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 Fredman'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.

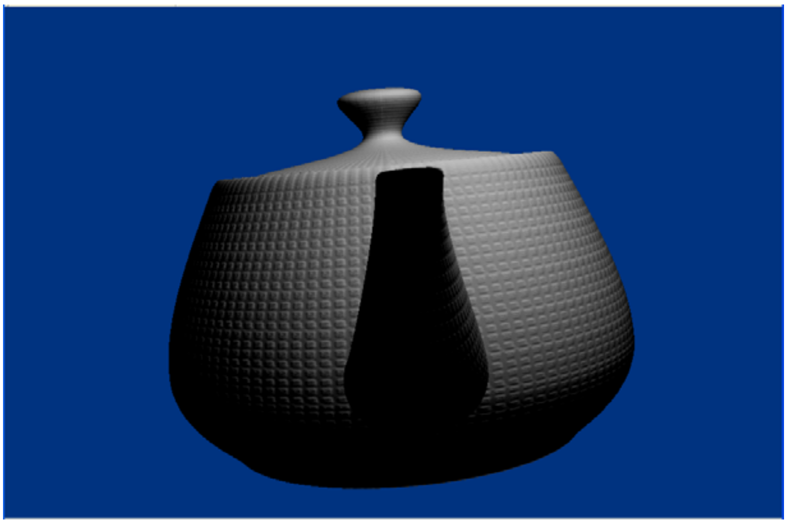
7.3. Discussion

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.



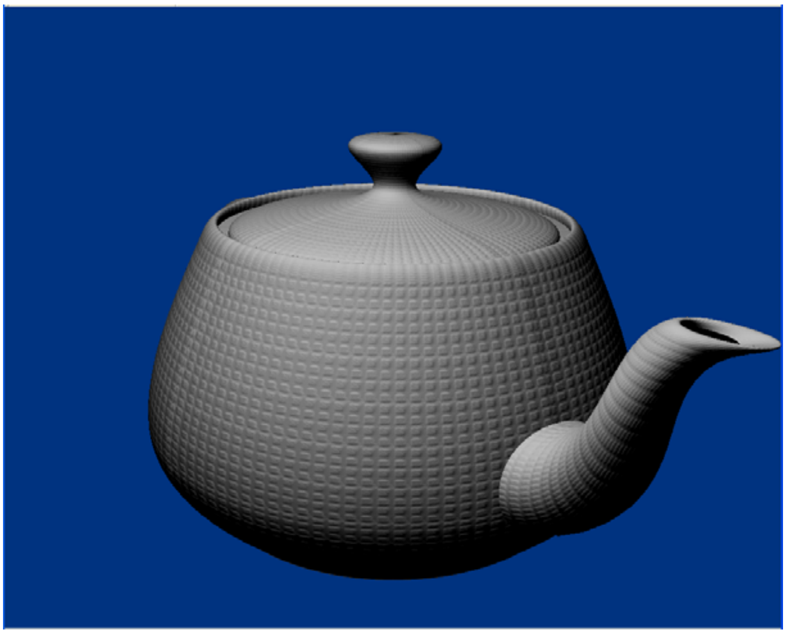
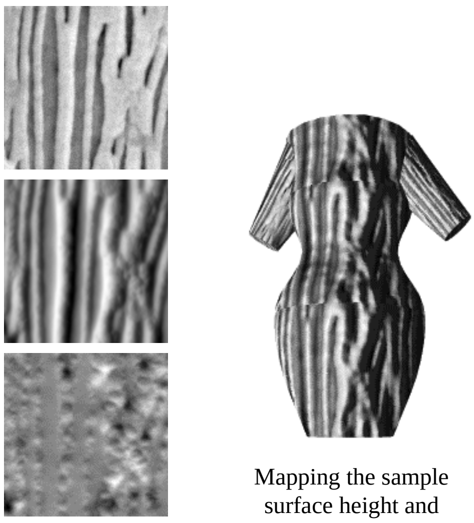


Figure 7.3.1 Two still images of a real-time sequence produced by rendering synthesised surface gradient and albedo maps using the method described in [Robb2003]. The images were generated by Michael Robb using synthesised surface gradient and albedo maps supplied by the author.

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

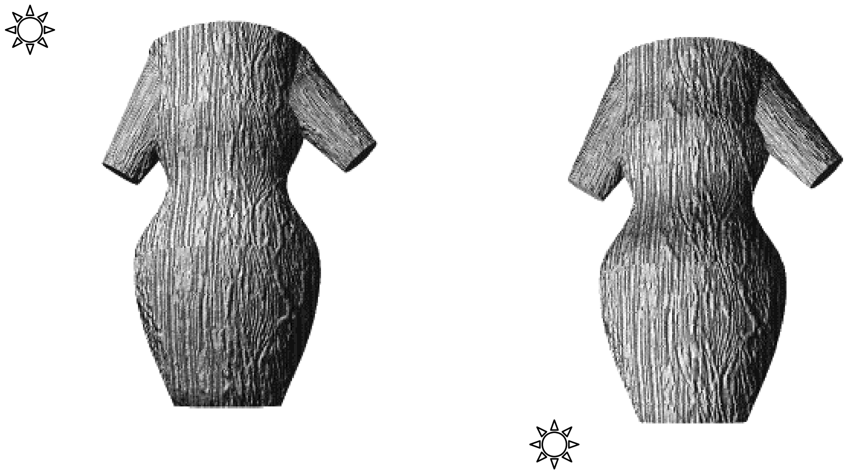
The synthesised representations can be input into graphics software packages to perform texture mapping on 3D models. If the packages can not directly use the synthesised representations, certain transformation is required. We briefly introduce the use of the synthesised 3D surface gradient and albedo 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.



Sample surface gradient and albedo maps

Mapping the sample surface height and albedo maps on a 3D model

(a)



(b)

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].