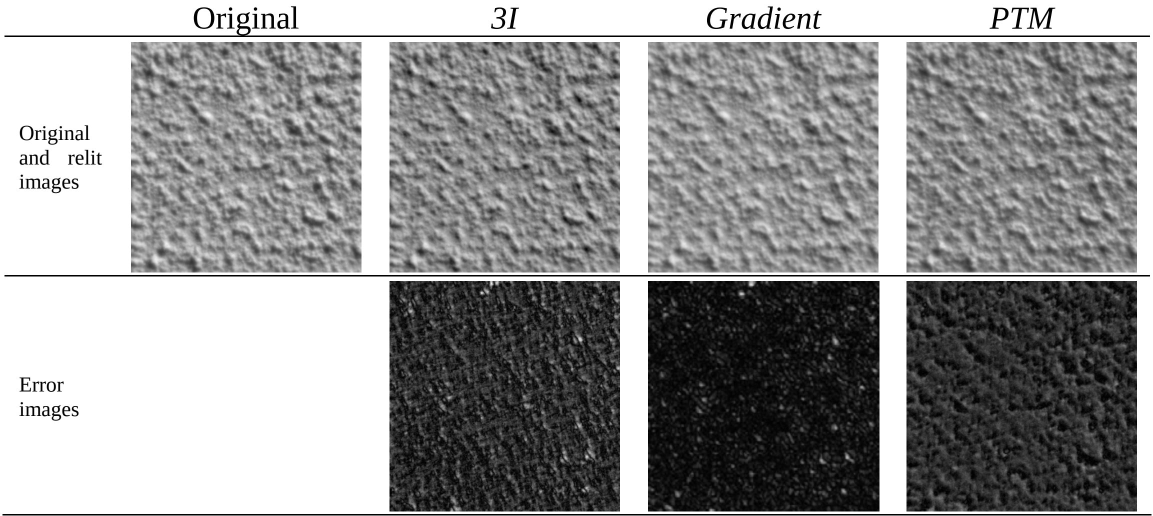


Figure 4.5.2 Subtracting normalised rms *errors produced by testing ability-of-prediction from those produced by testing ability-of-reconstruction.*

Example output images and their absolute difference images are shown in Figure 4.5.3 to Figure 4.5.8. We select three textures from the PhoTex database for the illustration. They represent Lambertian, Lambertian with shadows, and specular surfaces respectively. For each texture, we show the reconstructed and predicted images together with their corresponding error images (difference between original and relit images).



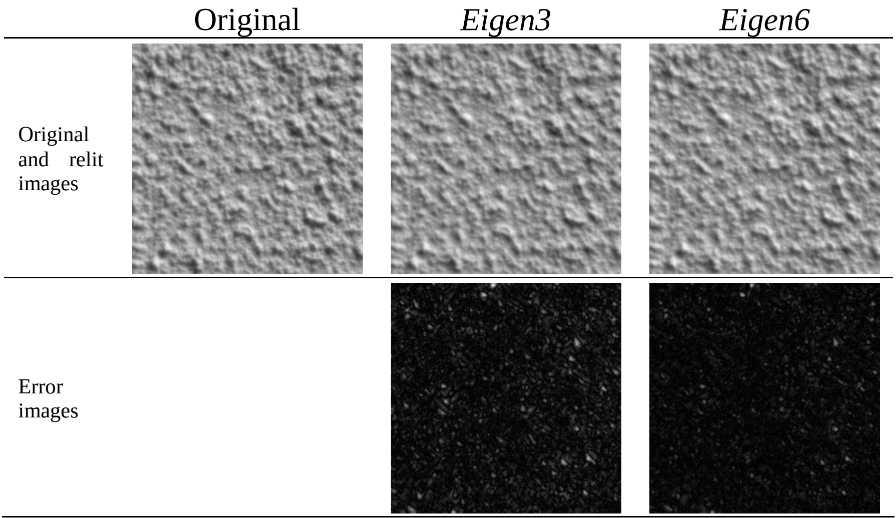
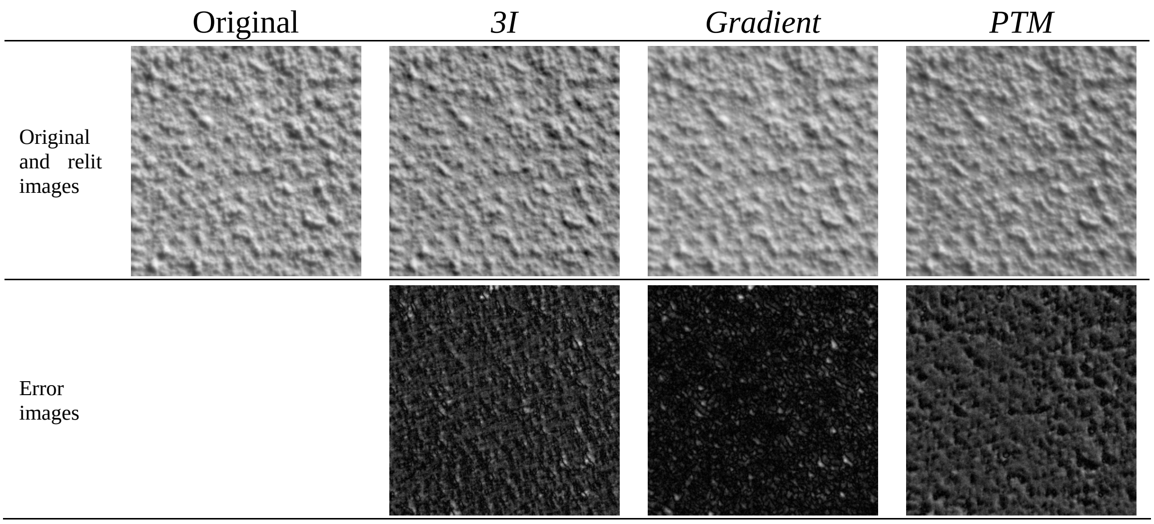


Figure 4.5.3 Texture "aar": Reconstructed images and their error (difference between original and rendering) images. Bright areas in the error images represent reconstruction inaccuracies.



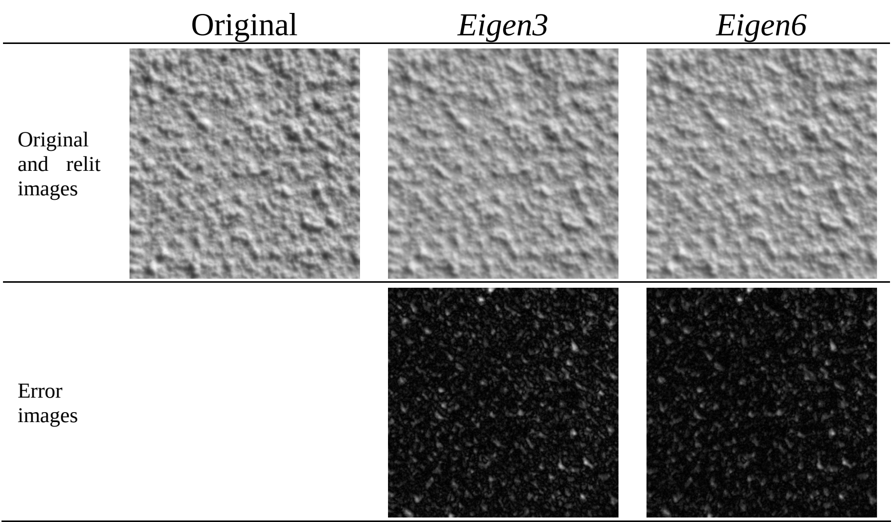
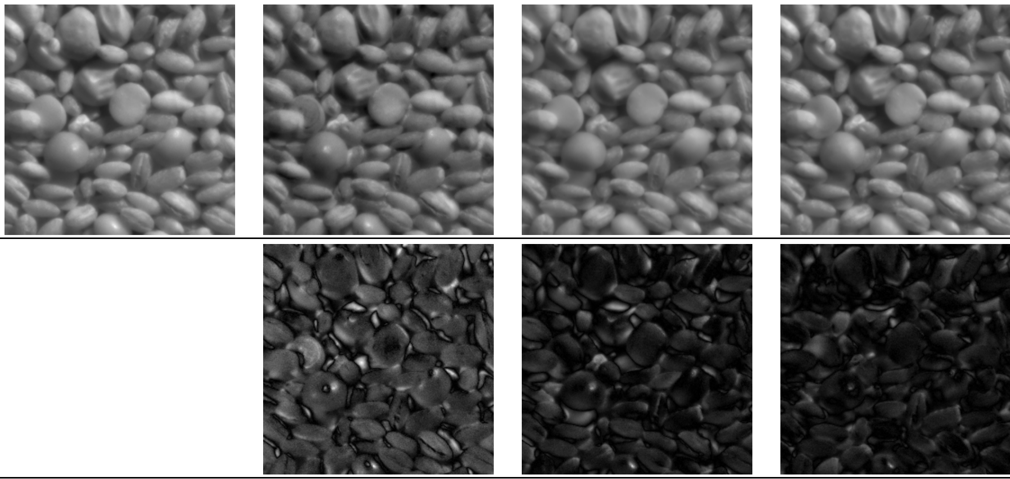


Figure 4.5.4 Texture "aar": Predicted images (produced by using leave-one-out) and their error (difference between original and rendering) images. Bright areas in the error images represent prediction inaccuracies.

	Original	*3I*	*Gradient*	*PTM*
Original and relit images				
Error images				



	Original	*Eigen3*	*Eigen6*
Original and relit images			
Error images			

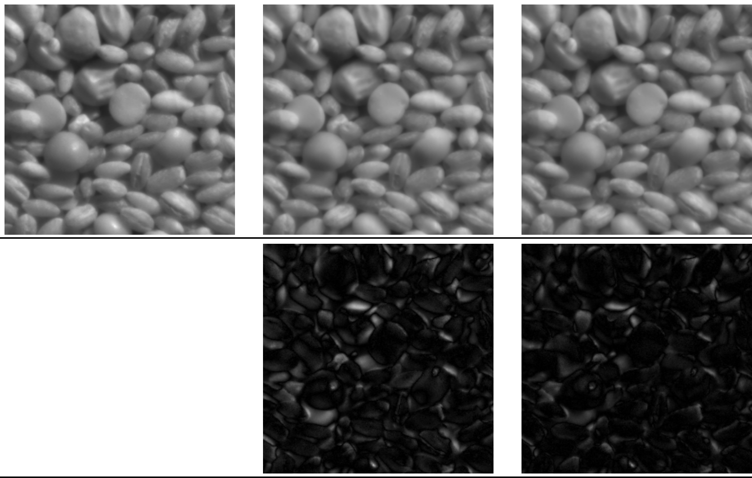


Figure 4.5.5 Texture “add”: Reconstructed images and error (difference between original and rendering) images. Bright areas in the error images represent reconstruction inaccuracies.

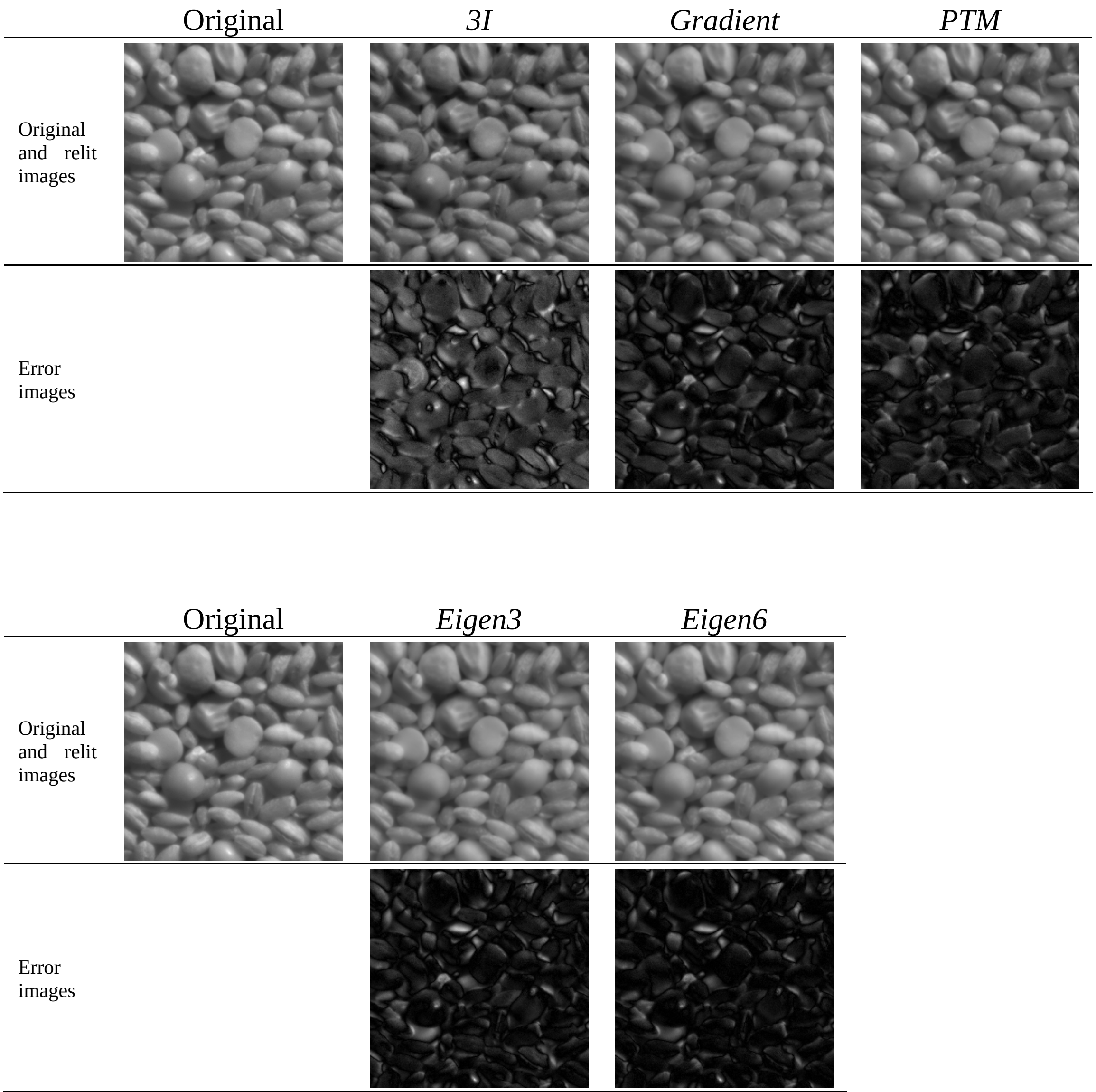


Figure 4.5.6 Texture "add": Predicted images (produced by using leave-one-out) and error (difference between original and rendering) images. Bright areas in the error images represent prediction inaccuracies.

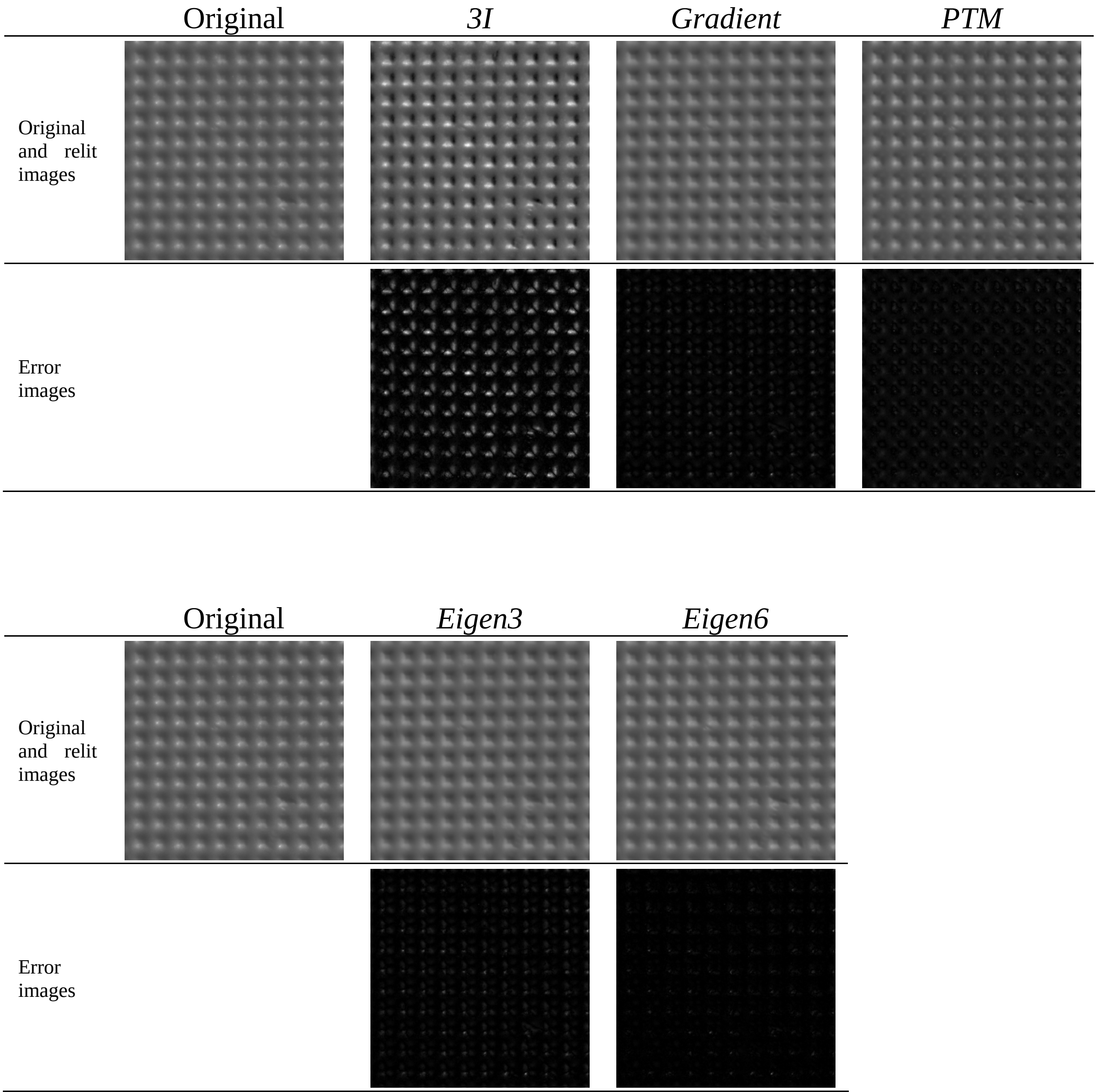


Figure 4.5.7 Texture "ach": Reconstructed images and error (difference between original and rendering) images. Bright areas in the error images represent reconstruction inaccuracies. The 3I *method produces very large errors. Because all error images are displayed in the same scale, errors produced by the other four methods are not noticeable comparing with those from the* 3I *method.*

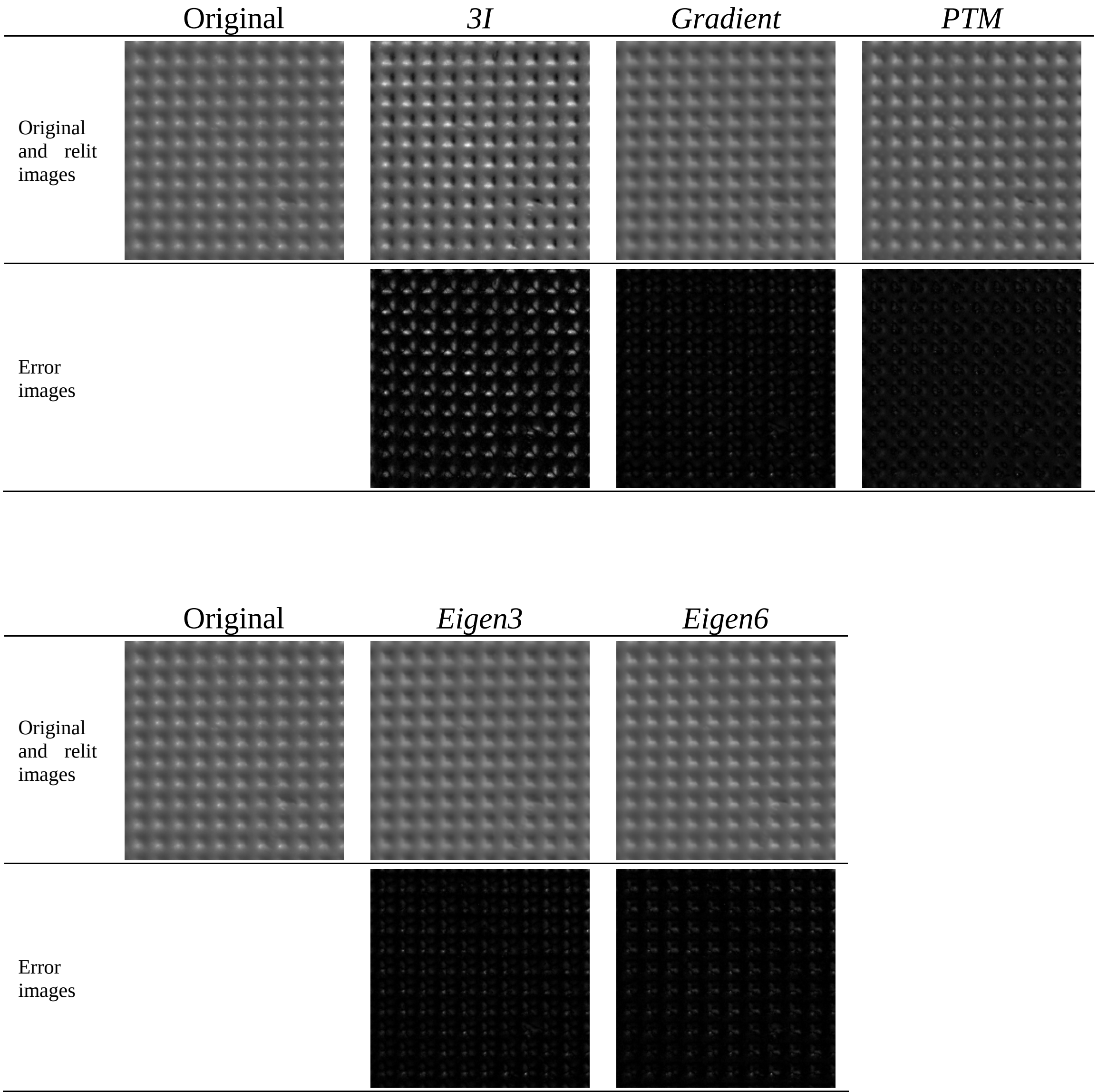


Figure 4.5.8 Texture "ach": Predicted images (produced by using leave-one-out) and error (difference between original and rendering) images. Bright areas in the error images represent prediction inaccuracies. The 3I *method produces very large errors. Because all error images are displayed in the same scale, errors produced by the other four methods are not noticeable comparing with those of the* 3I *method.*

4.5.3. Discussion of the assessment results

This section analyses the assessment results and discusses the relevant problems in different methods. In particular, we investigate the integration and differentiation algorithms when discussing the *Gradient* method. We also compare the relighting results of the selected five methods and a heightmap-based relighting method, in which surface gradient maps are integrated to generate the heightmap.

The 3I method

The *3I* method produced the worst performance in the assessment. It is obvious that it can only produce accurate results when the textures have pure Lambertian surfaces with no shadowing. However, since this method only uses three images, it provides the most economical way to approximate real textures.

The Gradient method

The *Gradient* method performs much better than the *3I* method in representing real textures, because it uses all 36 images of a sample texture and approximates these images in the least squares sense (Figure 4.5.1). However, its performance is affected by several factors: the approximation of Lambertian reflectance, noises in sample images and the intergratibility of surface gradient maps. These effects can be detected by testing the relationship between two surface gradient maps in frequency domain.

We first take Fourier Transform on the spatial surface gradient maps *p(x,y)* and *q(x,y)*. We use $\mathrm{P(u,v)}$ and $\mathrm{Q(u,v)}$ to denote *p(x,y)* and *q(x,y)* in frequency domain respectively, where $\mathrm{(u,v)}$ is the 2D spatial frequency co-ordinate. By Fourier theories, we have the following equations:

$$\mathrm{P(u,v)} = ju\,\mathrm{S(u,v)} \tag{4.5.2}$$

$$\mathrm{Q(u,v)} = jv\,\mathrm{S(u,v)} \tag{4.5.3}$$

where $\mathrm{S(u,v)}$ is the frequency domain denotation of the spatial surface height map *s(x, y)* and *j* is the square root of minus one.

Thus,

$$v\,\mathrm{P(u,v)} = u\,\mathrm{Q(u,v)} \tag{4.5.4}$$

However, most real textures do not have pure Lambertian surfaces, and the surface might not be integrable. These limitations cause equation (4.5.4) not to hold. Therefore, we can treat the surface gradient maps as images containing intergratibility noise. If we force the equation (4.5.4) to hold by changing $P(u,v)$ and $Q(u,v)$, we obtain the perfect synthetic surface gradient maps in frequency domain for a Lambertian surface. By taking inverse Fourier Transform, we can compare these synthetic surface gradient maps with their original counterparts. Figure 4.5.9 shows examples of a sample texture.

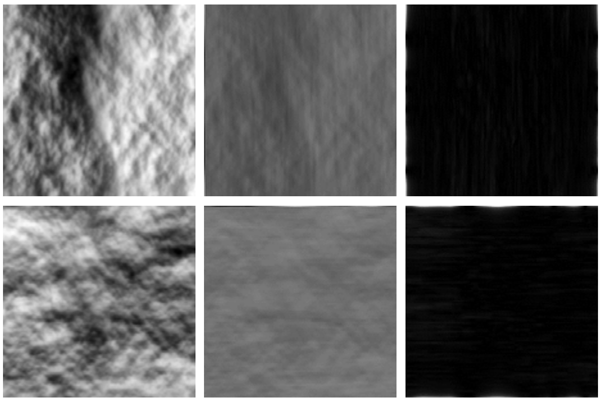


Figure 4.5.9 The comparison of real surface gradient maps and their synthetic counterparts. The first column shows the two surface gradient maps calculated using the Gradient *method; the second column shows the corresponding synthetic surface gradient maps generated using equation (4.5.4); the third column shows the absolute difference images, which are generated by subtracting synthetic maps (the second column) from corresponding real maps (the first column).*

The noise in surface gradient maps will further affect the height map generated by integrating surface gradient maps. In order to obtain the surface height map, surface integratibility is assumed. We have used a frequency domain approach to generate the surface height map from gradient maps [Frankot1988]. We evaluate the integration problem by relighting the surface height and albedo maps using the Lambertian model (4.4.6) and calculating the normalised *rms* errors as introduced in the previous section.

The surface height map in frequency domain can be expressed as:

$$S(u,v) = \frac{-ju\,P(u,v) - jv\,Q(u,v)}{u^2 + v^2} \qquad (4.5.5)$$

In order to use the Lambertian model (4.4.6), we need to differentiate the surface height map to obtain gradient maps. We have used two approaches when differentiating the surface height map: a frequency domain approach and a spatial domain approach. Equation (4.5.2) and (4.5.3) are used for the differentiation in the frequency domain, while the differentiation in spatial domain can be approximated by:

$$p(x,y) \cong s(x+1,y) - s(x,y) \qquad (4.5.6)$$

$$q(x,y) \cong s(x,y+1) - s(x,y) \qquad (4.5.7)$$

The two differentiation methods produce slightly different surface gradient maps. Figure 4.5.10 shows two pairs of example output surface gradient maps and their absolute difference images. Furthermore, we have found that smaller normalised *rms* errors are produced if we relight the gradient maps that are derived from differentiation in frequency domain. Figure 4.5.11 shows the comparison across 23 textures.

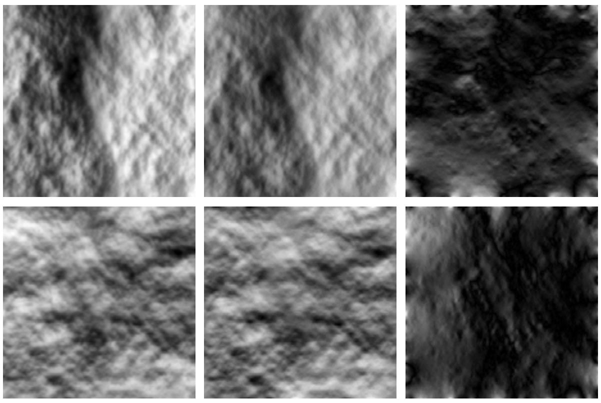
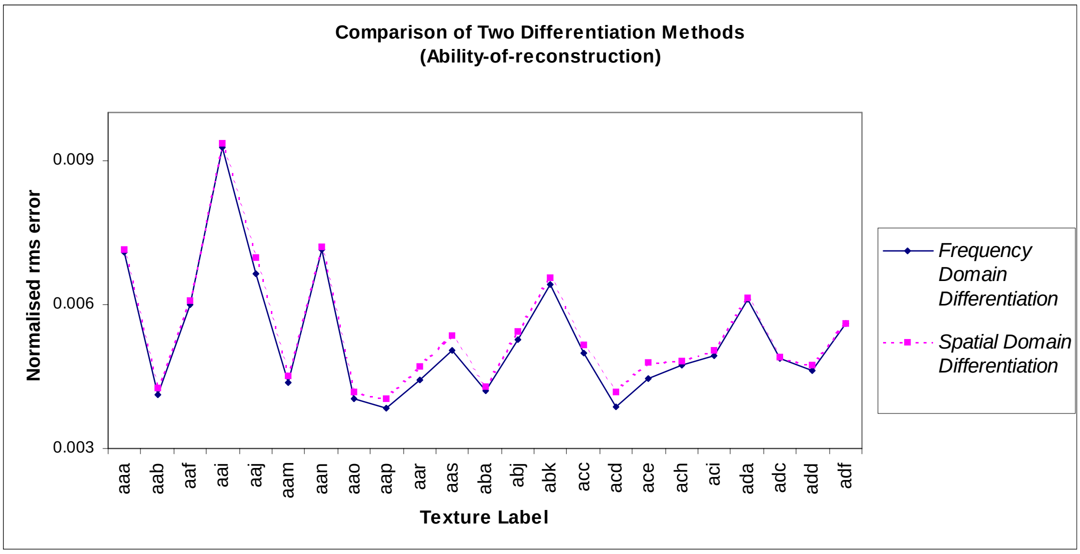
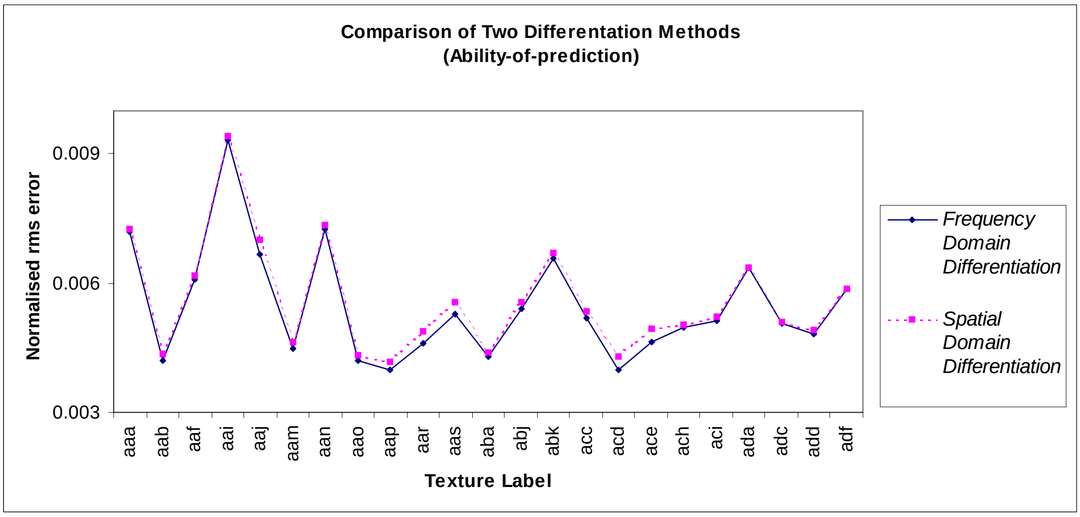


Figure 4.5.10 The comparison of differentiation methods. The first and second columns are gradient maps produced by differentiation of the surface height map in frequency and spatial domain respectively. The third column shows the absolute difference images.



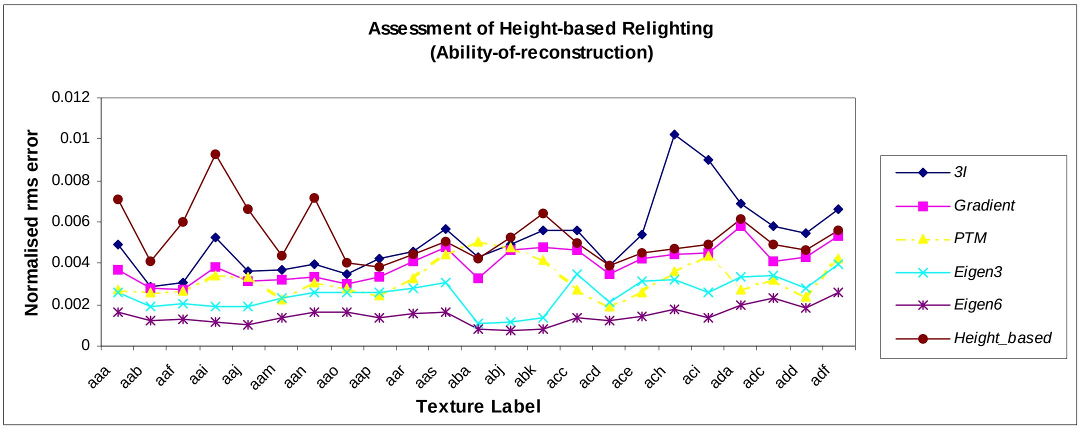
(a)



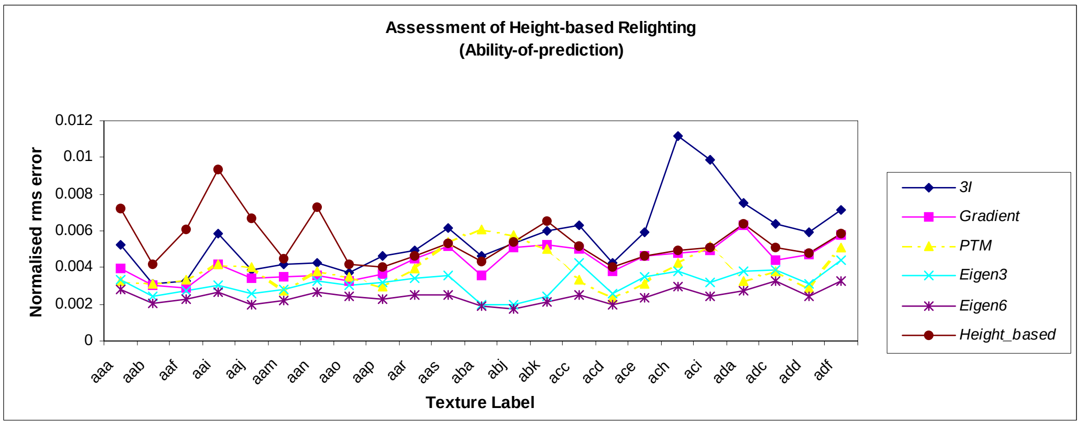
(b)

Figure 4.5.11 Comparison of two differentiation methods: (a)Ability-of-reconstruction; (b)Ability-of-prediction(Leave-one-out).

However, even if we use the differentiation method in frequency domain, the relighting results generated using surface height and albedo maps still have larger normalised *rms* errors compared with those produced by the *Gradient* method. In Figure 4.5.12 we show the comparison of the height map based method with the other five methods that we have introduced in the previous section. This comparison is also based on measuring ability-of-reconstruction and ability-of-prediction, which uses the *leave-one-out* method. It can be seen that the performance of the height map based method is even worse than the *3I* method for some textures. This is also the reason that we did not select a height based surface representation method in this thesis. Nevertheless, it provides the cheapest surface representation which only has two dimensions for a Lambertian surface.



(a)



(b)

Figure 4.5.12 Comparison of height-based relighting and other five methods: (a) Ability-of-reconstruction; (b) Ability-of-prediction (Leave-one-out).

The PTM method

Figure 4.5.1 shows that the PTM method performs better than the *3I* and *Gradient* methods in general. One possible reason for this is because it uses a quadratic lighting function, which employs an $\mathbf{R}^6$ representation—Polynomial Texture Maps (PTM). In contrast, the Lambertian model is a linear lighting function, which only uses an $\mathbf{R}^3$ representation. Furthermore, the *PTM* method was designed to capture the variation of image intensities due to surface self-shadowing and interreflections. It did perform well in our experiments: for texture "ada", "adc", "add" and "adf", which contain obvious self-shadowing and interreflections, the normalised *rms* errors are smaller than those produced by the *3I* and *Gradient* methods (Figure 4.5.1).

The Eigen3 and Eigen6 methods

The eigen-space based methods (*Eigen3* and *Eigen6*) are actually derived from the pure analysis of the image intensity matrix using the SVD method. Therefore, it will provide the best least square approximation to the original data matrix (Figure 4.5.1). It can also be observed that the bilinear interpolation method produced reasonable relighting results. The normalised *rms* errors produced by *Eigen6* are the smallest for all textures. The performance of the *Eigen3* method, which only uses three-dimensional representation maps, can not even be separated from that of the *PTM* method.

4.6. Conclusion

This chapter has selected five inexpensive methods for extracting surface relighting representations. This is the first stage in our overall framework for synthesis and relighting of 3D surface textures.

We first presented a review of available relighting representations of 3D surface textures. Since our main goal is to develop inexpensive approaches for synthesis and relighting of 3D surface textures, we select five low-dimensional relighting representations, comprising: a set of three photometric images (*3I*); surface gradient and albedo maps (*Gradient*); Polynomial Texture Maps (*PTM*); and two eigen-based representations using 3 and 6 base images (*Eigen3* and *Eigen6*). We presented a mathematical framework which summarises the common mathematical properties of these five methods. The *3I* and *Gradient* methods require the Lambertian model. The *PTM* method assumes the surface has Lambertian reflectance but uses a quadratic lighting function to model the variation of image intensities due to surface self-shadowing and interreflections. In contrast, *Eigen3* and *Eigen6* do not assume any reflectance models. The *Eigen6* method in particular is better able to cope with specular surfaces, although the surface geometry is required to be simple. These methods are compatible with modern graphics systems; the extracted surfaces representations can be programmed into graphics hardware so that relighting can be achieved in real-time by using linear combinations through texture units and register combiners in graphics processing chips.

We used 23 real textures to quantitatively assess the performances of the five methods by measuring the *ability-of-reconstruction* and the *ability-of-prediction*. The latter employs a *leave-one-out* test method. We compared relit images produced by different methods with original real images and calculated normalised *rms* errors. The results show that the *3I* method produces the worst performance and *Eigen6* method produces the best. The $\mathbf{R}^6$ *PTM* representations perform better than $\mathbf{R}^3$ *Gradient* representations, although it cannot be considered more superior to the cheaper *Eigen3* representations in $\mathbf{R}^3$ space.