



## A Grain Generation Method for Large Die Data Using the Out-of-Core Method

Cam Nguyen Tang<sup>1</sup>, Daijiro Uzuyama<sup>2</sup>, Kenjiro T. Miura<sup>3</sup>, Shin Usuki<sup>4</sup> and Mamoru Kikuta<sup>5</sup>

<sup>1</sup>Shizuoka University, [f0130039@ipc.shizuoka.ac.jp](mailto:f0130039@ipc.shizuoka.ac.jp)

<sup>2</sup>Shizuoka University, [f0930018@ipc.shizuoka.ac.jp](mailto:f0930018@ipc.shizuoka.ac.jp)

<sup>3</sup>Shizuoka University, [tmkmiur@ipc.shizuoka.ac.jp](mailto:tmkmiur@ipc.shizuoka.ac.jp)

<sup>4</sup>Shizuoka University, [dsusuki@ipc.shizuoka.ac.jp](mailto:dsusuki@ipc.shizuoka.ac.jp)

<sup>5</sup>Calsonic Kansei Corp., [KIKUTA-M@ck-mail.com](mailto:KIKUTA-M@ck-mail.com)

### ABSTRACT

The surfaces of automobile interior parts have geometrically microfabricated textures, which are called **grain textures**. The purpose of our research is to mill dies with grain by use of digital data. We generate microfabricated geometric data from a large CAD model of a die with the out-of-core method, texture synthesis, and displacement mapping. With these geometric data, the die with grain can be manufactured by milling or laser machining.

**Keywords:** digital grain generation, out-of-core, s-patch.

**DOI:** 10.3722/cadaps.2012.915-923

### 1 INTRODUCTION

The surfaces of home electric appliances, stationery, and automotive interior parts have geometrically microfabricated textures, which are called **grain textures**. These textures have a variety of purposes, such as to create premium accents (making a good visual and tactile impression) and to stop slippage. It is especially important to have premium accents for the surfaces of automobile interior parts. Figure 1 shows examples of grain texture in automobile interior parts. The high-quality grain texture is necessary to make the pattern of grain continuous and with no distortion. Grain texture on the surfaces of resin products has traditionally been generated by etching or electroforming techniques. In our research, we generate grain texture using digital data to make products of high quality with a simple process.



(a) Automobile interior parts (b) Grain texture

Fig. 1: Examples of automobile interior parts and grain.

The grain machining system proposed here consists of the digital data generation process to add a geometric model of the die with grain texture and the machining process using the digital data as show in Fig. 2 [1]. We propose a method to map grain texture to free-form surfaces, including non-developable surfaces. Texture synthesis [2, 3] was used to generate digital data of grain texture. After mapping, we generated mesh data with grain texture by displacement mapping [1]. In this paper, we propose a new out-of-core method to handle large mesh data, which is a problem for processing on a PC.



Fig. 2: The grain machining system proposed previously [1].

## 2 RELATED WORK

One of the existing methods to generate grain texture on a die is the electroforming technique. This is a process of synthesizing a metal object by controlling the electrodeposition of metal passing through an electrolytic solution onto a metal or metalized form. As the repeatability of this method is good, it is generally used to fabricate the dies for interior parts of luxury cars. The degree of accuracy is about 0.1 micrometer. However, this method is time consuming and incurs high costs to produce a metal object. The other existing method is etching, a process of using strong acid or mordant to cut into the unprotected parts of a metal surface to create a design in the metal. Although the processing time of the method is short, it is not of high quality. In addition, the method is bad for the environment because of the chemical treatment required.

In our research, we use images as grain texture captured by measuring natural materials. A computer graphic technique (texture synthesis) is used to expand artificial texture for grain [4, 5]. Miura et al. proposed a method to map texture to free-form surfaces by three-dimensional extension of texture synthesis [6]. They mapped texture even on a mesh including non-developable free-form surfaces by controlling the elongation of the texture by subdividing the surface into several regions using its normal vector.

## 3 GENERATING A LARGE GRAIN TEXTURE OF DIGITAL DATA

There are two major problems in generating grain texture for a large die. The first problem is that, as the surface area of the object is large, a single standard-sized texture image is not sufficient. It is not easy to capture a large grain image. Hence, the texture needs to be expanded or artificially generated.

We should minimize distortions during mapping of the expanded or artificially generated texture on the surface.

Another problem is that because of the vast amount of mesh data generated from a CAD model, batch processing is not possible. The CAD model of the die should be converted into a very fine mesh to maintain high quality of grain texture design. The vast amount of data is too large to store in the main memory of a PC. We propose an out-of-core method, which stores data in several files and only required partial data are loaded into the main memory.

### 3.1 Out-of-core Method for Generating Grain Texture

Polygonal mesh data of a model are frequently too large to load into the main memory of the PC. To avoid severe performance degradation due to virtual memory swapping, out-of-core [8] algorithms have been proposed that allow efficient decimation of polygon meshes without requiring the entire data set to be present in main memory. The challenge here is to design suitable data structures that avoid random access to parts of the model.

In our research, to efficiently process data from a large CAD model, we divide the CAD model into smaller regions (patches) for processing. With grain texture data and surface patch data, we have the following processes.

- 1) Exploring adjacency between the surfaces patches
- 2) Matching the boundary vertices of the adjacent patches
- 3) Texture synthesis and texture mapping
- 4) Blending texture along the boundary between the adjacent patches
- 5) Displacement mapping

Processes 1, 3, and 5 can be handled independently of each surface in order. Processes 2 and 4 match the boundary vertices of adjacent surfaces, which requires information regarding the adjacent surfaces. A patch under processing is loaded and its adjacent patches are loaded sequentially one by one (see Fig. 3). Therefore, we can reduce the data size to be loaded in the main memory at the same time with the data of at most two patches.

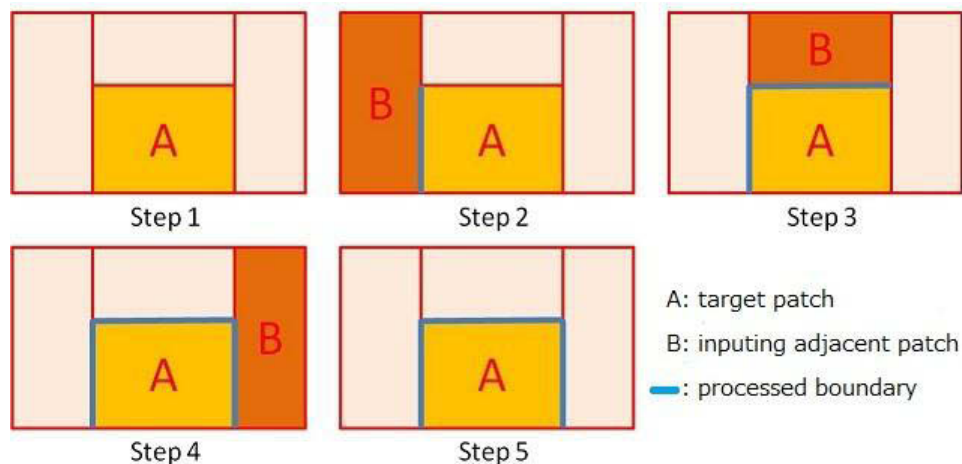
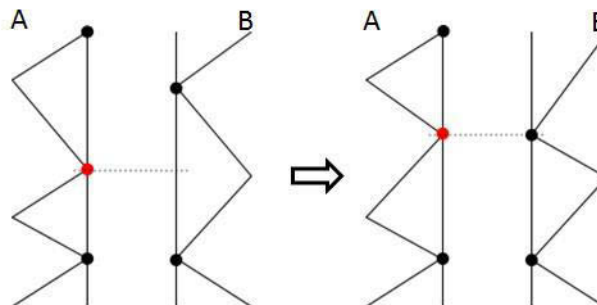


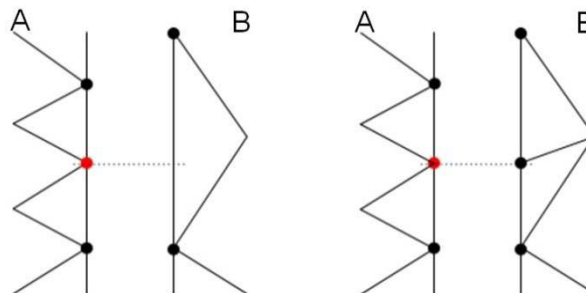
Fig. 3: The out-of-core method.

### 3.2 Matching the Boundary Vertices

The positions of the vertices of the polygonized patches do not match along the boundaries between two adjacent patches. If we map texture to these patches, they will have gaps along their boundaries. It is necessary to match all vertices on the boundaries between adjacent patches. As shown in Fig. 4, we change the position of the boundary vertex or insert a vertex in the boundary for matching.



(a) Moving a Vertex.



(b) Inserting a vertex

Fig. 4: Matching the boundary vertices.

### 3.3 Texture Synthesis and Texture Mapping

The method proposed by Miura et al. [6] allows smooth mapping of textures with an arbitrary precision for free-form surfaces, including non-developable surfaces. However, this requires a long processing time to generate complex data. Their method is not suitable for an out-of-core algorithm, which processes each region of the model individually. Therefore, in our research we use texture synthesis by blending with a selection of any part of the texture. First, we define texture coordinates  $(s, t, u)$  for each patch. The origin of this coordinate system is not important and, for example, we can use the origin of the world coordinate system. The texture coordinates  $(s, t)$  and the  $u$ -axis direction, which is perpendicular to both the  $s$ - and  $t$ -axes, are determined by the direction of the averaged normal vector of the patch (see Fig. 5).

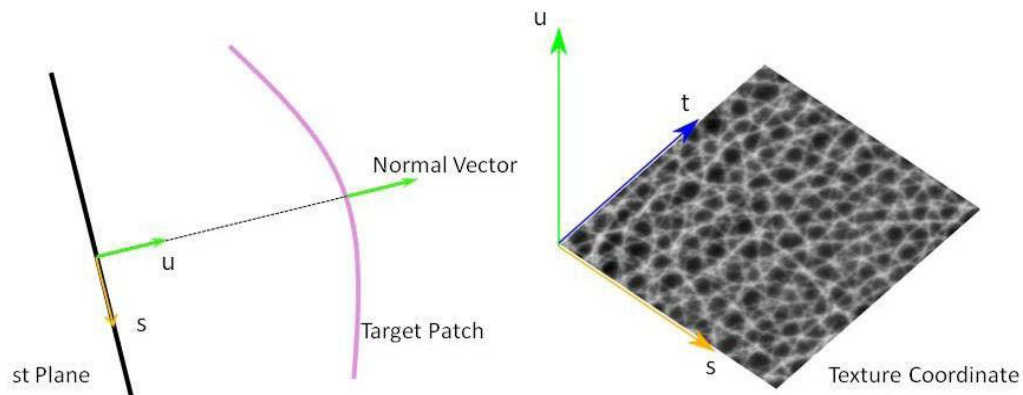


Fig. 5: Texture coordinates.

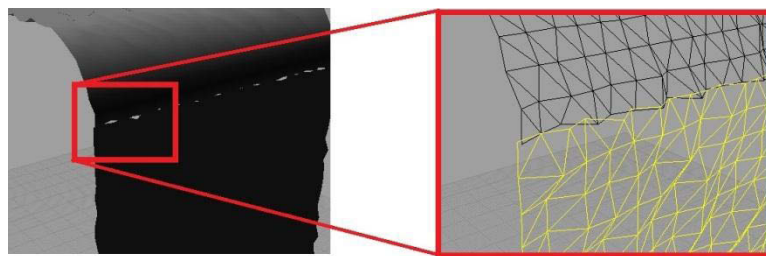


Fig. 6: Example of discontinuity of the meshes along the boundary between two adjacent patches.

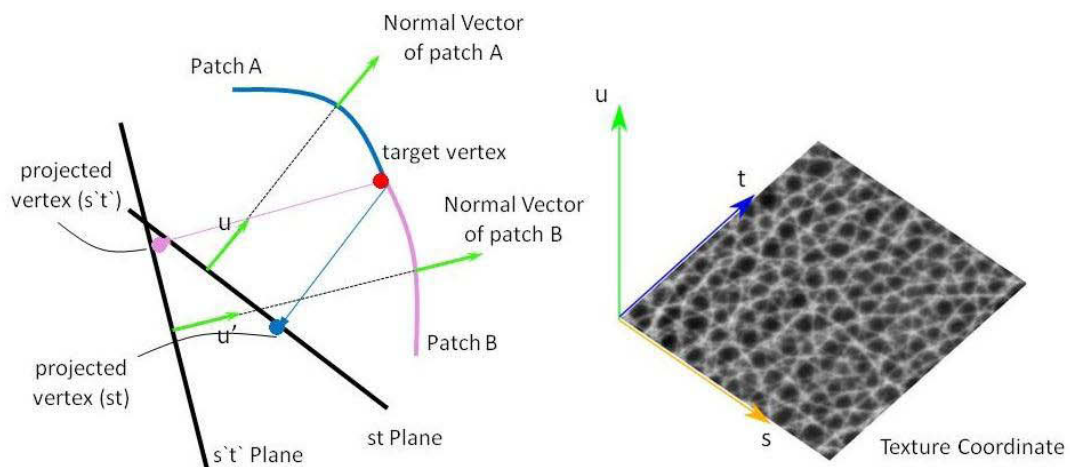


Fig. 7: Texture Mapping.

When mapping a texture, each vertex is projected on the  $st$  plane to obtain texture coordinates for the patch. With the texture coordinates of each vertex, we obtain a pixel value, which is the color information of the vertex. If we generate grain texture with these data, because the texture coordinate

systems for different patches are not the same and the texture coordinates for the vertices whose positions are the same but that belong to different patches are generally different, as illustrated in Fig. 7, the grain texture will become discontinuous along the boundaries between adjacent patches as shown in Fig. 6. Therefore, it is necessary to perform color blending for regions near the boundaries.

#### 4 BLENDING THE BOUNDARY REGIONS BETWEEN ADJACENT PATCHES

We perform simple blending with the method shown in Fig. 8. The method operates with the shortest distance from a vertex to the boundary and the normal vector of the patch. The blending region is adjusted by a specified distance ( $D_{max}$ ). The color information of the vertex is obtained from the weight of the target vertex and that of the corresponding vertex of the adjacent patch.

Let  $L$  denote the blended color information of the target vertex;  $l_A, l_B$  denote the weights of Patches A and B, respectively;  $v_A, v_B$  denote the colors of the vertex of the patches A and B, respectively; and  $d$  is the shortest distance from the target vertex to a vertex on the boundary. The blended color information  $L$  can be calculated by

$$L = l_A \cdot v_A + l_B \cdot v_B \quad (2.1)$$

$$= \frac{1}{2} \left(1 + \frac{d}{D_{max}}\right) \cdot v_A + \frac{1}{2} \left(1 - \frac{d}{D_{max}}\right) \cdot v_B$$

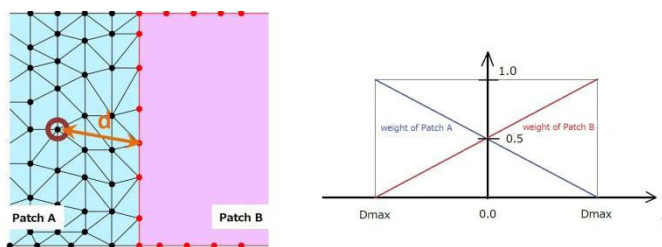


Fig. 8: Blending the boundary between patches.

However, this method is not applicable in cases where the region to be processed is close to more than two patches.

We have used the S-patch [7] method, which is built on the theory of Bezier simplexes, to solve the problem of multi-adjacent patches. To describe the blended color  $L$  (see Fig. 7), we map the multi-adjacent patches into an intermediate simplex  $\Delta$ . Let  $\alpha_i$  denote the signed area of the triangle  $PP_iP_{i+1}$  and  $v_i$  denote the control color of each patch. Let

$$\pi_i = \alpha_1 \cdot \alpha_2 \cdots \alpha_{n-2} \cdot \alpha_{n+1} \cdots \alpha_n \quad (2.2)$$

for  $i = 1, 2, \dots, n$ , denote the product of all areas except  $\alpha_{i-1}$  and  $\alpha_i$  and let

$$l_i = \frac{\pi_i}{\pi_1 + \pi_2 + \cdots + \pi_n} \quad (2.3)$$

With these definitions, every point  $p \in P$  is mapped by the following  $L$ :

$$L_{(p)} = l_1 v_1 + l_2 v_2 + \cdots + l_n v_n \quad (2.3)$$

The final step is to add grain texture to the mesh by calculating vertex displacements in the directions of the normal of each vertex.

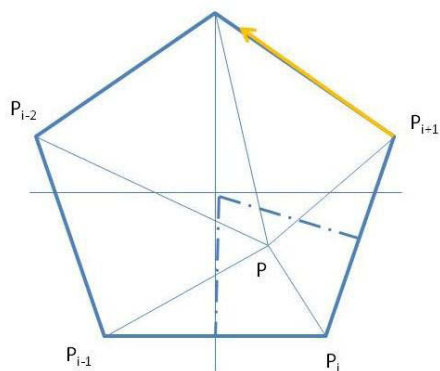


Fig. 9: Parameter of S-Patch.

With the S-Patch method we can blend the boundary region of multi-adjacent patches. After blending, the grain texture in the boundary region between two patches becomes continuous and smooth. Figure 9(a) shows a surface mesh example consisting of 5 patches and 9(b) shows the result of blending, which we map the same texture independently to each patch and blend them. The mapped texture exhibits no discontinuity.

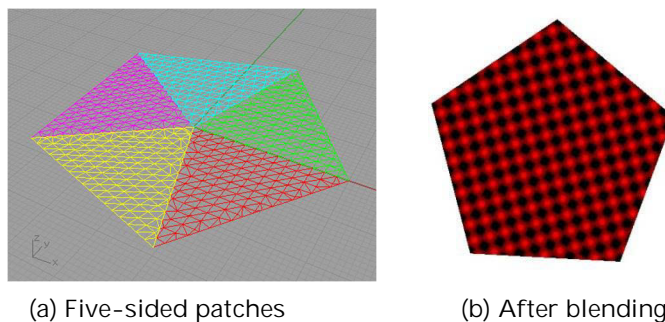


Fig. 10: Five-sided mesh consisting of 5 triangular patches blended with the S-patch method.

## 5 ADDITION OF GRAIN TEXTURE TO MESH

Figure 11 shows the results of applying the grain to an automotive interior part CAD model. The continuous grain is generated on the free-form surfaces or non-developable surfaces. We used a texture image with 950x950 pixels, and distance between two pixels corresponds to 100  $\mu\text{m}$  on the mesh and hence we adopted 100  $\mu\text{m}$  as the length of the triangles to generate the mesh. The number of patches is 86 and the mesh includes about 150 million vertices and 50 million polygons. The data size is 2.44 GB (binary STL file). It took about 172 minutes on a PC with a Core i5 2.8GHz CPU and 16 GB of main memory.

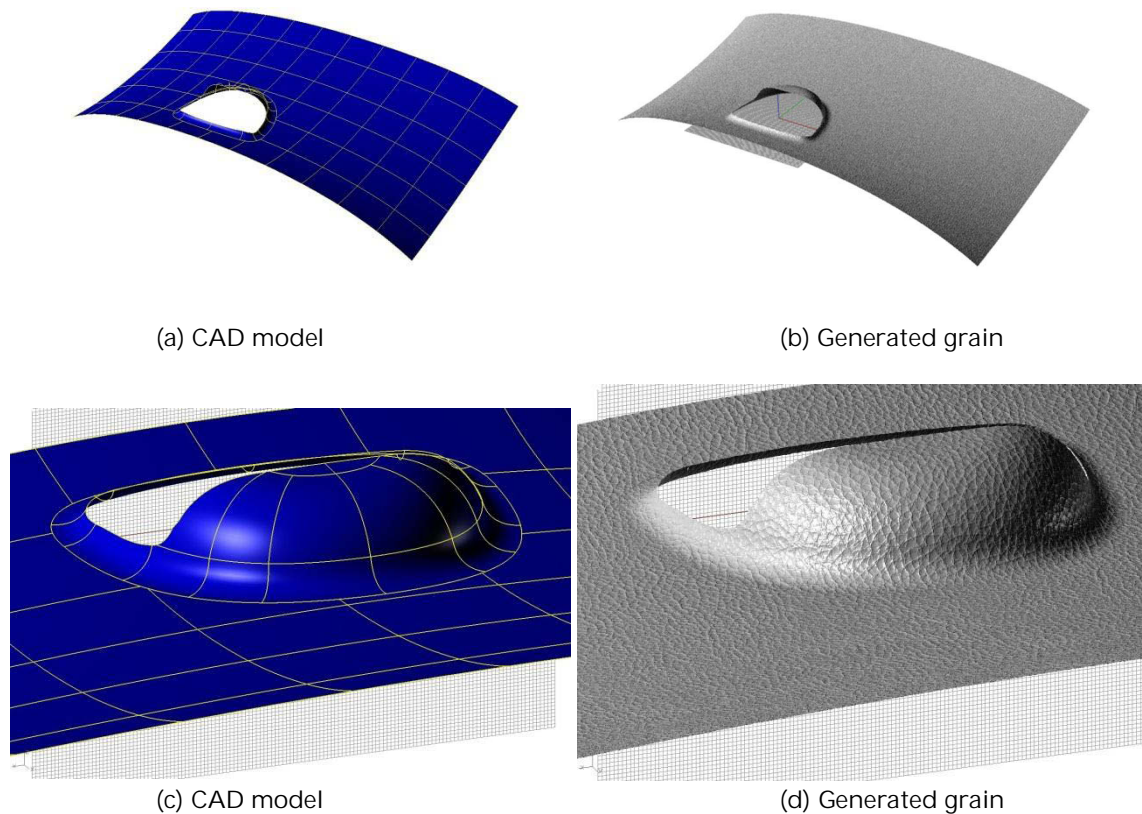


Fig. 11: Generated grain.

## 6 CONCLUSIONS AND FUTURE WORK

This paper proposed a method of digital grain texture generation for 3D mesh. Using this method, we can map grain texture that varies smoothly on free-form surfaces, including non-developable surfaces. With the out-of-core method we can process each patch for a large CAD model and with the S-Patch method we can make the grain texture continuous among more than four patches.

As future work, we will implement software to handle more complicated and large-scale CAD models to extend the applicability of our method for digital grain generation.

## REFERENCES

- [1] Uzuyama, D.; Kikuta M.; Miura, K.T.: Development of a grain milling system by use of digital data, *Computer-Aided Design & Applications*, 7(3), 2010, 291-296. DOI: 10.3722/cadaps.2010.291-296
- [2] Wei, Li-Yi; Levoy, M.: Fast texture synthesis using tree-structured vector quantization, *Proceedings of SIGGRAPH 00*, 2000, 479-488.



- [3] Efros, A.A.; Freeman, W.T.: Image quilting for texture synthesis and transfer, Proceedings of SIGGRAPH 01, 2001, 341-346.
- [4] Nakaishi, M.; Kobayashi, Y.; Shirai, K.: Creation of the wide-area texture which used texture synthesis, SICE Tohoku branch, No.246 meeting, 2008.
- [5] Kamibata, J.; Ohtake Y.; Suzuki H.: Shibo Design via Texture Synthesis, Proc. Spring Meeting of Japan Society of Precision Engineering, 2009.
- [6] Miura, K.T.; Uzuyama, D.; Kikuta M.: Extension of Image Quilting into 3D and Its Application to Grain Generation, Computer-Aided Design & Applications, 8(4), 2011. 545-555. DOI: 10.3722/cadaps.2011.545-555
- [7] Loop, C.; DeRose, T.: Generalized B-spline surface of arbitrary topology, Computer Graphics, 24(4), 1990.
- [8] Ahn, M.; Guskov, I.; Lee, S.: Out-of-core remeshing of large polygonal meshes Visualization and Computer Graphics, IEEE Transaction on, 2006.