www.cadanda.com Computer-Aided Design and Applications

Extension of Image Quilting into 3D and Its Application to Grain Generation

Kenjiro T. Miura[1], Daijiro Uzuyama[2] and Mamoru Kikuta[3]

[1]Shizuoka University, tmkmiur@ipc.shizuoka.ac.jp
[2]Shizuoka University, f0930018@ipc.shizuoka.ac.jp
[3]Calsonic Kansei Corp., KIKUTA-M@ck-mail.com

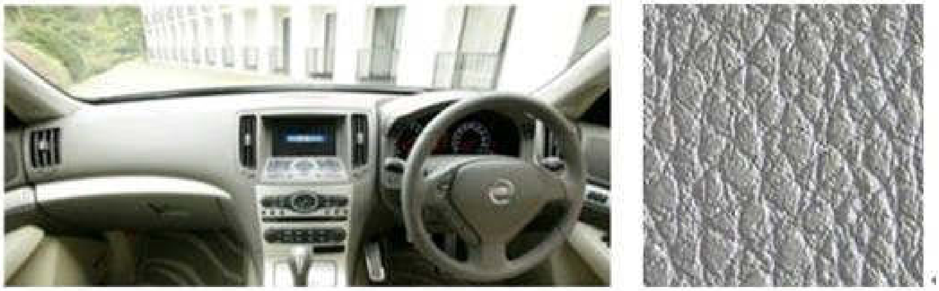
ABSTRACT

The surfaces of automobile interior parts have geometrically microfabricated, which are called grain. The purpose of our research is to mill dies with grain by use of their digital data. We generate microfabricated geometry data from a CAD model of a die with an image quilting method extended into 3D and displacement mapping. With these geometry data, the die with grain is manufactured by milling.

Keywords: digital grain generation, 3D image quilting, displacement mapping.
DOI: 10.3722/cadaps.2011.545-555

1 INTRODUCTION

The surfaces of automobile interior parts, stationery products and home electric appliances have geometrically microfabricated, which are called “grain”. This purpose is to have premium accents (making a good showing and tactile impression) and slip stopper. Especially it is important to have premium accents for the surfaces of automobile interior parts. Fig. 1 shows the example of automobile interior parts and grain. It is necessary to make the pattern of the grain continuous and no distortion. To generate the grain to the surfaces of resin products, the grain is heretofore generated to the die of them by use of etching or electroforming technique. In our research, we generate the grain by use of digital data in order to make the products high quality and to make the process simple.



(a) Automobile interior parts (b) Grain

Fig. 1: An example of automobile interior parts and grain.

The grain machining system proposed by our previous research consists of the digital data generation process to add the geometric model of the die with grain texture and the machining process by use of the digital data as shown in Fig.2 [1]. This paper proposes a method to map grain texture to a free-form surface including regions where the Gaussian curvature is not equal to 0, i.e. undevelopable. The method can control accuracy of the mapped texture. The texture mapping required for grain generation is not necessary to preserve the original texture and it is desirable to minimize its deformation when the surface is locally projected to a tangent plane of the surface. It is acceptable to locally synthesize texture as image quilting. After mapping the texture, we generate mesh data with grain texture by displacement mapping [1].

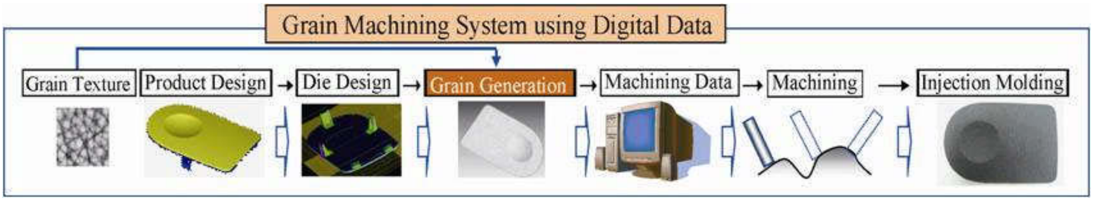


Fig. 2: The Grain Machining System Proposed by [1].

2 RELATED WORK

In this chapter, we review 1) digital grain generation and 2) texture synthesis.

2.1 Digital Grain Generation

One of the existing methods to generate the grain to the die is electroforming technique. It is a process of synthesizing a metal object by controlling the electro-deposition of metal passing through an electrolytic solution onto a metal or metalized form. Since the repeatability of this method is good, it is generally used to fabricate the die of interior parts of luxury cars. The degree of accuracy is about 0.1 micrometer. However, this method needs long time and high cost to synthesize a metal object. The other existing method is etching. Etching is the process of using strong acid or mordant to cut into the unprotected parts of a metal surface to create a design in the metal. Though the processing time of the method is short, this is not of high quality. In addition, the method is bad for the environment for chemical treatment.

The approach proposed our previous paper [1] is to mill dies with grain by use of their digital data. Existing methods can't be used without specialist equipments and elaborated human skills. However, the method which is to mill by use of the digital data is able to be used by general processing machine. In existing methods, it is difficult to control size, pattern and position of grain. However, it may be palliative to easily control them in our approach. The grain milling system which is developed by us is composed of two parts. One part is to generate grain to shape models of dies with parametric surfaces by use of digital data of grain. The other part is to process dies by milling or laser machining with the digital data of dies with grain.

In order to prevent dangerous work and to reduce the cost of labors and the environmental loads, the grain generation with digital data is gathering momentum recently [4]. The length between pixels in the grain texture is about from 25μm to 100μm and it is very small relatively to the size of the molded product. Hence it has been thought to be very difficult to machine a die with grain texture by ball-end milling. However, thanks to the advancement of the software technology, it becomes possible to machine the die represented by a large-scale mesh in the STL (Standard Triangulated Language) format.

2.2 Texture Synthesis

Although in our research, we use images as grain texture captured by measuring natural materials, several techniques were proposed to synthesize or expand artificial texture for grain [3, 4]. They used the image quilting technique [5] developed in the field of computer graphics.

In CG, instead of designing a new texture, several techniques were proposed to generate a texture with a specified size from a relatively small texture with preserving features of the original texture. Among them there are methods to extract necessary parameters to synthesize texture [11, 12]. These methods cannot be applicable to natural texture such as cattle skin which is frequently used for grain texture. Other methods generate sample collections of texture which consist of textures modified by scaling and/or rotation at first, then synthesize a new texture by detecting a pixel in the most similar region among the collections to a portion to be generated in the output texture [5, 13]. Although these methods have a higher reproducibility of the features of the original texture than the synthesis methods by parameter extraction and suppress distortion of the texture after mapping to a 3D mesh, they take a lot of processing time for large-scale texture synthesis necessary for grain texture generation.

Image quilting is a technique which synthesizes texture by copying rectangular regions from the sample texture [6]. It is more efficient than the pixel-based texture synthesis. It can reproduce features of the sample texture almost faithfully and synthesize texture in a practical time (15 seconds to a few minutes in MATLAB to output a 384×384-pixel image from a 192×192-pixel sample image [6]). Our proposed method extends image quilting to 3D meshes since our target is to generate actual grain texture of injection molding dies for manufacturing and the lengths of triangular polygons of the mesh are around 25-50 microns. Although the number of polygons can reach more than one million, the complexity of the shapes of the die is restricted because of injection molding process. There are a lot of texture mapping techniques such as [7, 8, 9, 10], but for the actual grain texture a relatively simple and robust texture mapping is preferable because of the simplicity of the part geometry. As the users for injection molded parts with grain texture can not only look at, but also touch their surfaces, it is strongly required to be able to control the elongation of the texture. We explain the image quilting technique in detail in the next chapter.

3 IMAGE QUILTING

Image quilting needs to remove discontinuity of the boundary between rectangular regions and the generated region. In order to remove the discontinuity, it makes their overlap regions, finds a path to minimize the difference of pixel values of the overlapped region on the path, and connects these regions. For a given sample texture shown in the left of Fig.3, Fig3.(a) shows a result of random dispositions, (b) a result of dispositions to minimize the difference of pixel values of the overlapped regions, and (c) a result after finding the boundary paths to minimize the difference and blending the pixel values for the deposition (b). A texture of high quality can be synthesized by image quilting.

4 EXTENSION OF IMAGE QUILTING FOR 3D MESHES

Image quilting is a technique for 2D texture and it is not possible to apply it for 3D meshes without modification. We have to solve the following three problems to perform image quilting for 3D meshes.

1. How do we define a rectangular region on the 3D mesh?
2. How do we calculate the difference of the pixel values in the overlapped region?
3. How do we find the best boundary path in the overlapped region for blending the pixel values?

Considering the above three problems, we adopt the following procedures.

1) Subdivide the 3D mesh into regions by use of the normal vectors of the triangular polygons.

2) Apply texture mapping for a certain region generated by Step 1) and call it the initial region.

3) For other regions,

(a) Select a region among them and search the boundary line of the region.

(b) Smooth out the boundary line.

(c) Generate multiple loops around the boundary line

(d) Search the best portion of the texture for the region by comparing the difference of the pixel values with the overlapped part of the initial region where the texture has been already mapped.

(e) Search the best boundary path in the overlapped part.

(f) Blend the pixel values along the boundary path.

4) Perform the displacement mapping.

In the following sections, we will explain the above steps in detail.

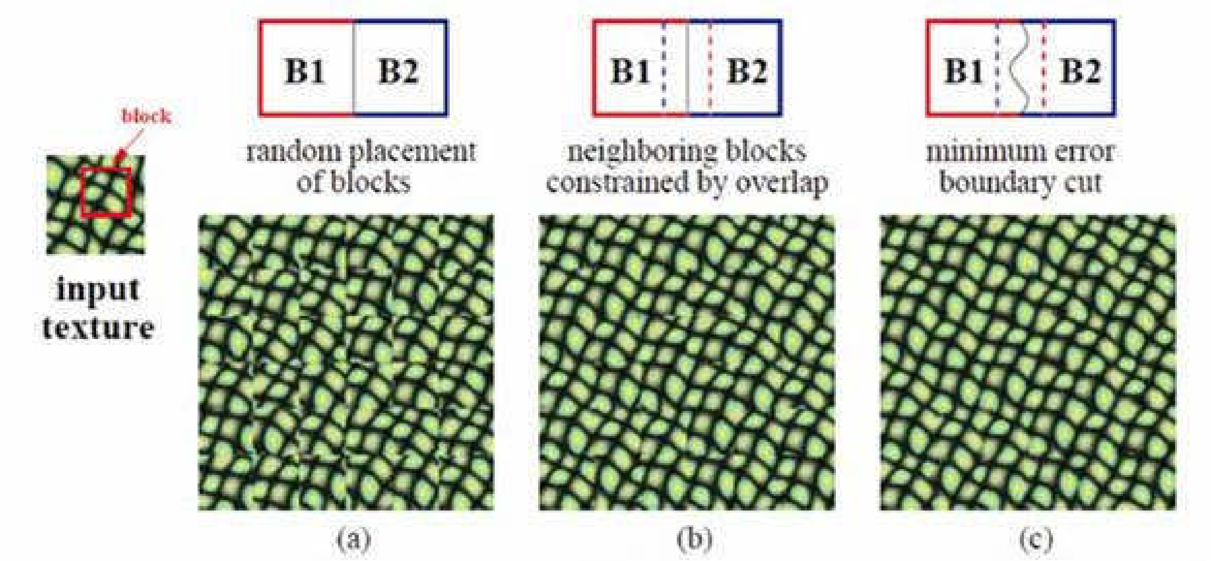

block
input texture
B1
B2
random placement of blocks
B1
B2
neighboring blocks constrained by overlap
B1
B2
minimum error boundary cut
(a)
(b)
(c)

Fig. 3: Image quilting [6].

4.1 Region Subdivision by the Normal Vectors of the Triangular Polygons

The texture on the 3D mesh representing an undevelopable surface is distorted because when the mesh is projected into a plane for texture mapping, if we try not to distort the texture in a specific direction, some distortion must be generated in its perpendicular direction since the Gaussian curvature is not equal to 0, i.e. both the maximum and minimum curvature are not equal to 0. The larger the angle between the normal vector of the mesh and that of the projection plane is, the larger the distortion becomes. The distortion (elongation) of the texture is given by a function of the angle θ as follows:

$$(\frac{1}{\cos\theta}-1)\times 100\% \tag{4.1}$$

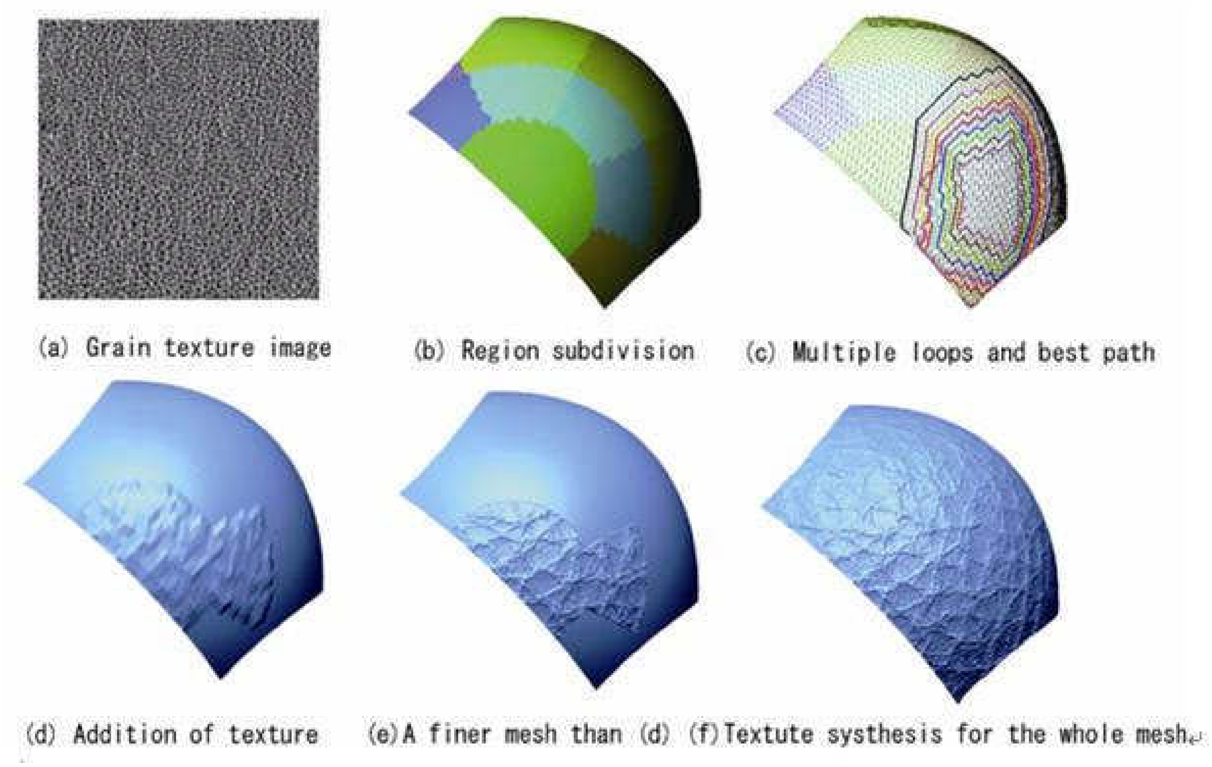
If we take into accounts the distortion in two directions which are perpendicular each other, the maximum distortion is given by multiplying the above function with $\sqrt{2}$.

In order to control the size of the distortion of the texture, as shown in Fig.4(b). the 3D mesh is subdivided into regions accruing to its normal vector and assign a unique ID to the polygons in each region. If the normal is calculated from a CAD model, any problems will seldom occur. However if the normal is calculated from the mesh, we might not obtain normal vectors which are smoothly changing over the mesh even when there are positional differences of a few microns between adjacent two patches. In such a case, it is effective to apply Laplacian smoothing to the normal vector of the mesh.

We correspond a subdivide region with a rectangular region of image quilting. The mesh in Fig.4 is generated from an 1/8 sphere approximated by a B-spline surface and then subdivided into a polygonal mesh and the north pole of the sphere where the z coordinate value is maximum is located at the center of the circular region.

For the mesh, the z axis is supposed to be the direction of ejection for grain texture generation and we subdivide the mesh into 4 portions by limiting the range of the unit normal vector of the Gaussian sphere from 0 degree to 90 degrees of the latitude and dissecting it by 22.5 degrees. We also subdivide the mesh into 8 octants by dissecting the range of the longitude, 360 degrees by 45 degrees. The region surrounding the north pole is regarded as a single region where the maximum distortion of the texture is given by 22.5 degrees when the region is projected onto the x-y plane. Hence from Eqn.(4.1) the elongation in a direction is equal to 8%. For other regions, if the region is projected along to the normal of the central polygon of the region, the maximum error is given by 22.5 degrees. We have not used the 45 degrees for the latitude because we prevent the region to be very long along the z direction.

To distribute errors uniformly on the mesh, it is desirable to subdivide the Gaussian sphere into the regions of the same shape. For example, we had better to adopt HEALPix [14], but we use the uniform subdivision of the two axes of the spherical coordinates for the ease of implementation.



(a) Grain texture image (b) Region subdivision (c) Multiple loops and best path

(d) Addition of texture (e)A finer mesh than (d) (f)Textute systhesis for the whole mesh

Fig. 4: Texture mapping.

4.2 Initial Texture Mapping

It is necessary to have the initial textured region since the texture is synthesized around the initial textured region by comparing the similarity between regions in image quilting. In the extension into 3D, the circular region where the north pole is at the center in Fig.4(b). is used as the initial region and at first, the grain texture image shown in Fig.4(a) is mapped there.

4.3 Processes For Each Region

After the initial texture mapping explained in the previous section, for a region adjacent to the initial region we search the boundary line of the region. As we have given a unique ID for polygons in each region, it is straightforward to find out the boundary line. The most inner loop in blue color in Fig.4(c) is the boundary line of the region next to the initial region.

4.3.1 *Overlapped Region*

To calculate the difference in the overlapped region, it is necessary to define the overlapped region in the 3D extension although the L-shaped overlapped region is used in 2D image quilting. Hence the boundary line is expanded regularly and multiple loops are generated by connecting vertices apart from the boundary line by an edge sequentially.

4.3.2 *Expansion of Boundary Line*

For the expansion of the boundary line, at first the boundary line is smoothed out. If two consecutive edges on the boundary line belong to the same triangle, we substitute the two edges with the other edge of the triangle as the boundary line. It makes simple the process of the expansion of the boundary line. We explain the expansion process using an example shown in Fig.6(a). In this figure, the inside of the region is assumed to be left-bottom. To avoid disconnection and multi-time tracing of the same edge, we replace each edge on the boundary line with the other two sides of the triangle sharing the edge. By this process, both of the end vertices of the vertex remain on the expanded boundary line and no disconnection of the expanded boundary line occurs. Since we have smoothed out the boundary line beforehand, we can avoid a loop around the triangle. Then the expanded boundary line is smoothed out. The processes are repeated to generate multiple loops as shown in Fig.4(c).

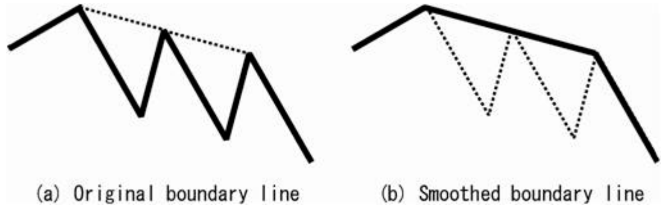
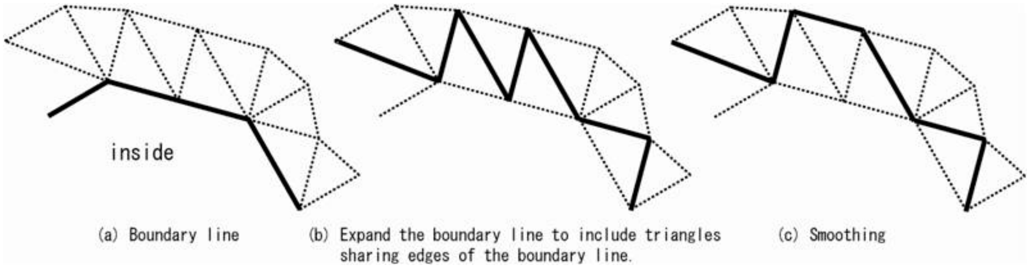


Fig.5: Smoothing of the boundary line.



(a) Boundary line

(b) Expand the boundary line to include triangles sharing edges of the boundary line.

(c) Smoothing

Fig. 6: Expansion of the boundary line.

4.3.3 *Difference of Overlapped Region*

The difference of the pixel values in the overlapped region is calculated by comparing those in the region to be texture-mapped with those in the initial region at the vertices on the multiple loops. To search texture portions the most similar to the pixel values in the initial region, the region is projected along its normal vector into a 2D plane and the texture is subdivided into several blocks to cover the projected region and find the best block which is the most similar to the initial region. We assign the texture block to the region. By this process, the texture is mapped to the region as well as the initial region.

4.3.4 *The Best Path*

The best path suitable for the boundary between the newly textured region and the initial region is generated by using the difference of the pixel values along the multiple loops. Starting from a vertex on each loop, a path is traced on the multiple loops to make a circuit. We assume that the number of the loops is *n*. When we are tracing the i-th loop, the path is traced to the next vertex on the same loop or to a vertex on the (i-1)-th or the (i+1)-th loop sharing the same vertex on the i-th loop. In this tracing method, the tracing does not enter an infinite loop because the path is traced in the ascendant order of the vertices on the multiple loops. When we trace the path, we select a vertex whose difference of the pixel values between the new and initial regions is a minimum among the vertices allowed to be traced. Among the *n*-possible paths, the path with the lowest score is selected as the boundary and the pink line in Fig.4(c) is detected by the process explained here.

4.3.5 Blending Pixel Values along the best Path

A new pixel value is assigned to each vertex of the initial region which exists inside the best path by blending the original pixel value with that of the newly-textured region according to the distance to the best path. The same process is performed to the vertices of the initial region outside the best path, but close to it.

It takes a long time to detect vertices of the initial region outside the best path. In our implementation, m inside loops and m outside loops are excluded from those traced to detect the best path and the vertices on these loops are examined to be candidates for blending.

The result of displacement mapping after the processes explained above is Fig. 4(d). Fig. 4(e) shows the result for a finer mesh (The length of the sides is reduced to 1/10/.). Then we regard the combined region as the initial region and repeat the same processes for other regions. Finally we obtain the fully textured mesh shown in Fig. 4(f).

5 ADDITION OF GRAIN TEXTUTE TO MESH

For a CAD model (a car part), we convert it into a 3D mesh model and add grain texture as shown in Fig.7. Although it includes undevelopable regions, a continuous grain texture is generated. We used an image with 2000×2000 pixels and the distance between two pixels corresponds to 50 μ m on the mesh and hence we adopted 50 μ m for the length of the triangles to generate the mesh. The number of the multiple loops is 21 and *m* is set to be 3 which is the number of the loops untraced for the best path detection. Since there are adjacent patches which have positional differences of a few microns, we performed Laplacian smoothing 10 times for normal vectors. The mesh includes about 1,6 million polygons and 8 million vertices and the size of its file is 812MB (binary STL file. The numbers of the region subdivision are 12 in the latitude and 12 in the longitude. The theoretical elongation is 1.2% maximum when two directions are considered. It took about 70 minutes on Xeon 3.4GHx with 64GB main memory.

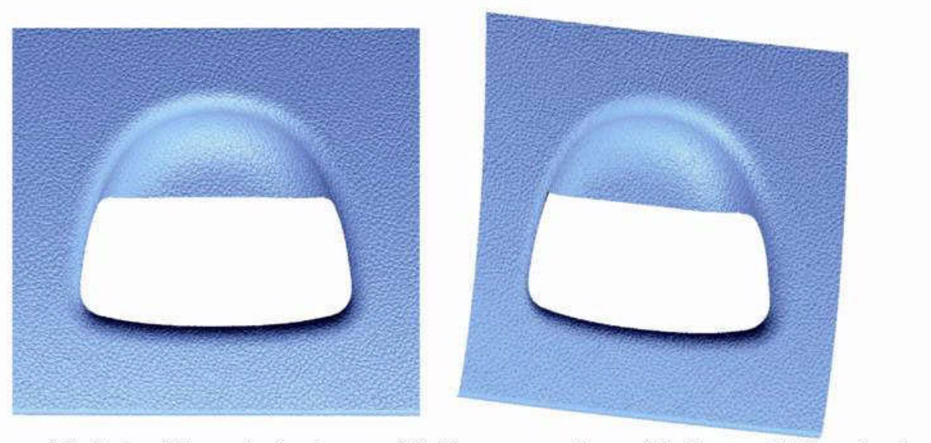

(a) Mesh with grain texture (b) The same mesh as (a) from a different view

Fig. 7: Mesh with grain texture.

6 MILLING EXPERIMENTS

We tried a milling experiment with the digital data which is generated by use of our method. The size of the part is about 153 mm X 147 mm which is a part of car interiors shown in Fig.7. Using a CAM system (Sescoi® WorkNC V19) which can process digital data, we performed 3-axis and 5-axis millings for the part by use of our digital data. For the 3-axis milling, we used a 3-axis milling machine (Fanuc Robodrill) and for the 5-axis milling, a 5-axis machining center (Makino Milling Machine Co. GF-6). Fig. 8 shows regions processed by 3-axis milling in gray and those done by 5-axis milling in yellow and Fig.9 shows the die under and after milling. Fig.10 shows a plastic part manufactured by injection molding using the die.

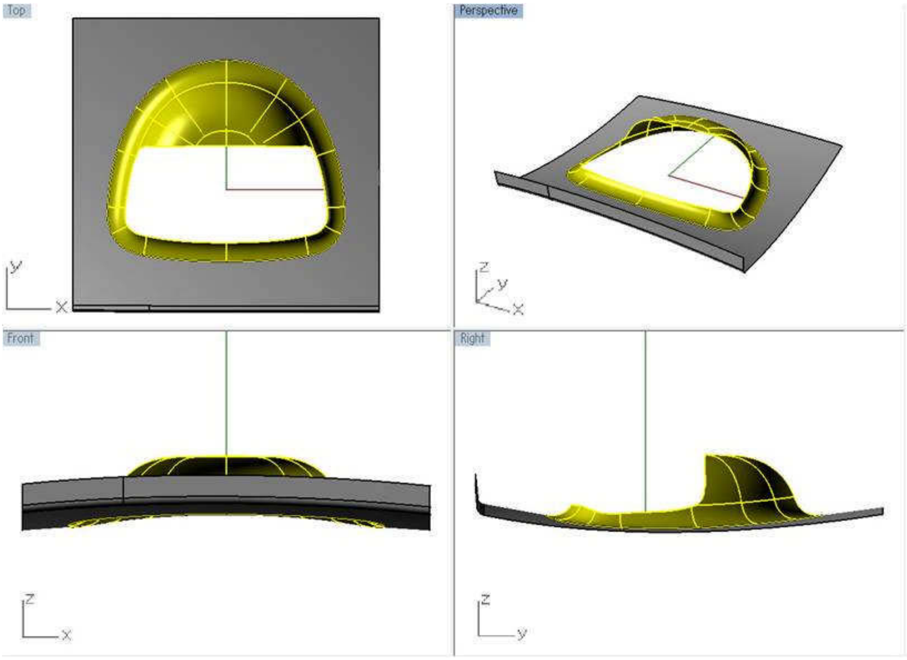


Fig. 8: Regions processed by 3-axis (gray) and 5-axis milling (yellow).

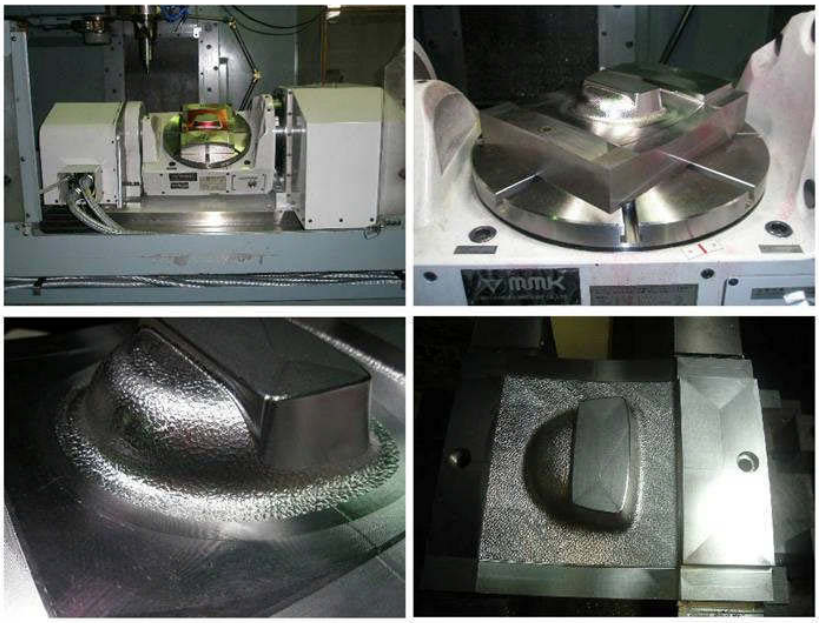
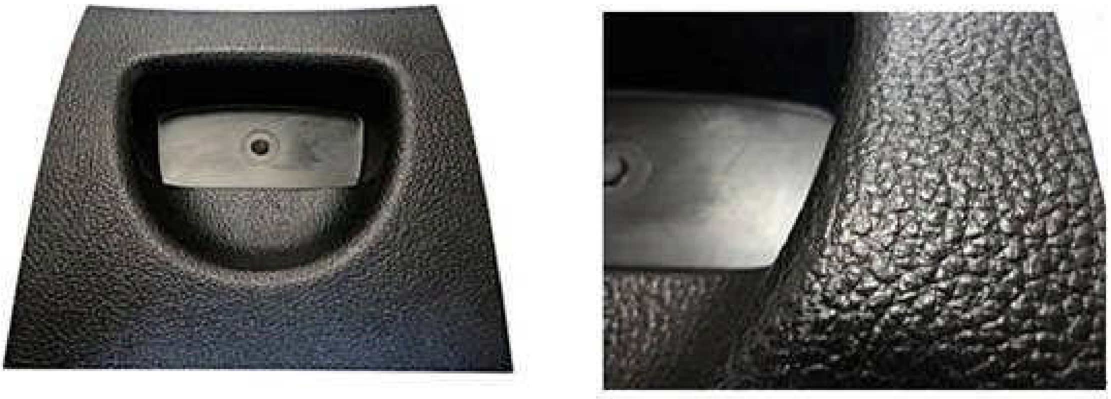


Fig. 9: The die under and after milling.



(a) An injection molded part

(b) A zoom-up

Fig. 10: An injection molded part using the die shown in Fig.9.

6.1 Milling Condition

Since the target model is the automobile interior parts which have complex free form surfaces and the grain need to be generated toward normal direction of the surface, 5 axis milling is needed. It is necessary to investigate whether 5-axis milling has repeatability as well as 3-axis milling. Therefore, 5-axis milling is used by use of the same tools, condition and tool path as 3-axis milling (Tab. 1).

No.	Process	Tool diameter [mm]	Remaining amount [mm]	Tolerance [mm]	Revolution [mm^{-1}]	Cutting speed [mm/min]	Pf [mm]
1	Rough-1	ϕ 1.0	0.03	0.005	50,000	1,000	0.037
2	Middle-1	ϕ 0.4	0.01	0.005	50,000	1,000	0.023
3	Middle-2	ϕ 0.2	0.01	0.005	50,000	1,000	0.016
4	Finishing-1	ϕ 0.1	0.005	0.001	50,000	750	0.025
5	Finishing-2	ϕ 0.1	0.005	0.001	50,000	750	0.025
6	Finishing-3	ϕ 0.1	0.000	0.001	50,000	750	0.025

Tab. 1: Milling condition.

6.2 Geometry Evaluation

Geometries are evaluated by use of the data processing software (INUS Technology Inc. RapidForm). Standard deviation which is outputted from this software is used as quantitative numerical value. Fig. 11 shows the result of accuracy evaluation of the grain digital data, the die and the injection molded product. For the curved surface milled by the 5-axis machine, the accuracy of the geometry is not so good as that of the planar surface. However the accuracy of the die and injection molded product are almost the same for the curved and planar surface. There are almost no difference of repeatability between 3-axis milling and 5-axis milling. The accuracies of several techniques to generate grain texture are summerized in Fig.12. Electroforming technique is the most reproducible method in existing methods. Although the repeatability of our method is inferior to the electroforming technique, there is no huge gap between electroforming technique and our milling-based technique.

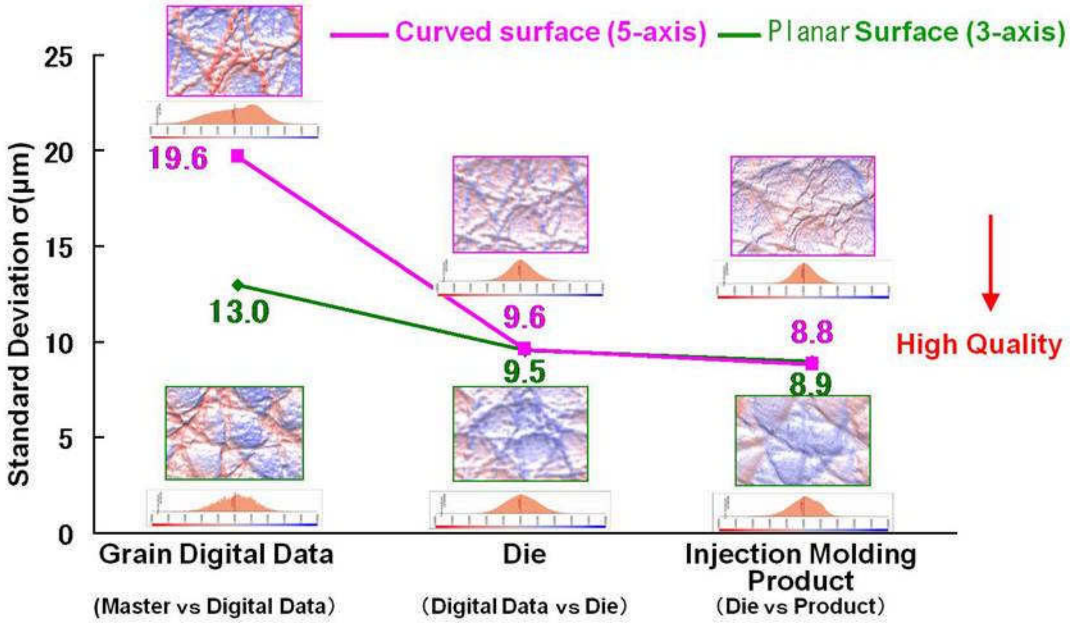


Fig. 11: Accuracies of the curved and planar surfaces of the digital data, the die and the injection molded product.

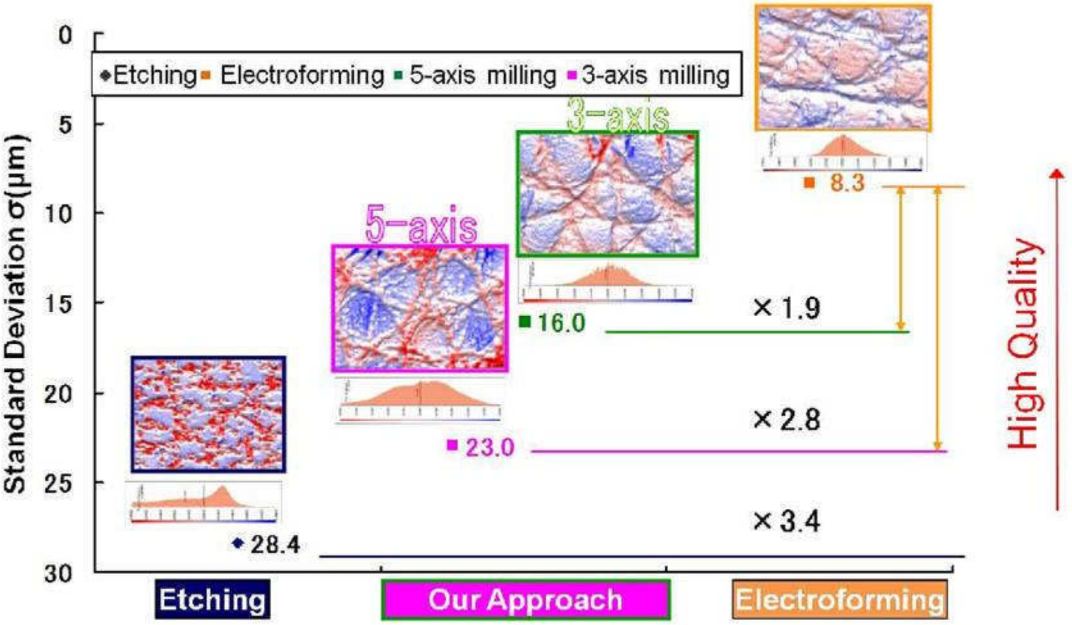


Fig. 12: Comparisons of the repeatability of our and other methods.

7 CONCLUSIONS AND FUTURE WORK

This paper has proposed an extension of image quilting into 3D for the purpose of digital grain texture generation. By using the method, we can map texture even on the mesh including undevelopable free-form surfaces by controlling the elongation of the texture. We have generated mesh data with grain texture by our method and performed displacement mapping based on the texture mapping. We have also machined a die using our mesh data. The level of the die quality is proved to be high enough for practical productions by actually doing injection molding with the die.

We will implement software to be able to handle more complicated and large-scaled 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] Smart Resin Product with Digital Grain, Nikkei Monodukuri, September Issue, 2009.

[3] Nakaishi, M.; Kobayashi, Y.; Shirai, K.: Creation of the wide-are texture which used texture synthesis, SICE Tohoku branch, No.246 meeting, 2008.

[4] Kamibata, J.; Ohtake Y.; Suzuki H.: Shibo Design via Texture Synthesis, Proc. Spring Meeting of Japan Society of Precision Engineering, 2009.

[5] Efros, A.A.; Freeman, W.T.: Image Quilting for Texture Synthesis, Proceedings of SIGGRAPH2001, 2001, 341-346. DOI:10.1145/383259.383296

[6] Efros, A.A.; Leung, T.K.: Texture Synthesis by Nonparametric Sampling, Proceedings of the International Conference on Computer Vision – 2, 1999, 1033.

[7] Turk, G: Texture Synthesis on Surfaces, SIGGRAPH 2001, pp. 347-354. DOI:10.1145/383259.383297

[8] Wei, L.Y; Levoy, M.: Texture synthesis over arbitrary manifold surfaces. SIGGRAPH 2001. DOI:10.1145/965506.965507

[9] Tong, X.; Zhang, J.; Liu, L.; Wang, X.; Guo, B.; Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces, SIGGRAPH 2002. DOI:10.1145/566570.566634

[10] Zhou, K.; Huang, X.; Wang, X.; Tong, Y.; Desbrun, M.; Guo B..; Shum, H.Y.: Mesh Quilting For Geometric Texture Synthesis, SIGGRAPH 2006. DOI:10.1145/1141911.1141942

[11] Heeger, D.J.; Bergen, J.R.: Pyramid-based Texture Analysis/Synthesis, Proceedings of SIGGRAPH95, 1995, 229-238. DOI:10.1109/ICIP.1995.537718

[12] Portilla, J.; Simoncelli, E.P.: Texture Modeling and Synthesis using Joint Statistics of Complex Wavelet Coefficients, Proceedings of the IEEE Workshop on Statistical and Computational Theories of Vision, 1999, 1-32.

[13] Wei, L.Y; Levoy, M.: Fast Texture Synthesis using Tree-structured Vector Quantization, Proceedings of SIGGRAPH2000, 2000, 479-488. DOI:10.1145/344779.345009

[14] Gorski, K.M.; Hivon, E.; Banday, A.J.; Wandelt, B.D.; Hansen, F.K.; Reinecke, M.; Bartelman, M.: HEALPix – A Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere, Astrophysics, 622(2), 2005, 759-771. DOI:10.1086/427976