

Three-dimensional Surface Texture Synthesis

Junyu Dong

Thesis submitted
for the
Degree of Doctor of Philosophy

Heriot-Watt University
Department of Computer Science



September 2003

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that the copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior consent of the author or the University (as may be appropriate)

Abstract

Texture synthesis has been extensively investigated by both computer vision and computer graphics communities during the past twenty years. However, the input and output are normally 2D intensity texture images. If the subjects are 3D surface textures (such as brick, woven or knitted textiles, embossed wallpapers etc.), these 2D synthesis techniques cannot provide the information required for rendering under other than the original illumination and viewpoint conditions. The aim of this thesis therefore is to develop inexpensive approaches for the synthesis of 3D surface textures. Few publications are available in this research area.

We first introduce an overall framework for the synthesis of 3D surface textures. The framework essentially combines surface representation methods with 2D texture synthesis algorithms to synthesise and relight new surface representations. Then we investigate five low-dimensional methods, namely the *3I*, *Gradient*, *PTM*, *Eigen3* and *Eigen6* methods, for extracting representations from a set of images of the 3D surface texture sample. The surface representations can be relit to generate new images under arbitrary lighting directions by linear combinations. These methods are quantitatively assessed by comparing the original and relit images. The results show that the *Eigen6* produces the best performance.

We select a 2D texture synthesis algorithm which is then extended into multi-dimensional space to use the five surface representations as input. In this way, we develop five approaches for the synthesis of 3D surface textures. The synthesised results are compatible with computer graphics systems and can be used in real-time rendering applications. The five synthesis approaches are qualitatively assessed by employing psychophysical experiments and non-parametric statistics. The results show that the two low-dimensional methods, the *Gradient* and *Eigen3*, on average offer as good a performance as of any of the other methods and incur low computational cost.

To my parents

Acknowledgement

First and foremost I would like to thank my supervisor, Dr. Mike Chantler, for his guidance, patience, encouragement, support, discussions, earlier work and everything I have learned from him. I would also like to thank Dr. McGunnigle for his earlier work, discussions and proofreading of this thesis. My thanks are also due to Mr. Andrew Spencer, Mr. Michael Robb and Mrs. Agnes Dannreuther for their proofreading and help in the preparation of this thesis.

I would like to thank Dr. Yvan Petillot and Dr. Katia Lebart for their helpful discussions about this research.

Many thanks to Mike, Ged, Cristina, Jing, Jerry, Dave, Jennifer, Andy, Mike, Ivan, Michael, Andreas and all people who have ever worked in the Texture Lab and ISL for creating a great and fun research environment that make my life in Edinburgh more interesting. I am also grateful to Mr Huiyu Zhou, Mr Shida Xu, Mr Zhenyu Lin, Mr Shumu Li, Mr Chuxu Liu and other fellow students who helped me in the psychophysical experiments.

Finally, I would like to express my gratitude to my wife, my parents, my parents-in-law and the rest of the family for their consistent support, encouragement and everything that they have been doing for me.

Table of Contents

Table of Contents.....	iv
List of Figures	viii
List of Tables	x
Principle Symbols	xi
Abbreviations	xiv
Chapter 1 Introduction	1
1.1. Motivation	1
1.2. Scope of the research	4
1.2.1. Definitions of terms	4
1.2.2. Scope of the research	5
1.3. Thesis organisation	7
1.4. Original work.....	8
Chapter 2 Literature Survey	10
2.1. Three-dimensional surface texture synthesis.....	11
2.2. Two-dimensional texture synthesis	15
2.2.1. Texture synthesis methods based on global sampling strategies	15
2.2.2. Texture synthesis methods based on local sampling strategies	19
2.2.3. Summary.....	21
2.3. Surface representation methods for relighting.....	21
2.3.1. Extracting surface relighting representations using reflectance models.....	22
2.3.2. Extracting surface relighting representations using other techniques.....	23
2.3.3. Extracting 3D surface texture representations for relighting	24
2.3.4. Summary.....	25
2.4. Conclusion	26
Chapter 3 Framework.....	27
3.1. Introduction	27
3.2. A framework for the synthesis of 3D surface textures	28

3.2.1.	Framework.....	28
3.2.2.	An example: the approach of [Liu2001].....	30
3.2.3.	Discussion.....	31
3.3.	The image data environment for the thesis.....	32
3.4.	Conclusion	34
Chapter 4 Surface Texture Representations for Relighting		35
4.1.	Introduction	35
4.2.	Criteria	36
4.3.	A detailed review and selection of surface representation and relighting methods	37
4.4.	The selected methods.....	43
4.4.1.	Mathematical framework.....	43
4.4.2.	Lambertian methods-- <i>3I</i> and <i>Gradient</i>	45
4.4.3.	The <i>PTM</i> method	49
4.4.4.	The eigen-based methods (<i>Eigen3</i> and <i>Eigen6</i>)	51
4.4.5.	Summary.....	55
4.5.	Quantitative assessment of 3D surface texture representation methods.....	56
4.5.1.	Normalised root mean-squared errors.....	56
4.5.2.	Assessment results	58
4.5.3.	Discussion of the assessment results	66
4.6.	Conclusion	71
Chapter 5 Synthesis Algorithms.....		73
5.1.	Introduction	73
5.2.	A detailed survey of synthesis algorithms	74
5.2.1.	Texture synthesis methods based on global sampling strategies	74
5.2.2.	Texture synthesis methods based on local sampling strategies	76
5.2.3.	Summary.....	79
5.3.	Two Approaches.....	80
5.3.1.	The first approach and modification—a pixel-based multi-resolution approach.....	80
5.3.2.	The second approach and modification—a patch-based approach.....	85
5.3.3.	Comparison of the two approaches.....	89

5.3.4. Summary	90
5.4. Analysis of the selected synthesis algorithm.....	90
5.4.1. Sample image size	91
5.4.2. Block size	92
5.4.3. Overlap size	92
5.4.4. Error tolerance	95
5.4.5. Strength and weakness.....	96
5.4.6. Summary.....	97
5.5. Conclusion.....	98
Chapter 6 Synthesis and Relighting.....	99
6.1. Introduction	99
6.2. The five synthesis approaches	101
6.2.1. The general algorithm for the synthesis of surface texture representations	101
6.2.2. The <i>3I</i> synthesis approach	105
6.2.3. The <i>Gradient</i> synthesis approach	105
6.2.4. The <i>PTM</i> synthesis approach.....	106
6.2.5. The <i>Eigen3</i> and <i>Eigen6</i> synthesis approaches	106
6.2.6. Summary.....	107
6.3. Qualitative assessment of the five approaches	108
6.3.1. Design of the psychophysical experiments.....	109
6.3.2. The test of significant difference—Friedman’s nonparametric two-way Analysis of Variance.....	114
6.3.3. The multiple comparison	114
6.4. Conclusion	116
Chapter 7 Conclusion and Discussion	118
7.1. Summary.....	118
7.2. Conclusion	121
7.3. Discussion.....	122
7.3.1. Using the synthesised 3D surface texture representations in real-time graphics programming	122

7.3.2. Using the synthesised 3D surface texture representations in graphics software packages	125
Appendix A: Texture samples	128
Appendix B: Synthesis and relighting results from the five methods for 23 textures ($\tau = 60^\circ$ and $\tau = 120^\circ$).....	129
Appendix C: Rank data of the five synthesis approaches	137
Appendix D: List of publications by the author	140
References.....	144

List of Figures

Figure 1.1.1 The tiling effects produced by mapping a texture image of an inadequate size using standard OpenGL functions.....	2
Figure 1.1.2 Texture synthesis using the algorithm proposed in [Efros2001].	3
Figure 1.1.3 Two images of a 3D surface texture.....	3
Figure 1.1.4 A 3D surface texture can be described using the surface height and albedo maps...	3
Figure 1.2.1 The scope of research in this thesis	6
Figure 2.1.1 The flowchart of the method introduced in [Liu2001]	13
Figure 3.2.1 The overall framework for the synthesis of 3D surface textures.....	30
Figure 3.2.2 The method introduced in [Liu2001] can be represent within our framework.	31
Figure 3.3.1 The imaging set-up and definitions of the slant and tilt angle.....	33
Figure 4.3.1 Different representations v.s. criteria	41
Figure 4.4.1 Information accounted for by the first ten eigenvectors	53
Figure 4.5.1 Relighting error vs texture for the five approaches.....	58
Figure 4.5.2 Subtracting normalised rms errors.....	59
Figure 4.5.3 Texture “aar”: Reconstructed images and their error (difference between original and rendering) images	60
Figure 4.5.4 Texture “aar”: Predicted images and their error (difference between original and rendering) images.....	61
Figure 4.5.5 Texture “add”: Reconstructed images and error (difference between original and rendering) images.....	62
Figure 4.5.6 Texture “add”: Predicted images and error (difference between original and rendering) images.....	63
Figure 4.5.7 Texture “ach”: Reconstructed images and error (difference between original and rendering) images.....	64
Figure 4.5.8 Texture “ach”: Predicted images and error (difference between original and rendering) images.....	65
Figure 4.5.9 The comparison of real surface gradient maps and their synthetic counterparts..	67
Figure 4.5.10 The Comparison of differentiation methods.	68
Figure 4.5.11 Comparison of two differentiation methods.....	69
Figure 4.5.12 Comparison of height-based relighting and other five methods.....	70
Figure 5.1.1 The selection of synthesis algorithm in the overall framework.	73
Figure 5.3.1 The next column neighbour of last best-matched pixel can be used as the current best match	81
Figure 5.3.2. The neighbourhood defined by Wei and Levoy [Wei2000].....	82
Figure 5.3.3. Three cases that must perform exhaustive search.....	83

Figure 5.3.4 Comparison of synthesis results.	84
Figure 5.3.5 The boundary cut process of Efros' 2D texture synthesis approach.....	87
Figure 5.3.6. The neighbour of previous best-matched blocks.	87
Figure 5.3.7 The comparison of results produced by the modified and original algorithms	87
Figure 5.3.8 Two example synthesised images	90
Figure 5.4.1 Synthesis results produced by using different input sample sizes	91
Figure 5.4.2 Synthesis results produced by using different input block sizes.....	92
Figure 5.4.3 Synthesis results produced by using different input overlap sizes	95
Figure 5.4.4 Synthesis results produced by using different error tolerances	96
Figure 5.4.5 Example synthesis results of two highly structured textures	96
Figure 5.4.6 A failed example (Texture “add”).	97
Figure 6.1.1 The final stage of the overall framework.....	99
Figure 6.2.1 Each group of best-matched blocks in synthesised results comes from the same location in samples	102
Figure 6.2.2 The group of best-matched blocks produced in R^3 space does not guarantee each individual in the group is the same as the best-matched block produced in R^1 space	104
Figure 6.2.3 The 3I synthesis approach	105
Figure 6.2.4 The Gradient synthesis approach	106
Figure 6.2.5 The PTM synthesis approach	106
Figure 6.2.6 The Eigen3 and Eigen6 approaches.....	107
Figure 6.3.1 Multiple comparison test of the five approaches.	115
Figure 7.3.1 Two still images of a real-time sequence produced by rendering synthesised surface gradient and albedo maps using the method described in [Robb2003]	124
Figure 7.3.2 Texture mapping using Micrografx Simply 3D 2	126
Figure 7.3.3 Texture mapping using Micrografx Simply 3D 2	127

List of Tables

Table 1.2.1 Definition of terms	5
Table 2.2.1 Characteristics of typical global sampling methods.....	18
Table 5.3.1 The pseudocode of the first approach	84
Table 5.3.2 The pseudocode of the first approach	88
Table 5.3.3 The comparison of two 2D texture synthesis algorithms.....	89
Table 6.2.1 Summary of the 5 approaches.....	108
Table 6.3.1 Synthesis and relighting results from the five methods for 11 textures.....	111

Principal Symbols

Symbol	Meaning	Section first introduced
\mathbf{R}^1	One-dimensional real space	3.2
\mathbf{R}^3	Three-dimensional real space	3.2
\mathbf{R}^6	Six-dimensional real space	4.5
\mathbf{R}^n	N-dimensional real space	3.2
(x, y)	Pixel location	3.2
(x_0, y_0)	Pixel location	3.2
(x'_0, y'_0)	Pixel location	3.2
$p(x, y)$	Surface derivative in x direction in spatial domain	3.2
$q(x, y)$	Surface derivative in y direction in spatial domain	3.2
$al(x, y)$	Albedo map	3.2
\mathbf{M}	Matrix	4.4.1
\mathbf{U}	Column-orthogonal matrix produced by decomposing a matrix \mathbf{M} using SVD	4.4.2
\mathbf{W}	Diagonal matrix containing singular values produced by decomposing a matrix \mathbf{M} using SVD	4.4.2
\mathbf{V}^T	Transpose of the orthogonal matrix produced by decomposing a matrix \mathbf{M} using SVD	4.4.2
\mathbf{I}	Image data matrix	4.4.1
$i_{11}, i_{12}, \dots, i_{mn}$	Pixel intensity values in certain images.	4.4.1
\mathbf{M}_1	Surface representation matrix	4.4.1
\mathbf{M}_2	Known matrix for extracting surface representation matrix \mathbf{M}_1	4.4.1

\mathbf{c}	Coefficient vector for relighting	4.4.1
$i(x, y)$	Intensity of an image pixel at (x, y)	4.4.2
λ	Incident intensity to the surface	4.4.2
ρ	Albedo value of the Lambertian reflection	4.4.2
\mathbf{l}	Lighting vector	4.4.2
\mathbf{n}	Normalised surface normal	4.4.2
τ	Tilt angle of illumination	4.4.2
σ	Slant angle of illumination	4.4.2
$s(x, y)$	Surface height map in spatial domain	4.4.2
\mathbf{N}	Surface normal matrix	4.4.2
\mathbf{A}	Albedo matrix	4.4.2
\mathbf{N}_a	Scaled surface normal matrix	4.4.2
\mathbf{L}	Lighting matrix	4.4.2
\mathbf{L}_{ptm}	Lighting matrix in the <i>PTM</i> method	4.4.3
\mathbf{l}_{ptm}	Lighting vector in the <i>PTM</i> method	4.4.3
\mathbf{A}_{ptm}	Polynomial Texture Map matrix	4.4.3
w_i	Singular value of the image data matrix \mathbf{I} , $i=1,2,\dots$	4.4.4
$\mathbf{\tilde{W}}_i$	The approximation matrix of the diagonal matrix \mathbf{W} containing the first few singular values	4.4.4
$\mathbf{i}_{(\tau_i, \sigma_j)}$	Image obtained under illumination tilt angle τ_i and slant angle σ_j .	4.4.4
η	Normalised <i>root mean-squared</i> errors	4.5.1
$P(u, v)$	Denotation of $p(x, y)$ in frequency domain	4.5.3
$Q(u, v)$	Denotation of $q(x, y)$ in frequency domain	4.5.3
$S(u, v)$	Denotation of the spatial surface height map $s(x, y)$ in frequency domain	
(u, v)	2D frequency co-ordinate	4.5.3
L	The level of the lowest scale of an image pyramid	5.3.1
$\{X, (m, n)\}$	Pixel location at level X of the result pyramid	5.3.1
$\{X, (k, l)\}$	Pixel location at level X of the sample pyramid	5.3.1

$O(n)$	The computational complexity is the order of n	5.3.3
Ω_j	The overlapping area covered by block j in the sample image and the already synthesised pixels	5.4.3
(x_i, y_i)	The i^{th} pixel of the sample image covered by the overlapping area Ω_j	5.4.3
(x'_i, y'_i)	The i^{th} pixel of the result image covered by the overlapping area Ω_j	5.4.3
$\min\{\dots\}$	Function to calculate the minimum value	5.4.3
$m_i(x, y)$	A pixel value at (x, y) in the i^{th} sample representation map	6.2.1
$m'_i(x', y')$	A pixel value at (x', y') in the i^{th} result representation map	6.2.1
α	Confidence level	6.3.1

Abbreviations

Abbreviations	Meaning	Section first introduced
SSD	Sum of Square Differences	2.1
BTF	Bidirectional Texture Functions	2.1
BRDF	Bidirectional Reflectance Distribution Function	2.3.1
SVD	Singular Value Decomposition	2.3.2
PCA	Principal Component Analysis	2.3.2
CUReT	Columbia-Utrecht Reflectance and Texture Database	2.3.2
<i>3I</i>	The method that uses three images of the sample as input for the synthesis and relighting	4.1
<i>Gradient</i>	The method that uses surface gradient and albedo maps as input for the synthesis and relighting	4.1
<i>PTM</i>	The method that uses Polynomial Texture Maps as input for the synthesis and relighting	4.1
<i>Eigen3</i>	The method that uses the first three eigen base images as input for the synthesis and relighting	4.1
<i>Eigen6</i>	The method that uses the first six eigen base images as input for the synthesis and relighting	4.1
<i>rms</i>	Root mean-squared errors	4.5
SAD	Sum of Absolute Differences	5.3
ANOVA	Friedman's nonparametric two-way Analysis of Variance	6.3.2

Chapter 1

Introduction

1.1. Motivation

The work described in this thesis is motivated by the desire for realistic texture synthesis in augmented and virtual reality applications, which play important roles in film and computer game industries. For example, as a commonly used technique to enhance realism, texture mapping normally requires an input texture of an adequate size. If the size is inadequate, texture synthesis techniques can be employed to generate a large texture using the small sample. Although repeatedly tiling the sample can produce a large image, notable seams and discontinuities will appear in the result image for many textures. The result therefore can not be perceived as a homogeneous texture. Figure 1.1.1 shows a simple example of the tiling effects and seams produced by repeating a sample image on the surface of a cubic box. Thus, the main purpose of texture synthesis is to synthesise a large texture image that is perceptually identical to the small sample for the human vision system. Recent texture synthesis techniques have been able to efficiently synthesise a wide range of real-world textures. Figure 1.1.2 shows an example; the result image is synthesised using the algorithm proposed in [Efros2001].

However, real-world textures are seldom “flat” and normally comprise rough surface geometry and various reflectance properties, which can produce dramatic effects on the appearance of the sample surfaces under varied illumination and viewing conditions. Figure 1.1.3 shows two example images of a 3D surface

texture—a piece of wallpaper illuminated from two directions. The difference is obvious. This presents challenges in both computer vision and computer graphics. It is therefore important to capture the characteristics of 3D surface textures so that new images illuminated from different directions can be produced. Photometric Stereo (PS) is one of the commonly used methods and can generate surface gradient and albedo maps from three images of a non-shadowed Lambertian surface [Woodham1981]. The surface gradient maps can be further integrated to produce a surface height map (surface profile). With the albedo and height or gradient maps, new images of a Lambertian surface under arbitrary illuminant directions can be generated. Figure 1.1.4 shows the surface height and albedo maps of a wallpaper patch.

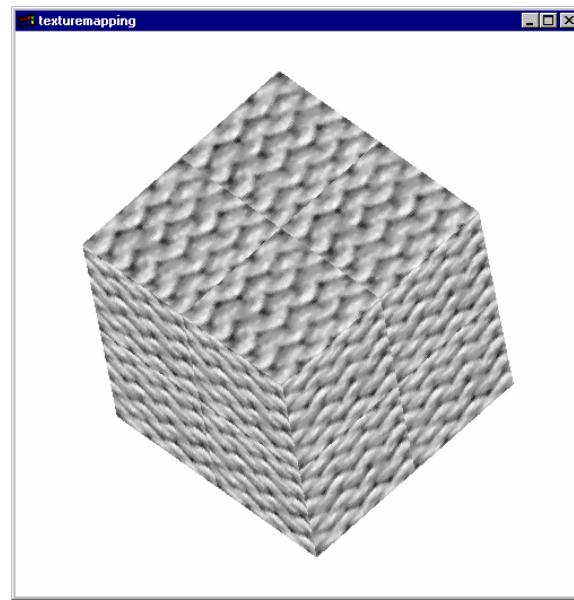


Figure 1.1.1 The tiling effects produced by mapping a texture image of an inadequate size using standard OpenGL functions. On each surface of the cubic box, the texture is repeated four times in order to cover the whole surface. Seams are obvious.

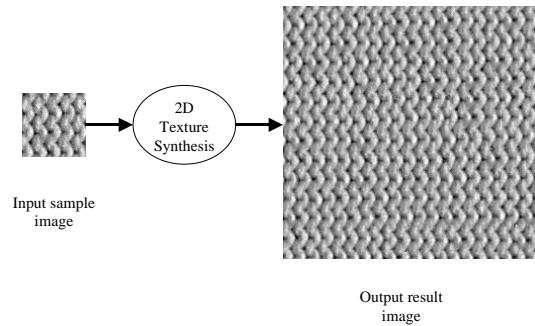


Figure 1.1.2 Texture synthesis using the algorithm proposed in [Efros2001].

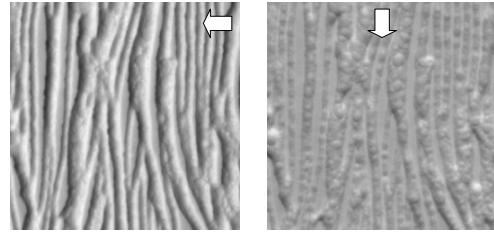


Figure 1.1.3 Two images of a 3D surface texture illuminated from different directions. The block arrows show the illuminant directions.

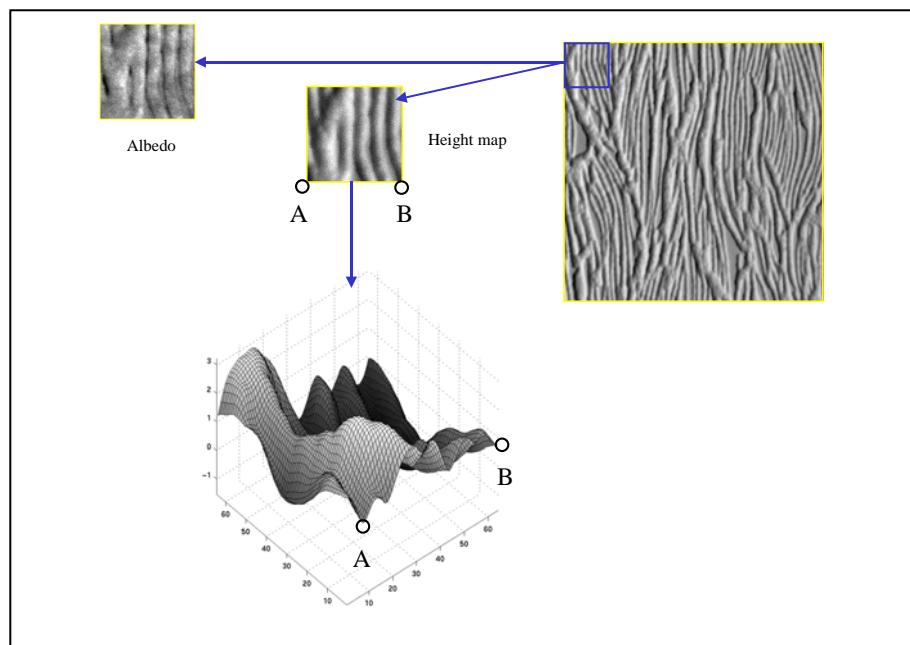


Figure 1.1.4 A 3D surface texture can be described using the surface height and albedo maps.

In recent years, texture synthesis has been extensively investigated by both computer vision and computer graphics communities. However, the input and output are normally 2D intensity images. If the subjects are 3D surface textures (such as brick, woven or knitted textiles, embossed wallpapers etc.), these 2D synthesis techniques cannot provide the information required for rendering under other than the original illumination and viewpoint conditions. This presents difficulties in realistic rendering in many augmented and virtual reality applications and has inspired the work described in this thesis—we wish to develop reliable and inexpensive methods for the synthesis of 3D surface textures.

In the research field of 3D surface texture synthesis, few publications are available [Zalesny2000, Zalesny2001, Liu2001, Shum2002 and Leung2001]. The aim of this thesis is therefore to develop inexpensive approaches for synthesising and relighting 3D surface textures. The synthesised results should be compatible with the input requirement of computer graphics packages and modern graphics systems so that real-time rendering can be achieved.

1.2. Scope of the research

In order to explicitly describe the work in this thesis and avoid confusions, we first summarise and list the definitions of commonly used terms. We then introduce the scope of our research.

1.2.1. Definitions of terms

Table 1.2.1 shows the definitions of terms used in this thesis.

Terms	Definition	First introduced
3D surface texture	Topological texture comprising 3D variation of surface geometry and reflectance	Section 1.1
2D texture	An intensity image of the sample texture.	Section 1.2
2D texture synthesis	Synthesising a large image using a small intensity image of the sample texture.	Section 1.2

	This is identical to the term <i>texture synthesis</i> .	
3D surface texture synthesis	Synthesising new texture images under different viewing and lighting conditions. The input sample data for 3D surface texture synthesis can be a set of intensity images or surface representations of the sample texture.	Section 1.1
Texture images	Images produced by illuminating 3D surface textures	Chapter 2
Input sample images	The intensity images used for 2D or 3D surface texture synthesis	Section 1.1
Output result images	The synthesised images output by 2D or 3D surface texture synthesis algorithms	Section 1.1
Photometric image set	A set of images captured under varied illumination directions using a fixed camera. Also called <i>photometric images</i>	Section 3.3
Surface representations	The set of representations extracted from a set of photometric images. They can be used to produce new images under different illumination directions. Also called surface relighting representations or representation maps.	Section 1.2
Relit images	The images produced by relighting surface representations	Section 1.3

Table 1.2.1 Definition of terms

1.2.2. Scope of the research

The work described in this thesis involves the following research:

- (i) selecting and investigating suitable surface representations for the synthesis of 3D surface textures under arbitrary illumination directions,

- (ii) selecting and investigating suitable 2D texture synthesis algorithms that can be efficiently extended to synthesise surface representations in multi-dimensional space, and
- (iii) developing and assessing 3D texture synthesis approaches.

Figure 1.2.1 shows the scope of our work described in this thesis.

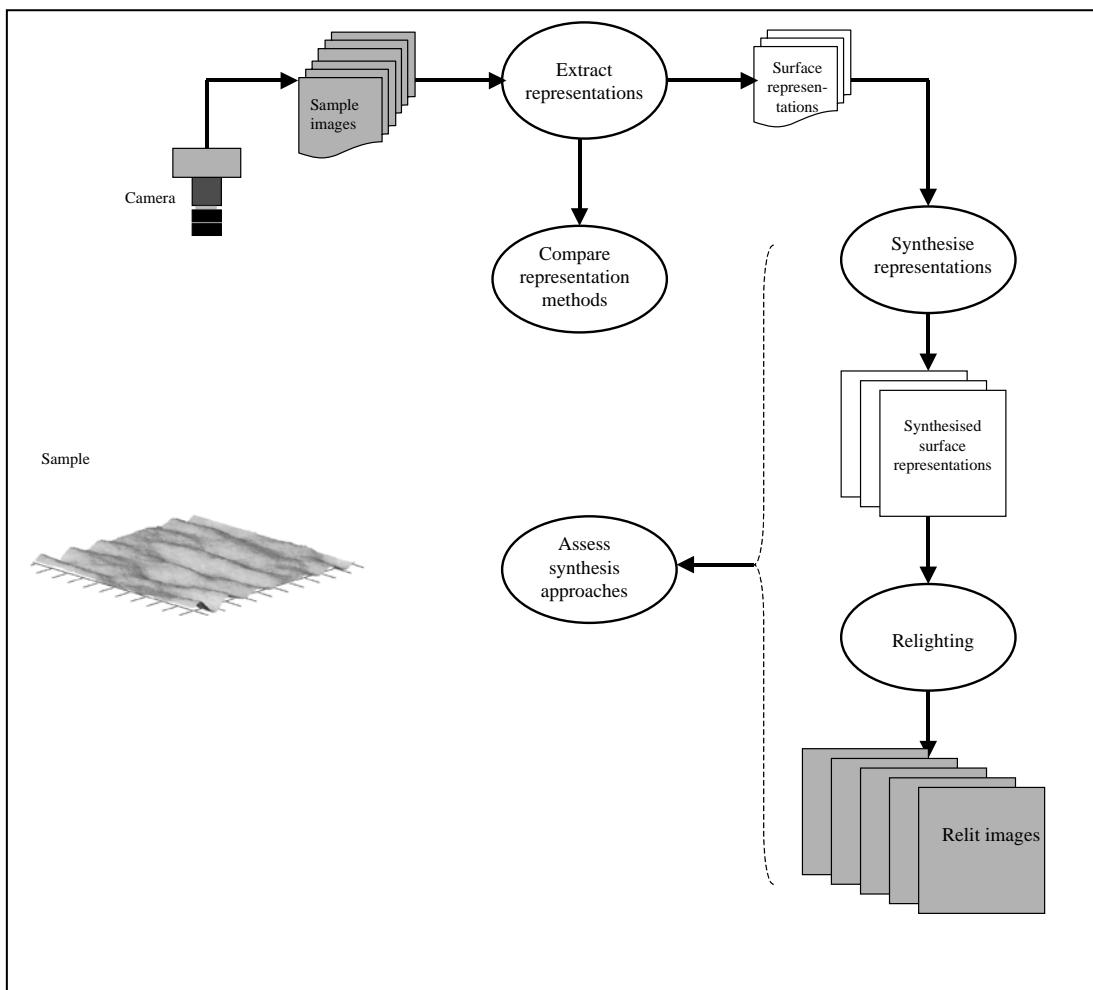


Figure 1.2.1 The scope of research in this thesis

This thesis concentrates on synthesising 3D surface textures with varied illumination directions. The illumination is assumed to be unidirectional. The viewpoint is assumed to be fixed and vertically above the surface textures, which are placed in a horizontal plane. Although using computer graphics programming techniques or software packages can achieve the effects produced by varying

viewpoints and illumination simultaneously, the description and research related to these topics are beyond the scope of this thesis. The synthesised 3D surface textures are compatible with computer graphics systems and can be effectively rendered using linear combinations in graphics hardware. However, this implementation will not be described in detail in this thesis. Readers can refer to [Robb2003] and [Burschka2003] for the latest developments on real-time graphics programming techniques using modern graphics hardware.

1.3. Thesis organisation

This thesis consists of seven chapters. Chapter 2 provides an overview of the research fields related to the thesis. Based on this survey, we present a framework for the synthesis of 3D surface textures in chapter 3. According to the framework, we then investigate and select suitable surface representations and 2D texture synthesis algorithms in chapter 4 and chapter 5. In chapter 6, we describe and compare five approaches for the synthesis of 3D surface textures. Chapter 7 summarises the work in this thesis and briefly discusses the use of synthesised results in computer graphics programming and software packages.

In chapter 2, we survey three research fields: (1) 3D surface texture synthesis, (2) 2D texture synthesis and (3) surface representation methods for relighting. Based on the survey, we conclude that there are only five publications available concerning 3D surface texture synthesis, while many techniques have been published in the other two research fields. Thus, we propose an overall framework for the synthesis of 3D surface textures in chapter 3. The framework essentially combines surface representation methods with 2D texture synthesis algorithms to synthesise 3D surface textures under arbitrary illuminant directions.

Based on the overall framework, chapter 4 reviews the available surface representations and selects five inexpensive methods for investigation. We propose a mathematical framework for the selected five methods and then describe each individual method. The performances of these methods are quantitatively assessed by comparing relit images with original real images under multiple illumination directions. We analyse the assessment results and in particular discuss the problem

associated with a heightmap-based representation, which is obtained by integrating surface gradient maps.

In chapter 5, we review the available 2D texture synthesis publications and select two methods based on [Wei2000] and [Efros2001] as the candidates of basic algorithms for 3D surface texture synthesis. We then investigate and compare the two methods in terms of the quality of synthesis results and computational complexity. The comparison shows that the algorithm based on [Efros2001] produces better performance. We therefore select it as our basic synthesis algorithm. The effects on output images produced by varying input parameters are also analysed.

In chapter 6, we describe five 3D surface texture synthesis approaches that combine the five surface representations with the basic synthesis algorithm. We then perform psychophysical experiments to qualitatively assess the performances of the five synthesis approaches, as no ground-truth data is available for a quantitative comparison.

Finally, we summarise and conclude our work described in this thesis in chapter 7. We also discuss the simple use of synthesised representations in graphics programming and software packages.

1.4. Original work

It is believed that this thesis contains the following original work:

1. An overall framework and five inexpensive approaches are proposed for the synthesis of 3D surface textures. In the literature, only five publications are available in this research field. This thesis, together with our previous publications, makes important contributions in the research field of 3D surface texture synthesis.
2. In chapter 4, a mathematical framework that summarises five surface representation methods is proposed. This framework exclusively reveals the relationships between the five surface representations using mathematical formulas.
3. In chapter 4, five surface representation methods are quantitatively compared. The comparison provides quantitative measurement for their performances in

representing 3D surface textures under different illumination directions. The problem of integrating surface gradient maps to generate surface height maps is also discussed. It is believed that it has not been investigated and reported before.

4. In chapter 6, an assessment method based on psychophysical experiments is proposed to qualitatively compare the five 3D surface texture synthesis approaches. It is believed that very little has been published on the systematic qualitative assessment of texture synthesis results.

Chapter 2

Literature Survey

The purpose of this chapter is to provide an overview of the research fields relevant to this thesis. Three fields will be surveyed; they are: (1) 3D surface texture synthesis, (2) 2D texture synthesis and (3) surface representation methods for relighting. These research fields will be reviewed in more detail later in the thesis when required by the context.

As introduced in chapter 1, *3D surface texture synthesis* techniques can synthesise new texture images under different viewing and lighting conditions. The input sample data for 3D surface texture synthesis can be a set of intensity images or representations of the sample texture. The synthesised results can be relit using illumination directions and viewing angles that differ from those used in original sample images. Few publications are available so far in this research area.

In contrast to 3D surface texture synthesis, the terminology *2D texture synthesis* is exclusively used in this thesis to refer to synthesising a large image from a small intensity image of the sample texture. Thus, this term is equivalent to *texture synthesis*, which is commonly used in computer vision and graphics communities. In this thesis, we also use *texture synthesis* to refer to *2D texture synthesis* since the former appeared in most relevant literature. There are many publications in this research area.

We use the terminology *surface representation methods for relighting* to refer to the techniques that can extract surface representations from a set of images and relight (render) these representations using illumination conditions that differ from those of the original. We also use the term *surface relighting representations*,

surface representations for relighting, or *surface representation maps* to refer to the extracted representations.

2.1. Three-dimensional surface texture synthesis

Since the main objective of this thesis is to develop inexpensive and reliable approaches for the synthesis of real-world 3D surface textures, we first present a detailed review in this area. There are only five publications that can be classified into this area. They are [Zalesny2000], [Zalesny2001], [Liu2001], [Tong2002] and [Leung2001].

Zalesny and Van Gool's work

Zalesny and Van Gool in [Zalesny2001] present a multi-view texture model which can synthesise new texture images under different viewpoints. These synthesised images can catch the effect of foreshortening due to changing viewpoints. They propose a compact model that captures the first and second order statistics of different pixel pairs, which are named *cliques* [Zalesny2000]. For each clique type, the histogram of pixel value difference is calculated. The sample texture is first modelled for a single viewpoint, typically a fronto-parallel one. The result image is initialised by an independent noise with pixel values uniformly distributed in the range of sample image. Then, different clique types are collected to form a neighbourhood structure. In order to synthesis a texture image with a novel viewpoint, the neighbourhood structure is deformed by contracting and stretching according to the angle between the two views. Clique types in the deformed neighbourhood structure are used to extract new statistical parameters—difference histograms—from the sample image with the desired viewpoint. Finally, these statistical parameters are combined with the deformed neighbourhood structure to generate the result image. During synthesis process, statistics of each clique type in the neighbourhood structure are forced to keep consistent between the result image and the sample image.

Their work did produce a compact multiview texture model that can capture viewpoint dependencies in the appearance of textures. They do not however, consider varying illumination which is the focus of this thesis.

Leung and Malik's work

The earliest publication that considers varying illumination in 3D surface texture synthesis probably is [Leung2001], in which Leung and Malik use 3D textons to represent the visual appearance of real-world surface textures. They first apply a set of linear Gaussian derivative filters on 20 images of a sample 3D surface texture with different viewing/lighting conditions (from CUReT database [Dana1999a]). Then they generate 3D textons that associate with appearance vectors containing the outputs of the filters. Each pixel in any sample image can be labelled with a 3D texton that associates with an appearance vector in a 960 dimensional space. The 3D textons can be used to reconstruct novel images under varying lighting/viewing conditions. Although they did mention that 3D textons can be used in the synthesis of 3D surface textures by modifying the 2D texture synthesis algorithm proposed in [Efros1999], the computation is very expensive because synthesis has to be performed in the 960 dimensional space. Furthermore, the algorithm in [Efros1999] uses Sum of Square Differences (SSD) as the similarity measurement, which produces large errors when matching is performed in a high-dimensional space. Few synthesis results are shown in their paper.

Liu et. al. 's work

Liu *et. al.* in [Liu2001] also exploit the CUReT database to develop a method for generating Bidirectional Texture Functions (BTFs). They firstly select and register four sample images from the CUReT image database, and then apply a shape-from-shading algorithm to recover the sample surface height and albedo maps by assuming the Lambertian reflectance. These are used to synthesise a larger height map and image *templates* by applying the 2D texture synthesis algorithm proposed in [Efros1999]. In order to produce the final image with a novel viewing/lighting condition, a reference image with the same viewing/lighting condition is selected from the BTF database and transformed into a grey scale image with the histogram equalised to that of the template image. Finally, the result image is synthesised by matching and copying blocks between the sample reference image and the template image.

Several limitations exist in Liu *et. al.* 's method [Liu2001]. Firstly, the method requires the registration of images because images in CUReT database are not registered. This is never a trivial task and can not guarantee every texture in the

database can be successfully registered. Secondly, they assume the Lambertian reflectance on the surface texture in order to perform shape-from-shading. Consequently, some real-world textures with non-Lambertian reflectance can not be used as input due to this assumption. Furthermore, applying shape-from-shading assumes integrability on the surface, which does not always hold for real-world surfaces [Tong2002]. Finally, a sample reference image has to be used to provide pixel values for the output synthesised BTFs with the desired viewing and lighting conditions. This requires additional computation and memory space to store the sample reference image. Nevertheless, this paper shows realistic rendering results for Lambertian surfaces and is the most relevant to our work described in this thesis. We show the flow chart of this work in Figure 2.1.1.

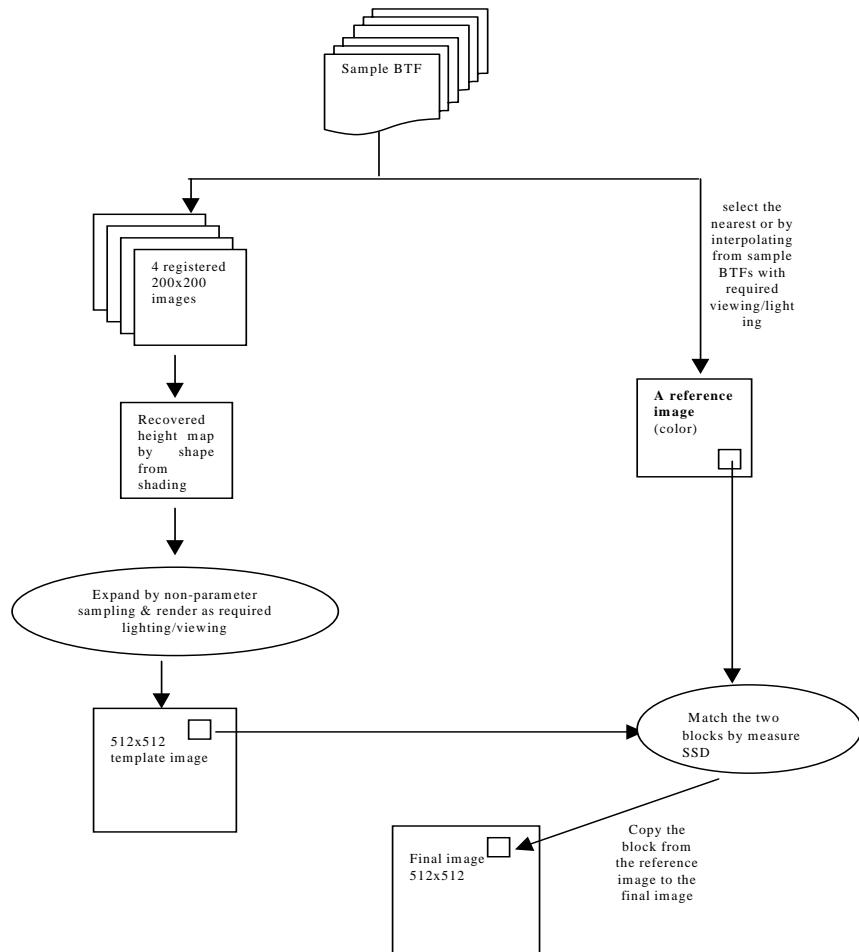


Figure 2.1.1 The flowchart of the method introduced in [Liu2001]

Tong et. al.'s work

Later work by Tong *et. al.* can synthesise BTFs on arbitrary surfaces by using surface textons [Tong2002]. Surface textons are defined by linear combinations of appearance vectors associated with 3D textons [Leung2001]. Tong *et. al.* suggest in [Tong2002] that the method proposed by [Liu2001] is not suitable for the synthesis of BTFs on arbitrary surfaces, because it is time consuming to reconstruct/render the appearance from the recovered sample geometry for all lighting and viewing settings. In addition, they suggest that it is impractical to directly synthesise 3D textons and reconstruct BTFs [Leung2001] on the surface of a 3D model because of the huge memory space required for storing appearance vectors. Thus, they pre-calculate the dot product for each pair of appearance vectors and store the results in a matrix. This matrix is then used for searching the best-matched pixel in sample BTFs for each vertex while the appearance vectors are discarded. Nevertheless, they still apply a fast searching algorithm for acceleration. The typical time consumed by their algorithm is 45 minutes for generating 3D textons and 21 minutes for synthesising a 96×96 image with 250k vertices on a 700Mhz Pentium III.

To summarise:

We have reviewed five available publications related to 3D surface texture synthesis. Zalesny and Van Gool present a multi-view texture model which can synthesise new texture images under different viewpoints with a fixed illumination direction [Zalesny2000 & Zalesny2001]. Leung and Malik propose the use of 3D textons to synthesise new images under arbitrary viewpoints and illuminations with expensive computation [Leung2001]. Liu *et. al.* apply a shape-from-shading technique to recover the surface heightmap under the Lambertian assumption and then use it for the synthesis of BTFs [Liu2001]. In later work, Tong *et. al.* introduce a method to synthesise BTFs on arbitrary surfaces by using 3D textons [Tong2002]. However, these techniques are computationally complex.

In contrast to previous work, our main objective in this thesis is to develop inexpensive approaches for the synthesis of 3D surface textures under varying illumination directions. We wish the synthesised texture representations to be capable of being loaded into graphics hardware and rendered in real-time on a modern desktop personal computer.

2.2. Two-dimensional texture synthesis

Although very few publications are available in the research field of 3D surface texture synthesis, many 2D texture synthesis techniques have been published during the past two decades. This section presents a brief survey of these 2D synthesis techniques. We will further review the relevant publications in more detail in chapter 5.

In [Xu2001], Xu *et. al.* present a short review on recent 2D texture synthesis approaches based on the underlying stochastic mechanisms employed by the sampling algorithms. Following their work, we also divide available publications on 2D texture synthesis into two groups according to sampling strategies. The first group employs global sampling strategies, which decide result pixel values by matching global statistics between the sample and result images in feature space. The second group uses local sampling strategies, which decide result pixel values by matching local statistics. Many different techniques have been used by the two sampling strategies. These techniques produce significantly different synthesis results and synthesis speeds. In later chapters, we will show that the taxonomy of 2D texture synthesis literature is related to the development of inexpensive approaches for 3D surface texture synthesis.

2.2.1. Texture synthesis methods based on global sampling strategies

A *global sampling strategy* means a texture synthesis algorithm generates result pixel values by matching global statistics between the sample and result images in feature space. The feature space is normally the multi-dimensional space spanned by feature images, which are produced by imposing a set of filters on the sample image; it may also be the 1D real space in which the pixel intensities of the sample image lies. This sampling strategy is called *ensemble sampling* in [Xu2001].

Two-dimensional texture synthesis is highly related to modelling a sample texture in terms of texture perception, which was pioneered by Julesz's conjecture. Julesz suggested that the Nth-order joint empirical densities of image pixels, e.g. the co-occurrence statistics for intensities, can statistically characterise a sample texture

[Julesz1962]. This has promoted a great deal of research in texture synthesis that employs global sampling strategies. These texture synthesis methods synthesise an output image according to statistical models. The models are derived from the sample image and employ a set of statistics. The output image is generated using the same statistics as those of the sample.

The majority of texture synthesis approaches employing global sampling strategies combine the use of statistical models with a bank of filters and multiresolution image representations. The multiresolution representations can capture long-range and nonlinear spatial interactions and therefore reduce the computational complexity. The sample image is first transformed into a multiresolution representation, and then the result image is synthesised by matching statistics across all resolutions. Heeger and Bergen use the steering pyramid and the Laplacian pyramid for texture synthesis by matching histograms between the sample and result pyramids [Heeger1995]. Their method fails to synthesise textures with distinguishable features, e.g. highly structured textures. De Bonet uses the Laplacian pyramid and analyses the input texture by computing the joint occurrence across multiple resolutions in the feature space [De Bonet1997]; the output texture is generated by sampling successive spatial frequency bands from the input texture, conditioned on the similar joint occurrence of features at all lower spatial frequencies. Van Nevel develops a texture synthesis method that relies on matching the first and second order statistics of wavelet subbands [Van Nevel 1998]. Based on joint statistics of complex wavelet coefficients in the multiresolution framework, Portilar and Simoncelli propose a parametric texture model that can synthesise a wide range of artificial and natural textures [Portilla2000]. In [Copeland2001], Copeland *et. al.* use the gray-level co-occurrence (GLC) model coupled with multiresolution data structure for texture synthesis. They also employed ten human observers to test the correlation between the synthesis results and their texture similarity metric by performing psychophysical experiments. In [Campisi2002], the Circular Harmonic Functions are used to develop a multiresolution approach for texture synthesis. It essentially extends previous work in [Jacovitti1998] by using multiresolution decomposition.

There are also several texture synthesis methods that employ statistical models derived from filtered images without explicitly using multiresolution image representations. Eom proposes a 2D moving average (MA) model for texture synthesis and analysis [Eom1998], and the result image is generated in frequency domain by using estimated parameters of the 2D MA model. Jacovitti *et. al.* use hard-limited Gaussian process to develop a twin stage texture synthesis-by-analysis [Jacovitti1998]. Zhu *et. al.* present a definition of texture as the *Julesz ensemble*, which is the set of all images sharing identical statistics, and texture synthesis is achieved by sampling the ensemble using a Markov chain Monte Carlo algorithm [Zhu2000]. Histograms of feature images are employed in their approach.

To summarise:

For 2D texture synthesis, a *global sampling strategy* decides result pixel values by matching global statistics between the sample and result images in feature space. Among 2D texture synthesis approaches employing global sampling strategies, the majority apply multiresolution decomposition techniques and impose filters in multiresolution domain to generate the statistical descriptions of the sample image. The synthesis is then performed by matching statistics across multiple resolutions in feature space [Heeger1995, De Bonet1997, Van Nevel1998, Portilla2000, Copeland2001 & Campisi2002]. Only few methods directly apply a bank of filters on the sample image without explicitly using multiresolution decomposition; the result image of these methods is synthesised by matching statistics in feature space [Eom1998, Jacovitti1998 & Zhu2000]. Table 2.2.1 shows the summary of typical texture synthesis methods employing global sampling strategies.

Table 2.2.1 Characteristics of typical global sampling methods

Reference	Global statistics	Number of pyramid levels	Iterations	Complexity/time-consumed/speed
[Heeger1995]	Marginal histograms	4	5	Faster than [Portilla2000]
[DeBonet1997]	Joint occurrence of features	Depends on the sample size	1	Slower than [Heeger1995]
[Eom1998]	Moving average model parameters, elongation and orientation parameters	1	1	unspecified
[Nevel1998]	Mean, histograms and the correlation matrix	3	1	2 minutes for 400 largest entries in the correlation matrix using a Sun UltraSparc
[Portilla2000]	Marginal Statistics, coefficient correlation, magnitude correlation and cross-scale phase statistics	3	50	20 minutes for a 256x256 image using 500Mhz Pentium workstation
[Zhu2000]	Marginal histograms of filtered responses	Unspecified	20 to 100	Slower than [Portilla2000] according to [Xu2001]
[Copeland2001]	Co-occurrence matrix	3	5 (spin-flip algorithm)	2.5 minutes using a Silicon Graphics Indy with a IP22 processor
[Campisi2002]	First and second order statistics	3 to 7	2 or 3	Computational complexity depends on the number of filters and iterations. Time-consumed is not specified.

2.2.2. Texture synthesis methods based on local sampling strategies

A *local sampling strategy* means the texture synthesis algorithm generates result pixel values solely by using local information in the sample and result images. A typical example is to compute local conditional distributions using certain neighbourhoods and synthesise pixels in the result image in raster order. The majority of texture synthesis methods with local sampling strategies make certain statistical assumptions. We further divide these synthesis approaches into two subclasses. One sub-class explicitly uses parametric statistical models for the synthesis. The other uses non-parametric methods.

Representative texture synthesis approaches using local sampling strategies and parametric models include [Cross1983, Popat1993, Bader1995, Zhu1998, Zhang1998b and Kokaram2002]. These methods first estimate the parameters of the assumed statistical models for the input sample image, and then synthesise the result image using the statistical models. Cross and Jain use Markov random field models to represent the sample image [Cross1983]. Popat and Picard present a method that first performs clustering analysis on the sample data and then calculates the probability mass function using Gaussian parameters for texture synthesis [Popat1993]. Bader *et. al.* propose the use of scalable data parallel algorithms for the 2D texture synthesis using Gibbs random fields [Bader1995]. Zhu *et. al.* develop a Markov random field model based on feature images, which are produced by a bank of filters with large image lattice; the result image is synthesised by using a Gibbs sampler [Zhu1998]. Zhang *et. al.* exploit the wavelet autoregressive model and radial basis functions in a multiresolution domain for texture synthesis [Zhang1998b]. Kokaram estimates the parameters of 2D autoregressive models and uses the models to synthesise missing gaps in images [Kokaram2002].

Non-parameteric texture synthesis approaches have the advantage that the estimation of parameters in statistical models is not necessary. Thus, the computational complexity is normally lower compared with their parametric counterparts. In particular, the method proposed by Efros and Leung is widely used

in texture synthesis research [Efros1999]¹. It assumes a Markov random field model and calculates the conditional distribution of a pixel given all its neighbours by querying the sample image and finding all similar neighbourhoods. It further inspired the work in [Wei2000], which improved the performance of the original algorithm by employing a multiresolution image representation and an accelerating algorithm. The methods in these two publications can produce excellent results while simplifying the whole synthesis process. Based on these two algorithms, several texture synthesis approaches have been developed and applied in different areas [Hertzmann2001, Efros2001, Parada2001, Ashikhmin2001, Harrison2001, Tonietto2002, Zelinka2002, Cohen2003, Zhang2003 and Nealen2003].

Other typical non-parametric approaches unrelated to the two algorithms proposed in [Efros1999 and Wei2000] include [Paget1998, Ashlock1999, Bar-Joseph2001, Xu2001, Liang2001 and Gousseau2002]. In [Paget1998], Paget and Longstaff propose a non-causal, non-parametric and multiscale Markov random field model for 2D texture synthesis; they employ the Parzen-window density to estimate the frequency of occurrence. In [Ashlock1999], generic algorithms are used to track the basic texture elements and produce a non-parametric partially ordered Markov random field model for texture synthesis. In [Bar-Joseph2001], Bar-Joseph *et. al.* construct a tree representation of the input signal in multiresolution domain and generate a new tree representation by learning and sampling the conditional probabilities of the paths in the original. Their method can synthesise static and time-varying textures. In [Xu2001], a patch-pasting algorithm is introduced for the fast texture synthesis. Later work in [Liang2001] extends it by sampling patches according to a non-parametric estimation of the local conditional MRF density function; the performance is also improved. More recently, Gousseau presents a texture synthesis method by sampling from level sets [Gousseau2002].

To summarise:

Texture synthesis approaches based on local sampling strategies have attracted much attention in recent years. Several parametric methods have been proposed to firstly model the sample image and then synthesise the result using the

¹ Note: in [Efros2001], it has been pointed that a nearly identical algorithm was proposed in [Garber1981] but discarded due to its then computational intractability.

parameters [Popat1993, Bader1995, Zhu1998, Zhang1998 and Kokaram2002]. However, many researchers employ non-parametric methods that are capable of producing promising results with less computation [Efros1999, Wei2000, Hertzmann2001, Efros2001, Parada2001, Ashikhmin2001, Tonietto2002, Bar-Joseph2001, Xu2001, Liang2001, Gousseau2002, Zelinka2002, Cohen2003 and Nealen2003]. In particular, the algorithms in [Efros1999 and Wei2000] have promoted further work in different research directions.

2.2.3. Summary

In section 2.2.1 and 2.2.2, we reviewed recent publications on 2D texture synthesis. These publications can be divided into two classes depending on whether global or local sampling strategies are used. Most texture synthesis approaches with global sampling strategies synthesise a result image by matching global statistics in feature space and multiresolution domain. Among texture synthesis methods with local sampling strategies, both parametric models and non-parametric models can be used. Recent publications suggest that some non-parametric texture synthesis methods can produce good synthesis results with less computation.

2.3. Surface representation methods for relighting

As introduced in chapter 1, varying the illumination directions can produce significant effects on images of a 3D surface texture. These images can exhibit remarkable differences, which present challenges in both computer vision and computer graphics. It is therefore important to extract surface representations of the sample texture under arbitrary illumination directions. Once the representations are available, they can be relit to generate new images with arbitrary lighting conditions. This section briefly reviews relevant publications in this research area, which involves reflectance distribution modelling, model-based and image-based relighting (rendering) techniques.

2.3.1. Extracting surface relighting representations using reflectance models

The most accurate surface relighting representations can be described by Bidirectional Reflectance Distribution Functions (BRDF) [Nicodemus1977]. With full BRDF data, images of the sample surface or object under arbitrary illumination can be produced. However, full BRDF data are difficult to obtain because the measurement of BRDF is very expensive and time-consuming. Various local-based reflectance models have been used in computer vision and computer graphics to describe how lights are reflected from a surface and reach to the observer. Commonly used models include the Lambertian model, the Torrance-Sparrow model [Torrance1967], the Phong model [Phong1975], the Cook-Torrance model [Cook1982], the Nayar model [Nayar1991] and other models [He1991 and Oren1994]. Obviously, extracting surface representations using reflectance models is equivalent to estimating the models' parameters, which normally represent surface geometric and material properties. However, these models can only be seen as approximations of the ground-truth, as the physics of light reflection involves extremely complicated nonlinear processes.

Methods for estimating the parameters of reflectance models has been extensively investigated in recent years. Photometric stereo is one of the major techniques used to obtain surface geometric and material properties [Woodham1981, Horn1989, Nayar1990, Kay1995, Rushmier1997, Saito1996 and Lin1999]. This approach requires a fixed camera, several lighting conditions and a static object. Traditional photometric stereo methods assume the Lambertian reflectance function and use three images to obtain surface gradient maps and an albedo map [Woodham1981 and Horn1989]. If the sample surface exhibits both diffuse and specular components, more complex reflectance models are required. Consequently, more images are needed in order to estimate the parameters [Nayar1990, Kay1995, Rushmier1997, Saito1996 and Lin1999]. By firstly separating diffuse and specular components, both diffuse and specular parameters can be estimated using photometric stereo techniques. The combined use of range and intensity data is another popular technique that can be used to extract surface relighting representations from existed reflectance models [Ikeuchi1991, Lu1995, Sato1997,

Ramamoorthi2001 and Nishino2001]. For example, Sato *et. al.* use multiple range images to recover surface shape and then estimate reflectance parameters of the Torrance-Sparrow model [Sato1997]. Polarisation techniques can also be used to separate reflection components so that surface representations can be estimated [Nayar1996].

The surface geometric representations estimated from reflectance models are usually expressed as surface normals or surface gradient maps. Extracting surface normals from an intensity image is also the aim of shape-from-shading [Horn1989]. Integration techniques can be further used to obtain the depth information or the height map from surface normals [Klette1996]. Both local and global integration approaches have been proposed in the past [Coleman1982 and Frankot1988]. Global approaches are more robust to noise than local approaches [Gullón2002].

2.3.2. Extracting surface relighting representations using other techniques

There are also a great number of other techniques that can be used to obtain surface relighting representations without directly employing reflectance models. The surface relighting representations derived from these techniques are not, in general, geometrical and material properties of the surface.

Image-based relighting (rendering) techniques can generate realistic images from pre-recorded images without using complex rendering processes as in geometry-based computer graphics [Kang1997, McMillan1999, Koudelka2001, Lin2002, Matusik2002 and Wong2002]. In [Kang1997], Kang presents a survey on early image-based rendering techniques. In [Matusik2002], Matusik *et. al.* introduce a system that can acquire and display high quality graphical models of objects using opacity hulls; both effects produced by changing view and illumination conditions are considered. In [Wong2002], Wong *et. al.* define the plenoptic illumination function that can relight images while supporting view interpolations. However, many image-based rendering techniques can only synthesise new images under different viewpoints, while the illumination remains fixed [Chen1995, Levoy1996 and Gortler1996].

The representation of varied BRDF on a surface requires numerous sample images. Researchers have developed several methods to approximate this model by projecting these images into general base functions so that the representation is more compact for practical applications [Lalonde1997, Lafourture1997 and McAllister2002]. Lalonde and Fournier use wavelet coefficients to represent large anisotropic BRDF data sets [Lalonde1997]. The Lafourture representation consists of a diffuse component and several specular lobes which are generalised Phong lobes [Lafourture1997]. McAllister *et. al.* employ the Lafourture representation for rendering the Spatial BRDFs using register combiners in an Nvidia Geforce 4 graphics card [McAllister2002].

Eigen-based methods are broadly used to extract surface relighting representation [Epstein1995, Zhang1998a, Georghiades1999 and Nishino2001]. These methods apply principal component analysis (PCA) or singular value decomposition (SVD) on a set of pre-recorded images and extract base images as the surface relighting representations. New images under arbitrary illumination directions can be generated by linearly combining these base images. Obviously, eigen-based approaches also belong to the class of image-based techniques. In addition, they can be used in pattern recognition and image impression [Nishino2001, Turk1991 and Belhumeur1997].

In the literature regarding surface representation methods, many other mathematical models are also exploited to express the sample images as linear or nonlinear combinations of a set of base functions, such as Fourier Series [Huang1984 and McGunnigle2001], spherical harmonics [Basri2001 and Ramamoorthi2001] and steering functions [Ashikhmin2002]. These base functions normally form base images and can be used to synthesise new images under arbitrary illumination conditions.

2.3.3. Extracting 3D surface texture representations for relighting

Rough surface textures can be seen as a finer scale geometric description with regular or random components. In theory, methods surveyed in section 2.3.1 and 2.3.2 can all be used to extract relighting representations of 3D surface textures.

Nevertheless, researchers have proposed special methods to represent 3D surface textures under arbitrary illumination directions.

Representing the appearance of 3D surface textures only received attention in recent years [Koenderink1996, Stavridi1997, Dana1999a, Dana1999b, Leung2001, Malzbender2001 and Ashikhmin2002]. In [Dana1999a], Dana *et. al.* define Bidirectional Texture Function (BTF) that can represent 3D surface textures under varied illumination and viewing directions; they construct the CUReT database that contains many images from over 60 samples. Dana and Nayar further investigate three BTF models, including the histogram model, the correlation model and PCA models [Dana1999b]. Leung and Malik exploit the CUReT database and employ a bank of 48 filters coupled with clustering analysis to derive 3D textons, which can be used to represent and recognise the visual appearance of 3D surface textures [Leung2001]. Malzbender *et. al.* propose a quadratic lighting model that uses Polynomial Texture Maps(PTM) to reconstruct the surface colour under varying lighting conditions [Malzbender2001]. Ashikhmin uses a set of steering basis functions for relighting bumpy surfaces [Ashikhmin2001].

2.3.4. Summary

We have presented a brief review of methods that can be used to extract surface relighting representations from a set of pre-recorded images. As the most compact representations, surface geometric and material properties can be obtained by estimating the parameters of various locally-based reflectance models [Woodham1981, Horn1989, Nayar1990, Kay1995, Rushmier1997, Saito1996, Lin1999, Ikeuchi1991, Lu1995, Sato1997, Ramamoorthi2001 and Nishino2001]. They can then be relit using corresponding reflectance models to generate new images under different illumination conditions. Image-based relighting/rendering are also commonly used techniques that can convert the pre-recorded images into relighting representations [Kang1997, McMillan1999, Koudelka2001, Lin2002, Matusik2002 and Wong2002]. Other methods employ mathematical models to express a set of sample images using linear or nonlinear combinations of basis functions, such as eigen-based methods [Epstein1995, Zhang1998a,

Georghiades1999, and Nishino2001], Fourier serious [Huang1984] and spherical harmonics [Basri2001].

In recent years, special interest has been given to the research into representing the appearance of 3D surface textures. Several methods have been proposed and shown great promise in computer vision and computer graphics [Koenderink1996, Stavridi1997, Dana1999a, Dana1999b, Leung2001, Malzbender2001 and Ashikhmin2002].

2.4. Conclusion

This chapter has briefly reviewed the related research fields to this thesis. These comprise the literature on:

- (1) 3D surface texture synthesis approaches,
- (2) 2D texture synthesis approaches, and
- (3) surface representation methods for relighting.

Based on this survey, we conclude that very few publications are available regarding 3D surface texture synthesis, while there are a great number of methods in the fields of 2D texture synthesis and the extraction of surface representations for relighting.

Among the 3D surface texture synthesis approaches, Zalesny and Van Gool's work can only synthesise new images with varied viewpoints, while the illumination direction is fixed [Zalesny2000 and Zalesny2001]. Liu *et. al.* use a 2D texture synthesis algorithm based on [Efros1999] and Lambertian surface representations for the synthesis of BTFs [Dana1999a]; In [Tong2002] and [Leung2001], a 2D texture synthesis algorithm based on [Efros1999] and the 3D texton representations are combined for the synthesis of BTFs. However, these methods require expensive computation.

In contrast, our main objective in this thesis is to develop inexpensive approaches for the synthesis and relighting of 3D surface textures. In next chapter, we will introduce a basic framework that can combine 2D synthesis approaches with surface representation methods in a methodical manner to synthesise new texture images under arbitrary illumination directions.

Chapter 3

Framework

3.1. Introduction

The goal of this thesis is to develop inexpensive approaches for synthesis and relighting of 3D surface textures. In chapter 2, we presented a survey and showed that few publications are available regarding 3D surface texture synthesis. However, many surface representation techniques and 2D texture synthesis methods have been published in recent years. The aim of this chapter is therefore to propose an overall framework for the synthesis and relighting of 3D surface textures. The framework will be capable of combining 2D texture synthesis methods with surface representation techniques in a methodical manner. Based on this framework, we will define the data environment that we need for all experiments in the thesis.

The overall framework comprises three parts:

1. extraction of a 3D surface representation from multiple images of the texture sample;
2. use of the representation to synthesise a description of a larger area of the surface texture; and
3. rendering (or relighting) of the synthesised surface representation according to a specified set of lighting conditions.

For assessment purposes, we employ a set of images selected from the PhoTex database [McGunnigle2001]. The database contains many images per

texture that have been captured under varied illumination directions. These images form the basic data environment for this thesis.

The chapter is organised as follows. Section 3.2 introduces the framework. Section 3.3 describes the data environment that we use for assessment. Finally we summarise the content of the whole chapter in section 3.4.

3.2. A framework for the synthesis of 3D surface textures

In this section, we introduce the overall framework for the synthesis of 3D surface textures.

3.2.1. Framework

The synthesis of 3D surface textures naturally deals with more information than its 2D counterpart. The latter requires consistent texture patterns to be generated in a single image that has no perceptual difference from the original. Synthesis is therefore performed in an \mathbf{R}^1 (monochrome) or \mathbf{R}^3 (colour) space. In contrast, 3D surface texture synthesis either implicitly or explicitly requires generation of surface geometric information and reflectance properties. A single sample image, which is used as input in 2D texture synthesis, does not normally provide enough information regarding surface topology and reflectance. Thus we have to employ multiple images (or their representations) of the sample 3D surface texture, which contain enough information regarding geometric and reflectance properties as the input data. Furthermore, we would like the sample image data set to be captured in an inexpensive way, i.e. using off-the-shelf digital cameras, and the synthesis results to be capable of being rendered in real-time on current desktop machines.

Our main goal is to develop inexpensive approaches for the synthesis and relighting of 3D surface texture. However, the original image set is normally of large dimension. It is impractical to directly synthesise the original image set in a large dimensional space, because the time consumed by texture synthesis algorithms increases linearly with the dimensionality of input sample vectors [Xin2002][Efros1999][Turk2001]. On the other hand, surface representation techniques can be used to convert the original image set into relighting representations of low dimension. For example, Malzbender *et. al.* used a method to

convert 50 images into six Polynomial Texture Maps [Malzbender2001]. A PCA-based method suggests 5 ± 2 eigenimages are sufficient to represent arbitrary lighting for the Lambertian and specular lobe although complex reflectance will increase the dimensionality [Epstein1995]. Thus, 3D surface texture synthesis can use the representations of the sample texture which are of lower dimension as input.

Therefore, our framework for the synthesis of 3D surface textures comprises 3 stages:

Stage 1: Extraction of the 3D surface representation

The first stage is to extract a suitable relighting representation of the sample 3D surface texture. Surface representation (i.e. image-based relighting) methods can be applied to a set of images captured under different illumination directions. The representation should be of a low dimension and preferably capable of per-pixel-rendering using a linear combination, which requires less computation and is compatible with modern graphics systems. Many techniques may be used for this stage, as reviewed in chapter 2. A simple example is that we may generate two surface gradient maps and an albedo map by applying traditional photometric stereo techniques [Woodham1981] and use them to represent a Lambertian surface. Thus, each pixel location will be represented by a multi-dimension vector.

Stage 2: Synthesis of the 3D surface representation

In the second stage, we need to select and modify a 2D texture synthesis algorithm so that it can use multi-dimensional vectors as input and perform the synthesis in \mathbf{R}^n space, where n is the dimensionality of the surface representation. In the case of a Lambertian surface, we can use two surface gradient maps $p(x, y)$, $q(x, y)$ and an albedo map $al(x, y)$ to represent the sample 3D surface texture; the synthesis will take the three dimensional vector $V = (p(x, y), q(x, y), al(x, y))$ as input, where (x, y) is the pixel co-ordinate.

During the synthesis process, it is important to preserve the correlation between representation maps. Suppose we are synthesising surface gradient and albedo maps and the pixel values in the result representations originate from those in the samples. At a pixel location (x'_0, y'_0) in the result map, the surface gradient values $p(x'_0, y'_0)$, $q(x'_0, y'_0)$ and albedo value $al(x'_0, y'_0)$ must derive from the sample

surface gradients $p(x_0, y_0)$, $q(x_0, y_0)$ and albedo $al(x_0, y_0)$ at the same location (x_0, y_0) on all input sample maps.

Stage 3: relighting

This is the final stage of the overall framework. We relight the synthesised representations using a specified set of lighting conditions to produce desired texture images. Relighting can be seen as an inverse process to the extraction of the representation maps. Recall the previous example. Since we extract surface gradient and albedo maps using the Lambertian reflectance model, the relighting will use the same model as well.

Figure 3.2.1 illustrates the overall framework.

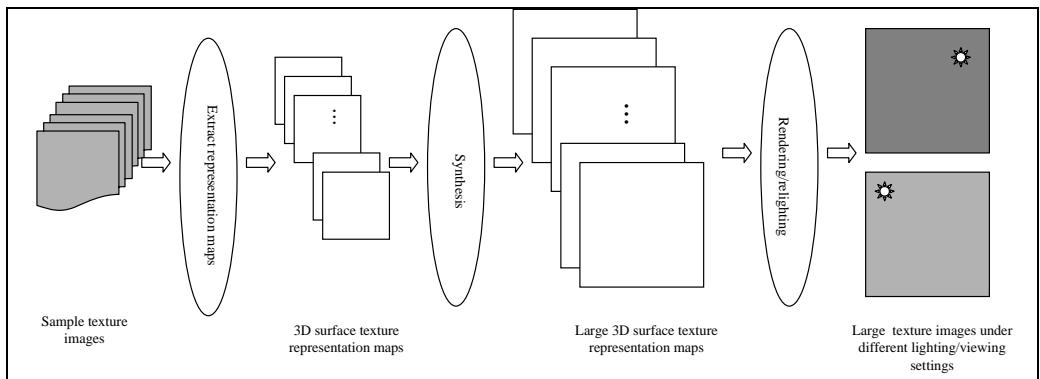


Figure 3.2.1 The overall framework for the synthesis of 3D surface textures

3.2.2. An example: the approach of [Liu2001]

Previous work on 3D surface texture synthesis can be represented within the proposed framework. We present an example by using the method introduced in [Liu2001].

Shum and his colleagues [Liu2001] exploited the CURet database to develop a method for the synthesis of Bi-directional Texture Functions. In the first stage, they applied a shape-from-shading algorithm to recover sample surface height and albedo maps under the assumption of Lambertian reflectance. Next, the 2D texture synthesis algorithm proposed in [Efros1999] is used to produce a large height map. In the final (relighting) stage, the synthesised height map is rendered to generate image *templates*. A reference image with desired lighting/viewing conditions is selected from the sample BTFs. The result image is obtained by matching and

copying blocks between template images and the reference sample images. Figure 3.2.2 shows how this method relates to our framework.

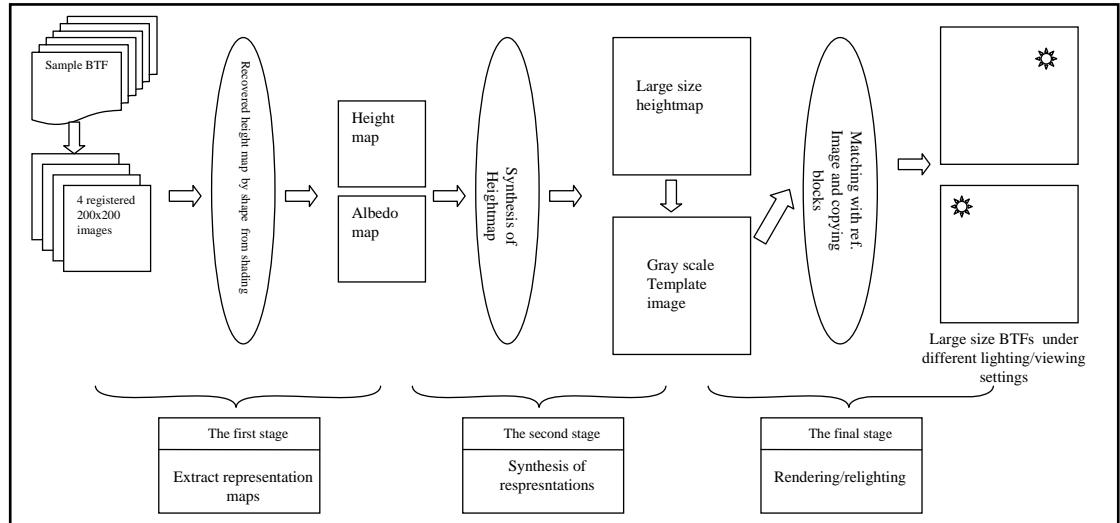


Figure 3.2.2 The method introduced in [Liu2001] can be represent within our framework.

3.2.3. Discussion

It should be noted that we could use many alternative techniques in different stages of the framework. In the first stage, many methods are available to directly obtain relighting representations of the sample surface (without applying image-based techniques) [Rushmire1998]. For example, 3D surface geometry can be acquired by using a 3D scanner; BRDF data of the sample texture might be measured although it is particularly difficult. Once we have the sample geometry and BRDF data, we can synthesise new geometry and corresponding BRDF data for a large surface area. Then the illumination techniques may be applied to produce the final images.

3.3. The image data environment for the thesis

According to the proposed framework, we firstly need to extract a suitable representation of the sample texture from a set of pre-recorded images. This section will therefore introduce the image data sets that we will use throughout the thesis.

In this thesis, we use a set of images selected from the PhoTex database (<http://www.cee.hw.ac.uk/texturelab/database/photex/>). The database contains many rough surface texture samples. For each texture sample, many images were captured using a fixed camera (a Vosskühler CCD 1300LN) while the rough surface was illuminated from various directions. Figure 3.3.1 shows the experimental set-up. The origin of the co-ordinate is at the centre of the sample surface. The camera is perpendicular to the sample surface, which is globally flat and lies in the x - y plane. The camera's line of sight (axis) overlaps with the z axis. Thus, the illuminant direction is defined by a slant angle and a tilt angle. Slant is the angle between the z axis and the illuminant vector; tilt is the angle between the x axis and the vector produced by projecting the illuminant vector onto the x - y plane. More details about the PhoTex database can be found on its website.

All images in the database are monochromatic with 1280x1024 resolution. Each pixel is stored in 12 bits. We call the images in the database *photometric* images, where the term *photometric* is in the context of photometric stereo: i.e. inferring information about a static scene by altering the lighting conditions [Woodham1981]. With image data sets selected from this database, surface representations can be extracted using various methods.

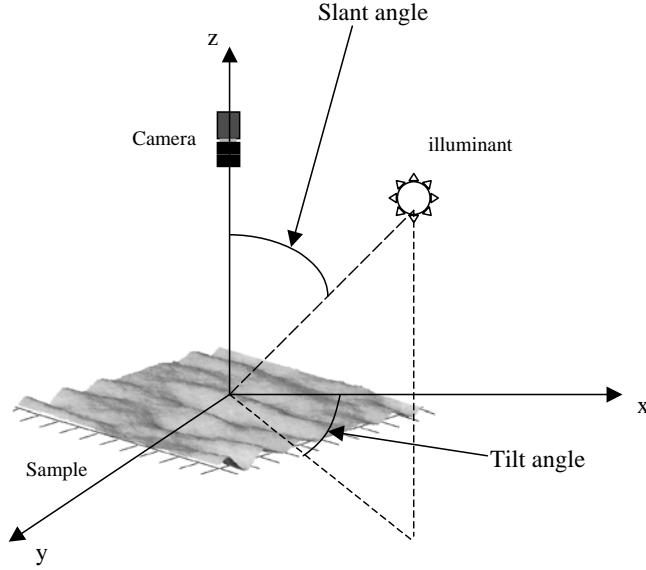


Figure 3.3.1 The imaging set-up and definitions of the slant and tilt angle.

We select image data sets according to two criteria. One is that the sample should comprise suitable granularity. Obviously, surface appearance can only be described by *texture* at certain scales [Dana1999a]. Large granularity in the sample tends to be perceived as individual elements. Suppose we are interested in the texture of rough rock surface. We would like the captured surface to cover a large area with small granularity on the rock. It makes no sense to focus on a tiny surface patch that has large granularity in the image. Thus, if original images in the database contain large granularity, it may be necessary to downsample. The other criterion is that selected texture types should cover a wide range. This is especially important if we want to evaluate our methods on real textures. The selected textures should include:

1. rough and smooth surfaces,
2. glossy and matte surfaces,
3. non-shadowing and shadowing surfaces, and
4. near-regular and stochastic patterns.

In Appendix A, we show sample images of all the selected textures.

3.4. Conclusion

In this chapter, we proposed a practical framework for the synthesis of 3D surface textures and introduced the image data environment for the thesis. The framework can combine surface representation techniques and 2D texture synthesis methods in a methodical manner to synthesise 3D surface textures. It comprises three stages: the first stage converts a set of pre-recorded images into a surface representation of a lower dimension; the second stage employs a 2D texture synthesis algorithm and extends it to \mathbf{R}^n space; the final stage renders the synthesised representations according to a set of lighting conditions. Many surface representation techniques and 2D texture synthesis methods can be used in the framework.

Based on the framework, we introduced the image data environment used for all experiments in this thesis. We exploit the PhoTex database and select image data sets according to two criteria: the granularity and the need for a range of texture types.

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-dimensional 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 Lafourture1997].

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-dimension 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 Lafourte BRDF representations, which is capable of representing Fresnel reflection, off-specular peak and retro-reflection, to perform real-time rendering in graphics hardware [Lafourte1997 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.

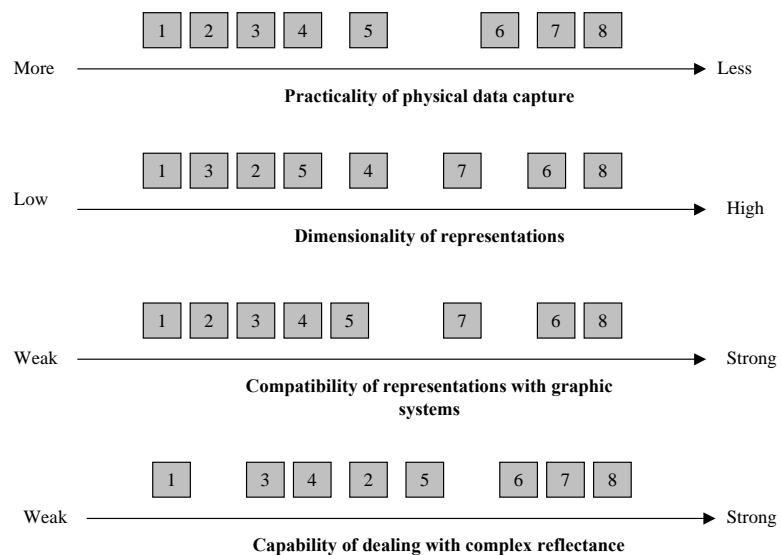


Figure 4.3.1 Different representations v.s. criteria. (1)Estimated surface geometry and reflectance parameters using reflectance models [Woodham1981]; (2) Eigen-based methods [Epstein1995]; (3) Polynomial Texture Maps

(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 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

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

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

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

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)

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

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 $1 \leq j \leq m$ and $1 \leq k \leq n$. 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

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} & 5 & i_{1n} \\ i_{21} & i_{22} & 5 & i_{2n} \\ 7 & 7 & 5 & 7 \\ i_{m1} & i_{m2} & 5 & i_{mn} \end{bmatrix} \quad (4.4.3)$$

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:

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

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 \mathbf{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 \mathbf{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 \mathbf{c} related to the required illumination direction:

$$\mathbf{i} = \mathbf{M}_1 \mathbf{c} \quad (4.4.5)$$

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

4.4.2. Lambertian methods--3I and Gradient

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

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

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:

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

$s(x, y)$ 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 \mathbf{I} can be expressed as:

$$\mathbf{I} = \mathbf{A} \mathbf{N} \mathbf{L} \quad (4.4.7)$$

where:

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

is the surface albedo matrix;

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

is the surface normal matrix;

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

is the lighting matrix.

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

$$\mathbf{N}_a = \mathbf{A} \mathbf{N}.$$

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

$$\mathbf{I} = \mathbf{N}_a \mathbf{L} \quad (4.4.8).$$

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 \mathbf{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

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

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

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ 7 & 7 & 7 \\ i_{m1} & i_{m2} & i_{m3} \end{bmatrix},$$

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

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

where $\mathbf{i} = (i_1, i_2, \dots, i_m)^T$ 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

$$\mathbf{i} = \mathbf{M}_1\mathbf{c} = \mathbf{I}\mathbf{c} \quad (4.4.10)$$

which means an image under a given lighting vector can be expressed as a linear combination of three images. The vector \mathbf{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 \mathbf{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:

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

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

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

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

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 \mathbf{I} becomes a known $m \times 36$ matrix and the lighting matrix \mathbf{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:

$$\mathbf{L} = \mathbf{U}_L \mathbf{W}_L \mathbf{V}_L^T.$$

Then we have

$$\mathbf{N}_a = \mathbf{I}\mathbf{L}^{-1} = \mathbf{I}\mathbf{V}_L \mathbf{W}_L^{-1} \mathbf{U}_L^T$$

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:

$$\mathbf{i} = \mathbf{N}_a \mathbf{l}$$

where $\mathbf{i} = (i_1, i_2, \dots, i_m)^T$ 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 lighting 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

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

The image data matrix is expressed as

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

$$\text{where } \mathbf{A}_{\text{ptm}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ 7 & 7 & 7 & 7 & 7 & 7 \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{m5} & a_{m6} \end{bmatrix} \text{ and } \mathbf{L}_{\text{ptm}} = \begin{bmatrix} 1 & 1 & 6 & 1 \\ l_{y1} & l_{y2} & 6 & l_{yn} \\ l_{x1} & l_{x2} & 6 & l_{xn} \\ l_{x1}l_{y1} & l_{x2}l_{y2} & 6 & l_{xn}l_{yn} \\ l_{y1}^2 & l_{y2}^2 & 6 & l_{yn}^2 \\ l_{x1}^2 & l_{x2}^2 & 6 & l_{xn}^2 \end{bmatrix}$$

Each row of matrix \mathbf{A}_{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}_{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 (4.4.4) 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 \mathbf{I} and the lighting matrix \mathbf{L}_{ptm} are known, we can use SVD to solve the over-determined system (4.4.11) and obtain the PTM matrix \mathbf{A}_{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

$$\mathbf{i} = \mathbf{M}_1 \mathbf{c} = \mathbf{A}_{\text{ptm}} \mathbf{l}_{\text{ptm}}$$

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

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

$$\mathbf{I} = \mathbf{U}_I \mathbf{W}_I \mathbf{V}_I^T$$

Each column in \mathbf{U}_I therefore is an eigen vector of $\mathbf{I}\mathbf{I}^T$ corresponding to the singular value in \mathbf{W}_I . \mathbf{U}_I is used to construct eigen base images and \mathbf{V}_I^T contains coefficients for linear combinations. We can write $\mathbf{W}_I = \text{diag}(w_1, w_2, \dots, w_n)$, 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

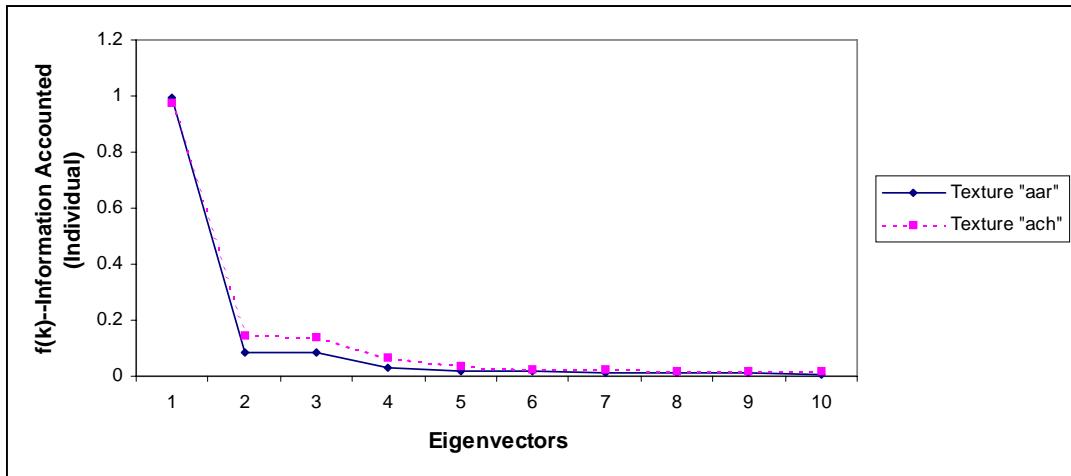
definitions as the measurement for information accounted for by individual eigenvector [Epstein1995]:

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

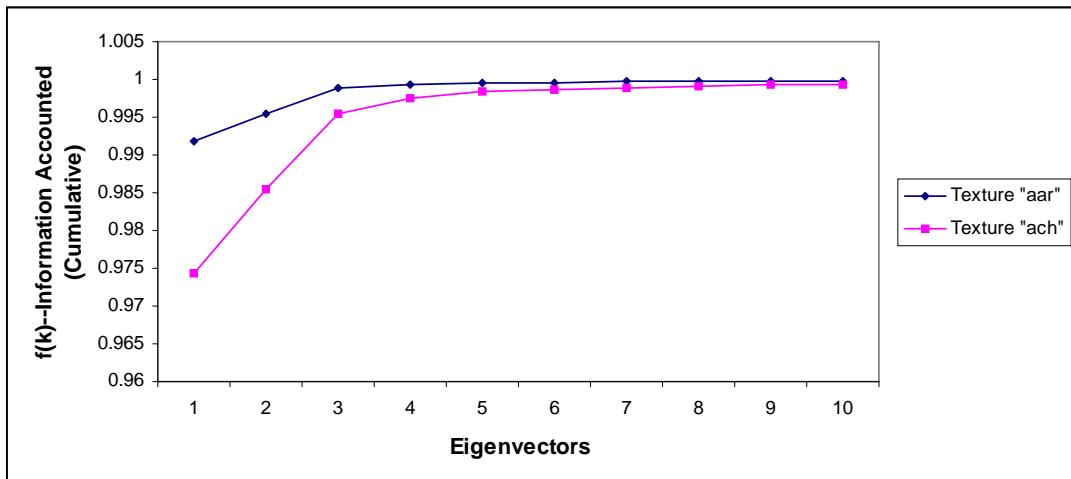
and cumulative eigenvectors [Zhang1998a]:

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

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

$$\hat{\mathbf{W}}_I = \text{diag}(w_1, w_2, \dots, w_k, 0, 0, \dots, 0),$$

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

$$\hat{\mathbf{I}} = \mathbf{U}_I \hat{\mathbf{W}}_I \mathbf{V}_I^T \quad (4.4.14)$$

Recalling equation (4.4.4) in the mathematical framework we can write $\mathbf{M}_1 = \mathbf{U}_I \hat{\mathbf{W}}_I$. We let \mathbf{M}_1 be an $m \times k$ matrix, since the last $n - k$ columns of $\mathbf{U}_I \hat{\mathbf{W}}_I$ are zeroes. Similarly, we create a $k \times n$ matrix \mathbf{M}_2 , which only contains the first k rows of \mathbf{V}_I^T , because the last $n - k$ rows of \mathbf{V}_I^T can be assigned zeroes due to the fact that the last $n - k$ diagonal elements of $\hat{\mathbf{W}}_I$ 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

$$\mathbf{I} = (\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n) = \mathbf{M}_1 \mathbf{M}_2 \quad (4.4.15)$$

where $\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n$ 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 $0 \leq \tau \leq 2\pi$ and $0 \leq \sigma \leq \pi/2$. 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 $\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n$ of (4.4.15). Firstly, we search for the intervals that contain τ and σ such that $\tau_i \leq \tau \leq \tau_{i+1}$ and $\sigma_j \leq \sigma \leq \sigma_{j+1}$. 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]:

$$\mathbf{i}_{(\tau, \sigma)} = (1-t_1)(1-t_2)\mathbf{i}_{(\tau_i, \sigma_j)} + t_1(1-t_2)\mathbf{i}_{(\tau_{i+1}, \sigma_j)} + t_1t_2\mathbf{i}_{(\tau_{i+1}, \sigma_{j+1})} + (1-t_1)t_2\mathbf{i}_{(\tau_i, \sigma_{j+1})} \quad (4.4.16)$$

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

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

where:

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

$$e_k = \frac{1}{NM} \sqrt{\sum_{x=1}^M \sum_{y=1}^N (r(x,y) - i(x,y))^2}$$
 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.

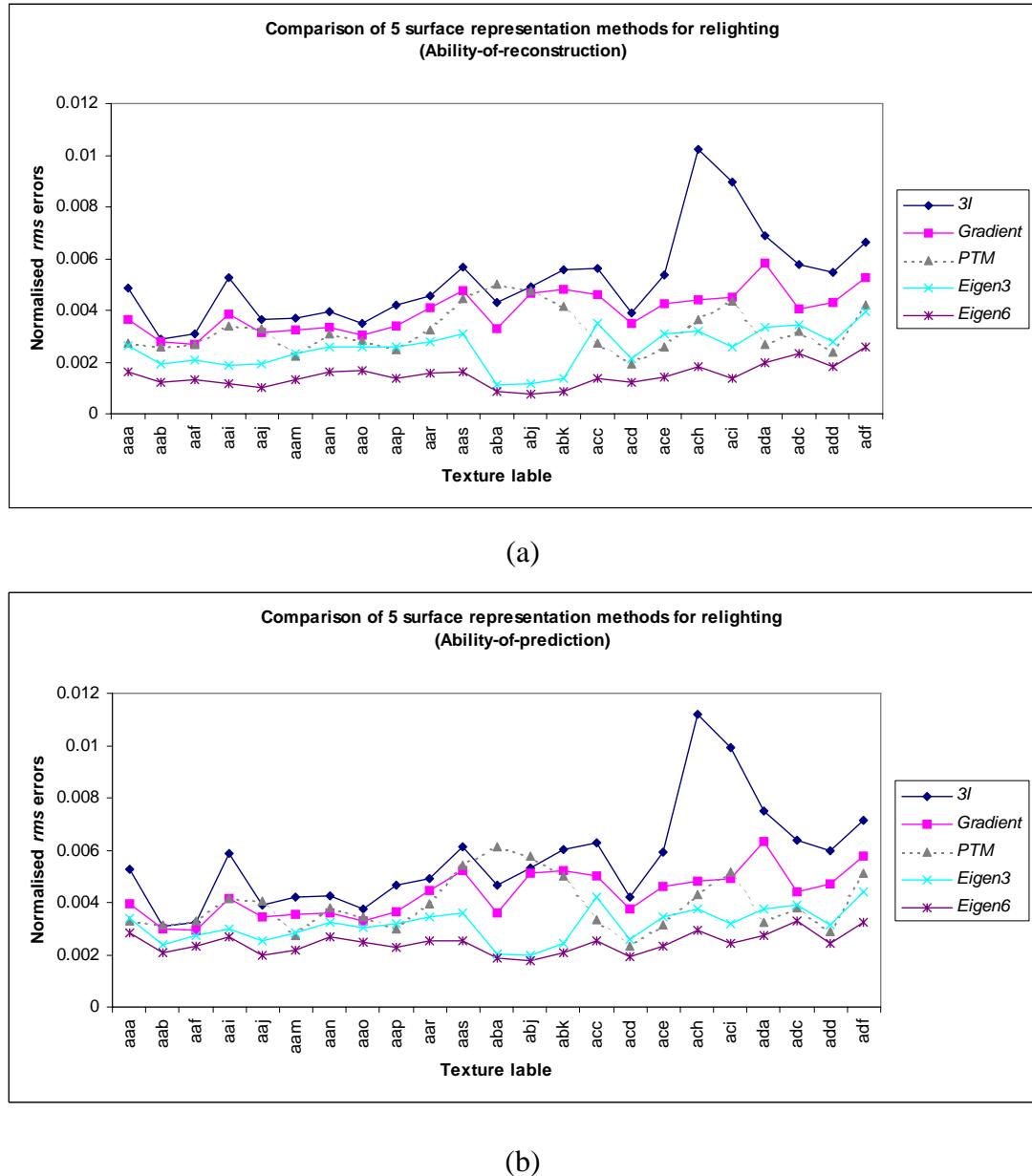


Figure 4.5.1 Relighting error vs texture for the five approaches: (a)Ability-of-reconstruction; (b)Ability-of-prediction(Leave-one-out).

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.

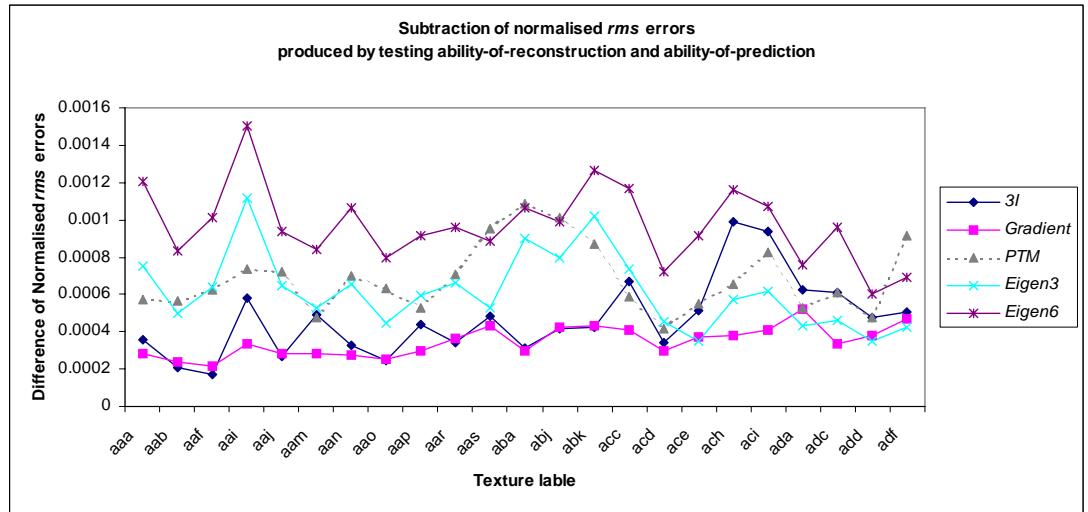


Figure 4.5.2 Subtracting normalised rms errors produced by testing ability-of-prediction from those produced by testing ability-of-reconstruction.

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).

	Original	$3I$	Gradient	PTM
Original and relit images				
Error images				
Error 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.

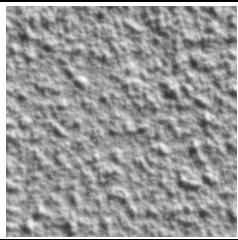
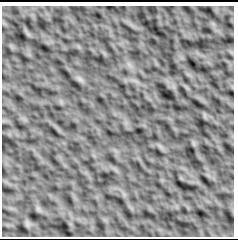
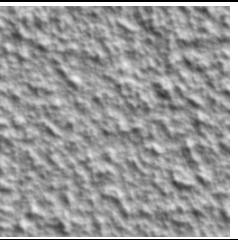
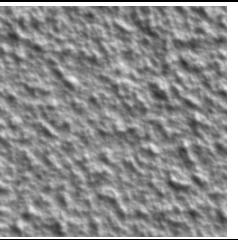
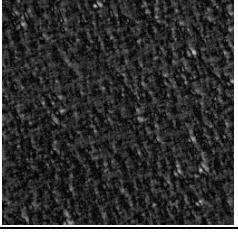
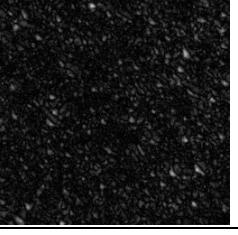
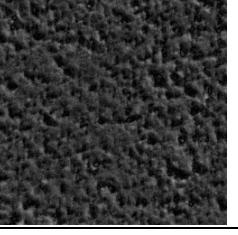
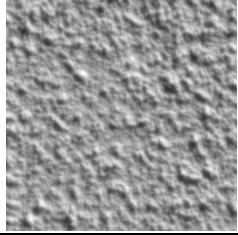
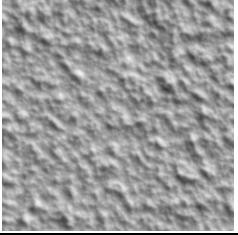
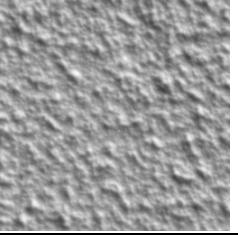
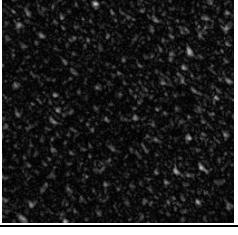
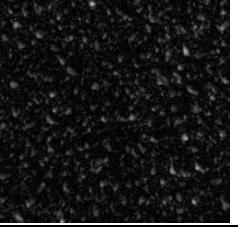
	Original	$3I$	Gradient	PTM
Original and relit images				
Error images				
	Original	$Eigen3$	$Eigen6$	
Original and relit images				
Error images				

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.

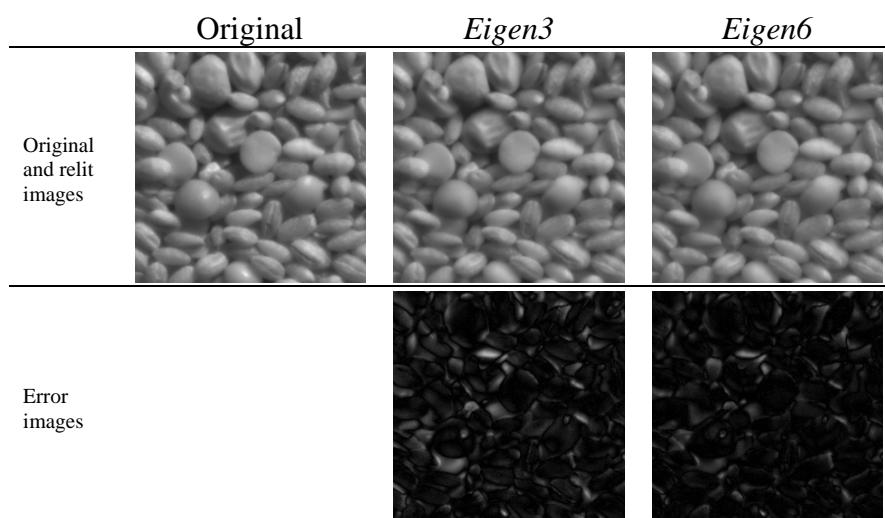
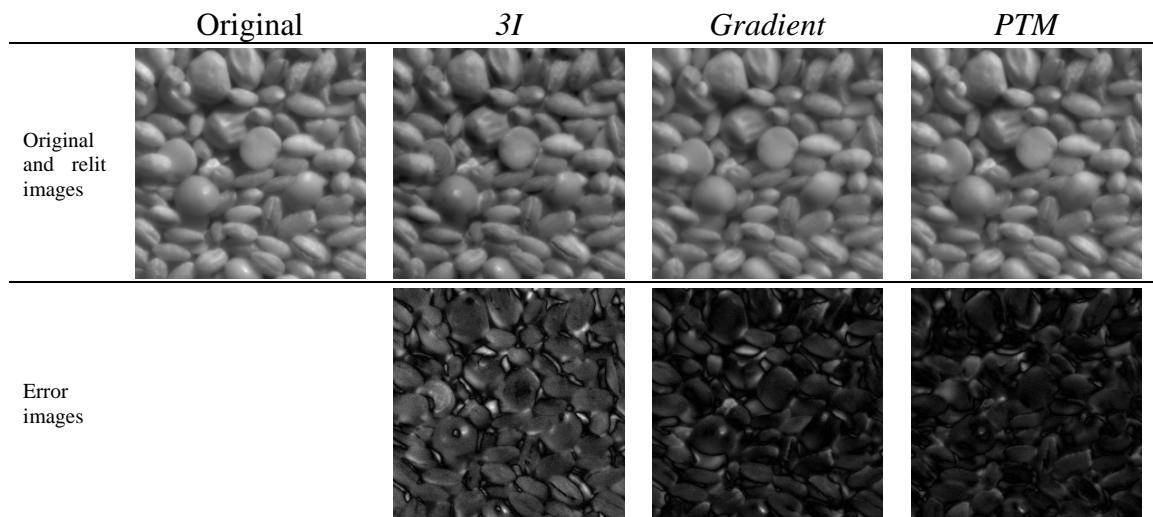


Figure 4.5.5 Texture “add”: Reconstructed images and error (difference between original and rendering) images. Bright areas in the error images represent reconstruction inaccuracies.

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

	Original	$Eigen3$	$Eigen6$
Original and relit images			
Error images			

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.

	Original	<i>3I</i>	Gradient	<i>PTM</i>
Original and relit images				
Error images				

	Original	<i>Eigen3</i>	<i>Eigen6</i>
Original and relit images			
Error images			

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.

	Original	<i>3I</i>	Gradient	PTM
Original and relit images				
Error images				

	Original	<i>Eigen3</i>	<i>Eigen6</i>
Original and relit images			
Error images			

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 $P(u,v)$ and $Q(u,v)$ to denote $p(x,y)$ and $q(x,y)$ in frequency domain respectively, where (u,v) is the 2D spatial frequency co-ordinate. By Fourier theories, we have the following equations:

$$P(u,v) = ju S(u,v) \quad (4.5.2)$$

$$Q(u,v) = jv S(u,v) \quad (4.5.3)$$

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

Thus,

$$v P(u,v) = u Q(u,v) \quad (4.5.4)$$

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.

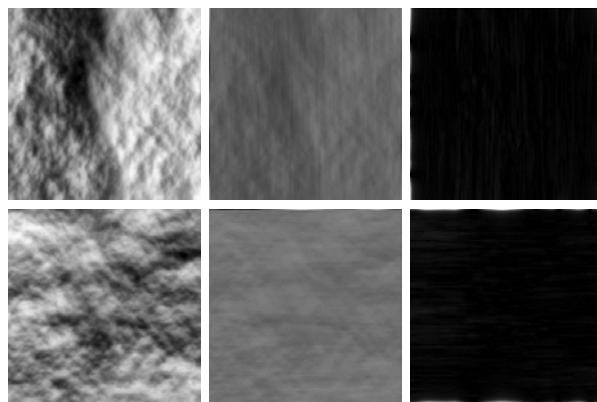


Figure 4.5.9 The comparison of real surface gradient maps and their synthetic counterparts. The first column shows the two surface gradient maps calculated using the Gradient method; the second column shows the corresponding synthetic surface gradient maps generated using equation (4.5.4); the third 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 integrability 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:

$$S(u, v) = \frac{-juP(u, v) - jvQ(u, v)}{u^2 + v^2} \quad (4.5.5)$$

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:

$$p(x, y) \approx s(x+1, y) - s(x, y) \quad (4.5.6)$$

$$q(x, y) \approx s(x, y+1) - s(x, y) \quad (4.5.7)$$

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.

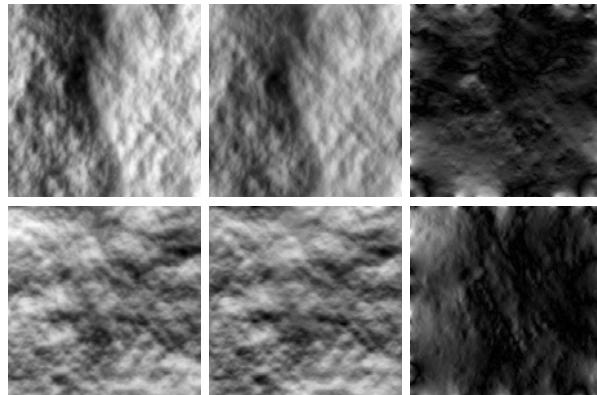


Figure 4.5.10 The comparison of differentiation methods. The first and second columns are gradient maps produced by differentiation of the surface height map in frequency and spatial domain respectively. The third column shows the absolute difference images.

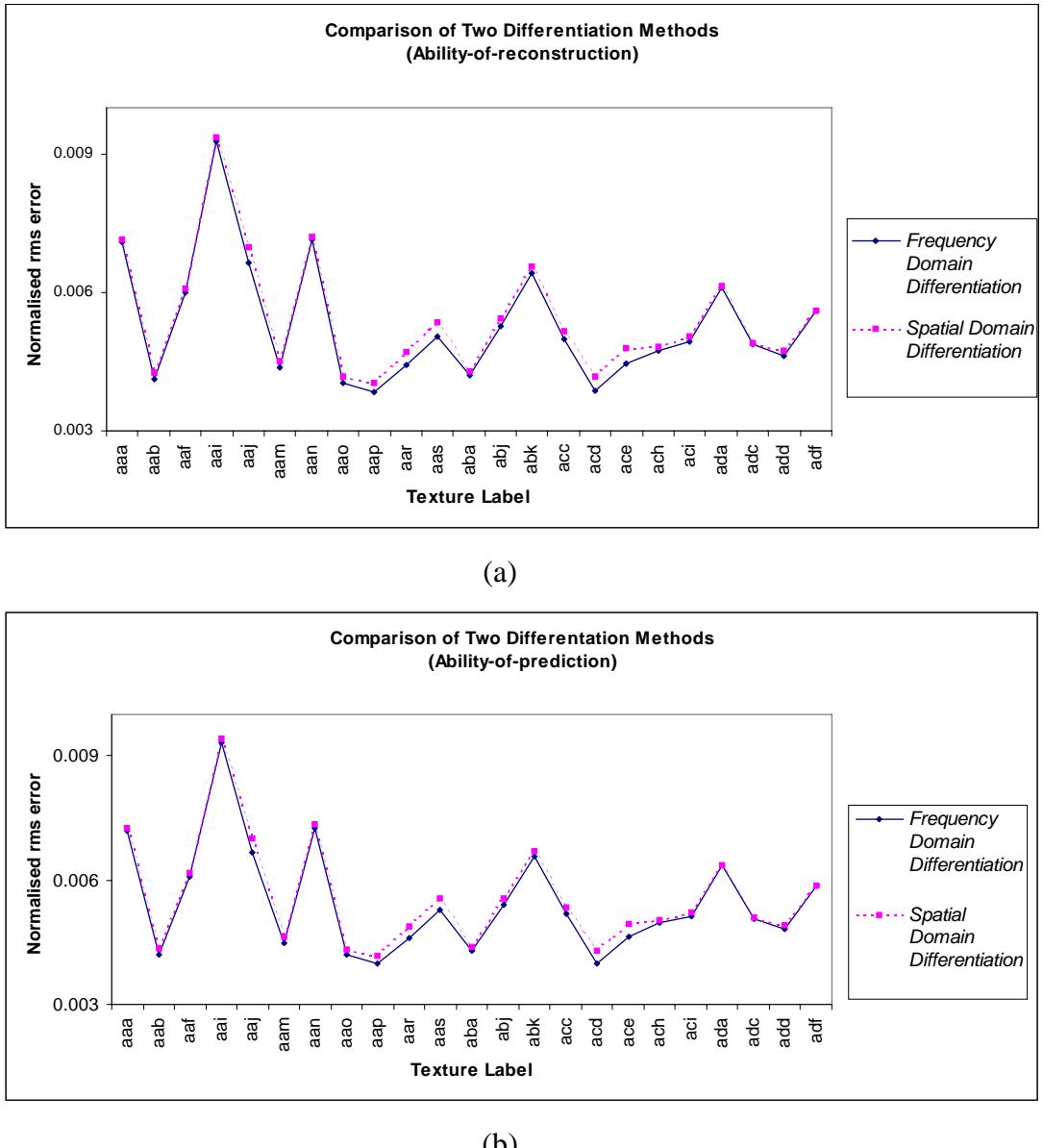


Figure 4.5.11 Comparison of two differentiation methods: (a)Ability-of-reconstruction; (b)Ability-of-prediction(Leave-one-out).

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.

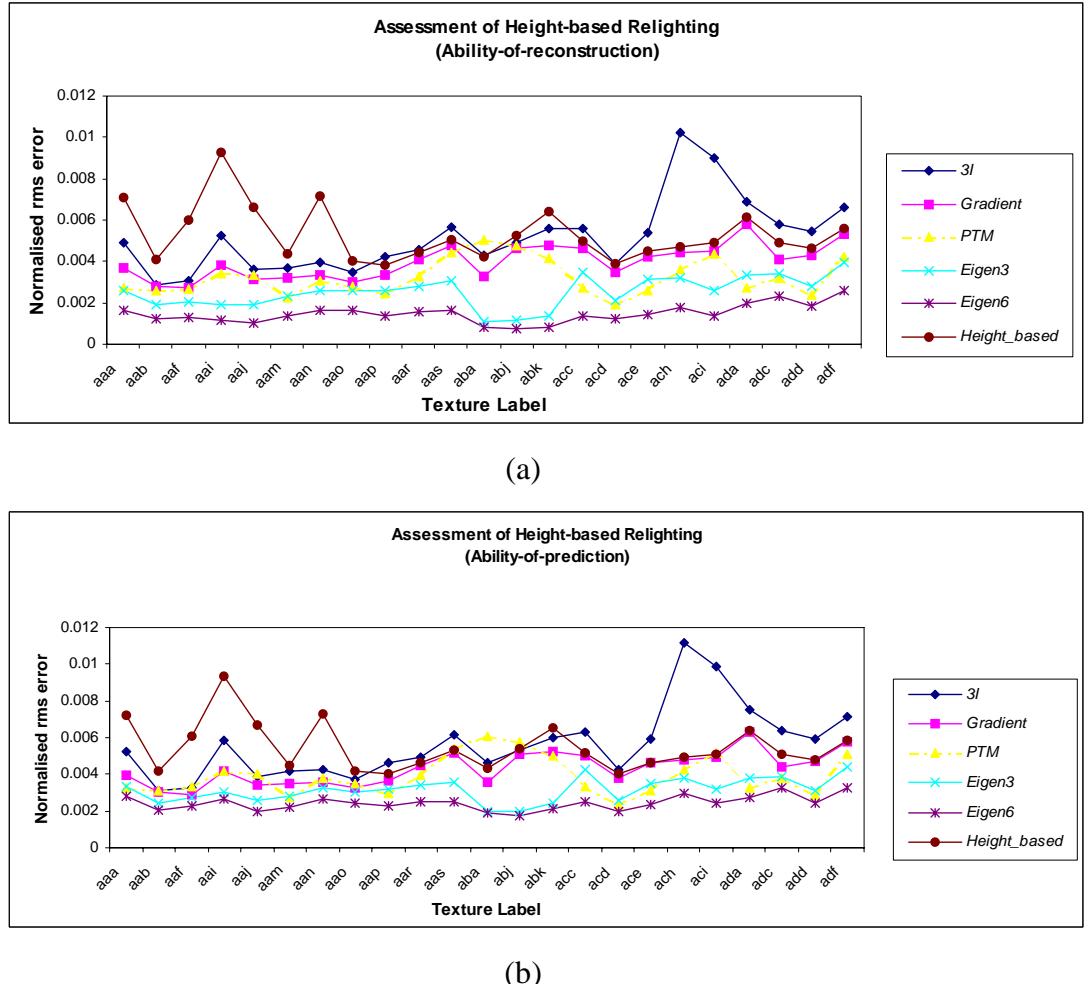


Figure 4.5.12 Comparison of height-based relighting and other five methods: (a) Ability-of-reconstruction; (b) Ability-of-prediction (Leave-one-out).

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.

Chapter 5

Synthesis Algorithms

5.1. Introduction

In Chapter 4, we investigated five inexpensive methods for extracting surface texture representations from a set of sample images. The aim of this chapter is to select an efficient 2D texture synthesis algorithm that can be easily extended for the synthesis of 3D surface texture representations. This is therefore equivalent to the second stage of our overall framework, as highlighted in

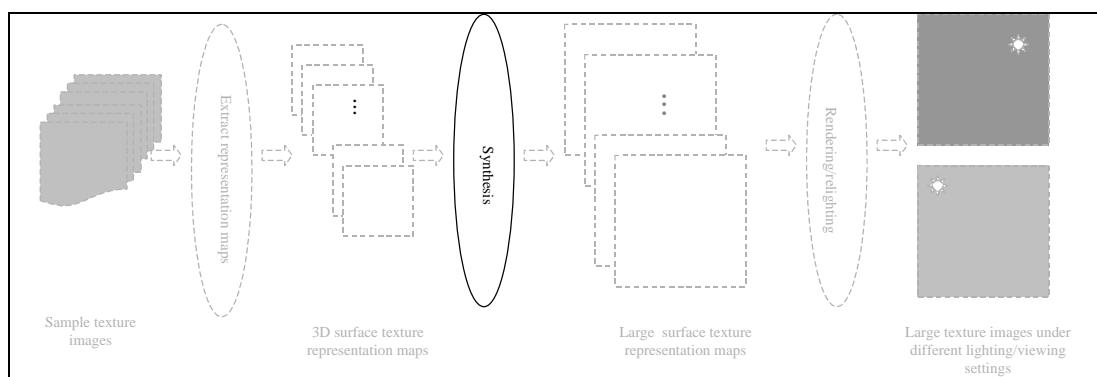


Figure 5.1.1 The selection of synthesis algorithm in the overall framework.

Since the main objective of this thesis is to develop inexpensive approaches for the synthesis of 3D surface textures, the choice of 2D synthesis algorithms is particularly important with respect to computational complexity and quality of final results. We set two criteria for the selection of 2D texture synthesis algorithms: (1) the suitability of the algorithm for extension to deal with multi-dimensional

representations, and (2) the capability of producing good results while requiring little computation.

We first present a detailed survey of recent publications on 2D texture synthesis. Then we investigate two popular approaches based on [Wei2000] and [Efros2001] respectively. The first approach synthesises images from small sample textures at pixel scale by employing a multi-resolution decomposition technique. The second approach synthesises the result image by ‘stitching’ together small patches selected from the sample image. We propose simple modifications to these two methods, which can reduce the computation and produce similar synthesis results to the originals. On comparing the two approaches, we select the modified Efros’ 2D texture synthesis algorithm as our basic algorithm, as it can produce better results while requiring less computation. In particular, we analyse the effects on output images produced by varying the four input parameters of the selected algorithm.

The rest of this chapter is organised as following. Section 5.2 presents a detailed survey on 2D texture synthesis methods. Section 5.3 describes the two selected approaches and compares them in terms of computational complexity and quality of results. Section 5.4 analyses the input parameters of the selected algorithm, and finally we conclude our work of this chapter in section 5.5.

5.2. A detailed survey of synthesis algorithms

The goal of this section is to survey 2D texture synthesis algorithms using the criteria that we proposed in the previous section. In Chapter 2, we divided 2D synthesis algorithms into two groups according to the sampling strategies. Accordingly, this survey is also based on this taxonomy. Two algorithms are selected for further investigation at the end of this section.

5.2.1. Texture synthesis methods based on global sampling strategies

As discussed in Chapter 2, texture synthesis algorithms employing global sampling strategies synthesise new images by matching global statistics between the sample and result images in feature space. In general, these texture synthesis methods are

not preferable for the use of synthesising surface relighting representations in multi-dimensional space. One reason is that the surface relighting representations normally consist of multi-dimensional vectors with correlation existing between the elements. For the synthesised surface representations, the correlation is unlikely to be maintained during global sampling process. Meanwhile, these methods essentially amount to a multi-parameter and non-linear optimisation process over a single image. As shown in Table 2.1.1, two typical methods [Zhu2000 and Portilla200], which produced good synthesis results over a wide range of sample textures, require more than 20 minutes for computing. Extending these approaches to multi-dimensional surface representations would require iteration and optimisation over inter as well as intra image parameters. Consequently, the computation may be expensive.

Many 2D texture synthesis methods synthesise result images by matching marginal or joint histograms between the sample and result images [Heeger1995, Van Nevel 1998, Zhu2000 and Copeland2001]. For 3D surface texture synthesis, the input consists of multi-dimensional vectors that represent the sample surface texture under arbitrary illumination. Thus, the one-dimensional histograms in 2D texture synthesis algorithms will become multi-dimensional histograms in 3D surface texture synthesis. During the matching process, the element values of the result multi-dimensional vectors are changed according to the sample histograms. This might destroy the correlation between the surface relighting representations. For example, if we use surface gradients and albedo maps as the representation of a Lambertian surface, the matching process will change the result surface gradients and albedo values by purely comparing the sample and result histograms. In order to maintain the relighting characteristics, the interrelationship [e.g. cross correlation between components (elements)] of the multi-dimensional data must be kept for an iteration method. This results in a complex multi-dimensional optimisation problem. Meanwhile, if the number of bins is large or the dimensionality is high, there might be too few pixels in each bin for a multi-dimensional histogram. For instance, if we calculate a six-dimensional histogram using six 64x64 representation maps and each dimension is divided into 10 bins, we only have, on average, $6*64*64/10^6=0.025$ pixels in each bin. Thus, it is difficult to accurately estimate the multi-dimensional

histograms. A similar problem might exist in those methods that synthesise new images by matching various statistics, e.g. [Jacovitti1998, Portilla2000 and Campisi2002].

In general, 2D synthesis algorithms that employ global sampling strategies become more complex when being extended to use multi-dimensional vectors as input. The method proposed in [Eom1998] estimates the parameters of a 2D moving model; it would be more difficult if implemented in high dimensional space. Although De Bonet's method can be easily extended to take surface relighting representations in multi-dimensional space as input, it is not clear whether the filter bank is sufficient to capture the characterisations of the sample representations [De Bonet1997].

To summarise:

Two-dimensional texture synthesis algorithms that employ global sampling strategies are generally not suitable as the basis of algorithms in 3D surface texture synthesis approaches. The main reason is that these algorithms become too complex or have difficulty to preserve the correlation between surface relighting representations when they are extended to multi-dimensional space.

5.2.2. Texture synthesis methods based on local sampling strategies

As introduced in Chapter 2, texture synthesis methods based on local sampling strategies synthesise new images by matching local information between the sample and result images. These methods can be further divided into two sub-classes depending on whether they employ a parametric and non-parametric model.

In general, parametric methods require expensive computation due to the estimation of the parameters. Zhu *et. al.* estimate the parameters of the FRAME model for texture synthesis; they report that the computational cost increases proportionally with the size of the filter window and long iterations are required to achieve accuracy [Zhu1995]. Bader *et. al.* implement parallel algorithms for the synthesis in order to reduce the computing time [Bader1995]. Zhang *et. al.* estimate the parameters of the wavelet autoregressive model and the radial basis function network for modelling and synthesising texture images [Zhang1998b]. Their multiresolution AR model has a total of 91 parameters. If multi-dimensional surface

representations are used as input in these methods, both the models and the computation become more complex.

Non-parametric texture synthesis methods are less complex compared with their parametric counterparts because they do not need to estimate the parameters of statistical models [Efros1999, Wei2000, Hertzmann2001, Efros2001, Parada2001, Ashikhmin2001, Harrison2001, Tonietto2002, Zelinka2002, Cohen2003 Nealen2003, Paget1998, Ashlock1999, Bar-Joseph2001, Xu2001, Liang2001 and Gousseau2002]. Thus, these methods are more suitable for extension to use multi-dimensional representations as input. However, several methods still require expensive computation. Paget and Longstaff require parallel algorithms for the synthesis using non-causal, non-parametric and multiscale Markov random field [Paget1998]. Ashlock and Davidson apply tandem generic algorithms for texture synthesis based on non-parametric partially ordered Markov models; their method need several hours to compute [Ashlock1999]. On the other hand, recent non-parametric synthesis approaches have been reported to be able to produce good results with less computation [Efros1999, Wei2000, Bar-Joseph2001, Xu2001, Liang2001, Hertzmann2001, Efros2001, Ashikhmin2001, Harrison2001, Tonietto2002, Zelinka2002, Cohen2003 and Nealen2003]. In these approaches, pixel values in the synthesised results are obtained from the sample images. The correlation between synthesised surface representations can be kept. Therefore, these methods are more suitable for the synthesis of surface relighting representations in multi-dimensional space. In particular, several patch-based synthesis algorithms ([Efros2001, Xu2001 and Liang2001]) have one of the smallest requirements in terms of computational complexity.

The algorithm proposed in [Efros1999] is a highlight in the research field of texture synthesis. It assumes a Markov random field model and calculates the conditional distribution of a pixel given all its neighbours by querying the sample image and finding all similar neighbourhoods. The conditional probability density function $p(I_{result}(x, y) | I_{N_R})$ can be estimated using the following set:

$$\Psi(I_{result}(x, y)) = \{I_{N_s} \subset I_{sample} : G * \|I_{N_s} - I_{N_R}\| \leq (1 + \varepsilon) * d_{\min}\} \quad (5.2.1)$$

where:

$I_{result}(x, y)$ is the intensity value of the pixel (x, y) to be synthesised in the output result image

I_{N_R} is the neighbourhood centred at pixel (x, y) in the output image

I_{N_S} is a neighbourhood in the input sample image

$G^* \|(I_{N_R} - I_{N_S})\|$ is a weighted Sum of Squared Differences (SSD) by a Gaussian kernel G between pixel values in an sample neighbourhood N_s and the result neighbourhood N_R , which is centred at $I_{result}(x, y)$

d_{min} is the minimum SSD between pixel values in the input and the output neighbourhood, weighted by a Gaussian kernel G

ε is the error threshold and is set to 0.1

The centred pixel values of neighbourhoods in $\Psi(I_{result}(x, y))$ provide an estimated histogram for $I_{result}(x, y)$. Thus, the algorithm first finds the best-matched neighbourhoods (within certain error tolerance ε) in the sample image for the result neighbourhood N_R centred by $I_{result}(x, y)$. Then a best-matched neighbourhood is randomly selected and its centred pixel value is assigned to $I_{result}(x, y)$. Although the algorithm is simple and not fast, it can produce promising synthesis results. Based on this algorithm, Wei and Levoy employed image pyramid representations to develop a new synthesis algorithm and used the tree-structured vector quantization for acceleration [Wei2000].

The work in [Efros1999 and Wei2000] has received broad attention in the computer vision and computer graphics communities. Later work based on these two algorithms includes [Ashikhmin2001, Hertzmann2001, Efros2001, Parada2001, Tonietto2002, Zelinka2002, Cohen2003 and Nealen2003]. In [Ashikhmin2001], Ashikhmin modifies the algorithm of [Wei2000] and achieves faster synthesis speeds, which allow direct user input for interactive control over the synthesis process. In [Hertzmann2001], Hertzmann *et. al.* propose an image processing framework called image analogies, which can learn the analogy between the original and filtered input images to produce new image pairs. Their algorithm is based on [Wei2000 and Ashikmin2001]. In [Parada2001], Parada and Ruiz-del-Solar use self-organizing maps to improve the algorithm of [Wei2000]. In [Efros2001], Efros and

Freeman develop a patch-based texture synthesis algorithm, which is based on [Efros1999] but produces better results with much less computation. In [Tonietto2002], a local-controlled synthesis algorithm is proposed that can generate texture in which the basic elements have different sizes, e.g. the skin of a cheetah. In [Zelinka2002], a jump map is first generated to store the matching input pixels and then used to synthesise a new texture image in real-time. In [Cohen2003], Wang tiles are employed and combined with the algorithm of [Efros2001] for texture synthesis. In [Nealen2003], a pixel-based algorithm and a patch-based algorithm are combined to improve previous synthesis methods.

To summarise:

Since estimating the parameters of statistical models in multi-dimensional space is complex, parametric texture synthesis methods with local sampling strategies are not suitable for synthesising multi-dimensional surface relighting representations. On the other hand, most non-parametric synthesis approaches can be easily extended to dealing with multi-dimensional representations, and they can produce good results with little computation. Thus, we select two non-parametric texture synthesis approaches based on [Wei2000 and Efros2001] for future investigation.

5.2.3. Summary

We have surveyed 2D texture synthesis approaches using the two criteria: (1) the suitability of the algorithm for extension to deal with multi-dimensional representations, and (2) the capability of producing good results while requiring little computation. Texture synthesis algorithms employing global sampling strategies have difficulty to synthesise the multi-dimensional surface representations because they tend to become excessively complex, and the correlation between the result representations may be damaged. On the other hand, non-parametric synthesis algorithms with local sampling strategies are capable of taking multi-dimensional vectors as input and producing good results with less computation. We therefore select two non-parametric approaches based on [Wei2000 and Efros2001] as candidate basic algorithms for 3D surface texture synthesis.

5.3. Two Approaches

This section investigates two 2D texture synthesis approaches based on [Wei2000] and [Efros2001].

5.3.1. The first approach and modification—a pixel-based multi-resolution approach

The first approach employs a pixel-based multi-resolution texture synthesis algorithm, which is based on a non-parametric sampling method [Wei2000]. The algorithm in [Wei2000] can be seen as the extension of the work in [Efros1999]. It also assumes a Markov random field texture model, which means a pixel value at a certain location only depends on its immediate neighbourhood. If we recall the expression (5.2.1), the algorithm in [Wei2000] essentially uses neighbourhoods across different resolutions and synthesises pixel values from lower to higher resolutions incrementally. The size of the neighbourhood is a parameter of the algorithm and must be chosen taking into account the granularity of the subject texture. When choosing the value of the next pixel in the output image the algorithm uses the populated portion of the pixel's neighbourhood to exhaustively search for the ‘best’ matched region in the sample image.

However, in our approach, for a certain percentage of the selections we use the ‘next column neighbour pixel’. Supposing we have just found a best-matched pixel and stored this in the result image, since we are synthesising texture in raster order, an obvious candidate for the next best match is the neighbouring pixel located in the next column of the sample image. Figure 5.3.1 shows an example.

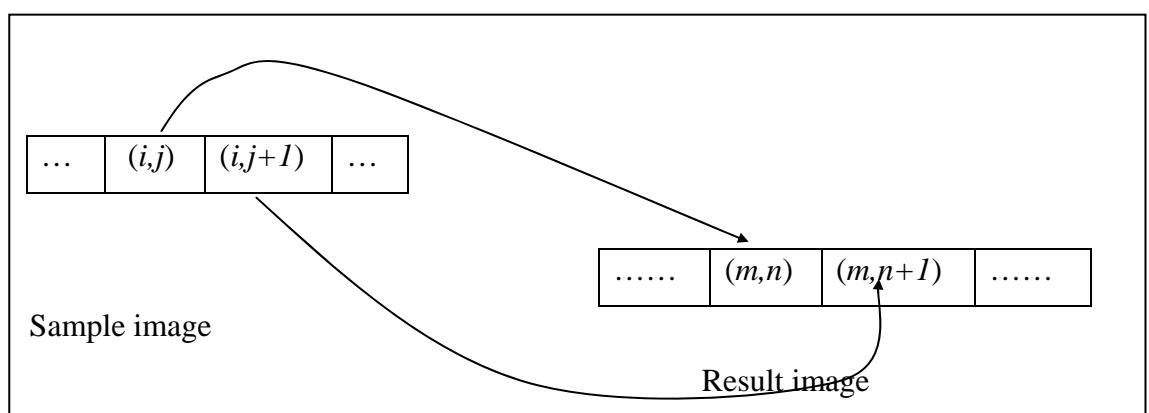


Figure 5.3.1 The next column neighbour of last best-matched pixel can be used as the current best match. Pixel (i,j) in the sample image is the best match of pixel (m,n) in the result image. When we are synthesising pixel $(m, n+1)$ in result image, we grant pixel $(i,j+1)$ in the sample image is the best-matched without performing an exhaustive search.

The use of the ‘next column neighbour pixel’ as opposed that derived by exhaustive search is controlled. It cannot be used for boundary conditions. In these cases we always perform an exhaustive search. In addition for certain randomly selected pixels we force the algorithm to use exhaustive search. The percentage of the random selections is controlled by a parameter set by the experimenter. If we set the exhaustive search rate to 100%, the algorithm is the same as Efros and Leung’s [Efros1999] and Wei and Levoy’s [Wei2000]. We can also trade off synthesis speeds against synthesis quality. This modification approach is similar to the synthesis algorithm in [Ashikhmin2001] and can be seen as a simplified version.

The whole synthesis process

First we decompose the input sample image to obtain a set of multi-scale images by applying a Low-pass filter, i.e. Gaussian filter [Burt1983] to obtain a pyramid data structure. Let L represent the level of the lowest scale in each pyramid and 0 represent the level of the highest scale. Corresponding to the sample pyramid, we construct a result pyramid data structure, in which all elements are 0 . The synthesis process begins from the lowest scale (level L), pixel by pixel, in raster order. For an output pixel, we first construct a neighbourhood as defined in [Wei2000]. The neighbourhood is shown in Figure 5.3.2. In the top pyramid level (the lowest scale), the neighbourhood uses only local populated neighbour pixels to perform exhaustive search. In the lower pyramid levels, it uses local populated neighbour pixels plus pixels immediately above (i.e. in the upper level). A neighbourhood is also constructed for each pixel in the sample pyramid. All of the pixels involved in the neighbourhood form the neighbourhood vector, which is used to perform exhaustive search to find the best matches for pixels in the result pyramid. During the exhaustive search, in order to determine the pixel value at a location (x,y) in the result pyramid, its neighbourhood is compared against all possible neighbourhoods in the sample pyramid. If pixel (i,j) has the most similar

neighbourhood, the value of pixel (i,j) in the sample pyramid is assigned to pixel (x,y) in the result pyramid. We use the Sum of Absolute Differences (SAD) to measure the similarity between neighbourhoods. More details about the exhaustive search algorithm can be found in [Wei2000].

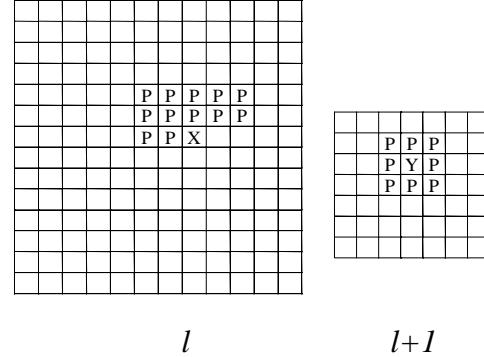


Figure 5.3.2. The neighbourhood defined by Wei and Levoy [Wei2000]. The current level of pyramid “ l ” is shown at left and the upper level “ $l+1$ ” is shown at right. It uses local populated neighbour pixels (marked as “P” in level “ l ”) plus pixels immediate above in the upper level (marked as “P” in level “ $l+1$ ”). All of marked pixels form the sub-neighbourhood. The current output pixel is marked as X, which locates at (x, y) in the l^{th} pyramid level. Its “parent” pixel in the $l+1$ pyramid level locates at $(x/2, y/2)$, which is marked as Y. Since the level “ $l+1$ ” is complete, this sub-neighbourhood can contain all pixels around Y. The sub-neighbourhood is constructed for each sample pyramid and result pyramid.

We use the ‘next column neighbour pixel’ as the best-matched pixel whenever allowed. Now suppose we have synthesised the pixel located at (m,n) in level X ($X \leq L$), and its best-matched pixel locates at (k,l) in level X of the sample pyramid. Let $\{X, (m, n)\}$ represent the pixel location in the result pyramid and $\{X, (k, l)\}$ for the pixel location of the sample pyramid. We are going to find the best match for next pixel. Suppose the next pixel locates at $\{X, (m, n+1)\}$ of the result pyramid. Intuitively, we consider the next column neighbour pixel of $\{X, (k, l)\}$ in the sample pyramid as the candidate of the best match of pixel $\{X, (m, n+1)\}$. If $\{X, (k, l+1)\}$ exists in the sample pyramid, we grant the neighbourhood of $\{X, (k, l+1)\}$ as the best match for that of $\{X, (m, n+1)\}$ in the result pyramid. The pixel value of $\{X, (k, l+1)\}$ of the sample pyramid is assigned to the pixel value of $\{X, (m, n+1)\}$ of the result pyramid (Recall Figure 5.3.1). However, there are three cases in which exhaustive searches must be performed. They are: (1) pixel $\{X, (m, n+1)\}$ of the result pyramid is randomly selected for exhaustively searching; (2) pixel $\{X, (m, n+1)\}$ does not

exist in the result pyramid, which means $\{X, (m, n)\}$ is the last pixel of the m^{th} row; and (3) pixel $\{X, (k, l+1)\}$ of the sample pyramid does not exist. Figure 5.3.3 shows these three cases.

The synthesis process will continue until all pixels in the result pyramid are assigned values from the lowest scale to the highest scale. In the highest scale (level 0), the required result image is synthesised. The pseudocode is shown in Table 5.3.1. For most textures, the ratio of exhaustive search is from 40% to 70% given good results. The quality of synthesis results is similar to previous work by using 100% exhaustive search [Wei2000][Efros1999], but the computational complexity is reduced. Figure 5.3.4 shows example results from using 100% exhaustive search algorithm and our algorithm. The acceleration technique can still be applied in the modified algorithm [Wei2000].

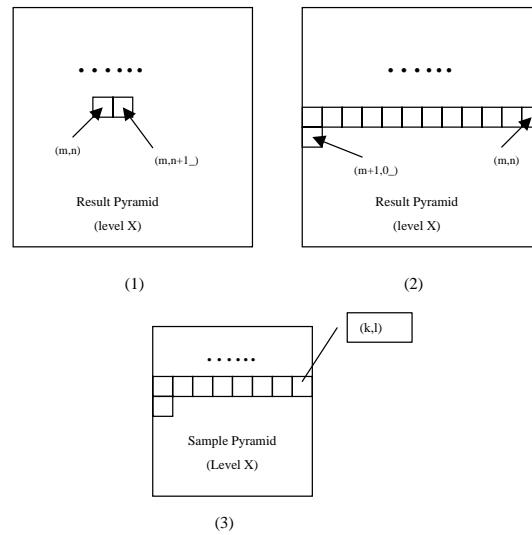


Figure 5.3.3. Three cases that must perform exhaustive search. (1) Pixel at $\{X, (m, n+1)\}$ of the result pyramid is randomly selected for exhaustive search, (2) Pixel at $\{X, (m, n+1)\}$ does not exist in the result pyramid, which also means $\{X, (m, n)\}$ is the last pixel of the m^{th} row and (3) Pixel at $\{X, (k, l+1)\}$ of the sample pyramid does not exist.

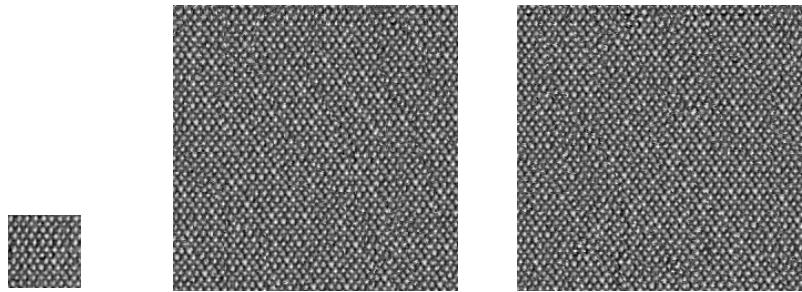


Figure 5.3.4 Comparison of synthesis results. The image in the left is the sample. The image in the middle is the synthesised result by using 100% exhaustive search. The image in the right is the synthesised result by only using 40% exhaustive search. All of other parameters are same.

```

SamplePyramid = buildPyramid(SampleImage);
ResultPyramid = Null;           //result image pyramid

Input rate = exhaustive search percentage;
RandomPixels = randomSelectPixels(rate);

Loop l := the highest pyramid level L to the lowest pyramid level 0
    Loop through all pixel locations (x, y) in result pyramids in level l,
        (i, j) = FindBestMatchLocation ( SamplePyramid,
                                            x, y, l, ResultPyramid);
        Result_PixelValue (x, y) = Sample_PixelValue (i, j);
    While (i, j+1) exists in sample pyramids
        && (x, y+1) exists in result pyramids
        && (x, y+1) not belong to RandomPixels
    {
        Result_PixelValue (x, y+1) = Sample_PixelValue (i, j+1);
        j := j+1;
        y := y+1;
    } End while

End loop

End loop

ResultImage = writeImage (ResultPyramid);

```

Table 5.3.1 The pseudocode of the first approach

To summarise:

We have investigated a 2D texture synthesis approach proposed by [Wei2000]. It assumes a Markov Random Field texture model, which means a pixel value at a certain location only depends on its immediate neighbourhood. A multi-resolution scheme is applied to construct the neighbourhood around a given pixel. The algorithm synthesises a result image in pixel scale by finding the best-matched neighbourhoods in the sample image. We modified the original algorithm by using the ‘next column neighbour pixel’ as the best-matched pixel for a certain percentage of pixel locations. The modification can produce similar results with less computation.

5.3.2. The second approach and modification—A patch-based approach

The second approach is based on the image quilting method proposed by Efros and Freeman [Efros2001]. The method synthesises a new image by ‘stitching’ together small patches from the sample image. It requires little computation and can produce remarkable synthesis results. This method is also an extension of the previous work in [Efros1999].

The method in [Efros2001] synthesises a result image block by block in raster order. Square blocks are used to capture the primary pattern in the sample texture. The size of the block is a parameter of the algorithm and must be chosen taking into account the granularity of the subject texture. First, a block is randomly selected from the sample image and pasted into the new image beginning at the first row and the first column. Then another block is selected as a candidate neighbour. It is placed next to the first block so that they overlap one another. The overlapping area between the two blocks is used to test the goodness of fit of the candidate using an L2 norm (Sum of Squared Differences). This is repeated for different candidates to find the minimum difference metric (distance). The final neighbour is randomly selected from those blocks whose distance lies in a certain range of the minimum distance. The range is controlled by a predefined error tolerance. A minimum error boundary cut is calculated in the overlapping area between the overlapping pixels so that the boundary looks smooth, as shown in Figure 5.3.5. Both vertical and

horizontal overlapping areas are used for selecting best-matched blocks inside the new image. This whole process is repeated until an output image of the required size has been generated.

We have made two modifications to this quilting algorithm. First, instead of locating the best-matched block using exhaustive search, we select the ‘next column neighbour block’, which is the corresponding neighbour of last selection, and assign it as the current best-matched block, providing it exists in the sample image. This modification is similar to that introduced for the first approach. During the synthesis process, after a best-matched block is found in the sample image, we store its location in an array. When a new block in the result image is being synthesised, we check the best-matched block locations of its already generated neighbours. If there exists a block that is adjacent to all the best-matched block locations in the sample image, this block is selected as the current best-matched block. Figure 5.3.6 illustrates this process. Suppose we are going to synthesise block h' in the result image. We first check the best-matched blocks of its existing neighbour blocks e' , f' and g' . If their best-matched blocks e , f and g are adjacent in the sample image, then block h , which is the neighbour of e , f and g , is selected as the best-matched block for h' . Obviously, for the first block row or column in the result image, only one neighbour block is checked. This simplification can increase the speed of the algorithm without apparently affecting the output. It can also be seen as an extension of the method used in [Ashikhmin2001].

The second modification to the original algorithm is that we use an error metric based on the Sum of Absolute Differences (SAD) rather than more expensive L2 norm (square of SAD). They produced similar results in our experiments. Although both the L2 norm and the SAD are not perfect as perceptual metrics, the existing perceptual metrics might not be completely reliable and require more expensive computations [Sebe2000, Bolin1998, Ramasubramania1999 and Ashikhmin2001]. Figure 5.3.7 shows example output images produced by the modified and original algorithms respectively. The pseudocode of the whole algorithm is listed in Table 5.3.2.

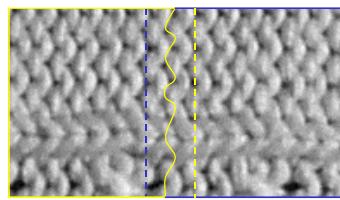


Figure 5.3.5 The boundary cut process of Efros' 2D texture synthesis approach. The curve shows the best boundary cut.

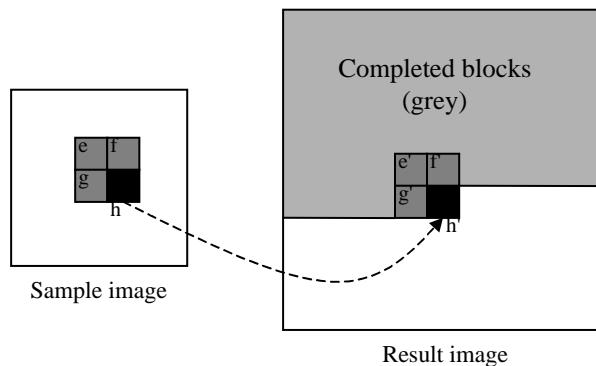


Figure 5.3.6. The neighbour of previous best-matched blocks. The grey area in the result image represents those blocks that have already been synthesised.

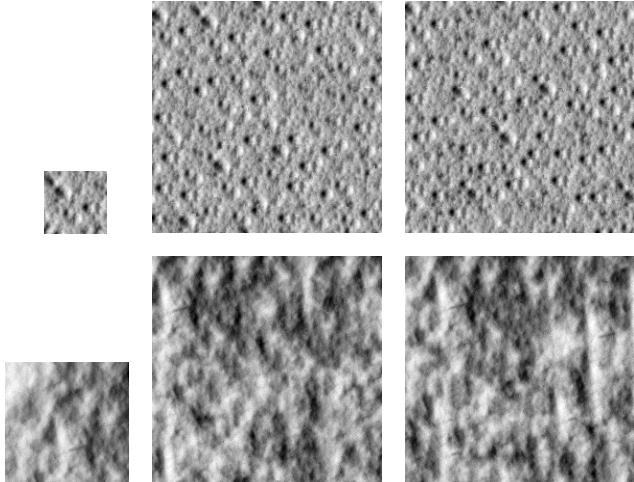


Figure 5.3.7 The comparison of results produced by the modified and original algorithms. The first column shows sample images (texture "aar" and "aaf"). The second column shows synthesis results produced by the original algorithm. The third column shows synthesis results produced by the modified algorithm.

```

ResultImage = Null;      //result larger image

Input SampleImage, BlockSize, OverlapSize

A_RandomBlock = randomSelectBlock(SampleImage);

PixelValue(ResultImage, FirstBlockLocation) = PixelValue(SampleImage,
                                                       A_RandomBlock);

Loop through the ResultImage in raster order in steps of one block

    SampleBlockLocation = FindBestMatchLocation ( SampleImage, ResultImage,
                                                   ResultBlockLocation, BlockSize, OverlapSize);

    SampleBlock=PixelValue(SampleImage, SampleBlockLocation);

    PixelValue (ResultImage, ResultBlockLocation) = BestBoundaryPath
                                                (ResultBlockLocation, NeighbourResultBlocks,
                                                 SampleBlock, OverlapSize) ;

While (SampleBlockLocation+1) exists in sample pyramids
        && (ResultBlockLocation+1) exists in result pyramids

    {

        SampleBlock=PixelValue(SampleImage, SampleBlockLocation+1);

        PixelValue (ResultImage, ResultBlockLocation+1) =
            BestBoundaryPath(ResultBlockLocation+1,
                             NeighbourResultBlocks, SampleBlock, OverlapSize);

        ResultBlockLocation := ResultBlockLocation+1;

        SampleBlockLocation := SampleBlockLocation+1;
    } End while

End loop

```

Table 5.3.2 The pseudocode of the first approach

To summarise:

The second 2D texture synthesis approach is based on the image quilting algorithm proposed by [Efros2001]. This approach can produce high-quality synthesis results while requiring little computation. We made two modifications to

the original algorithm. The modified algorithm can produce similar results to those from original algorithm while the computation is reduced.

5.3.3. Comparison of the two approaches

In section 5.3.2, we investigated two 2D texture synthesis approaches. Since the main goal of this thesis is to develop inexpensive approaches for the synthesis of 3D surface texture, we need to select one method which requires less computation while producing reasonable results. Therefore, we first compare the two approaches according to the computational complexity and synthesis results.

The computational complexity of the original algorithm in [Wei2000] without acceleration is $O(N)$, where N is the number of pixels in sample image. It is obvious that our modified algorithm has the computation $O(a\%*N)$, where a the input percentage of total pixels that the algorithm should perform exhaustive search. In contrast, the second approach, which is based on [Efros2001], has the computational complexity at most $O(B)$, where B is the number of blocks in the sample image. The number of blocks is usually much smaller than the number of image pixels. For example, if the block size is 13×13 , for a 64×64 sample image, the block number is only 24 compared with the pixel number 4096. Even if we set the percentage of total pixels that perform exhaustive search as 40% ($a=40$), the computational complexity of the first approach is $O(1638)$ whereas that of the second approach is $O(24)$. Obviously, the second approach requires much less computation. We report the time consumed in a typical experiment without using the acceleration technique in Table 5.3.3. The experiment was performed on a normal desktop PC with a 600MHz Intel Pentium III CPU. Note same acceleration techniques are available for both algorithms [Efros2001 & Wei2000].

Approach	The first	The second
Sample size	65×65	65×65
Result size	129×129	129×129
Computational complexity	$O(4225)$	$O(6)$ —With block size 26×26
Time consumed	5374 seconds	6 seconds
Platform where experiments performed	A 600MHz desktop PC with a Pentium III CPU, Linux OS.	

Table 5.3.3 The comparison of two 2D texture synthesis algorithms

Efros *et. al.* have already shown the comparison of some synthesis results produced by their method [Efros2001] and the method of [Wei2000]. They report that their algorithm is particularly effective for *semi-structured* textures, which were always difficult for statistical texture synthesis methods. In Figure 5.3.8, we show two synthesis result images produced by the two approaches using a sample texture from our database.

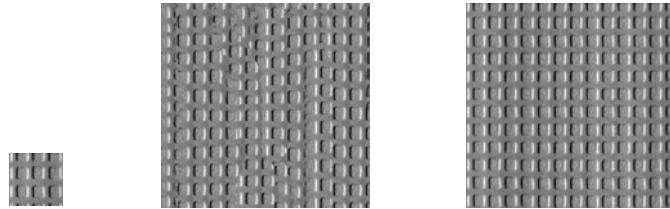


Figure 5.3.8 Two example synthesised images produced by the two approaches using a sample texture “acd” from our database. The image in the left is the input sample (65×65). The image in the middle is the synthesis result of the first approach. The image in the right is the synthesis result of the second approach. The size of result images is 256×256 .

5.3.4. Summary

We have investigated two 2D texture synthesis approaches. The first approach synthesises a new image by decomposing the input sample into a multi-resolution image set and searching the best-matched neighbourhood for every pixel in the result image. The second method generates a new image by ‘stitching’ together small patches from the sample image. Since the second approach can produce better synthesis results while requiring less computation, we select it as our basic algorithm for the synthesis of 3D surface textures.

5.4. Analysis of the selected synthesis algorithm

In last section, we selected a patch-based 2D texture synthesis approach as our basic algorithm for the synthesis of 3D surface texture. The selected approach requires four parameters as input, comprising: (1) a sample image, (2) a block size, (3) an overlap size and (4) an error tolerance. The four input parameters will affect the computation required by the algorithm and the quality of final synthesis results.

These effects are very important to the synthesis of 3D surface textures. This section will therefore analyse the algorithm in terms of computation and synthesis results by varying the input parameters.

5.4.1. Sample image size

Efros *et. al.* suggest that the input sample texture should contain enough variability [Efros2001]. Thus, the more stochastic patterns the sample texture contains, the larger sample image the algorithm should use. The reason is that a larger sample can provide more information and more choice when searching for best-matched blocks. If the sample texture contains many irregular elements, e.g. different beans of different sizes and shapes, we should use a larger sample. Otherwise, a smaller sample will cause mismatching between blocks and then lead to discontinuities in the result image. However, since the computational complexity of the synthesis algorithm is proportional to the total number of blocks contained in the input sample, a large sample requires more computations. We may trade off the quality of synthesis results against synthesis speeds by selecting an appropriate sample size. Figure 5.4.1 shows synthesis results using sample images of different sizes.

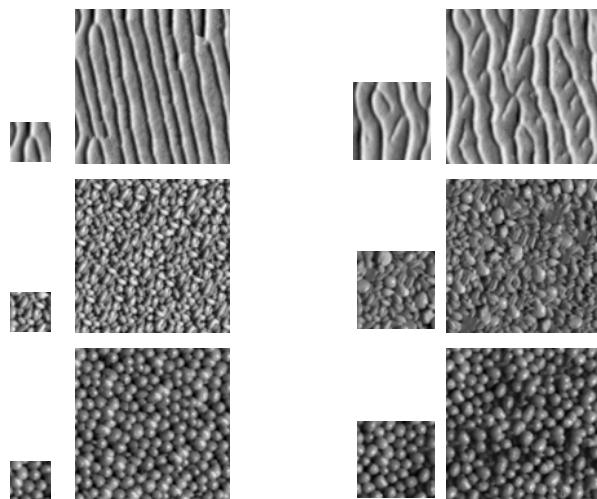


Figure 5.4.1 Synthesis results produced by using different input sample sizes. The size of sample images is 65×65 in the first column and 128×128 in the third column. The size of result images is 256×256. The three textures from top to bottom are “abj”, “add” and “adf” respectively.

5.4.2. Block size

The second input parameter required by the synthesis algorithm is *block size*. This parameter is crucial to the quality of synthesis results and speed. As reported in [Efros2001], the block should be large enough to capture the relevant structures or pattern in the sample texture. However, it must also be small enough so that the interaction between these structures is left to the algorithm. The overlarge block size will introduce more matching errors and may cause the result image losing the stochastic properties. On the other hand, it will reduce the computation required by the synthesis algorithm, since a large block size results in the sample image containing fewer blocks. Figure 5.4.2 shows the synthesis results using two example textures of different input block sizes. All other input parameters remain constant throughout the experiments.

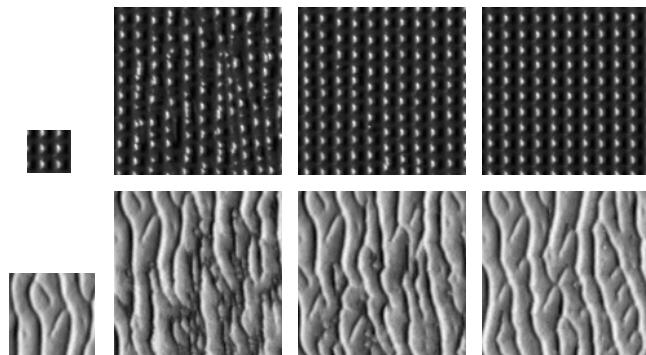


Figure 5.4.2 Synthesis results produced by using different input block sizes. The images in the first column are input sample images. In the first row (texture “ach”), the second to the fourth image uses block size of 4,8 and 17 respectively. In the second row (texture “abj”), the second to the fourth image uses block size of 5,10 and 20 respectively.

5.4.3. Overlap size

The third input parameter of the selected algorithm is the *overlap size*—the size of overlapping areas between neighbour blocks. We use the ‘width’ to represent the size of a vertical overlapping area and use the ‘height’ to represent the size of a horizontal one. The reason we can make this simplification is that the length of the other edges of overlapping areas is decided by the block size, which is constant in the synthesis process. Efros and Freeman report that they use 1/6 of a block size as the proper overlap size in their experiments [Efros2001].

Inappropriate overlap sizes will cause faulty matching during synthesis which will introduce discontinuities in the synthesis results. The reason is that pixels in overlapping areas are used for searching for the best-matched blocks. The algorithm calculates Sum of Absolute Differences (SAD) using those pixels in the overlapping areas; a block with the minimum SAD will be selected as the best-matched block. If the overlap size is too small or too large, there are either too few or too many pixels that can be used to calculate SAD. In either case, the minimum SAD might not represent the real best-matched blocks due to the sum effect. For example, suppose the best synthesis results are achieved by using a size that leads to each overlapping area containing 200 pixels. For each block location, the algorithm calculates

$$\min_j \left\{ \sum_{i=1}^{200} |I(x_i, y_i) - I(x'_i, y'_i)|_{\Omega_j} \right\} \quad (5.4.1)$$

where:

Ω_j is the overlapping area containing 200 pixels covered by block j in the sample image and the already synthesised pixels in the result image

(x_i, y_i) represents the i^{th} pixel in the sample image covered by the overlapping area Ω_j

(x'_i, y'_i) represents the i^{th} pixel in the result image covered by the overlapping area Ω_j

$I(x_i, y_i)$ is the i^{th} pixel value in the sample image

$I'(x'_i, y'_i)$ is the i^{th} pixel value in the result image.

Suppose another overlap size that makes the overlapping area contain 600 pixels. Then the following statement is not guaranteed to hold:

$$\min_j \left\{ \sum_{i=1}^{200} |I(x_i, y_i) - I(x'_i, y'_i)|_{\Omega_j} \right\} = \min_j \left\{ \sum_{i=1}^{600} |I(x''_i, y''_i) - I(x'''_i, y'''_i)|_{\Omega'_j} \right\} \quad (5.4.2)$$

where:

Ω'_j is the overlap area containing 600 pixels covered by block j in the sample image and the already synthesised pixels in the result image

(x_i'', y_i'') represents the i^{th} pixel in the sample image covered by the overlapping area Ω'_j

(x_i''', y_i''') represents the i^{th} pixel in the result image covered by the overlapping area Ω'_j

$I(x_i'', y_i'')$ is the i^{th} pixel value in the sample image

$I'(x_i''', y_i''')$ is the i^{th} pixel value in the result image.

Furthermore, small overlap sizes can not provide enough choice for the boundary cut, which is designed to produce smooth transitions in overlapping areas. All of these will lead to discontinuities or even ‘garbage’ in the result image. Figure 5.4.3 (a) and (b) show example synthesis results of two sample textures using different input overlap sizes. The results contain discontinuities and artefacts due to either oversmall and overlarge overlap sizes.

However, varying overlap size does not have significant impact on the synthesis results of *semi-structured* textures. This is obvious because *semi-structured* textures contain simple patterns which can be easily ‘stitched’ together. An example is shown in Figure 5.4.3 (c). In general, we found based on our experiments that using an overlap size between $1/6$ to $1/3$ of the block size can produce reasonable results.

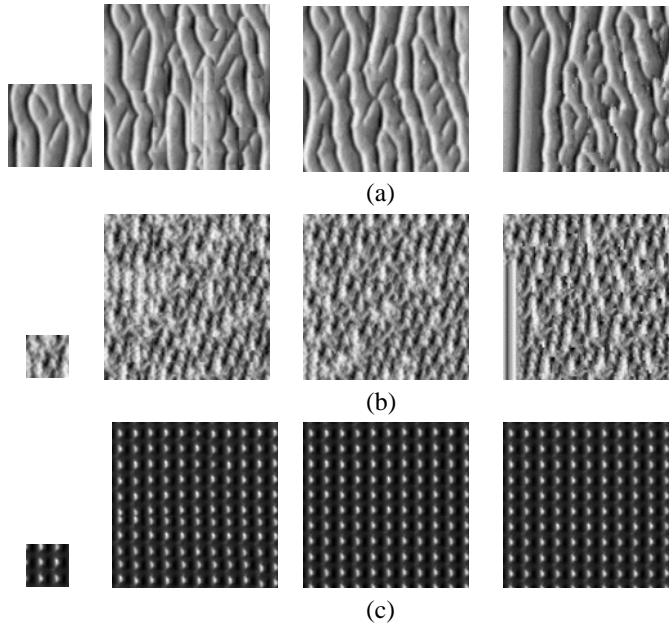
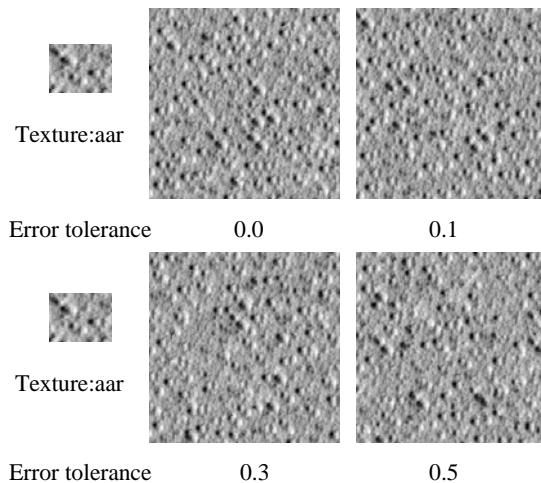


Figure 5.4.3 Synthesis results produced by using different input overlap sizes. In each row, the first image is the sample image; the second to the fourth images are result images produced by using different overlap sizes: (a) (Texture “abj”) 1, 6 and 15; (b)(Texture “aam”) 1, 6 and 15 and (c)(Texture “ach”) 1, 5 and 10 respectively. All other input parameters are kept constant.

5.4.4. Error tolerance

The fourth parameter of the algorithm is the error tolerance, which allows the algorithm to randomly choose a block from those that have similarity metrics within a certain range of the minimum one. Thus, more randomness may be introduced in the synthesis results. However, larger error tolerances will introduce more matching errors. Efros and Freeman used 0.1 in their experiments as the error tolerance when selecting best-matched blocks [Efros2001]. In our experiments, we have found that using the error tolerance between 0.0 to 0.1 does not produce much difference for synthesis results. Figure 5.4.4 shows two examples with a set of error tolerances.



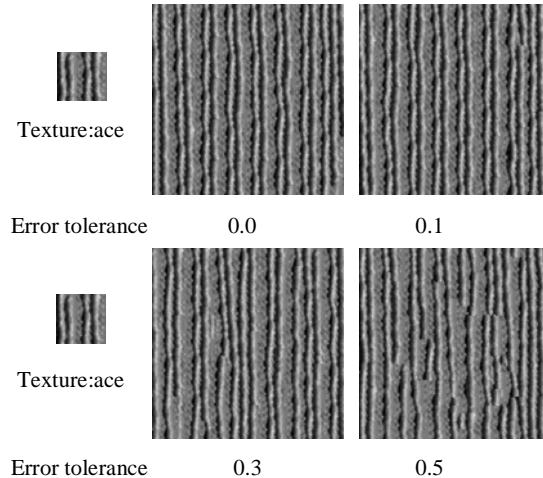


Figure 5.4.4 Synthesis results produced by using different error tolerances. The small images are samples (64x64); the large images (256x256) are synthesis results with different error tolerances. The error tolerances are listed under the result images.

5.4.5. Strength and weakness

As reported in [Efros2001], this algorithm performs remarkably well on *semi-structured* textures, which normally contain obvious boundaries between repeated near-regular patterns. These obvious boundaries and near-regular patterns can simplify the matching between blocks. Therefore better synthesis results can be produced. Figure 5.4.5 shows two highly structured textures and their synthesis results.

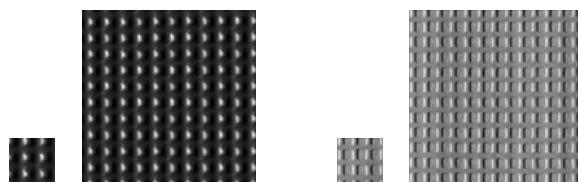


Figure 5.4.5 Example synthesis results of two highly structured textures (texture “ach” and “acd”).

The algorithm has problems when synthesising textures with irregular elements, e.g. a texture that comprises beans of different sizes and shapes. Figure 5.4.6 shows an example. Texture “ada” comprises lentils, which have different individual shapes and are randomly layout. Discontinuities and artefacts are obvious in the result image. The reason is that the algorithm employs a square block with a

constant size. Irregular granularities make matching and obtaining the best boundary cut difficult; more flexible patch shapes should be used to produce seamless boundary cuts. Recent research has shown promising results to solve this issue. Kaplan and Salesin developed an algorithm to solve the “Escherization” problem [Kaplan2000]. Their algorithm can find a new closed figure similar to the sample and use it to tile the plane. Kwatra *et. al.* use graph cuts for choosing irregular patches and can seamlessly paste the patches during texture synthesis [Kwatra2003]. However, it is still difficult to develop efficient methods for selecting auto-adaptive block during synthesis process. This algorithm remains one of the best choices for the synthesis of 3D surface textures in terms of synthesis speeds and the quality of results.

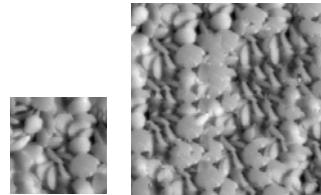


Figure 5.4.6 A failed example (Texture “add”). The algorithm has problems when synthesising textures that comprise irregular elements.

5.4.6. Summary

We analysed the selected 2D texture synthesis algorithm in terms of the inputs required: the sample image size, block size, overlap size and error tolerance. The larger sample image contains more information and can produce better synthesis results, but it also increases computations. The block size should be big enough to capture the basic structures or patterns in the sample image. The overlap size should be between 1/6 to 1/3 of the block size. Inappropriate overlap sizes will introduce discontinuities or artefacts to the result image. The error tolerance between 0.0 and 0.1 does not produce much effect on the synthesis results.

The selected algorithm can produce remarkable synthesis results for semi-structured textures, whereas it has difficulties to synthesise textures with irregular elements or granularities. The reason is that the algorithm uses a fixed square block in order to capture the basic texture structures.

5.5. Conclusion

The aim of this chapter is to investigate available 2D texture synthesis methods and select an efficient algorithm that can be easily extended for use with relighting representations of 3D surface textures. This is the second stage of our overall framework.

We first presented a detailed review of 2D texture synthesis approaches based on two criteria: (1) the suitability of the algorithm for extension to deal with multi-dimensional representations, and (2) the capability of producing good results while requiring little computation. Then we investigated two popular algorithms proposed by Wei *et. al.* [Wei2000] and Efros *et. al.* [Efros2001]. Since the latter produces better results while requiring less computation, we selected it as our basic synthesis algorithm. In addition to the sample image, the algorithm requires a block size, an overlap size and an error tolerance as inputs. We analysed the effect on the quality of synthesis results when varying the four input parameters. Based on [Efros2001] and our experiments, the primary conclusion is that the block should be bigger than basic texture patterns/granularities perceived by human vision and the overlap size should be between 1/6 to 1/3 of block size.

In next chapter we will describe how to combine surface relighting representations that we introduced in the previous chapter with the synthesis algorithm described in this chapter to synthesise and relight 3D surface textures.

Chapter 6

Synthesis and Relighting

6.1. Introduction

In chapter 4, we introduced five methods for representing and relighting surface textures. In chapter 5, we selected a 2D texture synthesis algorithm. In this chapter we present five approaches that combine the surface representation methods with the 2D texture synthesis algorithm to synthesise images of 3D surface textures under arbitrary lighting directions. We will compare these synthesis approaches according to the quality of their output results. The criterion for the comparison is the resemblance, as perceived by human vision, between output results and input samples. The work described in this chapter corresponds to the final stage in our overall framework, as highlighted in Figure 6.1.1.

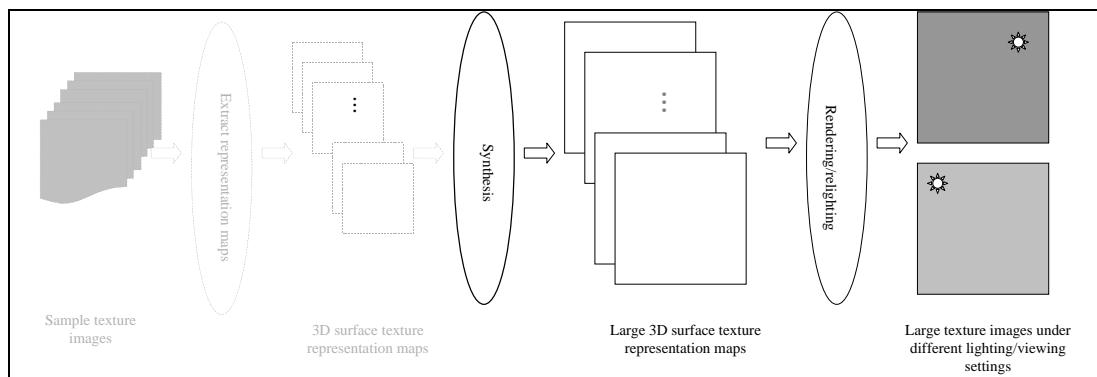


Figure 6.1.1 The final stage of the overall framework

We modify the 2D texture synthesis algorithm selected in chapter 5 so that it can take sample surface representations as input and perform synthesis in multi-dimensional space. We propose five approaches for the synthesis of 3D surface textures that correspond to the five surface representation and relighting methods introduced in chapter 4:

The *3I* synthesis approach: This approach uses three images of the sample texture as input and it relights the synthesised images using the *3I* relighting method. Synthesis is performed in \mathbf{R}^3 space.

The *Gradient* synthesis approach: The second approach uses surface gradient and albedo maps as input and it relights the synthesised surface gradient and albedo maps using the *Gradient* relighting method. Synthesis is also performed in \mathbf{R}^3 space.

The *PTM* synthesis approach: This approach uses Polynomial Texture Maps (PTM) as input and it relights synthesised PTMs using the *PTM* relighting method. Synthesis is performed in \mathbf{R}^6 space.

The *Eigen3* synthesis approach: The fourth approach uses the first three eigen base images as input and it relights synthesised base images using the *Eigen3* relighting method.

The *Eigen6* synthesis approach: This is identical to the previous approach except that it uses the first six base images as input. Thus, synthesis is performed in \mathbf{R}^6 space.

For our experiments we use the same 23 textures as those used in chapter 4.

We are also interested in the performances of these five approaches concerning the quality of their synthesised results. In chapter 4, we performed a quantitative assessment of the five surface representation and relighting methods. However, we can not perform a similar quantitative comparison here because ground truth data is not available. We therefore qualitatively assess the five synthesis approaches. We perform psychophysical experiments to rank these five approaches based on human perception. Based on the rank data, we use Fredman's nonparametric two-way Analysis of Variance followed by a multi-comparison method to test their significance. The conclusion is that the *Gradient* and *Eigen3*

approaches outperform any of the other approaches if both the synthesised results and computational cost are considered.

The chapter is organised as follows. Section 6.2 introduces the five synthesis approaches. Section 6.3 describes the psychophysical experiments for the qualitative comparison of the five approaches. Finally in section 6.4 we draw conclusions from the results of this chapter.

6.2. The five synthesis approaches

This section introduces five synthesis approaches: *3I*, *Gradient*, *PTM*, *Eigen3* and *Eigen6*. They employ the same basic algorithm—the modified Efros and Freeman’s 2D texture synthesis algorithm. However, they use different inputs, which comprise different multi-dimensional vectors that represent a sample surface texture under arbitrary illumination directions. During the synthesis process, each pixel location on the sample surface is represented by multi-dimensional vectors that are extracted using the surface representation methods introduced in chapter 4. The synthesis algorithm uses the multi-dimensional vectors as input to synthesise new surface representation maps. They are finally relit using the relevant relighting methods to obtain new images under different illumination directions.

6.2.1. The general algorithm for the synthesis of surface texture representations

The general algorithm for the synthesis of surface texture representations is an extension of the 2D synthesis algorithm that we selected in chapter 5. The algorithm synthesises a result representation by ‘stitching’ together small blocks from a sample representation. It uses a Sum of Absolute Differences (SAD) as the metric for selecting best-matched blocks in the sample. For 2D texture synthesis, the calculation of SAD only uses pixel intensity values. In the case of 3D surface texture synthesis, each pixel location on the sample surface is expressed as a multi-dimensional vector. The general algorithm therefore uses multi-dimensional vectors. The SAD that we use for multi-dimensional surface representations is:

$$\text{SAD} = \sum_{i=1}^n \sum_{\substack{(x,y) \in \Omega_j \\ (x',y') \in \Omega_j}} |m_i(x, y) - m'_i(x', y')| \quad (6.2.1)$$

where:

(x, y) represents a sample pixel location

(x', y') represents a result pixel location

$m_i(x, y)$ is a pixel value at (x, y) in the i^{th} sample representation map

$m'_i(x', y')$ is a pixel value at (x', y') in the i^{th} result representation map

Ω_j is an overlapping area covered by block j

n is the dimensionality or the total number of sample representation maps.

The best-matched blocks are found by minimising the SAD between the overlapping windows of the sample and result representation maps.

The sample surface and output representations are stored as multiple images. The number of images is equal to the dimension of the representations. Thus synthesis in \mathbf{R}^3 space involves three input images and three output images, as shown in Figure 6.2.1.

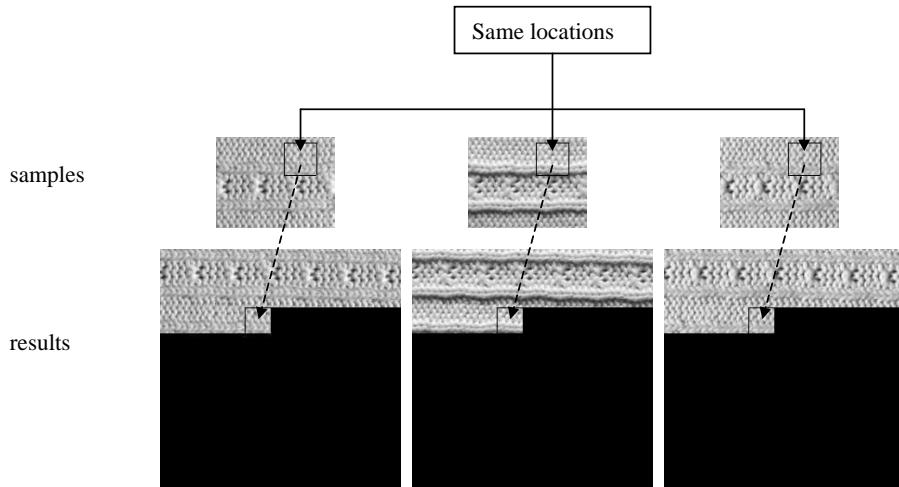
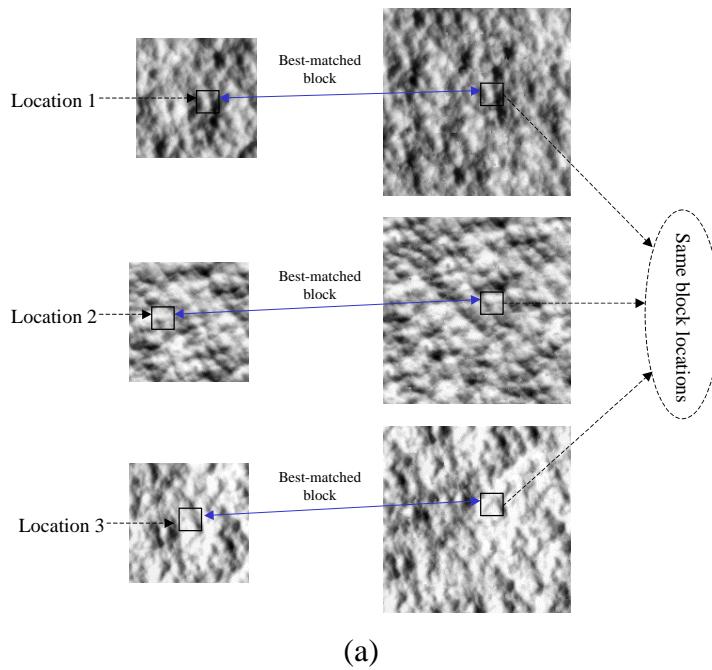


Figure 6.2.1 Each group of best-matched blocks in synthesised results comes from the same location in samples

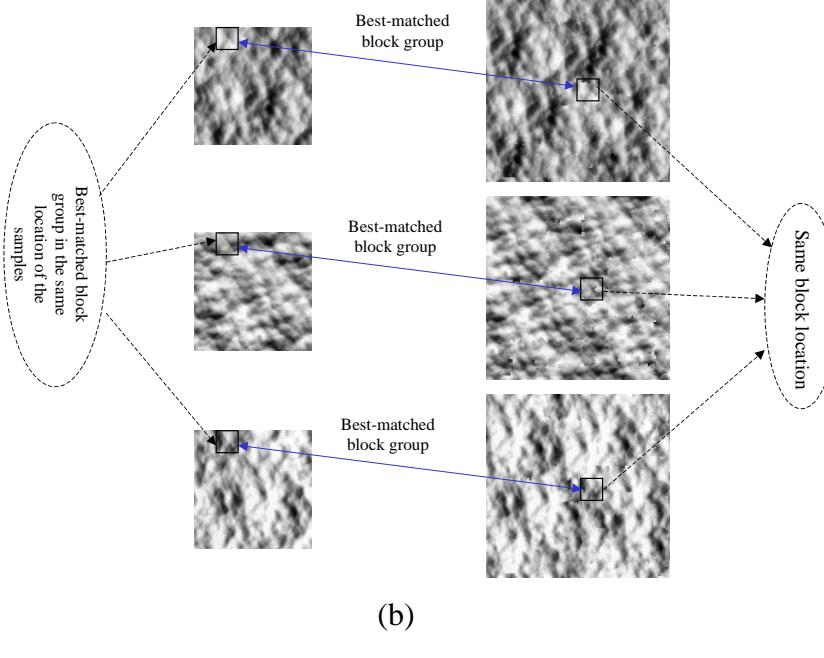
The synthesised representation maps are then relit using corresponding relighting methods to produce the final results.

Matching errors

It should be noted that *matching errors* exist during the selection of best-matched blocks by calculating the minimum SAD in \mathbf{R}^n space. Suppose we are observing two synthesis processes. The first process synthesises only one representation map in \mathbf{R}^1 space using pixel values as input; the second synthesises all representation maps simultaneously in \mathbf{R}^n space. All other parameters are identical. At the same locations of two output representation maps, the best-matched block obtained in \mathbf{R}^1 space might be different from its counterpart in the group of best-matched blocks that are produced simultaneously in \mathbf{R}^n space (using n -dimensional vectors as input). In the other words, the group of best-matched blocks produced in \mathbf{R}^n space does not guarantee each individual in the group is the same as the best-matched block produced in \mathbf{R}^1 space. Figure 6.2.2 illustrates this process. Each large image (output) in Figure 6.2.2 (a) is synthesised independently in \mathbf{R}^1 space. For the framed blocks in output images, their best-matched blocks in the samples have different locations. These locations also differ from those in the sample images of (b), in which synthesis is performed in \mathbf{R}^3 space. In (b), all framed blocks in output images lie in the same location.



(a)



(b)

Figure 6.2.2 The group of best-matched blocks produced in \mathbf{R}^3 space does not guarantee each individual in the group is the same as the best-matched block produced in \mathbf{R}^1 space. (a) Each large image (output) is synthesised separately in \mathbf{R}^1 space; all framed blocks in the output images lie in the same location but their best-matched blocks have different locations in the samples. (b) Synthesis in \mathbf{R}^3 space. All framed blocks lie in the same location in output images and are identical to those in (a), but their best-matched block group has the same location in the samples. This location differs from each of those in (a).

The reason for producing matching errors is that the minimum SAD, which decides the best-matched blocks, is normally greater than zero when synthesising real-world surface texture representations. Thus, the following mathematical statement is obvious:

$$\min_{\Omega_j} \left\{ \sum_{(x,y) \in \Omega_j} \sum_{(x',y') \in \Omega_j}^n |m_i(x,y) - m'_i(x',y')| \right\} \geq \sum_{i=1}^n \min_{\Omega_j} \left\{ \sum_{(x,y) \in \Omega_j} |m_i(x,y) - m'_i(x',y')| \right\} \quad (6.2.2)$$

The left side of equation (6.2.2) represents the minimum SAD calculated using n -dimensional vectors, while the right side is the sum of the minimum SAD calculated in \mathbf{R}^1 space. The *matching error* can be seen as the difference between the two sides of equation (6.2.2). The higher the dimensionality of input vectors is, the larger the

matching errors might be. *Matching errors* will introduce discontinuities in the result representation maps.

6.2.2. The *3I* synthesis approach

The *3I* synthesis approach first synthesises three output images from three sample *photometric images*, which are captured under linearly independent illumination directions. The synthesis is therefore performed in \mathbf{R}^3 space. The three synthesised *photometric images* are then relit to generate new images under arbitrary illumination directions using a linear combination—the *3I* relighting method, as introduced in chapter 4. Figure 6.2.3 shows the process in \mathbf{R}^3 space.

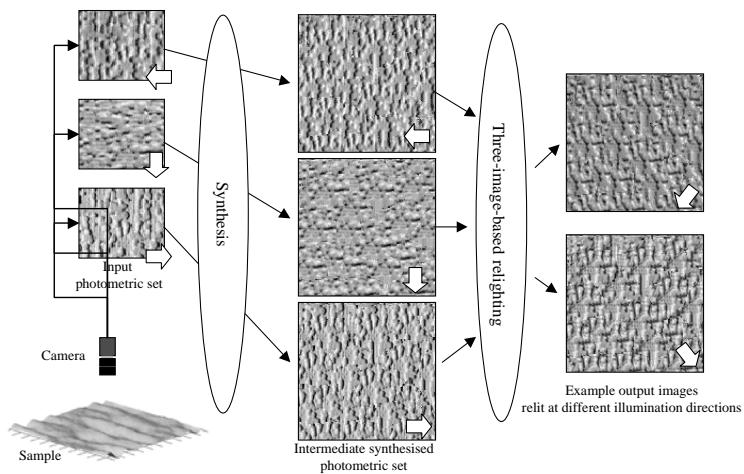


Figure 6.2.3 The *3I* synthesis approach

6.2.3. The *Gradient* synthesis approach

The *Gradient* synthesis approach synthesises output surface gradient and albedo maps from sample maps. These are generated using the *Gradient* representation method. Synthesis is also performed in \mathbf{R}^3 space. Since pixel values in surface gradient maps are normally smaller than those in the albedo map, all pixel values are transformed into same scale during synthesis process. This gives the surface gradient and albedo maps the same weight when calculating Sum of Absolute Difference (SAD). However, the synthesised surface gradient and albedo maps still use pixel values from the corresponding original sample maps. They are relit using the Lambertian model to generate final images under arbitrary illumination directions. Figure 6.2.4 shows the whole synthesis process.

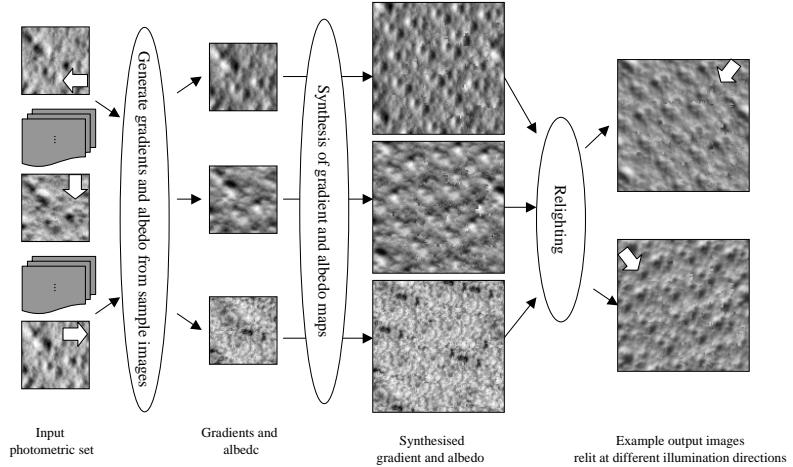


Figure 6.2.4 The Gradient synthesis approach

6.2.4. The *PTM* synthesis approach

This *PTM* synthesis approach performs synthesis in \mathbf{R}^6 space. The six-dimensional sample Polynomial Texture Maps are also transformed into same scale so that they have the same weight when calculating SAD. The synthesised PTMs are relit using the *PTM* relighting method [Malzbender2001] to produce final images under different illumination directions.

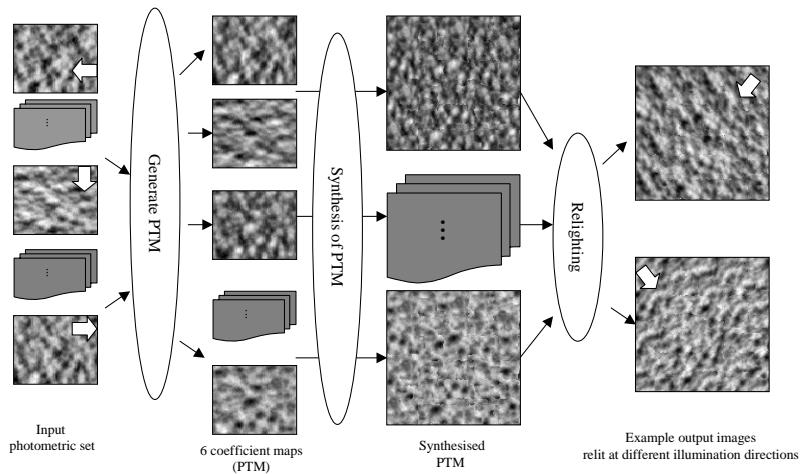


Figure 6.2.5 The PTM synthesis approach

6.2.5. The *Eigen3* and *Eigen6* synthesis approaches

The *Eigen3* or *Eigen6* approach uses the first 3 or 6 eigen base images as input to synthesise output eigen base images. The sample eigen base images are generated

using the *Eigen3* or *Eigen6* surface representation method. They are also transformed into the same scale during synthesis process so that they have equal weight in calculating SAD between samples and results. The synthesised base images are relit using a bilinear interpolation—the eigen-based relighting methods described in chapter 4 to generate new images under varied illumination directions.

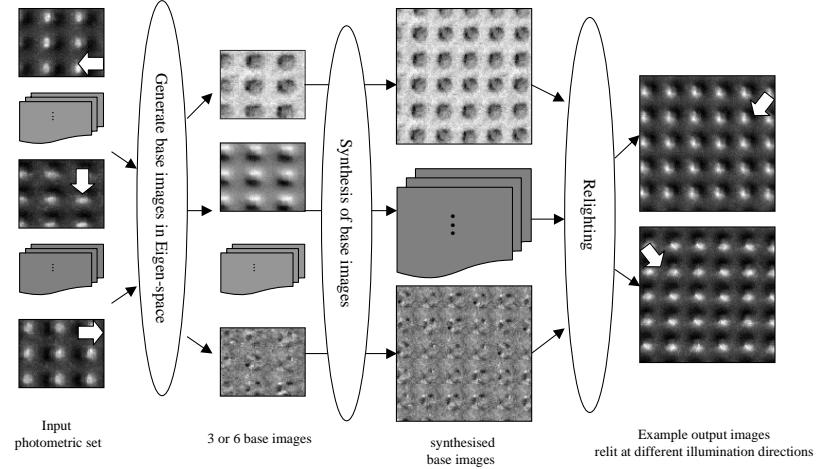


Figure 6.2.6 The Eigen3 and Eigen6 approaches

6.2.6. Summary

We have presented five approaches for the synthesis and relighting of 3D surface textures. They use surface representation maps extracted from a set of sample images as input to synthesise new surface representations. The synthesised representations are then relit using the corresponding relighting methods to generate final result images under arbitrary illumination directions. We summarise the five approaches in Table 6.2.1. Synthesis results of 23 textures with illumination angles of ($\tau = 60^\circ$, $\sigma = 60^\circ$) and ($\tau = 120^\circ$, $\sigma = 60^\circ$) are shown in Appendix B.

Table 6.2.1 Summary of the 5 approaches

Approach	1 st phase	2 nd phase	3 rd phase
<i>3I</i>	No processing required in this phase as the three images are used directly	\mathbf{R}^3 synthesis (produces 3 output photometric images)	Image-based relighting (produces final image)
<i>Gradient</i>	Produces sample gradient(p, q) and albedo maps (al) using all sample images	\mathbf{R}^3 synthesis (produces output gradient and albedo maps)	Gradient-based relighting
<i>PTM</i>	Generates sample Polynomial Texture Maps	\mathbf{R}^6 synthesis (produces output Polynomial Texture Maps)	PTM- based Relighting
<i>Eigen3</i>	Generates 3 base images of sample in eigen-space	\mathbf{R}^3 synthesis (produces output eigen base images)	Eigen-based relighting
<i>Eigen6</i>	Generates 6 base images of sample in eigen-space	\mathbf{R}^6 synthesis (produces output eigen-base images)	Eigen-based relighting

6.3. Qualitative assessment of the five approaches

Section 6.2 described five approaches for the synthesis and relighting of 3D surface textures. This section evaluates the performances of these methods concerning the quality of their synthesis results. In chapter 4, we have quantitatively assessed the surface representation and relighting methods. The conclusion is that the *3I* representation method produces the worst performance and the *Eigen6* method produces the best. The \mathbf{R}^6 *PTM* representations perform better than \mathbf{R}^3 *Gradient* representations, although it can not be considered superior to the computationally cheaper *Eigen3* representations in \mathbf{R}^3 space. We are interested in whether the qualitative performance¹ of the five synthesis approaches is consistent with the quantitative assessment results of relighting methods.

Despite the significant quantity of research on texture synthesis approaches little has been published concerning their assessment. The majority of researchers therefore simply display their results alongside those of their competitors and leave

¹ Note that unlike the assessment of surface representation and relighting methods, we can not perform a quantitative comparison because no ground-truth data is available.

the comparison to readers [DeBonet1997, Wei2000, Efros1999, Xu2001, Efros2001 and shikhmin2001]. Few provide any experimental support. Copeland *et. al.* did use a psychophysical experiment with ten observers to assess the ability of a numerical error metric to model the perceptual differences between texture patterns [Copeland2001] but very little has been published on the systematic qualitative assessment of texture synthesis results *per se*. In this section, we introduce a simple qualitative approach which uses nonparametric statistical tests and psychophysical experiments.

6.3.1. Design of the psychophysical experiments

Since we are interested in comparing the performances of the five synthesis approaches concerning the quality of synthesis results, we use rank (ordinal) data as the scale of statistic measurement. An ordinal scale of measurement represents an ordered series of relationships or rank order. In our case, we wish to know which methods outperform others or which one can achieve the best, second, or third performance. Unlike precise measurement, rank data is suitable for qualitative measurement. Furthermore, the advantage of using rank data is that it can be simply obtained from observation.

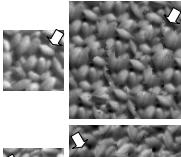
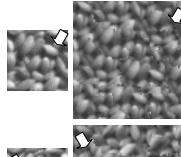
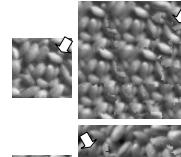
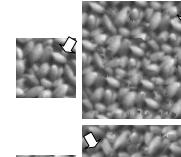
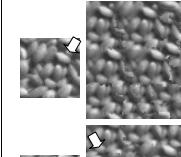
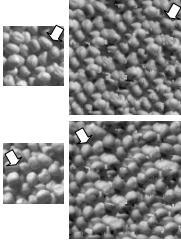
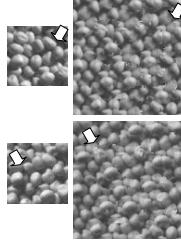
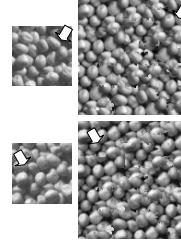
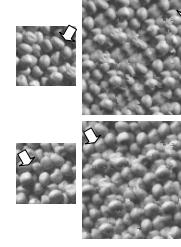
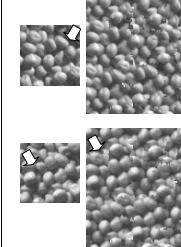
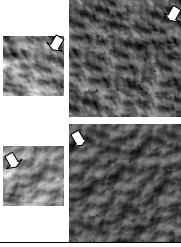
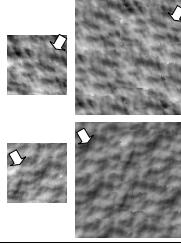
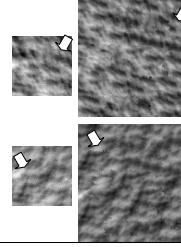
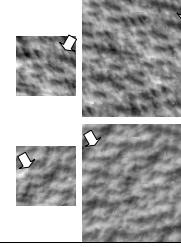
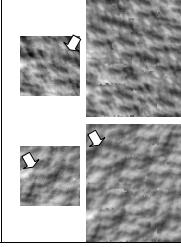
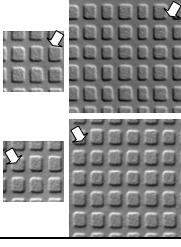
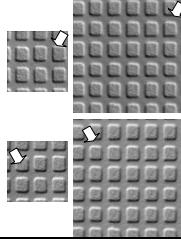
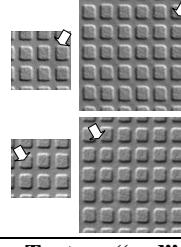
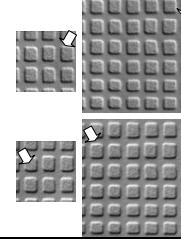
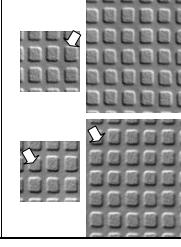
We asked a set of ten human observers to rank different synthesis approaches by comparing output images with input samples. The main concern is the resemblance between the samples and results under multiple illumination directions. In order to avoid distraction from other effects during comparison, we simply place the sample images alongside results with same illumination conditions. Although we have performed the synthesis on 23 sample textures and we can generate images with arbitrary illumination directions, we only select a representative subset from the results for the psychophysical experiments so that observers are relieved from exhaustive comparison. The subset comprises eleven textures (near 50% of all textures) with two illumination directions. These textures include surfaces that exhibit near Lambertian reflectance, Lambertian reflectance with shadows and interreflections, and specular reflectance. These textures also include surfaces with stochastic and structured patterns.

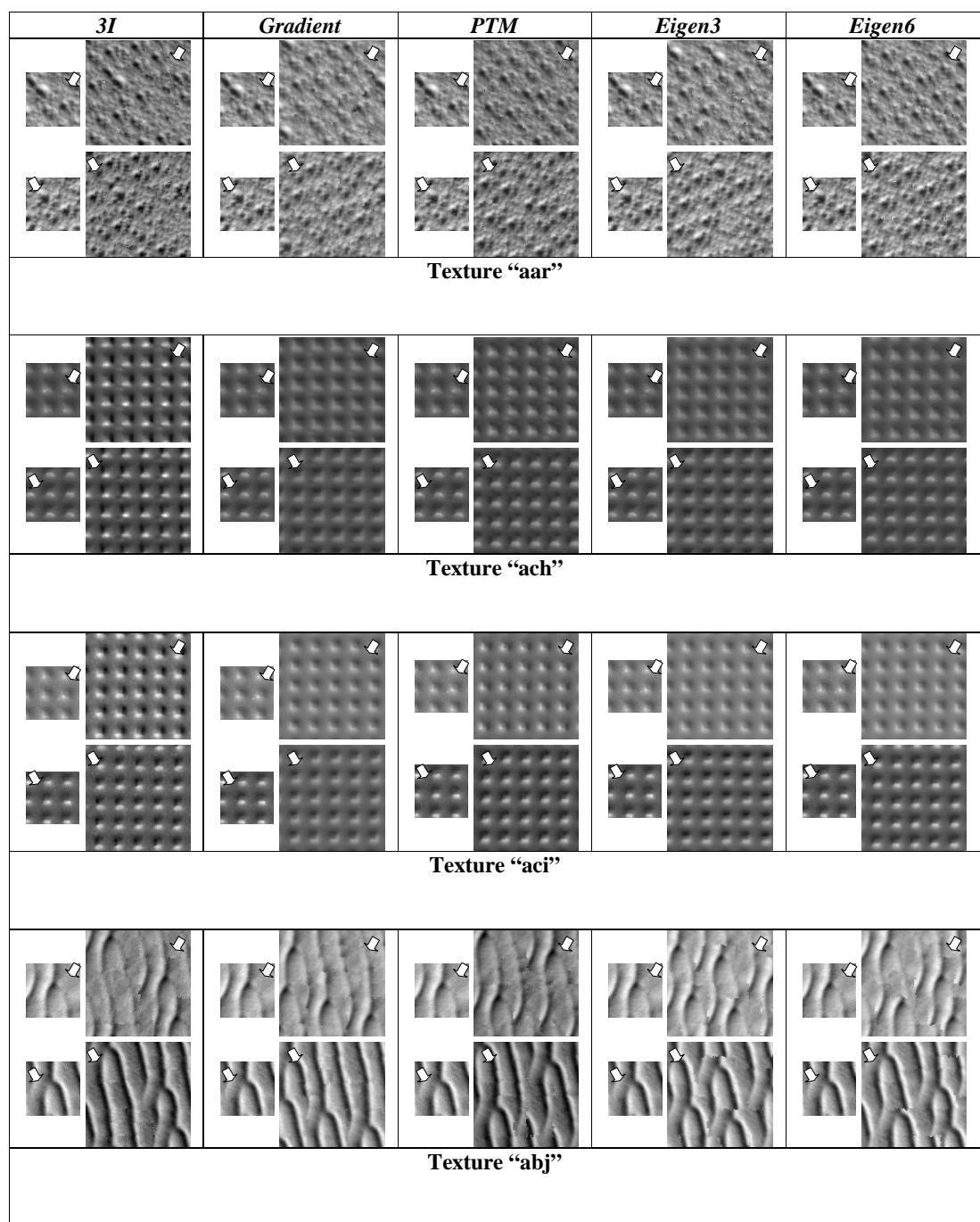
For each texture, we used each of the five approaches to synthesise two output images under illumination angles of ($\tau = 60^\circ$, $\sigma = 60^\circ$) and ($\tau = 120^\circ$, $\sigma = 60^\circ$). These

images are shown in Table 6.3.1 and labelled as “aa^j”, “aas”, “ace”, “adc”, “add”, “aar”, “acd”, “aai”, “ach”, “aci” and “abj”. Observers were asked to compare real sample images with synthesised images and rank the results for each of the eleven textures from the best to the worst. The illumination directions are indicated by block arrows in the figure. No other instructions were given concerning as to what qualities to look for when comparing methods. Thus we collect 110 sets of rankings (10 observers x 11 textures).

Table 6.3.1Synthesis and relighting results from the five methods for 11 textures. The small images in each cell are the samples; the large images are synthesis results. Arrows indicate illumination directions ($\tau = 60^\circ$ and $\tau = 120^\circ$).

<i>3I</i>	<i>Gradient</i>	<i>PTM</i>	<i>Eigen3</i>	<i>Eigen6</i>
Texture "aaj"				
Texture "aas"				
Texture "ace"				

<i>3I</i>	<i>Gradient</i>	<i>PTM</i>	<i>Eigen3</i>	<i>Eigen6</i>
				
Texture "adc"				
				
Texture "add"				
				
Texture "aa'i"				
				
Texture "acd"				



6.3.2. The test of significant difference—Friedman’s nonparametric two-way Analysis of Variance

We firstly would like to know whether there are significant differences between the performances of these approaches according to the rankings. Since observers performed their rankings independently, we use Friedman’s nonparametric two-way Analysis of Variance (ANOVA) to test for significance.

Friedman’s nonparametric two-way Analysis of Variance (ANOVA) is designed to determine if we may conclude from sample evidence that there are differences between treatment effects (which in our case are the five approaches). We therefore construct a matrix which contains one column for each approach. Each column contains 110 rank data (10 observers x 11 textures). Friedman’s test compares the means of these columns (see [Daniel1990] for more details). The null hypothesis H_0 is that there are no significant differences between the five methods, while the alternative hypothesis H_1 is that at least one is different. The test statistic is defined as:

$$\chi_r^2 = \frac{12}{bk(k+1)} \sum_{j=1}^k \left[R_j - \frac{b(k+1)}{2} \right]^2 \quad (6.3.1)$$

where:

b is total number of rank data for each method (110)

k is the number of methods to be compared (5), and

R_j is the sum of rank data for each method.

The test results indicated that there is at least one method which performs significantly differently from the others at a confidence level of 100%.

6.3.3. The multiple comparison

Since there is significant difference between the performances of these approaches, we are interested in which approaches perform better than others. We therefore use a multiple comparison test of means that is designed to provide an upper bound on the probability that any comparison will be incorrectly found to be significant [Hochberg1987]. The multiple comparison compares each pair of approaches and outputs the confidence interval for the difference at certain confidence level.

We use the Statistic Toolbox in Matlab to perform the multiple comparison. The result is shown in Figure 6.3.1. Each group mean is represented by a small circle within an interval. Two means are significantly different if the associated intervals are disjoint, and are not significantly different if their intervals overlap.

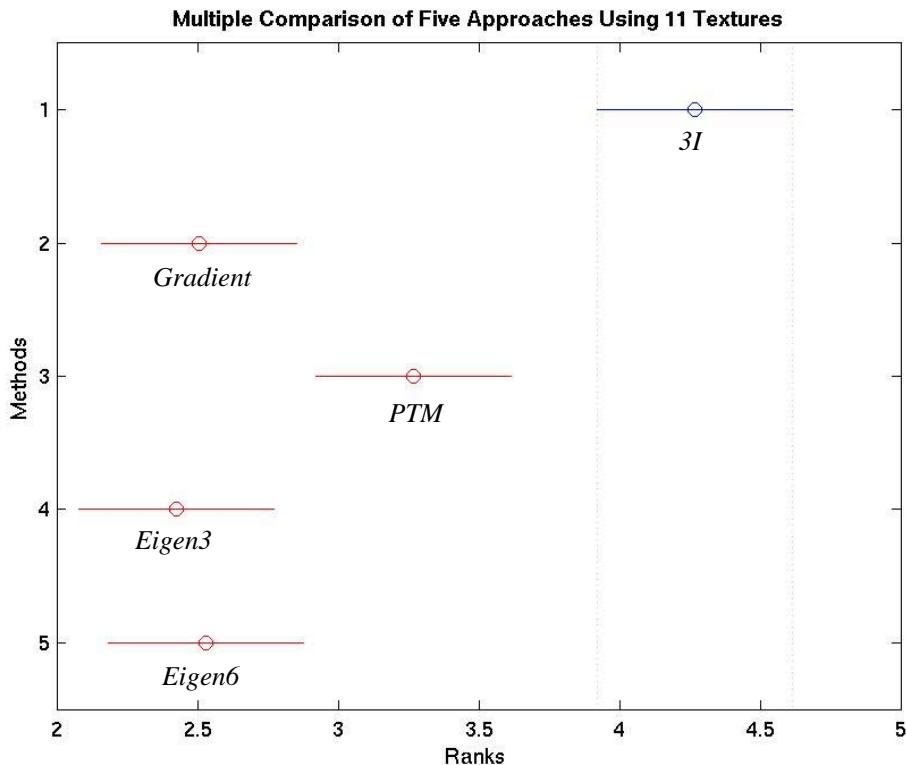


Figure 6.3.1 Multiple comparison test of the five approaches. Small circles and lines represent the group means and their intervals. The horizontal axis indicates rank values. Two means are significantly different if their intervals are disjoint.

Based on the results of this test in which the confidence levels of the intervals are 99% ($\alpha = 0.01$) we make the following observation. There are no significant differences between the performances of the *Gradient*, *Eigen3*, and *Eigen6* approaches. However, each of these methods does outperform both *3I* and *PTM*, while the *PTM* method outperforms the *3I*.

Although *Eigen6* produced the best quantitative relighting results, its qualitative performance in the synthesis experiments was not significantly better than its two nearest competitors: *Gradient* and *Eigen3*. This is maybe because synthesis is

performed in \mathbf{R}^6 space which is more prone to matching errors. These errors often introduce discontinuities, which are particularly noticeable to human observers. Consequently, when the samples and results with same illumination directions are being compared, the effect due to discontinuities might counteract the good performance produced in relighting. Therefore, the overall performance of *Eigen6* is lowered to the same level as *Eigen3* and *Gradient* in the qualitative assessment. Correspondingly, although *PTM* performed better than *Gradient* in the relighting assessment, it failed to outperform *Gradient* in the qualitative comparison of synthesis results.

If we take computation complexity into account, we find that synthesis in \mathbf{R}^6 space is of course the most expensive. It exactly doubles the computation time compared with \mathbf{R}^3 synthesis. Thus we conclude that the *Gradient* and *Eigen3* approaches on average offer as good a performance as of any of the other methods and incur low computational cost. However, if image-acquisition requirements have to be kept low then the *3I* synthesis approach, which uses only three photometric images, provides relighting at the cost of lower quality output.

6.4. Conclusion

In this chapter, we proposed five approaches for the synthesis and relighting of 3D surface texture. The five approaches—*3I*, *Gradient*, *PTM*, *Eigen3* and *Eigen6* use the corresponding surface representations of a sample texture as input to a modified version of Efros and Freeman’s image quilting method. The synthesised surface representations are relit to produce new images under arbitrary illumination directions. For the *3I*, *Gradient*, and *Eigen3* approaches, synthesis is performed in \mathbf{R}^3 space, while the *PTM* and *Eigen6* approaches perform synthesis in \mathbf{R}^6 space.

We qualitatively compared the five approaches by employing psychophysical experiments. We asked ten observers to rank different synthesis approaches by comparing output images with input sample images. The ranked data were first tested using Friedman’s nonparametric two-way Analysis of Variance. The test suggests that there is at least one significant difference between the performances of these five approaches. A multiple comparison was then applied to determine which approaches outperform others. The conclusion is that, at the confidence level 99%,

the *Gradient*, *Eigen3* and *Eigen6* approaches perform better than *3I* and *PTM*. If computation complexity is taken into account, the *Gradient* and *Eigen3* approaches are preferable.

Chapter 7

Conclusion and Discussion

7.1. Summary

The aim of this thesis is to develop inexpensive approaches for 3D surface texture synthesis. This is motivated by the desire for realistic texture synthesis in augmented and virtual reality applications. The synthesised results should be able to be rendered under varied illumination directions. They should also be compatible with the input requirement of computer graphics programming and software packages so that real-time rendering can be achieved using personal computers with modern graphics cards.

In chapter 2, we presented an overview of the research fields related to this thesis. We have surveyed three fields: (1) 3D surface texture synthesis, (2) 2D texture synthesis, and (3) surface representation methods for relighting. The research into 3D surface texture synthesis only received attention in the past three years. Among the available five publications [Zalesny2000, Zalesny2001, Liu2001, Tong2002 and Leung2001], the methods described in [Zalesny2000 and Zalesny2001] aim to synthesise new texture images under different viewpoints with a fixed illumination direction. In [Liu2001], Liu *et. al.* develop a method that can synthesise Bidirectional Texture Functions (BTF) of Lambertian surfaces by combining a shape-from-shading technique with a 2D texture synthesis algorithm. In later work [Tong2002], Tong *et. al.* define surface textons by linearly combining appearance vectors associated with 3D textons [Leung2001] and use them for

synthesising BTFs on surfaces of 3D models. Although there are only five publications regarding 3D surface texture synthesis, a great number of techniques have been published in the research fields of both 2D texture synthesis and extraction of surface representations.

Therefore, we proposed an overall framework for the synthesis of 3D surface textures in chapter 3. The framework essentially combines surface representation methods with 2D texture synthesis algorithms to synthesise new surface representations. They then can be relit to generate new images under arbitrary illumination directions. In chapter 3, we also defined the data environment for all experiments in the thesis. We selected 23 textures according to two criteria: one is the requirement of suitable granularities; the other is the coverage of different texture types. Thus, the selected textures comprise rough and smooth surfaces, glossy and matte surfaces, non-shadowing and shadowing surfaces as well as near-regular and stochastic patterns.

In chapter 4, we selected five low dimensional methods for extracting representations of the 3D surface texture sample **and** investigated the relighting of these representations. We first introduced our criteria for the selection of surface representations. These criteria include the practicality of physical data capture, the low dimensionality of representations, the compatibility of representations with graphics systems and the capability of dealing with complex reflectance including shadows and specularities. Then we surveyed the literature and selected five surface representations, namely the **3I**, **Gradient**, **PTM**, **Eigen3** and **Eigen6** methods. The **3I** uses three images of the sample texture taken at an illumination slant angle of 45° and tilt angles of 0°, 90° and 180° as surface representations. The **Gradient** method uses surface gradient and albedo maps derived from photometric stereo techniques. The **PTM** method employs Polynomial Texture Maps (PTM) to represent Lambertian surfaces exhibiting shadows and interreflections. The **Eigen3** and **Eigen6** methods use the first three and six eigen base images respectively to represent a surface with complex reflectance. These five methods were evaluated 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. The evaluation results were analysed. Our overall conclusion in chapter 4 is 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 can not be considered more superior to the cheaper *Eigen3* representations in \mathbf{R}^3 space.

In chapter 5, we selected an efficient 2D texture synthesis algorithm as the basis algorithm for the synthesis of 3D surface texture representations. We first surveyed available 2D texture synthesis algorithms according to two criteria: (1) the suitability of the algorithm for extension to deal with multi-dimensional representations, and (2) the capability of producing good results while requiring little computation. Then we selected two popular 2D texture synthesis algorithms based on [Wei2000 and Efros2001] as candidates. We investigated the two algorithms and proposed our simple modifications that can improve the synthesis speeds without affecting synthesis results. By comparing the two algorithms, we finally chose the algorithm based on [Efros2001] as the basis algorithm for the synthesis of 3D surface texture representations. We analysed the effects on output images produced by changing input parameters to the basis algorithm.

In chapter 6, we proposed five 3D surface texture synthesis approaches by extending the basis algorithm in multi-dimensional spaces. The five synthesis approaches use the five surface representations introduced in chapter 4—*3I*, *Gradient*, *PTM*, *Eigen3* and *Eigen6*—as input. The synthesised representations are then relit to generate new images under different illumination directions. In order to assess the performances of the five synthesis approaches, we employed psychophysical experiments to qualitatively compare the relighting results. We asked ten human observers to rank these five approaches according to the resemblance between the sample and synthesised images under same illumination directions. Based on the rank data, we used Friedman's nonparametric two-way Analysis of Variance followed by a multi-comparison method to test their significance. The conclusion is that there are no significant differences between the performances of the *Gradient*, *Eigen3*, and *Eigen6* approaches. However, each of

these methods does outperform both *3I* and *PTM*, while the *PTM* method outperforms the *3I*.

7.2. Conclusion

We have developed five inexpensive approaches for the synthesis of 3D surface textures. Unlike conventional 2D texture synthesis techniques, these approaches allow the synthesised results to be relit under arbitrary lighting directions. In literature, there are only five relevant publications in this research field. Our approaches essentially extend a 2D texture synthesis algorithm into multi-dimensional spaces and use five inexpensive surface representations as input. The synthesised representations can be linearly combined to generate new images under arbitrary illumination directions [Dong2002a]. These approaches require inexpensive computation. The synthesised results are compatible with computer graphics systems and therefore can be applied in real-time rendering applications.

We have investigated five surface representation methods [Dong2002b]. A mathematical framework has been developed to describe these methods. We quantitatively assessed the five surface representation methods by comparing the original and relit images. It has been shown that the *Eigen6* method, which employs the first six eigen base images to represent the sample texture, outperforms all other methods. The *3I* method, which uses three photometric images as surface representations, produces the worst performance. The *Eigen3* (using the first three eigen base images) and *PTM* (using Polynomial Texture Maps) methods outperform the *Gradient* method, which employs surface gradient and albedo maps to represent Lambertian surfaces. However, the performance of the *PTM* representations can not really be separated from that of its cheaper *Eigen3* competitor. We also discussed the problem of integration and showed that a heightmap-based representation, which is obtained from the *Gradient* method, produces even worse performance than the *3I* method.

We have developed a simple method that can qualitatively compare the five synthesis approaches by employing psychophysical experiments based on the rank data [Dong2003a]. The experiments showed that although the *Eigen6* surface representation method produced the best performance in representing sample

surfaces in the quantitative assessment, there are no significant differences between the *Gradient*, *Eigen3* and *Eigen6* synthesis approaches. However, each of these approaches does outperform both the *3I* and *PTM* approaches, while the *PTM* approach outperforms the *3I*. Therefore, if we take into computational complexity into account, the *Gradient* and *Eigen3* synthesis approaches, in general, provide better performances.

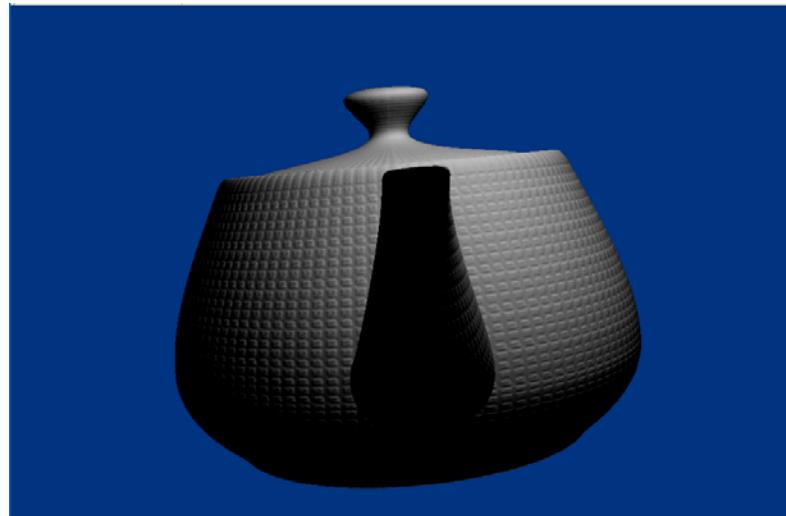
7.3. Discussion

In this section—the last section of this thesis, we discuss the use of the synthesised representations or images in computer graphics applications. We briefly introduce relevant references and basic techniques in real-time graphics programming regarding rendering the synthesised surface texture representations. We will also illustrate the use of the synthesised results in a simple computer graphics package.

7.3.1. Using the synthesised 3D surface texture representations in real-time graphics programming

For the synthesised surface gradient and albedo maps, per-pixel bump-mapping can be applied using consumer-level graphics cards to achieve real-time rendering. In [Robb2003], Robb *et. al.* introduced the method of rendering surface gradient and albedo maps using the NVIDIA GeForce Ti4600 graphics accelerator. First, the two surface gradient maps are converted to surface normal vectors. Then a vertex program is used to obtain the tangent normal and binormal of each vertex as well as the location and direction of the current light source in the tangent space. Finally, per-pixel lighting is performed using the register combiner units of the Ti4600 graphics chip, where the diffuse colour is calculated in the form of dot product between the lighting vector and the surface normal. In addition, the ambient, diffuse and specular lighting values can also be calculated and added together to achieve realistic rendering results. Figure 7.3.1 shows two still images from a real-time sequence of rendering synthesised surface gradient and albedo maps on a 3D teapot model using the method described in [Robb2003]. Both the lighting and viewing conditions are different in the two images.

For the *3I*, *PTM*, *Eigen3* and *Eigen6* methods, the synthesised surface representations can be relit by linear combinations. Given a lighting direction, the coefficients for the linear combinations (that are used to generate the relit image) can be calculated using the methods introduced in chapter 4. The linear combinations can be seen as the dot products between the coefficients and surface representations. Thus, the synthesised representation maps together with the coefficients can be firstly loaded into texture units. Then register combiners can be used to calculate dot products. However, depending on the graphics hardware, multi-pass implementations may be required. For example, the NVIDIA GeForce3 chip does not support signed addition. Thus, two passes are needed to achieve the whole linear combination process. More detail can be found in [Burschka2003] regarding the implementation of linear combinations using NVIDIA graphics cards.



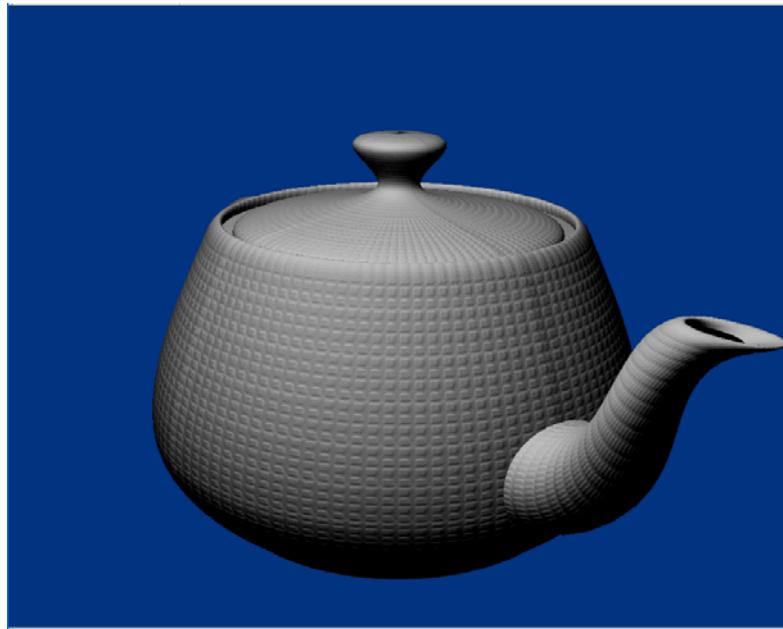
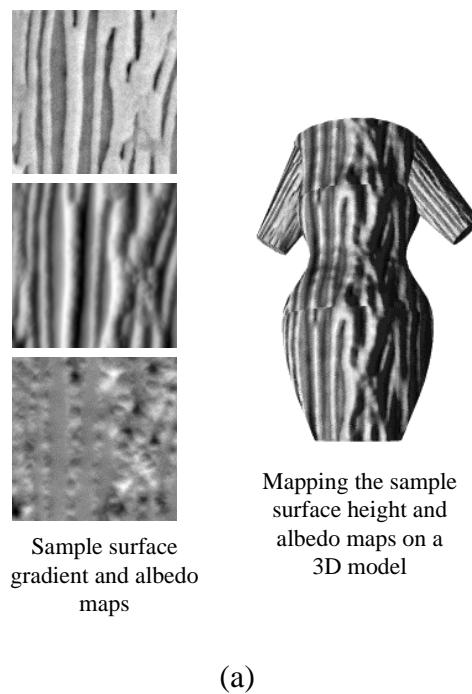


Figure 7.3.1 Two still images of a real-time sequence produced by rendering synthesised surface gradient and albedo maps using the method described in [Robb2003]. The images were generated by Michael Robb using synthesised surface gradient and albedo maps supplied by the author.

Three-dimensional surface textures with specularities can also be represented by surface geometrical and material parameters of certain reflectance models. Many methods can be used to estimate these parameters [Nayar1990, Kay1995, Rashmier1997, Saito1996, Lin2001 and Dong2003b]. The estimated parameters can then be used as input for the synthesis according to our overall framework described in chapter 3. In [Dong2003b], we introduce a simple method for the capture and synthesis of 3D surface textures with specularites. The synthesised representations can also be programmed into graphics hardware for real-time rendering. However, while the diffuse component can be calculated using dot products in graphics chips, current consumer-level graphics hardware can not directly perform the exponential calculation involved in the specular components of the reflectance models. To solve this problem, a lookup table storing the pre-calculated exponentiation can be used for the acceleration. More detail can be found in [Kautz2000 and McAllister2002].

7.3.2. Using the synthesised 3D surface texture representations in graphics software packages

The synthesised representations can be input into graphics software packages to perform texture mapping on 3D models. If the packages can not directly use the synthesised representations, certain transformation is required. We briefly introduce the use of the synthesised 3D surface gradient and albedo maps (output of the *Gradient* synthesis approach) for texture mapping in a simple 3D graphics package—Micrografx simply 3D 2. This package can accept height (displacement) and albedo maps for bump mapping. Thus, we first integrate the synthesised surface gradient maps to generate the surface heightmap (displacement map) using the method described in chapter 4. Then, we use the height map together with the albedo map for the rendering on 3D models. Figure 7.3.2 (b) shows two example output images produced by mapping synthesised height and albedo maps on a 3D model with different illumination directions. For the purpose of comparison, Figure 7.3.2 (a) shows the sample surface gradient and albedo maps, alongside the mapping results using the sample height and albedo maps, which is generated by integrating sample gradient maps.



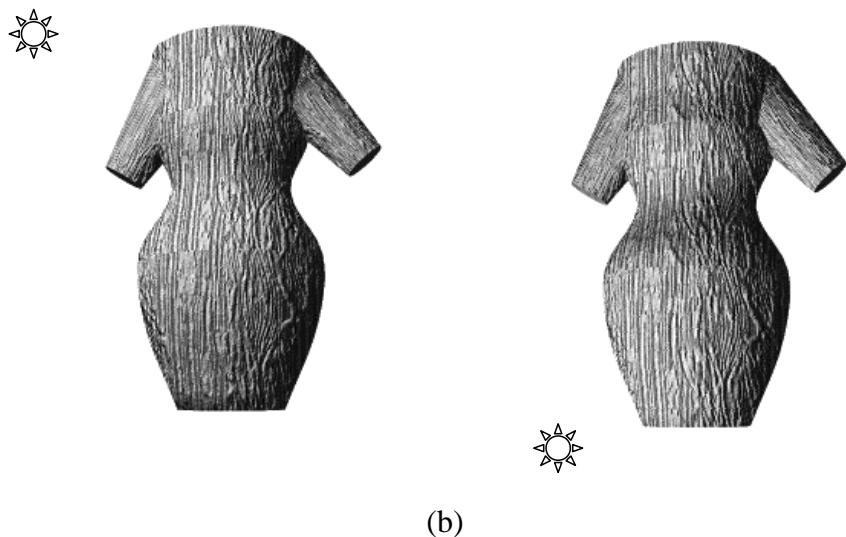


Figure 7.3.2 Texture mapping using Micrografx Simply 3D 2. (a) Left: the sample surface gradient and albedo maps; right: mapping the sample surface height and albedo maps on a 3D model. The sample height map is generated by integrating gradient maps. The sample size is 128×128. (b) Mapping the synthesised surface height and albedo maps on a 3D model. The height map is generated by integrating synthesised gradient maps (size: 512×512). The texture label is “acc”.

Alternatively, we can firstly integrate sample surface gradient maps to generate a sample height map. Then, the sample height and albedo maps can be used as input for synthesising large height and albedo maps. This method is described in more detail in [Dong2002a]. Figure 7.3.3 shows two example images of mapping the synthesised height and albedo maps on the 3D model.

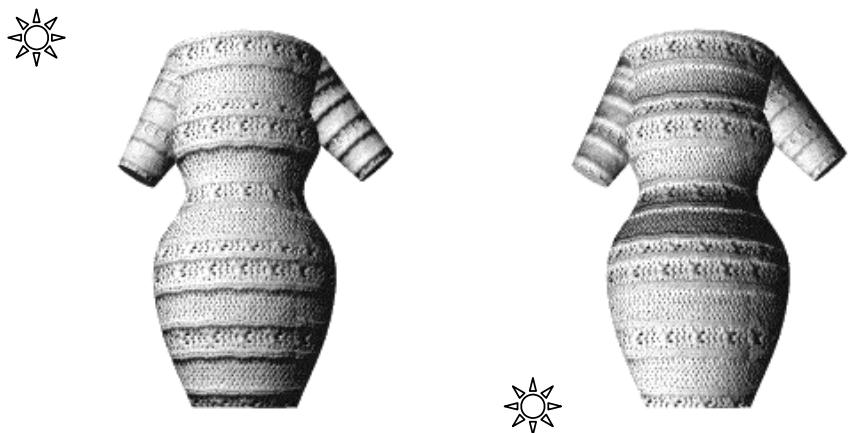
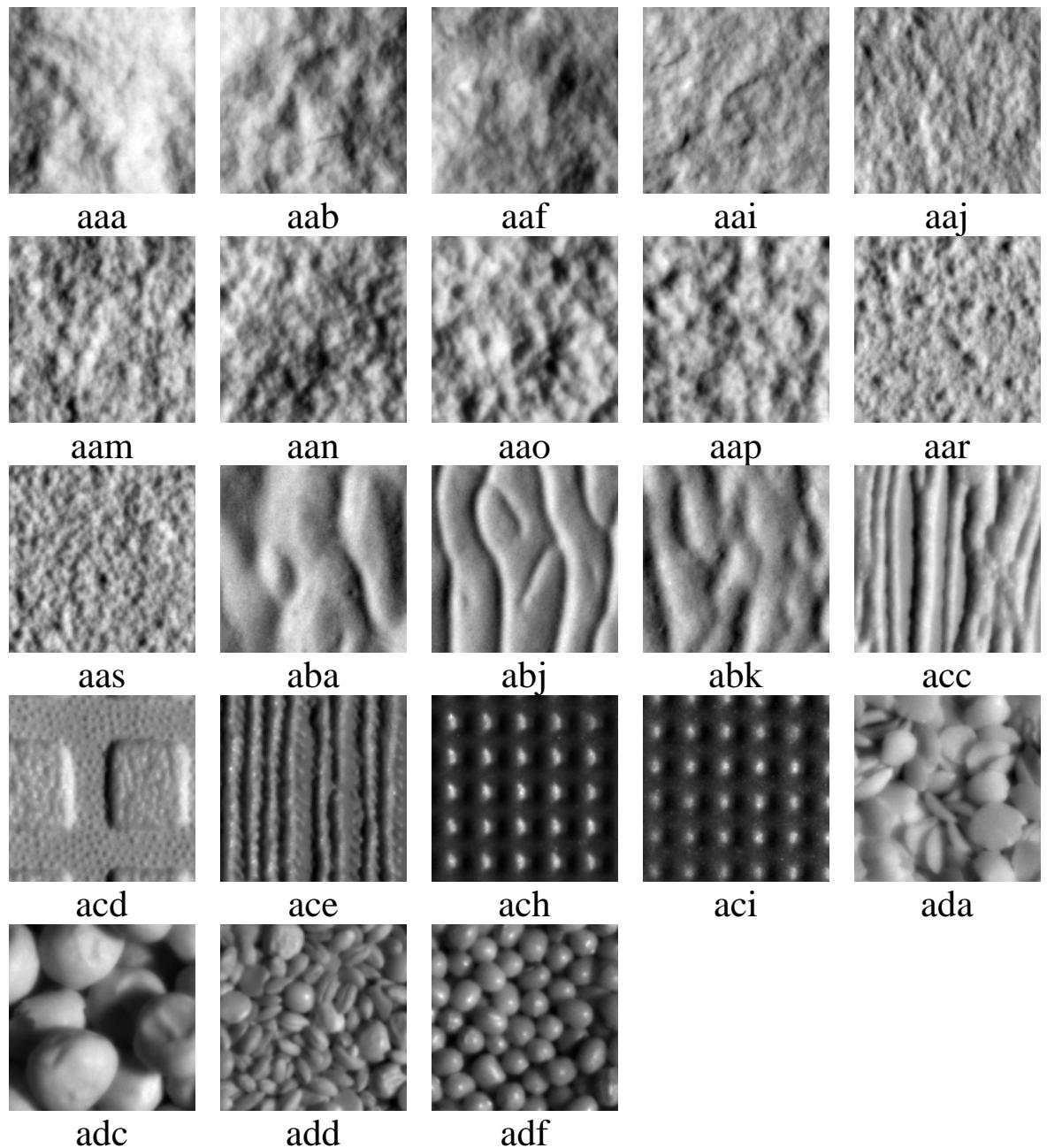


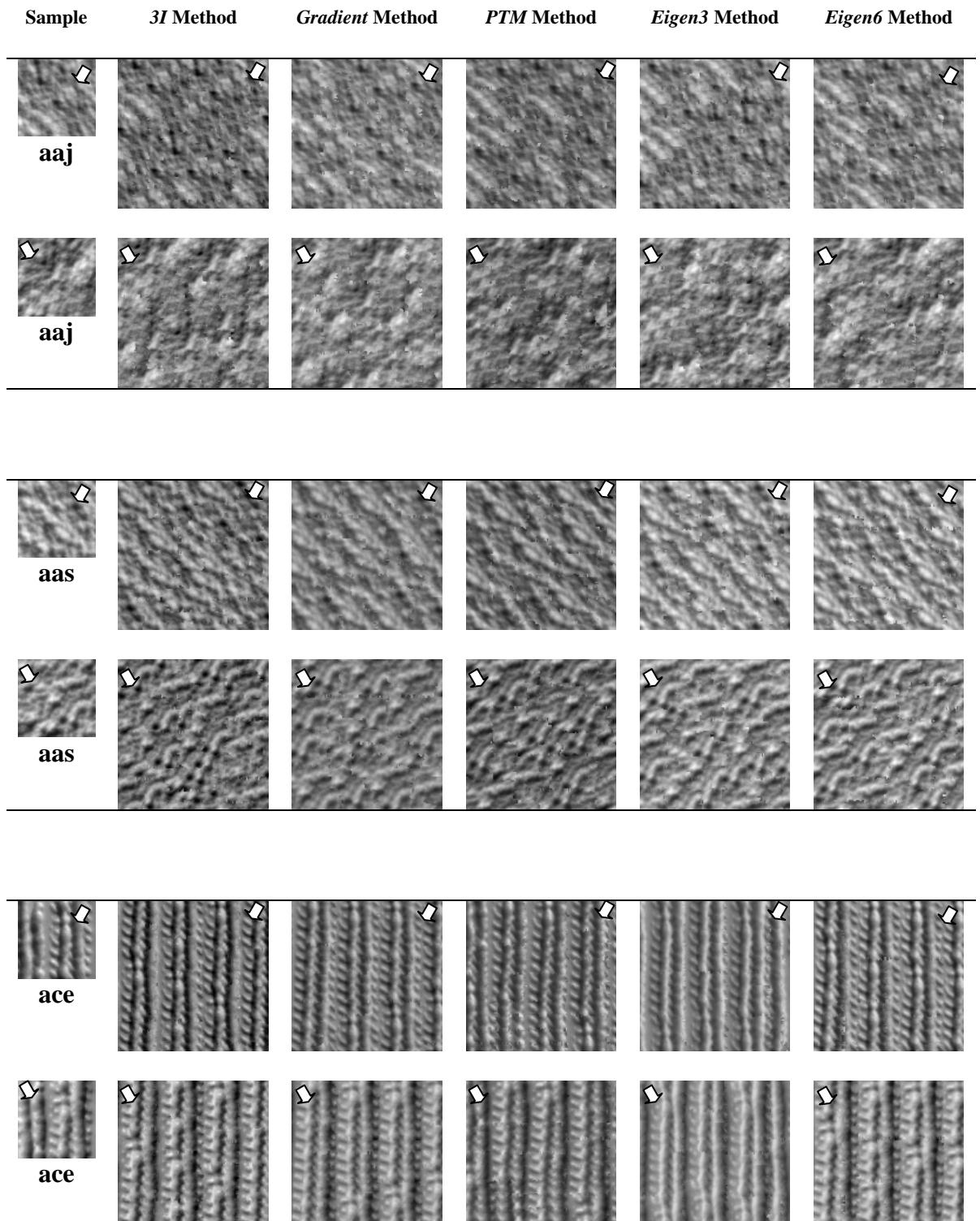
Figure 7.3.3 Texture mapping using Micrografx Simply 3D 2. The inputs are synthesised surface height and albedo maps (size: 512×512). They are generated using the sample albedo map and height map, which is produced by integrating sample gradient maps. The size of all samples is 128×128. These images are taken from [Dong2002a].

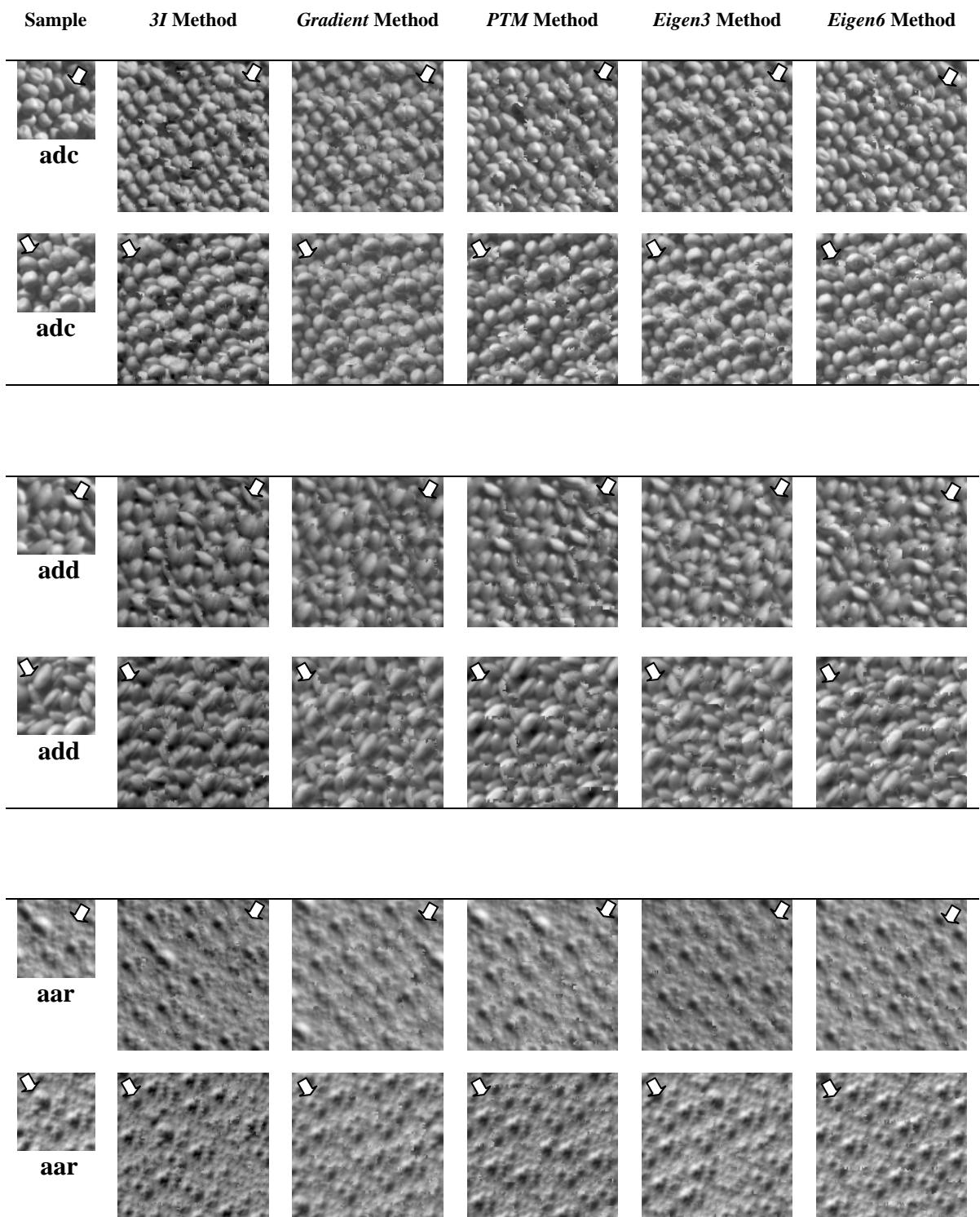
Appendix A: Texture samples

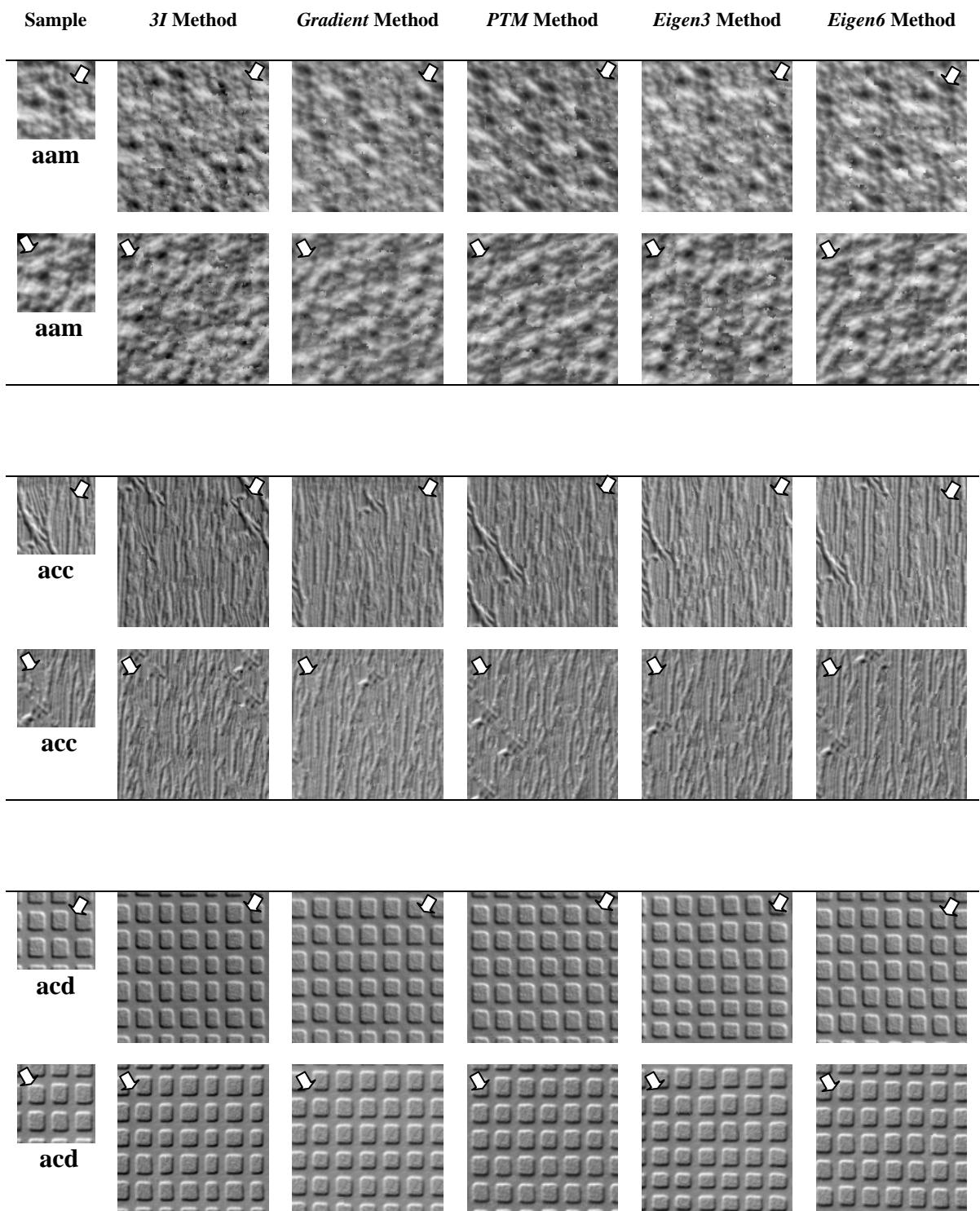


Appendix B:

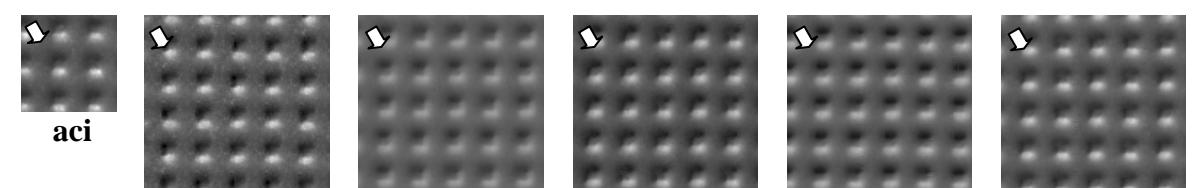
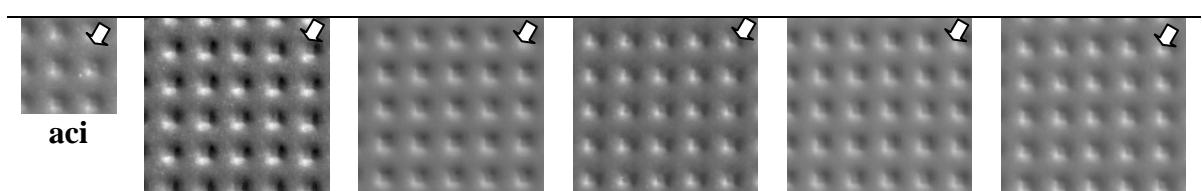
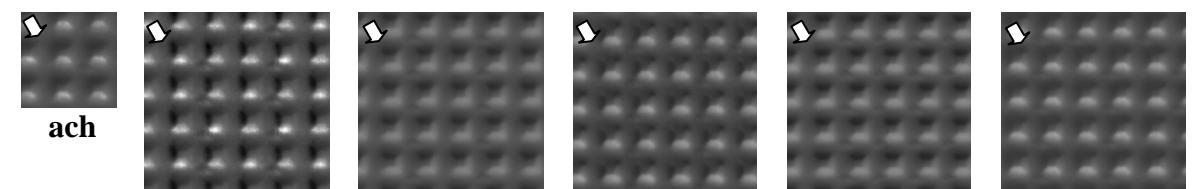
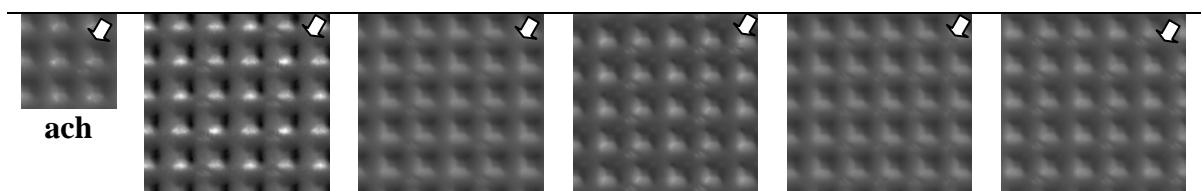
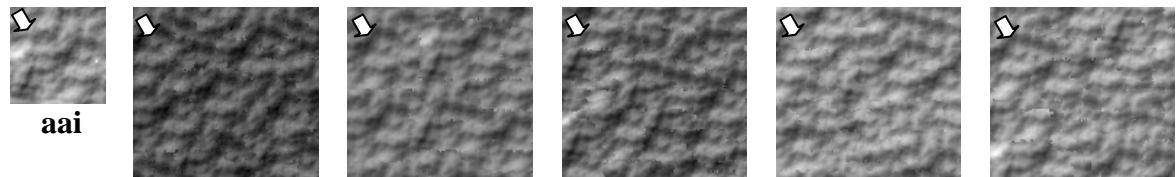
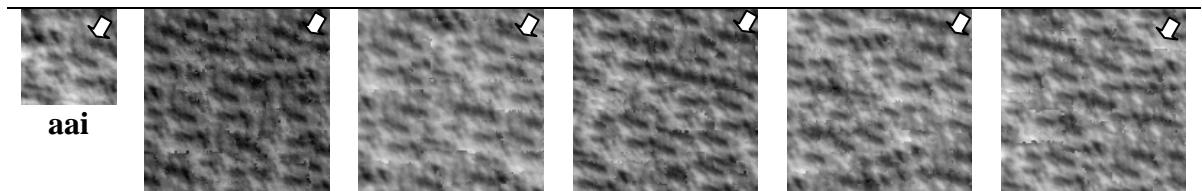
Synthesis and relighting results from the five methods for 23 textures. Arrows indicate illumination directions ($\tau = 60^\circ$ and $\tau = 120^\circ$).

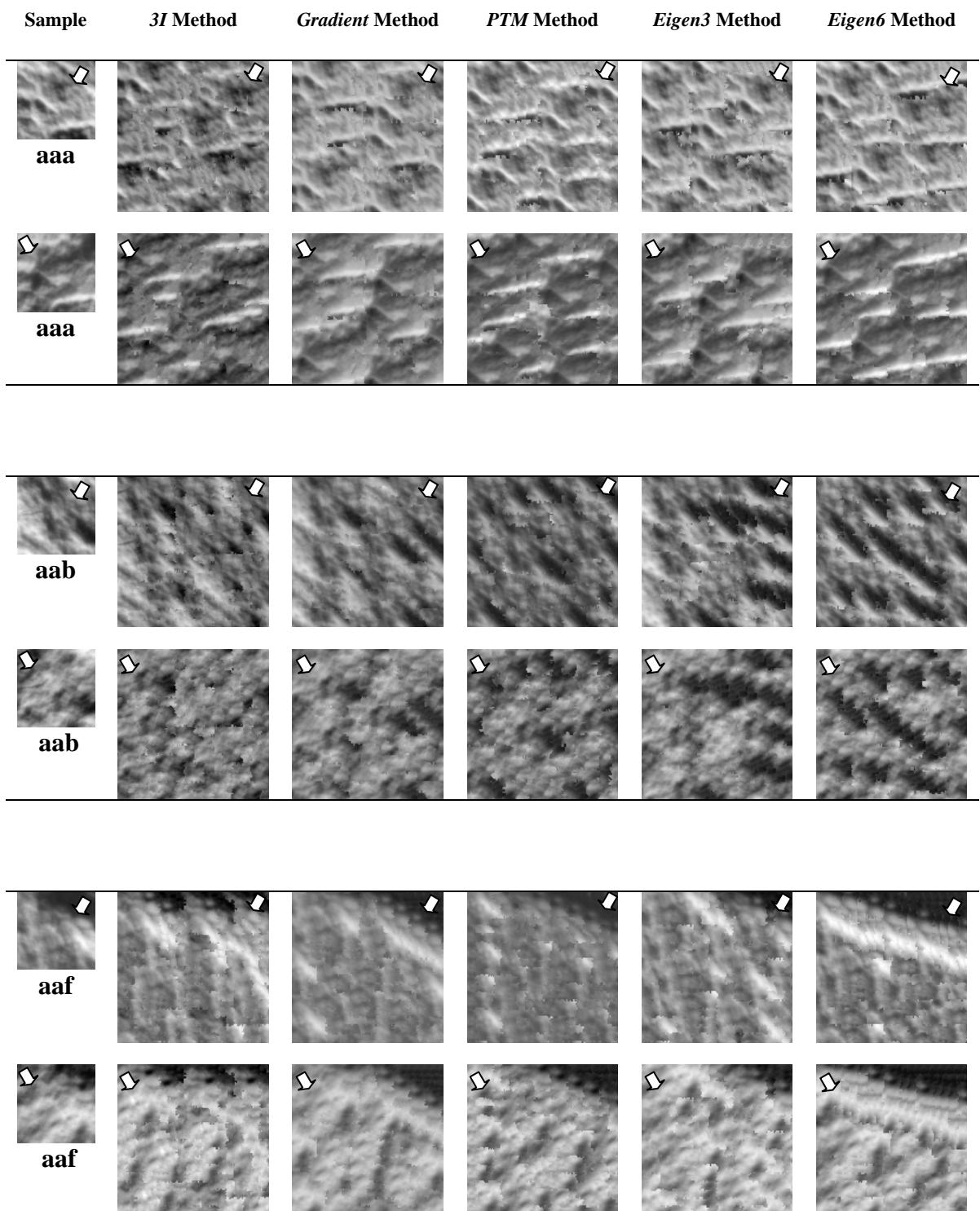


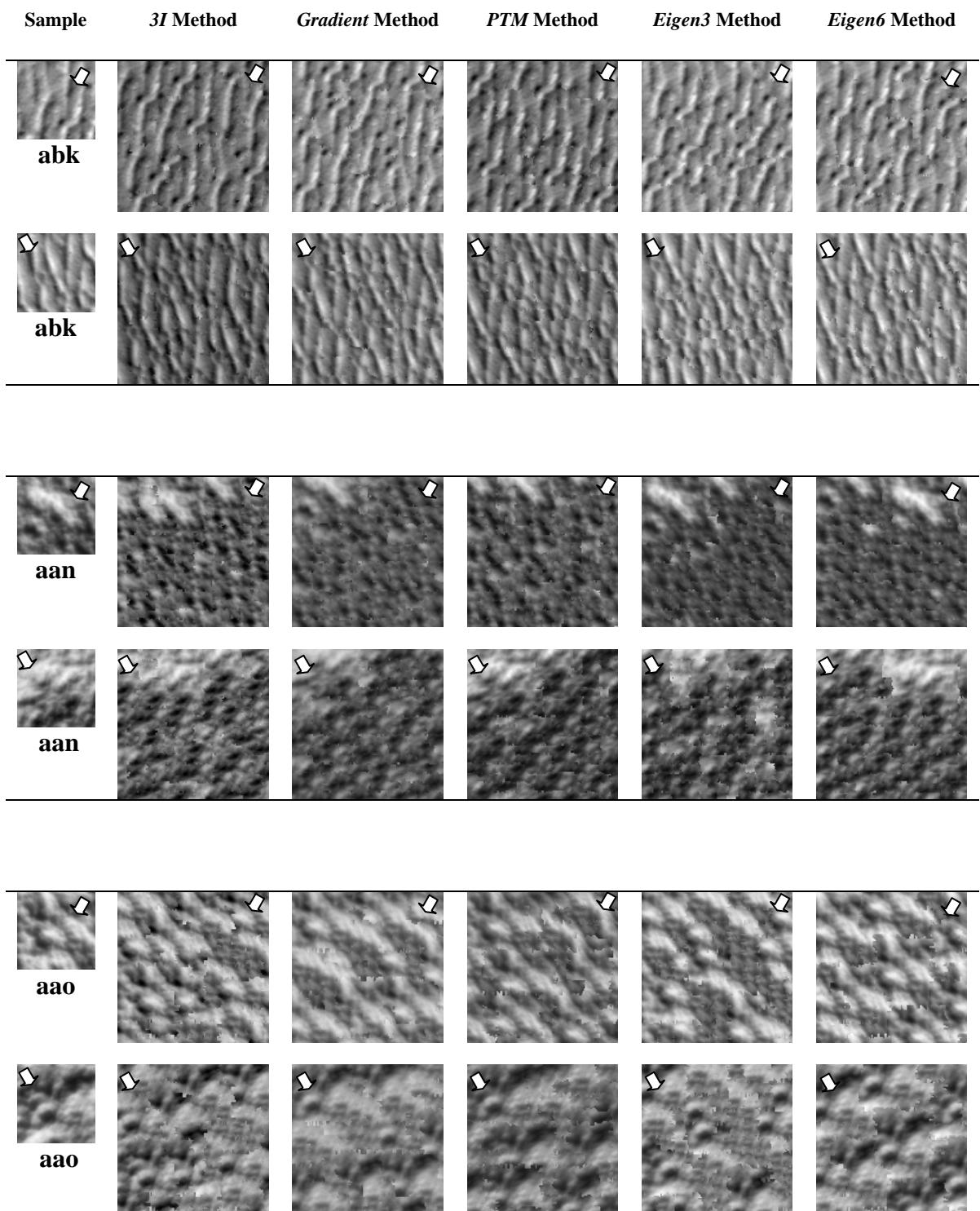


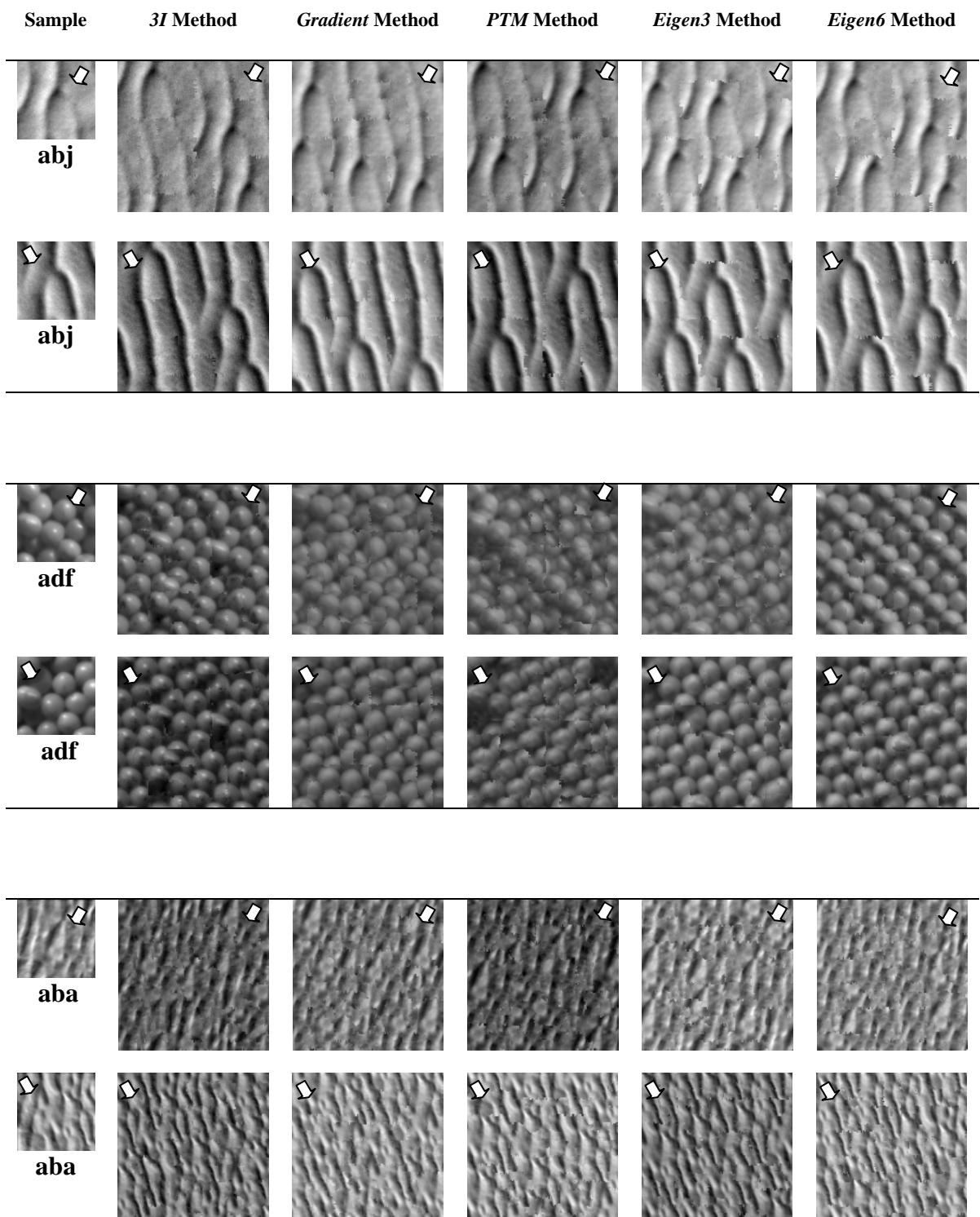


Sample	<i>3I</i> Method	<i>Gradient</i> Method	<i>PTM</i> Method	<i>Eigen3</i> Method	<i>Eigen6</i> Method
--------	------------------	------------------------	-------------------	----------------------	----------------------

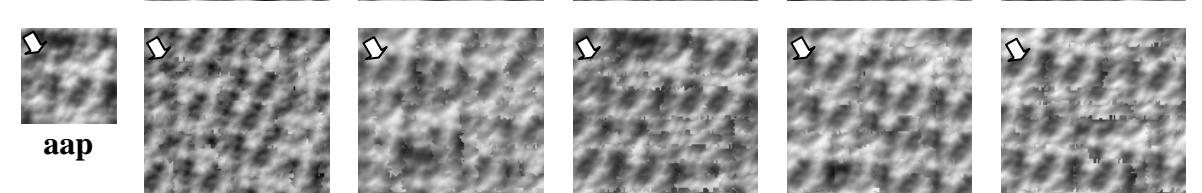
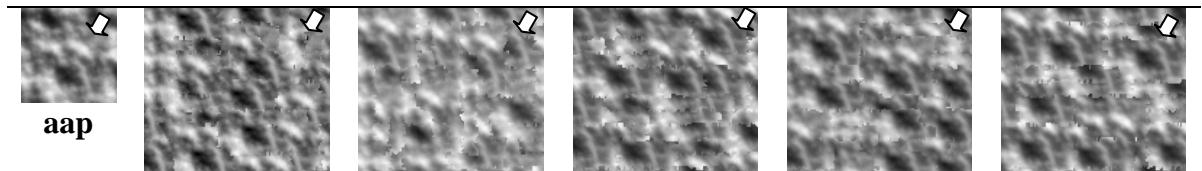
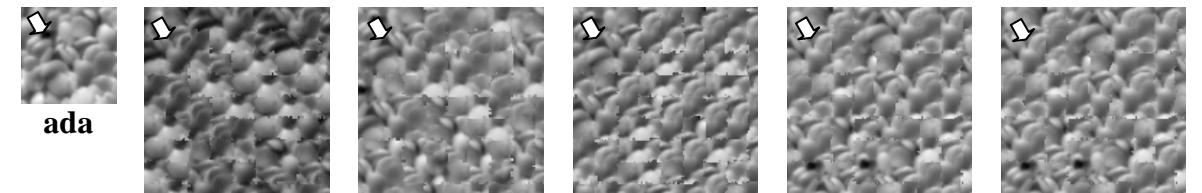
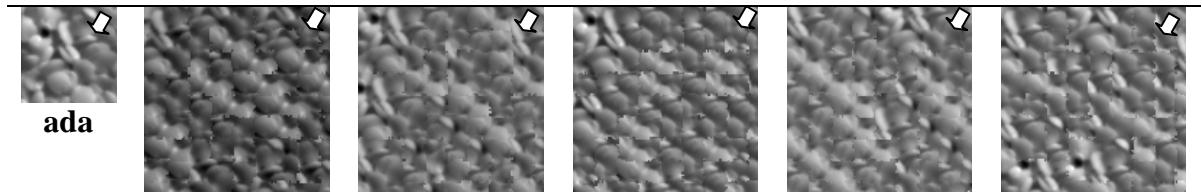








Sample	<i>3I</i> Method	<i>Gradient</i> Method	<i>PTM</i> Method	<i>Eigen3</i> Method	<i>Eigen6</i> Method
--------	------------------	------------------------	-------------------	----------------------	----------------------



Appendix C: Rank data of the five synthesis approaches

Texture Label	Rank of Methods				
	<i>3I</i>	<i>Gradient</i>	<i>PTM</i>	<i>Egen3</i>	<i>Egen6</i>
aaj	1	3	2	5	4
	5	3	4	1	2
	5	2	4	1	3
	4	3	5	1	2
	4	3	5	1	2
	4	2	5	1	3
	1	5	2	4	3
	5	3	4	1	2
	4	1	5	2	3
	5	1	4	3	2
aas	4	3	5	1	2
	5	2	4	1	3
	4	5	3	1	2
	4	5	2	3	1
	5	2	4	3	1
	5	2	4	3	1
	4	1	5	2	3
	4	3	5	1	2
	4	5	2	1	3
	5	4	1	2	3
ace	1	2	5	4	3
	5	4	3	1	2
	5	1	4	2	3
	4	1	3	5	2
	5	3	4	1	2
	4	1	5	3	2
	4	2	1	5	3
	5	1.5	4	3	1.5
	5	2	3	1	4
	5	2	1	4	3
add	5	2	4	3	1
	4	2	5	1	3
	5	1	4	2	3
	4	3	2	1	5
	5	1	4	2	3
	5	2	4	1	3
	4	2	5	3	1
	4	1	5	3	2
	4	1	5	2	3
	4	5	1	2	3

Texture Label	Rank of Methods				
	<i>3I</i>	<i>Gradient</i>	<i>PTM</i>	<i>Eigen3</i>	<i>Eigen6</i>
adc	5	3	4	2	1
	4	1	5	2	3
	5	4	1	2	3
	4	1	5	2	3
	5	2	4	1	3
	5	1	4	2	3
	5	3	4	2	1
	5	3	4	2	1
	3	1	5	2	4
	4	5	2	3	1
aai	5	1	4	2	3
	5	4	1	3	2
	4	1	5	3	2
	5	2	4	1	3
	5	3	4	1	2
	5	3	4	1	2
	5	3	4	2	1
	5	1	4	3	2
	5	3	4	2	1
	4	2	3	1	5
acd	5	1	3	2	4
	5	2	3	1	4
	5	1	3	4	2
	5	3	4	2	1
	5	1	3	2	4
	4	5	3	2	1
	4	3	1	5	2
	5	2	1	4	3
	5	4	1	3	2
	1	4	2	3	5
aar	2	5	1	4	3
	5	2	1	3	4
	4	1	3	2	5
	5	1	4	2	3
	5	4	3	1	2
	3	4	5	2	1
	5	3	4	1	2
	3	2	1	5	4
	1	5	2	3	4
	2	4	1	3	5

Texture Label	Rank of Methods				
	<i>3I</i>	<i>Gradient</i>	<i>PTM</i>	<i>Eigen3</i>	<i>Eigen6</i>
ach	4	1	3	2	5
	5	1	2	4	3
	4	3	1	2	5
	5	1	4	3	2
	5	2	4	3	1
	5	4	3	2	1
	5	3	2	4	1
	5	4	3	2	1
	5	1	4	3	2
	5	1	2	4	3
aci	5	1	2	4	3
	5	3	4	1	2
	5	3	1	2	4
	5	3	4	1	2
	5	3	4	2	1
	5	3	4	2	1
	5	2	4	3	1
	5	2	4	3	1
	4	5	3	1	3
	5	4	1	2	3
abj	2	1	4	5	3
	5	1	4	3	2
	4	1	2	3	5
	5	2	1	4	3
	4	3	5	2	1
	3	4	5	2	1
	2	1	3	5	4
	1	2	3	4	5
	3	5	1	4	3
	3	4	2	5	1

Note: 1-best; 5-worst.

Appendix D: List of publications by the author

Capture and synthesis of 3D surface texture

Junyu Dong and Mike Chantler

Abstract

This paper presents and compares six novel approaches for capturing, synthesising and relighting real 3D surface textures. Unlike 2D texture synthesis these techniques allow the captured textures to be relit using illumination conditions, and viewing angles, that differ from those of original. Our approaches each comprise two stages: synthesis and relighting. Synthesis can be applied either before or after relighting. The relighting stage is implemented in three different ways: using image-based, gradient-based, and height-based approaches. Thus there are a total of six different ways in which we may combine these functions. We present a representative set of results selected from our experiments with 30 textures. The best images are obtained when image-based or gradient-based relighting is used after synthesis.

Published in the Proceeding of the 2nd International Workshop on Texture Analysis & Synthesis. 1 June 2002, Copenhagen, Denmark, pp.41-45.

Capture and synthesis of 3D surface texture

Junyu Dong and Mike Chantler

Abstract

We present and compare five approaches for capturing, synthesising and relighting real 3D surface textures. Unlike 2D texture synthesis techniques they allow the captured textures to be relit using illumination conditions that differ from those of the original. We adapted a texture quilting method due to Efros and combined this with five different relighting representations, comprising: a set of three photometric images; surface gradient and albedo maps; polynomial texture maps; and two eigen based representations using 3 and 6 base images.

We used twelve real textures to perform quantitative tests on the relighting methods in isolation. We developed a qualitative test for the assessment of the complete synthesis systems. Ten observers were asked to rank the images obtained from the five methods using five real textures. Statistical tests were applied to the rankings.

The six-base-image eigen method produced the best quantitative relighting results and in particular was better able to cope with specular surfaces. However, in the qualitative tests there were no significant performance differences detected between it and the other two top performers. Our conclusion is therefore that the cheaper gradient and three-base-image eigen methods should be used in preference, especially where the surfaces are Lambertian or near Lambertian.

Submitted to International Journal of Computer Vision: special issue on texture analysis and synthesis, November, 2002.

Comparison of five 3D surface texture synthesis methods

Junyu Dong and Mike Chantler

Abstract

We present and compare five approaches for synthesizing and relighting real 3D surface textures. We adapted Efros's texture quilting method and combined it with five different relighting representations, comprising: a set of three photometric images; surface gradient and albedo maps; polynomial texture maps; and two eigen based representations using 3 and 6 base images. We used twelve real textures to perform quantitative tests on the relighting methods. We develop a systematic qualitative test for the assessment of the complete synthesis systems. Our conclusion is that the cheaper gradient and three-base-image eigen methods should be used in preference to the other methods, especially where the surfaces are Lambertian or near Lambertian.

Published in the Proceeding of the 3rd International Workshop on Texture Analysis & Synthesis. 17 October 2003, Nice, France.

Estimating Parameters of Illumination models for the synthesis of 3D surface texture

Junyu Dong Andrew Spence Mike Chantler

This paper proposed a method to estimate the parameters of an illumination model and then use these parameters for the synthesis of specular surface textures. We used the relationship between surface gradient maps in frequency domain as a constraint for the separation of diffuse and specular components. During the estimation, we always keep errors between the real images and reconstructed images as small as possible. The estimated parameters form sample surface representation maps, which are then used as inputs for the synthesis of large representation maps. The synthesised representations are finally relit using the illumination model to produce new images under arbitrary illumination directions.

Research memoriam 2003/03, Department of Computer Science, Heriot-Watt University, Edinburgh, UK.

References

- [Ashikhmin2001] Ashikhmin, M. 2001. Synthesizing Natural Textures. In the proceedings of 2001 ACM Symposium on Interactive 3D Graphics, Research Triangle Park, North Carolina March 19-21, pp. 217-226.
- [Ashikhmin2002] Ashikhmin, M. and Shirley, P. 2002. Steerable Illumination Textures. ACM Transactions on Graphics, v.21, no. 1, pp. 1-19.
- [Ashlock1999] Ashlock, D. and Davidson, J. 1999. Texture synthesis with tandem genetic algorithms using nonparametric partially ordered Markov models. Evolutionary Computation, CEC 99. Proceedings of the 1999 Congress on, Volume: 2 , 1999 -1163 Vol. 2.
- [Bader1995] Bader, D. A.; JaJa, J. and Chellapa, R. 1995. Scalable data parallel algorithms for texture synthesis using Gibbs random fields. IEEE Transactions on Image Processing. Vol. 4, No.10. pp.1456-1460.
- [Bar-Joseph2001] Bar-Joseph, Z.; El-Yaniv, R.; Lischinski, D. and Werman, M. 2001. Texture mixing and texture movie synthesis using statistical learning. IEEE Transactions on visualization and computer graphics, Vol. 7, No. 2, April-June 2001.
- [Basri2001] Basri, R. and Jacobs, D. 2001. Lambertian reflectance and linear subspaces. In Proceedings. Eighth IEEE International Conference on Computer Vision. ICCV 2001. Volume: 2 , 2001. Page(s): 383 –390.
- [Belhumeur1997] Bellhumer, P. N.; Hespanha, J. and Kriegman, D. 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Face Recognition, 17(7):711--720.
- [Blinn1978] Blinn, J. F. 1978. Computer Display of Curved Surfaces. Ph.D. Thesis, University of Utah.
- [Bolin1998] Bolin, M. R. and Meyer, G. W. 1999. A perceptually based adaptive sampling algorithm. Proceedings of

- SIGGRAPH 98, 299-309.
- [Burschka2003] Burschka, D.; Cobzas, D.; Dodds, Z.; Hager, G.; Jagersand, M. and Yerex, K. 2003. IEEE Virtual Reality 2003 tutorial 1: Recent Methods for Image-based Modelling and Rendering. March 2003.
- [Campisi2002] Campisi, P. and Scarano, G. 2002. A multiresolution approach for texture synthesis using the circular harmonics functions. IEEE Transactions on Image Processing, Vol. 11, No. 1.
- [Chen1995] Chen, S. E. 1995. QuickTime VR—An image-based approach to virtual environment navigation. In Proceedings of Computer Graphics, Ann. Conf. Eries, SIGGRAPH, August 1995, pp29-38.
- [Cohen2003] Cohen, M. F.; Shade, J.; Hiller, S. and Deussen, O. 2003. Wang Tiles for Image and Texture Generation. SIGGRAPH2003.
- [Coleman1982] Coleman, E. N. and Jain, R. 1982. Obtaining 3-dimensional shape of textured and specular surfaces using a four-source photometry. Computer graphics and image processing, 18:309-328.
- [Cook1982] Cook, R. L. and Torrance, K. E. 1982. A Reflectance Model for Computer Graphics. ACM Transactions on Graphics (TOG). Volume 1, Issue 1. Pages: 7 – 24.
- [Copeland2001] Copeland, A. C.; Ravichandran, G. and Trivedi, M. M. 2001 Texture synthesis using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support, Opt. Eng. 40, pp. 2655.
- [Cross1983] Cross, G. R. and Jain, A. K. 1983. Markov random fields texture models. IEEE, PAMI, 5, pp. 25-39.
- [Dana1999a] Dana, K. J.; Van Ginneken, B.; Nayar, S. K. and Koenderink, J. J. 1999. Reflectance and Texture of Real-World Surfaces. ACM Transactions on Graphics, Vol. 18, No. 1, Pages 1-34.
- [Dana1999b] Dana, K.J. and Nayar, S. K. 1999. 3D Textured Surface

Modelling. Nayar WIAGMOR Workshop CVPR '99

- [Daniel1990] Daniel, W.W. 1990. Applied nonparametric statistics, 2nd ed. PWS-LENT Publishing Company. Boston.
- [De Benet1997] De Benet, J.S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In Computer Graphics Proceedings, SIGGRAPH 97. ACM. pp.361.
- [Debevec2000] Debevec, P.; Hawkins, T.; Tchou, C.; Duiker, H.; Sarokin, W. and Sagar, M. 2000. Acquiring the reflectance field of a human face. Computer Graphics (SIGGRAPH2000 proceedings), July 2000, pp. 145-156.
- [Dong2002a] Dong, J. and Chantler, M. 2002. Capture and synthesis of 3D surface texture. Proceeding of the 2nd International Workshop on Texture Analysis & Synthesis. 1 June 2002, Copenhagen, Denmark, pp.41-45.
- [Dong2002b] Dong, J. and Chantler, M. 2002. Capture and synthesis of 3D surface texture. Submitted to International Journal of Computer Vision: special issue on texture analysis and synthesis. November 2002.
- [Dong2003a] Dong, J. and Chantler, M. 2003. Comparison of five 3D surface texture synthesis methods. Proceeding of the 3rd International Workshop on Texture Analysis & Synthesis. 17 October 2003, Nice, France.
- [Dong2003b] Dong, J.; Spence, A. and Chantler, M. 2003. Estimating Parameters of Illumination models for the synthesis of 3D surface texture. Research memoriam 2003/03, Department of Computer Science, Heriot-Watt University, Edinburgh, UK.
- [Drbohlav2002] Drbohlav, O. and Sára, R. 2002. Specularities Reduce Ambiguity of Uncalibrated Photometric Stereo. Proceedings of 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, pp. 46-62.
- [Efros1999] Efros, A. A. and Leung, T. K. 1999. Texture synthesis

- by non-parametric sampling. In Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE Comput. Soc. Part vol.2, pp.1033-8.
- [Efros2001] Efros, A. A. and Freeman, W. T. 2001. Image Quilting for Texture Synthesis and Transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 341 - 346 Los Angeles, California
- [Eom1998] Eom, K. B. 1998 2-D Moving Average models for texture synthesis and analysis. IEEE Transactions on Image Processing, Vol. 7, No12, pp.1741-1746.
- [Epstein1995] Epstein, R.; Hallinan, P.W.; Yuille, A.L. 1995. 5 ± 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In Proceedings of the Workshop on Physics-Based Modeling in Computer Vision. Page(s): 108.
- [Frankot1988] Frankot, R. T. and Chellappa, R. 1988. A method for enforcing integrability in shape from shading algorithms. IEEE transactions on pattern analysis and machine intelligence, 10(4),:439-451, July 1988.
- [Garber1981] Garber, D. D. 1981. Computational models for texture analysis and texture synthesis. PhD thesis, University of Southern California, Image Processing Institute.
- [Georghiades1999] Georghiades, A. S.; Belhumeur, P. N. and Kriegman D. J. 1999. Illumination-Based Image Synthesis: Creating Novel Images of Human Faces Under Differing Pose and Lighting. In Proc. Workshop on Multi-View Modeling and Analysis of Visual Scenes pp. 47-54.
- [Gortler1996] Gortler, S. J.; Grzeszczuk, R.; Szeliski, R. and Cohen, M. F. 1996. The lumigraph. In Proc. Computer Graphics, Ann. Conf. Series, SIGGRAPH, August 1996. Pp.43-54.
- [Gousseau2002] Gousseau Y. 2002. Texture synthesis through level sets. Proceedings of the 2nd international workshop on texture analysis and synthesis. pp. 53-58, Copenhagen, Denmark.

- [Gullón2002] Gullón, C. 2002. Height recovery of rough surfaces from intensity images. PhD thesis, School of Engineering and Physical Sciences, Heriot-Watt University.
- [Harrison2001] Harrison, P. 2001. A non-hierarchical procedure for re-synthesis of complex textures. In WSCG'2001, pages 190-197, Plzen, Czech Republic.
- [He1991] He, X. D.; Torrance, K.E.; Sillion, F. X. and Greenberg, D. P. 1991. A comprehensive physical model for light reflection. Proceedings of SIGGRAPH 1991, pp. 175-185.
- [Heeger1995] Heeger, D.J. and Bergen, J.R. 1995. Pyramid-based texture analysis/synthesis. In Proceedings International Conference on Image Processing (Cat. No.95CB35819). IEEE Comput. Soc. Press. Part vol.3, pp.648-51.
- [Hertzmann2001] Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B. and Salesin, D. H. 2001. Image analogies. Proceedings of the 28th annual conference on Computer graphics and interactive techniques Pages: 327 – 340.
- [Hochberg1987] Hochberg, Y. and Tamhane, A. C. Multiple Comparison Procedures, 1987, Wiley.
- [Horn1989] Horn, B. and Brooks, M. 1989. Shape from Shading. MIT Press, Cambridge, MA.
- [Huang1984] Edited by Huang, T. S. Image reconstruction from incomplete observations. Published by Jai, 1984.
- [Ikeuchi1991] Ikeuchi, K. and Sato, K. 1991. Determining reflectance properties of an object using range and brightness images Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 13 Issue: 11, Nov. 1991. Page(s): 1139 –1153.
- [Iversen1994] Iversen, H. and Lonnestad, T. 1994. An evaluation of stochastic models for analysis and synthesis of gray-scale texture. Pattern Recognition Letters 15 (1994) pp. 575-585.

- [Jacovitti1998] Jacovitti, G.; Neri, A. and Scarano G. 1998 Texture synthesis-by-analysis with hard-limited Gaussian process. *IEEE Transactions on Image Processing*, Vol 7, No.11, pp. 1615-1621.
- [Julesz1962] Julesz, B. 1962. Visual pattern discrimination. *IRE Transaction on Information Theory*. IT-8:84-92, 1962.
- [Kang1997] Kang S. B. 1997. A survey of image-based rendering techniques. Technical Report. Cambridge Research Laboratory, CRL 97/4. August 1997.
- [Kaplan2000] Kaplan, C. S. and Salesin, D. H. 2000. Escherization. Proceedings of the 27th annual conference on Computer graphics and interactive techniques Pages: Pages: 499 – 510 Year of Publication: 2000 ISBN:1-58113-208-5
- [Kautz2000] Kautz, J. and Seidel H.-P. 2000. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware.
- [Kay1995] Kay, G. and Caelli, T. 1995. Estimating the parameters of an illumination model using photometric stereo. *Graphical models and image processing*, Vol. 57, No. 5, September.
- [Klette1996] Klette, R. and Deguchi, K. 1996. Height data from gradient fields. *Proceedings of machine vision: applications, architectures and system integration V*, SPIE, 2908:204-215.
- [Koenderink1996] Koenderink, J. J.; van Doorn, A. J.; Dana, K. J. and Nayar, S. 1999. Bidirectional reflection distribution function of thoroughly pitted surfaces *International Journal Computer Vision*, vol. 31, pp. 129--144.
- [Kokaram2002] Kokaram, A. 2002. Parametric texture synthesis for filling holes in pictures. In *Proceedings of International Conference on Image Processing*. 2002., Volume:1, Page(s): 325 –328.
- [Koudelka2001] Koudelka, M. L.; Belhumeur, P. N.; Magda, S. and Kriegman, D. J. 2001. Image-based modeling and rendering of surfaces with arbitrary BRDFs. *Computer*

- Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Volume: 1, 8-14. Page(s): I-568 -I-575 vol.1.
- [Kwatra2003] Kwatra, V.; Schodl, A.; Essa, I.; Turk, G. and Bobick, A. 2003. Graphcut textures: image and video synthesis using graph cuts. SIGGRAPH2003.
- [Lafortune1997] Lafortune, E. P. F.; Foo, S.-C.; Torrance, K. E. and Greenberg, D. P. 1997. Non-linear approximation of reflectance functions. International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 24th annual conference on Computer graphics and interactive techniques. Pages: 117 – 126. 1997
- [Leung2001] Leung, T. and Malik, J. 2001. Representing and Recognizing the Visual Appearance of Materials Using Three-dimensional Textons. International journal of Computer Vision, Vol 43, Number 1, pp 7-27.
- [Levoy1996] Levoy, M. and Hanrahan, P. 1996. Light field rendering. In Proc. Computer Graphics, Ann. Conf. Series, SIGGRAPH, August 1996, pp. 31-42.
- [Liang2001] Liang, L.; Liu, C.; Xu, Y.; Guo, B. and Shum, H. 2001. Real-time texture synthesis by patch-based sampling. ACM Transactions on Graphics, Vol. 20, No. 3, July 2001, pp. 127-150.
- [Lin1999] Lin, S. and Lee, S. W. 1999. Estimation of diffuse and specular appearance. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Volume: 2, 20-27 Sept. 1999. Page(s): 855 -860 vol.2.
- [Lin2001] Lin, Z.; Wong, T. T. and Shum, H. Y. 2001 Relighting with the reflected irradiance field: representation, sampling and reconstruction. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Volume: 1 , 2001. Page(s): I-561 -I-567 vol.1
- [Liu2001] Liu, X.; Yu, Y. and Shum, H.Y. 2001. Synthesizing Bidirectional Texture Functions for Real-World Surface.

- In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 97 – 106. Los Angeles, California.
- [Lu1995] Lu, J. and Little, J. 1995. Reflectance function estimation and shape recovery from image sequence of a rotating object. Computer Vision, 1995. Proceedings, Fifth International Conference on, 20-23 June 1995. Page(s): 80 –86.
- [Malzbender2001] Malzbender, T.; Gelb D. and Wolters, H. 2001. Polynomial Texture Maps, In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 519 – 528. Los Angeles, California.
- [Matusik2002] Matusik, W.; Pfister, H.; Ngan, A.; Beardsley, P.; Ziegler, R. and McMillan, L. 2002. Image-based 3D photography using opacity hull. ACM Transactions on Graphics (TOG) , Proceedings of the 29th annual conference on Computer graphics and interactive techniques July 2002, Volume 21, Issue 3. pp.427-437.
- [McAllister2002] McAllister, D. K.; Lastra, A. and Heidrich, W. 2002. *Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions*, Graphics Hardware 2002, Eurographics / SIGGRAPH Workshop Proceedings (Saarbrucken, Germany) September.
- [McGunnigle1998] McGunnigle G. 1998. The classification of textured surfaces under varying illuminant directions. PhD thesis, Department of Computing and Electrical Engineering, Heriot-Watt University.
- [McGunnigle2001] McGunnigle, G. and Chantler, M. 2001. Segmentation of rough surfaces using reflectance. Proceedings of British machine vision conference, pp. 323-332, 10-13 September, Manchester, UK.
- [McMillan1999] McMillan, L. and Gortler S. 1999. Image-based rendering: A new interface between computer vision and computer graphics. ACM SIGGRAPH Computer Graphics, Volume 33 Issue 4. Pages: 61 – 64. ISSN:0097-8930.
- [Nayar1990] Nayar, S. K., Ikeuchi, K. and Kanade, T. 1990. Determining Shape and Reflectance of Hybrid Surfaces

by Photometric Sampling, IEEE Journal of Robotics and Automation, Vol. 6, No. 4, pp. 418-431, August 1990.

- [Nayar1991] Nayar, S. K.; Ikeuchi, K. and Kanade, T. 1991. Surface Reflection: Physical and Geometrical Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 13, No. 7, pp. 611-634.
- [Nayar1996] Nayar, S. K.; Fang, X. and Boult, T. E. 1996. Separation of Reflection Components Using Color and Polarization. International Journal of Computer Vision, 1996.
- [Nealen2003] Nealen, A. and Alexa, M. 2003. Hybrid Texture Synthesis. Eurographics Symposium on Rendering 2003.
- [Nicodemus1977] Nicodemus, F. E.; Richmond, J.C. and Hsai, J. J. 1977. Geometrical considerations and nomenclature for reflectance. U.S. Dept. of Commerce, National Bureau of Standards, October 1977.
- [Nishino2001] Nishino, K.; Sato, Y. and Ikeuchi, K. 2001. Eigen-Texture method: appearance compression and synthesis based on a 3D model. Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 23 Issue: 11, Page(s): 1257 –1265.
- [Oren1994] Oren, M. and Nayar, S. K. 1994. Generalization of Lambert's Reflectance Model, Proceedings of ACM SIGGRAPH 94, Orlando, July 1994.
- [Paget1998] Paget, R. and Longstaff I. D. 1998. Texture synthesis via a noncausal nonparametric multiscale Markov random field. IEEE Transactions on Image Processing, Vol. 7, No.6, June 1998.
- [Parada2001] Parada, P. and Ruiz-del-Solar, J. 2001. Texture synthesis using image pyramids and self-organizing maps. Image Analysis and Processing, 2001. Proceedings. 11th International Conference on, 26-28 Sept. 2001. Page(s): 244 –249.

- [Popat1993] Popat, k. and Picard, R. W. 1993. Novel cluster-based probability model for texture synthesis, classification, and compression. In Proc. SPIE visual communications and Image Processing'93, Boston, Nov 3-11.
- [Portilla2000] Portilla, J. and Simoncelli, E.P. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision, volume 40, issue 1, pages 49-71, December.
- [Press1998] Press, W. H.; Flannery, B. P.; Teukolsky, S. A. and Vetterling, W. T. 1998 Numerical Recipes in C. Cambridge University Press.
- [Ramamoorthi2001] Ramamoorthi, R. and Hanrahan, P. 2001. A signal-processing framework for inverse rendering International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 117 – 128, Year of Publication: 2001.ISBN:1-58113-374-X
- [Ramamoorthi2002] Ramamoorthi, R. 2002. Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 24 Issue: 10, Page(s): 1322 –1333.
- [Ramasubramanian1999] Ramasubramanian, M.; Pattanaik, S. N.; and Greenberg, D. P. Perceptually based physical error metric for realistic image synthesis. Proceedings of SIGGRAPH 99, 73-82.
- [Robb2003] Robb, M.; Spence, A. D.; Chantler, M. J. and Timmins, M. Real-Time Per-Pixel Rendering of Bump-mapped Textures Captured using Photometric Stereo. Vision, Video, and Graphics 2003, Bath, UK. 10-11 July 2003.
- [Rushmeier1997] Rushmeier, H.; Taubin, G. and Gueziec, A. 1997. Applying Shape from Lighting Variation to Bump Map Capture. In Proceedings of the Eighth Eurographics Rendering Workshop, Saint-Etienne, France, pp. 35-44.
- [Rushmeier1998] Rushmeier, H. E. 1998. Global illumination input. Course notes 5-A basic guide to global illumination. 25th

International Conference on Computer Graphics and Interactive Techniques. SIGGRAPH98.

- [Saito1996] Saito, H.; Omata, K. and Ozawa, S. 1999. Recovery of Shape and Surface Reflectance of Specular Object from Rotation of Light Source, Second International Conference on 3-D Digital Imaging and Modeling (3DIM99), Ottawa, Oct 4-8, 1999, pp.526-535
- [Sato1997] Sato, Y.; Wheeler, M.D. and Ikeuchi, K. 1997. Object shape and reflectance modeling from observation. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques , Pages: 379 – 387
- [Sebe2000] Sebe, N.; Lew, M. S. and Huijsmans, D. P. 2000. Toward improved ranking metrics. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 22 Issue: 10, Oct. 2000. Page(s): 1132 –1143.
- [Shashua1992] Shashua, A. 1992. Geometry and Photometry in 3D visual Recognition. *PhD thesis, MIT*.
- [Spence2003] Spence, A. D. and Chantler, M. J. 2003. Optimal illumination for three-image photometric stereo acquisition of surface texture. The 3rd International workshop on texture analysis and synthesis. October, 2003, Nice, France.
- [Stavridi1997] Stavridi, M.; van Ginneken, B. and Koenderink, J.J. 1997. Surface bidirectional reflection distribution function and the texture of bricks and tiles. Applied Optics, Vol. 36, No. 16, June.
- [Tong2002] Tong, X.; Zhang, J.; Liu L.; Wang X.; Guo B. and Shum, H.-Y. 2002. Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Transactions on Graphics (TOG). Proc. of the 29th annual conference on Computer graphics and interactive techniques, Volume 21 Issue 3. Pages: 665 – 672.
- [Tonietto2002] Tonietto, L. and Walter, M. 2002. Towards local control for image-based texture synthesis. Computer Graphics and Image Processing, Proceedings. XV Brazilian Symposium on, 7-10 Oct. Page(s): 252 –258.

- [Torrance1967] Torrance, K. E. and Sparrow, E. M. 1967. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.* 57, (Sept. 1967), pp.1105-1114.
- [Turk1991] Turk M. and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71--86.
- [Van Nevel1998] Van Nevel, A. 1998. Texture Synthesis via Matching First and Second Order Statistics of a Wavelet Frame Decomposition. In Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP-98) , Chicago, Illinois. ICIP (1) pp. 72-76.
- [Wei2000] Wei, L. and Levoy, M. Fast texture synthesis using tree-structured vector quantization. In Computer Graphics Proceedings. Annual Conference Series 2000. SIGGRAPH 2000. Conference Proceedings. ACM. pp.479-88.
- [Wong2002] Wong, T.; Fu, C.; Heng P. and Leung, C. 2002. The Plenoptic Illumination Function, *IEEE Transactions on Multimedia*, Vol. 4, No. 3, September 2002, pp. 361-371.
- [Woo1999] Woo, M.; Neider, J.; Davis, T. and OpenGL Architecture Review Board. 1999. *The OpenGL Programming Guide* 3rd Edition. The Official Guide to Learning OpenGL Version 1.2. Addison-Wesley Pub Co; 3rd edition (August 6, 1999).
- [Woodham1981] Woodham, R. 1981. Analysing images of curved surfaces. *Artificial Intelligence*, 17:117-140.
- [Xu2001] Xu, Y.; Zhu, S. C.; Guo, B. and Shum, H.Y. 2001. Asymptotically Admissible Texture Synthesis, In Proc. Second International Workshop of Statistical and Computational Theories of Vision, Vancouver, Canada.
- [Zalesny2000] Zalesny, A. and Van Gool, L. 2000. A compact model for viewpoint dependent texture synthesis. SMILE 2000. Workshop on 3D Structure from Images, Lecture Notes in Computer Science 2018, M Pollefeys *et. al.* (Eds.), pp.123-143.

- [Zalesny2001] Zalesny, A. and Van Gool, L. 2001. Multiview Texture Models. In Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition, Volume: 1, Page(s): 615 –1180.
- [Zelinka2002] Zelink, S. and Garland, M. 2002. Towards real-time texture synthesis with the jump map. Proceedings of the 13th Eurographics workshop on Rendering, Pisa, Italy, Pages: 99 – 104. ISBN:1-58113-534-3.
- [Zhang1998a] Zhang, Z. 1998. Modeling Geometric Structure and Illumination Variation of a Scene from Real Images”, In Proc. International Conference on Computer Vision (ICCV’98), Bombay, India. Page(s): 1041 –1046.
- [Zhang1998b] Zhang, J.; Wang, D. and Tran Q. N. 1998. A wavelet-based multiresolution statistical model for texture. IEEE Transactions on Image Processing, Vol, 7, No. 11, November.
- [Zhang2003] Zhang, J.; Zhou, K.; Velho, L.; Guo, B. and Shum H. Y. 2003. Synthesis of Progressively Variant Textures on Arbitrary Surfaces. SIGGRAPH2003.
- [Zhu1998] Zhu, S. C.; Wu, Y. and Mumford, D. 1998. Filters, Random fields And Maximum Entropy (FRAME). International Journal of Computer Vision 27(2) 1-20, March/April.
- [Zhu2000] Zhu, S.C.; Liu, X.W. and Wu, Y.N. June 2000. Exploring texture ensembles by efficient Markov chain Monte Carlo-Toward a "trichromacy" theory of texture. IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.22, no.6, pp.554-69.

References

- [Ashikhmin2001] Ashikhmin, M. 2001. Synthesizing Natural Textures. In the proceedings of 2001 ACM Symposium on Interactive 3D Graphics, Research Triangle Park, North Carolina March 19-21, pp. 217-226.
- [Ashikhmin2002] Ashikhmin, M. and Shirley, P. 2002. Steerable Illumination Textures. ACM Transactions on Graphics, v.21, no. 1, pp. 1-19.
- [Ashlock1999] Ashlock, D. and Davidson, J. 1999. Texture synthesis with tandem genetic algorithms using nonparametric partially ordered Markov models. Evolutionary Computation, CEC 99. Proceedings of the 1999 Congress on, Volume: 2 , 1999 -1163 Vol. 2.
- [Bader1995] Bader, D. A.; JaJa, J. and Chellapa, R. 1995. Scalable data parallel algorithms for texture synthesis using Gibbs random fields. IEEE Transactions on Image Processing. Vol. 4, No.10. pp.1456-1460.
- [Bar-Joseph2001] Bar-Joseph, Z.; El-Yaniv, R.; Lischinski, D. and Werman, M. 2001. Texture mixing and texture movie synthesis using statistical learning. IEEE Transactions on visualization and computer graphics, Vol. 7, No. 2, April-June 2001.
- [Basri2001] Basri, R. and Jacobs, D. 2001. Lambertian reflectance and linear subspaces. In Proceedings. Eighth IEEE International Conference on Computer Vision. ICCV 2001. Volume: 2 , 2001. Page(s): 383 –390.
- [Belhumeur1997] Bellhumer, P. N.; Hespanha, J. and Kriegman, D. 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Face Recognition, 17(7):711--720.
- [Blinn1978] Blinn, J. F. 1978. Computer Display of Curved Surfaces. Ph.D. Thesis, University of Utah.
- [Bolin1998] Bolin, M. R. and Meyer, G. W. 1999. A perceptually based adaptive sampling algorithm. Proceedings of

- SIGGRAPH 98, 299-309.
- [Burschka2003] Burschka, D.; Cobzas, D.; Dodds, Z.; Hager, G.; Jagersand, M. and Yerex, K. 2003. IEEE Virtual Reality 2003 tutorial 1: Recent Methods for Image-based Modelling and Rendering. March 2003.
- [Campisi2002] Campisi, P. and Scarano, G. 2002. A multiresolution approach for texture synthesis using the circular harmonics functions. IEEE Transactions on Image Processing, Vol. 11, No. 1.
- [Chen1995] Chen, S. E. 1995. QuickTime VR—An image-based approach to virtual environment navigation. In Proceedings of Computer Graphics, Ann. Conf. Eries, SIGGRAPH, August 1995, pp29-38.
- [Cohen2003] Cohen, M. F.; Shade, J.; Hiller, S. and Deussen, O. 2003. Wang Tiles for Image and Texture Generation. SIGGRAPH2003.
- [Coleman1982] Coleman, E. N. and Jain, R. 1982. Obtaining 3-dimensional shape of textured and specular surfaces using a four-source photometry. Computer graphics and image processing, 18:309-328.
- [Cook1982] Cook, R. L. and Torrance, K. E. 1982. A Reflectance Model for Computer Graphics. ACM Transactions on Graphics (TOG). Volume 1, Issue 1. Pages: 7 – 24.
- [Copeland2001] Copeland, A. C.; Ravichandran, G. and Trivedi, M. M. 2001 Texture synthesis using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support, Opt. Eng. 40, pp. 2655.
- [Cross1983] Cross, G. R. and Jain, A. K. 1983. Markov random fields texture models. IEEE, PAMI, 5, pp. 25-39.
- [Dana1999a] Dana, K. J.; Van Ginneken, B.; Nayar, S. K. and Koenderink, J. J. 1999. Reflectance and Texture of Real-World Surfaces. ACM Transactions on Graphics, Vol. 18, No. 1, Pages 1-34.
- [Dana1999b] Dana, K.J. and Nayar, S. K. 1999. 3D Textured Surface

Modelling. Nayar WIAGMOR Workshop CVPR '99

- [Daniel1990] Daniel, W.W. 1990. Applied nonparametric statistics, 2nd ed. PWS-LENT Publishing Company. Boston.
- [De Benet1997] De Benet, J.S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In Computer Graphics Proceedings, SIGGRAPH 97. ACM. pp.361.
- [Debevec2000] Debevec, P.; Hawkins, T.; Tchou, C.; Duiker, H.; Sarokin, W. and Sagar, M. 2000. Acquiring the reflectance field of a human face. Computer Graphics (SIGGRAPH2000 proceedings), July 2000, pp. 145-156.
- [Dong2002a] Dong, J. and Chantler, M. 2002. Capture and synthesis of 3D surface texture. Proceeding of the 2nd International Workshop on Texture Analysis & Synthesis. 1 June 2002, Copenhagen, Denmark, pp.41-45.
- [Dong2002b] Dong, J. and Chantler, M. 2002. Capture and synthesis of 3D surface texture. Submitted to International Journal of Computer Vision: special issue on texture analysis and synthesis. November 2002.
- [Dong2003a] Dong, J. and Chantler, M. 2003. Comparison of five 3D surface texture synthesis methods. Proceeding of the 3rd International Workshop on Texture Analysis & Synthesis. 17 October 2003, Nice, France.
- [Dong2003b] Dong, J.; Spence, A. and Chantler, M. 2003. Estimating Parameters of Illumination models for the synthesis of 3D surface texture. Research memoriam 2003/03, Department of Computer Science, Heriot-Watt University, Edinburgh, UK.
- [Drbohlav2002] Drbohlav, O. and Sára, R. 2002. Specularities Reduce Ambiguity of Uncalibrated Photometric Stereo. Proceedings of 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, pp. 46-62.
- [Efros1999] Efros, A. A. and Leung, T. K. 1999. Texture synthesis

- by non-parametric sampling. In Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE Comput. Soc. Part vol.2, pp.1033-8.
- [Efros2001] Efros, A. A. and Freeman, W. T. 2001. Image Quilting for Texture Synthesis and Transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 341 - 346 Los Angeles, California
- [Eom1998] Eom, K. B. 1998 2-D Moving Average models for texture synthesis and analysis. IEEE Transactions on Image Processing, Vol. 7, No12, pp.1741-1746.
- [Epstein1995] Epstein, R.; Hallinan, P.W.; Yuille, A.L. 1995. 5 ± 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In Proceedings of the Workshop on Physics-Based Modeling in Computer Vision. Page(s): 108.
- [Frankot1988] Frankot, R. T. and Chellappa, R. 1988. A method for enforcing integrability in shape from shading algorithms. IEEE transactions on pattern analysis and machine intelligence, 10(4),:439-451, July 1988.
- [Garber1981] Garber, D. D. 1981. Computational models for texture analysis and texture synthesis. PhD thesis, University of Southern California, Image Processing Institute.
- [Georghiades1999] Georghiades, A. S.; Belhumeur, P. N. and Kriegman D. J. 1999. Illumination-Based Image Synthesis: Creating Novel Images of Human Faces Under Differing Pose and Lighting. In Proc. Workshop on Multi-View Modeling and Analysis of Visual Scenes pp. 47-54.
- [Gortler1996] Gortler, S. J.; Grzeszczuk, R.; Szeliski, R. and Cohen, M. F. 1996. The lumigraph. In Proc. Computer Graphics, Ann. Conf. Series, SIGGRAPH, August 1996. Pp.43-54.
- [Gousseau2002] Gousseau Y. 2002. Texture synthesis through level sets. Proceedings of the 2nd international workshop on texture analysis and synthesis. pp. 53-58, Copenhagen, Denmark.

- [Gullón2002] Gullón, C. 2002. Height recovery of rough surfaces from intensity images. PhD thesis, School of Engineering and Physical Sciences, Heriot-Watt University.
- [Harrison2001] Harrison, P. 2001. A non-hierarchical procedure for re-synthesis of complex textures. In WSCG'2001, pages 190-197, Plzen, Czech Republic.
- [He1991] He, X. D.; Torrance, K.E.; Sillion, F. X. and Greenberg, D. P. 1991. A comprehensive physical model for light reflection. Proceedings of SIGGRAPH 1991, pp. 175-185.
- [Heeger1995] Heeger, D.J. and Bergen, J.R. 1995. Pyramid-based texture analysis/synthesis. In Proceedings International Conference on Image Processing (Cat. No.95CB35819). IEEE Comput. Soc. Press. Part vol.3, pp.648-51.
- [Hertzmann2001] Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B. and Salesin, D. H. 2001. Image analogies. Proceedings of the 28th annual conference on Computer graphics and interactive techniques Pages: 327 – 340.
- [Hochberg1987] Hochberg, Y. and Tamhane, A. C. Multiple Comparison Procedures, 1987, Wiley.
- [Horn1989] Horn, B. and Brooks, M. 1989. Shape from Shading. MIT Press, Cambridge, MA.
- [Huang1984] Edited by Huang, T. S. Image reconstruction from incomplete observations. Published by Jai, 1984.
- [Ikeuchi1991] Ikeuchi, K. and Sato, K. 1991. Determining reflectance properties of an object using range and brightness images Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 13 Issue: 11, Nov. 1991. Page(s): 1139 –1153.
- [Iversen1994] Iversen, H. and Lonnestad, T. 1994. An evaluation of stochastic models for analysis and synthesis of gray-scale texture. Pattern Recognition Letters 15 (1994) pp. 575-585.

- [Jacovitti1998] Jacovitti, G.; Neri, A. and Scarano G. 1998 Texture synthesis-by-analysis with hard-limited Gaussian process. *IEEE Transactions on Image Processing*, Vol 7, No.11, pp. 1615-1621.
- [Julesz1962] Julesz, B. 1962. Visual pattern discrimination. *IRE Transaction on Information Theory*. IT-8:84-92, 1962.
- [Kang1997] Kang S. B. 1997. A survey of image-based rendering techniques. Technical Report. Cambridge Research Laboratory, CRL 97/4. August 1997.
- [Kaplan2000] Kaplan, C. S. and Salesin, D. H. 2000. Escherization. Proceedings of the 27th annual conference on Computer graphics and interactive techniques Pages: Pages: 499 – 510 Year of Publication: 2000 ISBN:1-58113-208-5
- [Kautz2000] Kautz, J. and Seidel H.-P. 2000. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware.
- [Kay1995] Kay, G. and Caelli, T. 1995. Estimating the parameters of an illumination model using photometric stereo. *Graphical models and image processing*, Vol. 57, No. 5, September.
- [Klette1996] Klette, R. and Deguchi, K. 1996. Height data from gradient fields. *Proceedings of machine vision: applications, architectures and system integration V*, SPIE, 2908:204-215.
- [Koenderink1996] Koenderink, J. J.; van Doorn, A. J.; Dana, K. J. and Nayar, S. 1999. Bidirectional reflection distribution function of thoroughly pitted surfaces *International Journal Computer Vision*, vol. 31, pp. 129--144.
- [Kokaram2002] Kokaram, A. 2002. Parametric texture synthesis for filling holes in pictures. In *Proceedings of International Conference on Image Processing*. 2002., Volume:1, Page(s): 325 –328.
- [Koudelka2001] Koudelka, M. L.; Belhumeur, P. N.; Magda, S. and Kriegman, D. J. 2001. Image-based modeling and rendering of surfaces with arbitrary BRDFs. *Computer*

- Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Volume: 1, 8-14. Page(s): I-568 -I-575 vol.1.
- [Kwatra2003] Kwatra, V.; Schodl, A.; Essa, I.; Turk, G. and Bobick, A. 2003. Graphcut textures: image and video synthesis using graph cuts. SIGGRAPH2003.
- [Lafortune1997] Lafortune, E. P. F.; Foo, S.-C.; Torrance, K. E. and Greenberg, D. P. 1997. Non-linear approximation of reflectance functions. International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 24th annual conference on Computer graphics and interactive techniques. Pages: 117 – 126. 1997
- [Leung2001] Leung, T. and Malik, J. 2001. Representing and Recognizing the Visual Appearance of Materials Using Three-dimensional Textons. International journal of Computer Vision, Vol 43, Number 1, pp 7-27.
- [Levoy1996] Levoy, M. and Hanrahan, P. 1996. Light field rendering. In Proc. Computer Graphics, Ann. Conf. Series, SIGGRAPH, August 1996, pp. 31-42.
- [Liang2001] Liang, L.; Liu, C.; Xu, Y.; Guo, B. and Shum, H. 2001. Real-time texture synthesis by patch-based sampling. ACM Transactions on Graphics, Vol. 20, No. 3, July 2001, pp. 127-150.
- [Lin1999] Lin, S. and Lee, S. W. 1999. Estimation of diffuse and specular appearance. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Volume: 2, 20-27 Sept. 1999. Page(s): 855 -860 vol.2.
- [Lin2001] Lin, Z.; Wong, T. T. and Shum, H. Y. 2001 Relighting with the reflected irradiance field: representation, sampling and reconstruction. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Volume: 1 , 2001. Page(s): I-561 -I-567 vol.1
- [Liu2001] Liu, X.; Yu, Y. and Shum, H.Y. 2001. Synthesizing Bidirectional Texture Functions for Real-World Surface.

- In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 97 – 106. Los Angeles, California.
- [Lu1995] Lu, J. and Little, J. 1995. Reflectance function estimation and shape recovery from image sequence of a rotating object. Computer Vision, 1995. Proceedings, Fifth International Conference on, 20-23 June 1995. Page(s): 80 –86.
- [Malzbender2001] Malzbender, T.; Gelb D. and Wolters, H. 2001. Polynomial Texture Maps, In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 519 – 528. Los Angeles, California.
- [Matusik2002] Matusik, W.; Pfister, H.; Ngan, A.; Beardsley, P.; Ziegler, R. and McMillan, L. 2002. Image-based 3D photography using opacity hull. ACM Transactions on Graphics (TOG) , Proceedings of the 29th annual conference on Computer graphics and interactive techniques July 2002, Volume 21, Issue 3. pp.427-437.
- [McAllister2002] McAllister, D. K.; Lastra, A. and Heidrich, W. 2002. *Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions*, Graphics Hardware 2002, Eurographics / SIGGRAPH Workshop Proceedings (Saarbrucken, Germany) September.
- [McGunnigle1998] McGunnigle G. 1998. The classification of textured surfaces under varying illuminant directions. PhD thesis, Department of Computing and Electrical Engineering, Heriot-Watt University.
- [McGunnigle2001] McGunnigle, G. and Chantler, M. 2001. Segmentation of rough surfaces using reflectance. Proceedings of British machine vision conference, pp. 323-332, 10-13 September, Manchester, UK.
- [McMillan1999] McMillan, L. and Gortler S. 1999. Image-based rendering: A new interface between computer vision and computer graphics. ACM SIGGRAPH Computer Graphics, Volume 33 Issue 4. Pages: 61 – 64. ISSN:0097-8930.
- [Nayar1990] Nayar, S. K., Ikeuchi, K. and Kanade, T. 1990. Determining Shape and Reflectance of Hybrid Surfaces

by Photometric Sampling, IEEE Journal of Robotics and Automation, Vol. 6, No. 4, pp. 418-431, August 1990.

- [Nayar1991] Nayar, S. K.; Ikeuchi, K. and Kanade, T. 1991. Surface Reflection: Physical and Geometrical Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 13, No. 7, pp. 611-634.
- [Nayar1996] Nayar, S. K.; Fang, X. and Boult, T. E. 1996. Separation of Reflection Components Using Color and Polarization. International Journal of Computer Vision, 1996.
- [Nealen2003] Nealen, A. and Alexa, M. 2003. Hybrid Texture Synthesis. Eurographics Symposium on Rendering 2003.
- [Nicodemus1977] Nicodemus, F. E.; Richmond, J.C. and Hsai, J. J. 1977. Geometrical considerations and nomenclature for reflectance. U.S. Dept. of Commerce, National Bureau of Standards, October 1977.
- [Nishino2001] Nishino, K.; Sato, Y. and Ikeuchi, K. 2001. Eigen-Texture method: appearance compression and synthesis based on a 3D model. Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 23 Issue: 11, Page(s): 1257 –1265.
- [Oren1994] Oren, M. and Nayar, S. K. 1994. Generalization of Lambert's Reflectance Model, Proceedings of ACM SIGGRAPH 94, Orlando, July 1994.
- [Paget1998] Paget, R. and Longstaff I. D. 1998. Texture synthesis via a noncausal nonparametric multiscale Markov random field. IEEE Transactions on Image Processing, Vol. 7, No.6, June 1998.
- [Parada2001] Parada, P. and Ruiz-del-Solar, J. 2001. Texture synthesis using image pyramids and self-organizing maps. Image Analysis and Processing, 2001. Proceedings. 11th International Conference on, 26-28 Sept. 2001. Page(s): 244 –249.

- [Popat1993] Popat, k. and Picard, R. W. 1993. Novel cluster-based probability model for texture synthesis, classification, and compression. In Proc. SPIE visual communications and Image Processing'93, Boston, Nov 3-11.
- [Portilla2000] Portilla, J. and Simoncelli, E.P. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision, volume 40, issue 1, pages 49-71, December.
- [Press1998] Press, W. H.; Flannery, B. P.; Teukolsky, S. A. and Vetterling, W. T. 1998 Numerical Recipes in C. Cambridge University Press.
- [Ramamoorthi2001] Ramamoorthi, R. and Hanrahan, P. 2001. A signal-processing framework for inverse rendering International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Pages: 117 – 128, Year of Publication: 2001.ISBN:1-58113-374-X
- [Ramamoorthi2002] Ramamoorthi, R. 2002. Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 24 Issue: 10, Page(s): 1322 –1333.
- [Ramasubramanian1999] Ramasubramanian, M.; Pattanaik, S. N.; and Greenberg, D. P. Perceptually based physical error metric for realistic image synthesis. Proceedings of SIGGRAPH 99, 73-82.
- [Robb2003] Robb, M.; Spence, A. D.; Chantler, M. J. and Timmins, M. Real-Time Per-Pixel Rendering of Bump-mapped Textures Captured using Photometric Stereo. Vision, Video, and Graphics 2003, Bath, UK. 10-11 July 2003.
- [Rushmeier1997] Rushmeier, H.; Taubin, G. and Gueziec, A. 1997. Applying Shape from Lighting Variation to Bump Map Capture. In Proceedings of the Eighth Eurographics Rendering Workshop, Saint-Etienne, France, pp. 35-44.
- [Rushmeier1998] Rushmeier, H. E. 1998. Global illumination input. Course notes 5-A basic guide to global illumination. 25th

International Conference on Computer Graphics and Interactive Techniques. SIGGRAPH98.

- [Saito1996] Saito, H.; Omata, K. and Ozawa, S. 1999. Recovery of Shape and Surface Reflectance of Specular Object from Rotation of Light Source, Second International Conference on 3-D Digital Imaging and Modeling (3DIM99), Ottawa, Oct 4-8, 1999, pp.526-535
- [Sato1997] Sato, Y.; Wheeler, M.D. and Ikeuchi, K. 1997. Object shape and reflectance modeling from observation. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques , Pages: 379 – 387
- [Sebe2000] Sebe, N.; Lew, M. S. and Huijsmans, D. P. 2000. Toward improved ranking metrics. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 22 Issue: 10, Oct. 2000. Page(s): 1132 –1143.
- [Shashua1992] Shashua, A. 1992. Geometry and Photometry in 3D visual Recognition. *PhD thesis, MIT*.
- [Spence2003] Spence, A. D. and Chantler, M. J. 2003. Optimal illumination for three-image photometric stereo acquisition of surface texture. The 3rd International workshop on texture analysis and synthesis. October, 2003, Nice, France.
- [Stavridi1997] Stavridi, M.; van Ginneken, B. and Koenderink, J.J. 1997. Surface bidirectional reflection distribution function and the texture of bricks and tiles. Applied Optics, Vol. 36, No. 16, June.
- [Tong2002] Tong, X.; Zhang, J.; Liu L.; Wang X.; Guo B. and Shum, H.-Y. 2002. Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Transactions on Graphics (TOG). Proc. of the 29th annual conference on Computer graphics and interactive techniques, Volume 21 Issue 3. Pages: 665 – 672.
- [Tonietto2002] Tonietto, L. and Walter, M. 2002. Towards local control for image-based texture synthesis. Computer Graphics and Image Processing, Proceedings. XV Brazilian Symposium on, 7-10 Oct. Page(s): 252 –258.

- [Torrance1967] Torrance, K. E. and Sparrow, E. M. 1967. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.* 57, (Sept. 1967), pp.1105-1114.
- [Turk1991] Turk M. and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71--86.
- [Van Nevel1998] Van Nevel, A. 1998. Texture Synthesis via Matching First and Second Order Statistics of a Wavelet Frame Decomposition. In Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP-98) , Chicago, Illinois. ICIP (1) pp. 72-76.
- [Wei2000] Wei, L. and Levoy, M. Fast texture synthesis using tree-structured vector quantization. In Computer Graphics Proceedings. Annual Conference Series 2000. SIGGRAPH 2000. Conference Proceedings. ACM. pp.479-88.
- [Wong2002] Wong, T.; Fu, C.; Heng P. and Leung, C. 2002. The Plenoptic Illumination Function, *IEEE Transactions on Multimedia*, Vol. 4, No. 3, September 2002, pp. 361-371.
- [Woo1999] Woo, M.; Neider, J.; Davis, T. and OpenGL Architecture Review Board. 1999. *The OpenGL Programming Guide* 3rd Edition. The Official Guide to Learning OpenGL Version 1.2. Addison-Wesley Pub Co; 3rd edition (August 6, 1999).
- [Woodham1981] Woodham, R. 1981. Analysing images of curved surfaces. *Artificial Intelligence*, 17:117-140.
- [Xu2001] Xu, Y.; Zhu, S. C.; Guo, B. and Shum, H.Y. 2001. Asymptotically Admissible Texture Synthesis, In Proc. Second International Workshop of Statistical and Computational Theories of Vision, Vancouver, Canada.
- [Zalesny2000] Zalesny, A. and Van Gool, L. 2000. A compact model for viewpoint dependent texture synthesis. SMILE 2000. Workshop on 3D Structure from Images, Lecture Notes in Computer Science 2018, M Pollefeys *et. al.* (Eds.), pp.123-143.

- [Zalesny2001] Zalesny, A. and Van Gool, L. 2001. Multiview Texture Models. In Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition, Volume: 1, Page(s): 615 –1180.
- [Zelinka2002] Zelink, S. and Garland, M. 2002. Towards real-time texture synthesis with the jump map. Proceedings of the 13th Eurographics workshop on Rendering, Pisa, Italy, Pages: 99 – 104. ISBN:1-58113-534-3.
- [Zhang1998a] Zhang, Z. 1998. Modeling Geometric Structure and Illumination Variation of a Scene from Real Images”, In Proc. International Conference on Computer Vision (ICCV’98), Bombay, India. Page(s): 1041 –1046.
- [Zhang1998b] Zhang, J.; Wang, D. and Tran Q. N. 1998. A wavelet-based multiresolution statistical model for texture. IEEE Transactions on Image Processing, Vol, 7, No. 11, November.
- [Zhang2003] Zhang, J.; Zhou, K.; Velho, L.; Guo, B. and Shum H. Y. 2003. Synthesis of Progressively Variant Textures on Arbitrary Surfaces. SIGGRAPH2003.
- [Zhu1998] Zhu, S. C.; Wu, Y. and Mumford, D. 1998. Filters, Random fields And Maximum Entropy (FRAME). International Journal of Computer Vision 27(2) 1-20, March/April.
- [Zhu2000] Zhu, S.C.; Liu, X.W. and Wu, Y.N. June 2000. Exploring texture ensembles by efficient Markov chain Monte Carlo-Toward a "trichromacy" theory of texture. IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.22, no.6, pp.554-69.