A HLAC Shape Descriptor Extraction Method for 3D Solid Textures

MOTOFUMI T. SUZUKI, YOSHITOMO YAGINUMA, NORITAKA OSAWA

National Institute of Multimedia Education
NIME, 2 – 12 Wakaba, Mihama-ku, Chiba 261-0014
JAPAN

Abstract: Texture mapping has become the most important method to increase the level of detail for a 3D computer graphics model without increasing its geometric complexity. Procedural texture synthesis methods have been used to generate solid textures for 3D models. These methods use noise functions and generate various solid texture patterns. The number of 3D solid texture databases on the web is increasing, and texture databases require systematic search methods for efficient retrievals. We have applied a 3D HLAC (Higher Order Local Autocorrelation) feature extraction technique to a search system which looks for similar patterns of the 3D solid textures from databases. 3D HLAC shape descriptors can classify similar texture patterns advancing the development of efficient search engines.

Key-Words: Procedural textures, HLAC, Shape descriptor, Similarity search, 3D model

1 Introduction

The use of 3D models is gaining wide popularity, and their use is very important in software applications for computer graphics. Recently, similarity search techniques for retrieving 3D models from the Internet [1-8] have been investigated intensively. Most of these similarity search techniques and systems are designed for comparing shapes, and they are not suitable for comparing textures, especially textures which are often used for 3D models. Texture mapping has become the most important method of increasing the level of detail for 3D computer graphics models. The number of available textures is increasing rapidly on various web sites, because textures are useful for improving realistic 3D computer graphics images without increasing their geometric complexity. We have implemented an experimental 3D solid texture search engine based on a pattern classification method using 3D HLAC (Higher Order Local Autocorrelation) shape descriptors. 3D solid textures are different from 2D image textures in dimension and are represented in voxels rather than pixels. Therefore, popular 2D HLAC masks have been extended to 3D HLAC masks to analyze 3D solid textures for our system. The system can search for similar patterns of 3D solid textures form databases. The system was tested by a database which is composed of artificially generated 3D solid textures based on Perlin’s noise functions and procedural methods.

2 Procedural Textures

Procedural methods are used for generating textures in 3D computer graphics. Various types of textures such as stone, wood, cloud, water and smoke are generated. In the procedural methods, texture functions are involved in order to synthesize various patterns. A variety of research has been conducted to generate texture images, including Fourier synthesis [23], fractal method [22] and a stochastic texture model [24]. Important introductions and concepts for procedural textures can be found in papers [19] and [20]. For realistic textures, stochastic functions have been applied to generate textures with irregular patterns. Often, these functions are called noise and return random numbers. Lattice noises are important noises for typical procedural texture synthesis. Perlin’s noise function [19] is one of the most efficient noise functions to generate procedural textures. The function is defined by the following equation where P represents the coordinate of lattice point \((x, y, z)\):

\[ \text{PerNo}(P) = \sum_{i=0}^{k} \frac{\text{Noise}(2^i P)}{2^i} \]

The function Noise(p) is a repeatable random function. The function returns noises when the coordinate of lattice point \(P (x, y, z)\) is passed to the function. By evaluating the function of the sequential 3D voxel coordinate \((x, y, z)\), a 3D solid texture is
generated. The function returns a floating number between -1.0 and +1.0 at each lattice coordinate. These floating numbers are scaled into 256 gray color codes between 0 and 255 for a CRT display. Figure 1 shows 64 slices of the 2D images (64 x 64 pixels) which are generated by the functions. Although these are slices of 2D images, actual 3D solid textures are represented in the form of three-dimensional voxels.

Figure 1: 64 slices of the 2D images (64 x 64 pixels)

In our system, two types of 3D solid textures are generated from noise functions which include (a) Perlin’s noise functions and (b) Perlin’s noise functions with fractal noises. Figure 2 shows slices of these two types of 3D solid textures. The left image shows stochastic texture with various intensities. The right image is also stochastic texture, but irregular patterns are included by reflecting the characteristics of fractal functions. Often fractal functions are essential to synthesize most natural textures.

Figure 2: A texture based on Perlin’s noise function (left) and one with fractal noises (right).

3 3D HLAC

Autocorrelation functions are translation invariants, and their unique properties have been used for various pattern recognition applications for 2D images. An important approach is presented in [9], in which higher order local autocorrelation (HLAC) is proposed which successfully reduces the computing costs. HLAC has been used as a feature descriptor for various pattern recognition methods including gesture recognition [14], image retrieval [16] and face recognition [10]. For such applications, HLAC mask patterns are essential for computing features of 2D images. We have extended the typical 2D HLAC mask patterns to 3D HLAC mask patterns to handle 3D solid textures. A 3D HLAC mask pattern file which was generated by a simulation program is available at the following web site: (http://open.nime.ac.jp/). The file contains 251 3D HLAC mask patterns which can be used for extracting 3D HLAC shape features. Figure 3a shows all the possible 251 3D HLAC mask patterns in a string sequence representation. Each alphabet represents the unique location of grid cells in a 3 x 3 x 3 cube. The relations of the cell positions and alphabet labels are defined in Figure 3b.

Figure 3a: 3D HLAC mask patterns in string sequence forms
The 3D HLAC mask patterns are applied to 3D solid textures as shown in Figure 4. In the figure, the large cube represents a 3D solid texture and the small cube represents a 3D HLAC mask. Intuitively, the mask is moving randomly inside of the large cube to extract 3D HLAC features. When the mask is moved to a new position, 251 mask patterns are used to test if the voxels exist at the mask position. When each mask pattern hits the voxels, the color intensity value of the voxels are added to the 3D HLAC features. The mask may be moved sequentially to process the entire large cube, but it requires high computation time. Thus, we have set the system to move the mask randomly so that the system can reduce the computation time. It is important that the mask moves to statistically random positions to capture 3D HLAC features.

Once all the required 3D HLAC features are extracted for each 3D solid texture in the database, the 3D HLAC features are stored into a histogram. The extracted 3D HLAC shape features consist of 251 vectors. Shape similarities of 3D models are compared by using a histogram of these 251 vectors for each 3D model. By comparing histogram values of each model, the system can sort 3D models based on pattern similarities. The advantage of using 3D HLAC features is that the number of vectors for comparison can be reduced. It is especially efficient for 3D volume data such as voxels.

4 Experiments and Results
This section describes (1) the search system, (2) the time needed for computing HLAC features, and (3) the 3D solid texture search system.

4.1 The Search System
A Pentium 4-2.66 MHz CPU with 1024 Mbytes of memory is used to generate 3D solid textures. A texture database has been created to test the pattern classification power of 3D HLAC features. Perlin's noise functions which are described in Section 2 are used to synthesize various patterns of 3D solid textures. A total of 600 3D solid textures are synthesized as a database whereas 50% of the 3D textures are generated by using fractal functions. Thus, the database contains two types of Perlin's noise function textures which include (a) a simple Perlin's noise function and (b) a Perlin's noise function with fractal noises. For 3D solid texture generation, a size of 64 x 64 x 64 voxels is chosen. About 900 seconds are needed to synthesize solid textures for type (a) function and 1500 seconds for type (b) function. Type (b) function requires more computation time compared to type (a) because fractal noise calculations are involved.

Figure 5 shows the interface of the search system which visualizes 3D solid textures. Textures with similar patterns are listed based on 3D HLAC feature computation. As shown in the figure, each 3D solid texture is sliced into 64 2D images to visualize texture patterns.
4.2 The Time Needed for Computing HLAC features

About 1600 seconds are needed to compute 3D HLAC features for 600 3D solid textures of a 64 x 64 x 64 voxel grid. Since 3D solid texture is three-dimensional, computation time is increased significantly for large voxel grid sizes. The number of cubes is raised to the third power against grid sizes. Exponential computation time is needed for large grid sizes of solid textures. However, the 3D HLAC computation should be carried out only one time as a preprocess, and 3D solid textures are compressed into 251 3D HLAC features. Thus, 3D HLAC feature representation of 3D solid textures allows the search system to respond extremely fast. The average query time for searching 30 solid textures from a database which contains 600 solid textures is about 1.7 seconds.

4.3 The 3D Solid Texture Search System

Figures 6a, 6b and 6c show examples of search results based on similarity patterns. In the examples, each 3D solid texture is decomposed into 64 slices of 2D images. In each figure, similar 3D solid textures are listed based on solid pattern similarities. The 3D HLAC features successfully classified type (a) and type (b) functions of texture patterns. As shown in the figures, 3D solid textures with similar patterns are listed against queried 3D solid textures. Although typical 2D HLAC shape descriptors have some difficulty handling 3D solid textures, our preliminary system successfully searched for 3D solid textures by extending 2D HLAC masks to 3D HLAC masks.
5 Conclusion and Future Work

The use of 3D solid textures is very important for many computer graphics applications. Various procedural methods have been used to synthesize 3D solid textures. Content-based search techniques are essential for managing databases of such 3D solid textures. In our system, 3D HLAC mask patterns are used to analyze shape descriptors of 3D solid textures. We have extended typical 2D HLAC mask patterns to 3D HLAC mask patterns which makes it possible for 3D HLAC shape descriptors to classify 3D solid texture patterns. We have implemented an experimental search system for a 3D solid texture database. The system includes classification capabilities based on 3D HLAC shape descriptors. Experimental search tests showed that the system successfully could find 3D solid textures with similar patterns against the queried textures. The system responds fairly fast by reflecting shape descriptor abstraction properties of 3D HLAC.

In the experiment, we have used 3D solid textures with 256 grey scale colors. In addition to these grey scale colors, we will examine 3D HLAC shape descriptors for 3D solid textures with RGB colors in a future work. The 3D solid textures with RGB colors may be able to be synthesized easily by adding noise functions to each R, G and B component. By using the synthesized color textures, we will investigate how 3D HLAC shape features are related to texture with RGB colors.

Acknowledgements:

This research was supported by research grants from the Okawa Foundation for Information and Telecommunication (#2002-02-29) and a grant-in-aid from the Ministry of Education, Science, Sports and Culture, Japan (#EYS-B(2)-15700115)

References:


