

Synthesis of Textures with Intricate Geometries using BTF and Large Number of Textured Micropolygons

sub047

Abstract

BTF has been studied extensively and much progress has been done for measurements, compression and rendering so far. However, with regard to representation of materials with intricate geometries such as furs, carpets or towels, many problems still remain. For example, complicated silhouettes of such materials are difficult to represent by BTF. Another problem is synthesis of BTF to arbitrary texture size. In this paper, we propose a method to synthesize and render objects, applying textures of such materials. Small-scaled geometries produce unique visual patterns on actual textures and produce complicated silhouettes. In the method, we draw a large number of small-sized polygons placed on the shape model, with visual patterns and silhouettes represented by alpha-channeled images mapped on the polygons as textures. The images are sampled from small pieces of the materials with various view directions. Using the proposed method, we can synthesize the patterns and silhouettes caused by small-scaled geometries. The final images are produced by compositing the visual patterns with a textures rendered by BTF, which can realize the effects of anisotropic reflections of the texture.

1. Introduction

The modeling and rendering of materials such as furs, carpets and towels which have intricate geometrical structures are challenging and important problems in the field of computer graphics. Since such complex geometries are difficult to model with polygon meshes, many methods have been researched. Although many effective methods such as using particles have been proposed, most of them still require high skills and a lot of time. Moreover, such materials tends to have complex, anisotropic reflection properties making the problem more difficult.

One potential solution for this purpose is sampling BTF from real materials. However, there remains some problems. Sampling BTF requires a special sampling equipment, and sampled BTF data tend to be enormous even if compressed by recently proposed methods. Moreover, it cannot represent complex silhouettes caused by intricate geometries of such materials; note that intricate geometries produce not only complex silhouettes, but also many other

visual effects such as hair strand patterns, etc. Another typical problem is seamless tiling of textures.

In this paper, a novel method for rendering arbitrary shaped objects, applying materials with intricate geometries such as furs are proposed. Normally, intricate geometries cause unique visual patterns to their appearances. For example, furs exhibits numerous stripe patterns along the direction of the hair strands. In the proposed method, visual patterns caused by intricate geometries are synthesized as a collection of small images sampled from a piece of the material. The visual patterns are combined with another image which is synthesized using low-pass filtered BTF, which expresses the effects of reflectance such as shadings or speculars. The advantages of the proposed method are: (1) since the method is based on sampling, modeling of intricate geometries is not necessary, (2) by using low-pass filtered BTF, quantity of the texture data is greatly reduced, (3) by using low-pass filtered BTF, anisotropic reflection properties can be synthesized, and (4) visual patterns and effects of silhouettes caused by intricate geometries can be rendered.

2. Related studies

Rendering and modeling fur, knit fabrics, hair and other intricate geometry and small structures of objects has been investigated extensively [?, ?, ?]. Lengyel et al. [?] used shell texture to render volumetric fur and fins to render the silhouettes of fur in real time.

Relatively small geometry, such as the surface of a carpet or clothing, is often represented using the bi-directional texture function (BTF). Since BTFs are effective to render materials realistically, efficient representations and rendering techniques have been intensively studied, such as polynomial texture maps (PTM) [?], Spherical harmonics (SH) representation [?], Precompiled radiance transfer (PRT)[?], bi-scale radiance transfer (BRT) [?]. or factorization-based methods [?].

Point-based approaches have also been applied in to model and render such mesoscopic geometry. For example, for the rendering of trees, Meyer [?] rendered complex repetitive scenes using a volume renderer with view-dependent texture, which switches textures according to the view direction. Max [?] used layered depth images to ren-

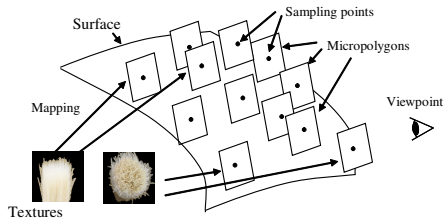


Figure 1: Representation of small-scaled geometry.

der trees. Yamazaki et al. [?] used billboarding microfacets textured with real images to render fur. Reche [?] extended Yamazaki’s method and successfully rendered trees from a number of images.

Our method to render visual patterns caused by complex geometries as a collection of small images is inspired by the Yamazaki’s method. Their work and ours have different purposes; the purpose of their work is to replay the appearance of the sampled object as it is, and the purpose of ours is to apply sampled material to arbitrary surface model.

3. Overview of rendering steps

In our method, two different types of rendering are applied to express visual effects caused by different scales of geometries. The effects caused by the smaller scaled geometries are visual patterns of intricate geometries of the material. These patterns are expressed by a large number of small images arranged at a point set sampled from the surface of the rendered shape. The effects caused by the larger scaled geometries are shadings and speculars, which are changed by the directions of the surface of the model, the light directions and the view direction. These effects are expressed by low-resolution BTFs, which are BTFs processed by a low-pass filter about texture coordinates. Final images are synthesized by compositing those two kind of images.

First, the method to synthesize visual effects of intricate geometries are explained. As already mentioned, the effects are rendered by drawing collection of images. These images are appearances of a piece of the material called a “material unit,” which can be seen as an unit of the visual patterns. To express overlapping effects between the patterns (for example, adjacent strands of a fur look overlapped each other, causing self-occlusions), these arranged images include the alpha channel which represents the transparency of each pixel. By the overlapping images with effects of transparency, the appearance of the surface covered by the intricate geometries are expressed. In this paper, drawing of the images are realized by drawing a large number of textured, small-sized polygons (micropolygons). Polygons are located at a set of points, sampled from the rendered surface. Those points are called “sampling points.” Figure 1

shows the basic idea of the method.

Complex silhouettes caused by intricate geometries are also synthesized with the same process. Since the transparency is applied, the geometrical silhouettes are rendered as the border of the patterns in the images drawn near the edges of the rendered shape.

Then, the effects of shadings and speculars which exhibits anisotropic reflection properties are rendered independently. It can be achieved by applying low-pass filtering about texture coordinates on BTFs. The shape model is rendered for designated view and the lighting conditions by applying the same method proposed by Furukawa et al.[?]. In the rendered image, the effects of the shadings and the speculars are represented.

At the final stage of rendering, the visual patterns and the result of BTF rendering are composited. For the composition method, several formulation can be considered. One simple method is just multiplying the two rendering results. Another effective method is more complicated is as follows; the BTF rendering is done first, and, when the micropolygons are drawn to express visual patterns, the color of each micropolygon is modulated by using the result of BTF rendering.

4. Implementation

4.1. Sampling appearances of intricate geometry

As mentioned previously, we take samples of images of a small piece of material called a material unit, and synthesize the visual patterns of small geometries appearances from the images. A material unit is a small piece of material that has the same geometrical structures as the material to be rendered. The surface model is synthesized as a collection of numerous material units. For some anisotropic materials, the material unit may be “tilted” as shown in figure 2. In such cases, the material unit occupies a region of a tilted column in the volume of the intricate geometries. To express this property, tilt direction is defined as shown in the figure. In the case of furs, the tilt direction is the direction of strands.

Sample images of the material unit can be made from real pictures or rendered synthetically. When the sample images are prepared from real pictures, images of a small piece of material are sampled while varying the view direction. For anisotropic materials, sampling for the entire hemisphere of the view direction is required. However, for many materials of furs, rotation around the tilt direction does not change the appearance much. In such cases, we could reduce the freedom of rotation around the tilt direction as a simple and useful approximation,

From the sampled images, the background of the image is set to be transparent. The opaqueness of each pixel is

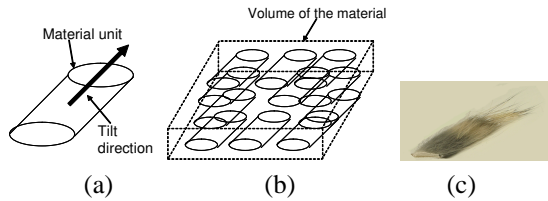


Figure 2: Tilted material unit: (a) a material unit and its tilt direction, (b) material units arranged in a volume of the material, (c) an example of a tilted material unit.

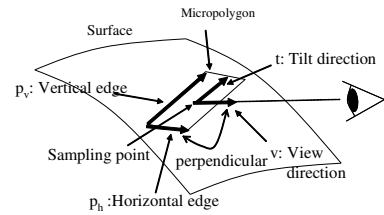


Figure 4: Direction of micropolygons.

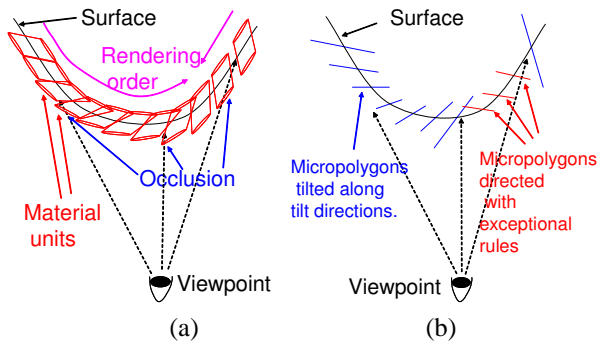


Figure 3: Order-free drawing: (a) rendering order, (b) order-free rendering using tilted micropolygons.

stored in the alpha channel. If the images have colors, they are converted to gray-scaled images.

4.2. Self-occlusions of intricate geometry

The center of each micropolygon is fixed at a point on the surface, which we call a “surface point.” The surface points are uniformly sampled from the surface model. Each micropolygons are texture-mapped by images with an alpha channel expressing opacity.

The most common method to render such scenes would be alpha-blending. However, to apply alpha-blending, overlapping micropolygons should be rendered in an appropriate order. Suppose two micropolygons corresponding to two adjacent pieces of intricate geometries. If one of the pieces occlude the other, the micropolygon expressing the occluded one should be rendered earlier.

This problem is not trivial. Suppose the case that we use fur as the material, and strands in the fur are tilted from the normal directions, which is a common case about furs. As the examples in figure 3(a) show, the order to render the micropolygons does not only depend on the distances from the viewpoint, but also by the tilts of the material units. This makes the decision of the rendering order difficult.

To cope with this problem, we use a rendering method which does not depend on order of rendering, realized

by combining simplified process of transparency and z-buffering. The basic idea of this method is shown in figure 3(b). As shown in the figure, each micropolygons are tilted along the tilt directions of the corresponding material unit. Because of these tilts, the order of occlusions of the geometries observed from the viewer for a fixed line of sight coincides with the z-values of rasterized pixels of the overlapping micropolygons. Thus, if both per-texel z-buffering and per-texel transparency could be used, order-free rendering would be realized. Although most of today’s rendering hardware does not support per-texel z-buffering, this function can be realized using pixel shader programming, which is explained in the next subsection.

Note that, in figure 3(b), the sizes of the micropolygons are spread, depending on their tilt directions. The purpose for this spreading is to compensate the changes in apparent sizes of the micropolygons seen from the viewer, which occurs depending on the tilt directions. Also note that there are several exceptional micropolygons which do not tilted along their tilt directions. These exception occurs when the tilt direction and the view direction becomes nearly parallel. These issues are discussed in section 4.3.

4.3. Creation of micropolygons

Figure 4 shows how micropolygons are directed. Creation of micropolygons are performed for each sampling point as follows. First, the tilt direction \mathbf{t} (a unit vector) is calculated. The vertical edge of the micropolygon \mathbf{p}_v is parallel with \mathbf{t} . The horizontal edge of the micropolygon \mathbf{p}_h is defined so that it is perpendicular to both \mathbf{t} and the view direction \mathbf{v} (a unit vector directed from the sampling point to the viewpoint). The real length of the vertical edge is changed so that the apparent vertical edge length seen from the viewer is constant length L . Then, $\mathbf{p}_h = N(\mathbf{t} \times \mathbf{v})$, $\mathbf{p}_v = \mathbf{t}/|\cos \phi|$, where N is an operator that normalizes a vector and ϕ is the angle between \mathbf{v} and \mathbf{t} .

Using the above form, vertical edge length approaches to the infinity when ϕ approaches to zero, which causes the algorithm unstable. When the angle ϕ is small, the effects of self-occlusion caused by intricate geometries are small in the situation, since the viewer sees the material from

the tilt direction. Considering this fact we apply another method to create the micropolygon when $|\cos \phi|$ is less than a threshold value. The second rule is, $\mathbf{p}_h = N(\mathbf{n} \times \mathbf{v})$, $\mathbf{p}_v = N(\mathbf{v} \times \mathbf{p}_h)$, where n is the outward normal direction of the surface at the sampling point. This rule means that the micropolygon is created so that it is perpendicular to the view direction \mathbf{v} and the apparent direction of the normal \mathbf{n} and the vertical edge \mathbf{p}_v is the same.

The sampled images for the material unit are used as the textures of the micropolygons to represent visual effects of small-scaled geometries. Selection of the texture mapped onto a micropolygon depends on the view direction expressed in the tangent space at the surface point. If the material is isotropic, the image sampled with the nearest latitude angle as the view direction is selected. If the material is anisotropic, the image sampled with the nearest direction in the hemisphere as the view direction is selected.

4.4. Per-texel transparency using pixel shaders

Most of today's rendering hardware does not support per-texel z-buffering as it is. Even using today's pixel shaders, full-fledged alpha-blending and z-buffering cannot be implemented, since the frame buffer are not accessible from the pixel shaders. However, by giving up on general alpha-blending, and by simplifying the problem to the case that all the texels are either completely transparency or completely opacity, then, the rendering process can be programmed by pixel shaders.

The concrete method is as the following. The micropolygons are rendered as normally done with the z-buffering on. In the pixel shader program, the alpha value of the texel is compared with a fixed threshold, for example 0.5. If the alpha value of the texel is less than the threshold, the drawing of the pixel is canceled by 'cancel' instruction, otherwise, the color of the pixel is updated using the texel value. Since the canceled pixel does not affect neither z-buffer or frame-buffer, per-texel z-buffering and per-texel transparency are realized.

In the above method, all the texels are processed as completely opacity or transparency. Since blending at the borders between the overlapping micropolygons are not performed, those borders does not look natural. To alleviate this problem, anti-aliasing the border by multi-path rendering is effective.

This process can be done as the following. First, rendering is processed several times, changing the threshold of alpha value to decide opacity or transparency (For example, rendering three times, changing the threshold to 0.4, 0.5 and 0.6.). Then, the result image is synthesized by averaging those rendered images for each pixel.

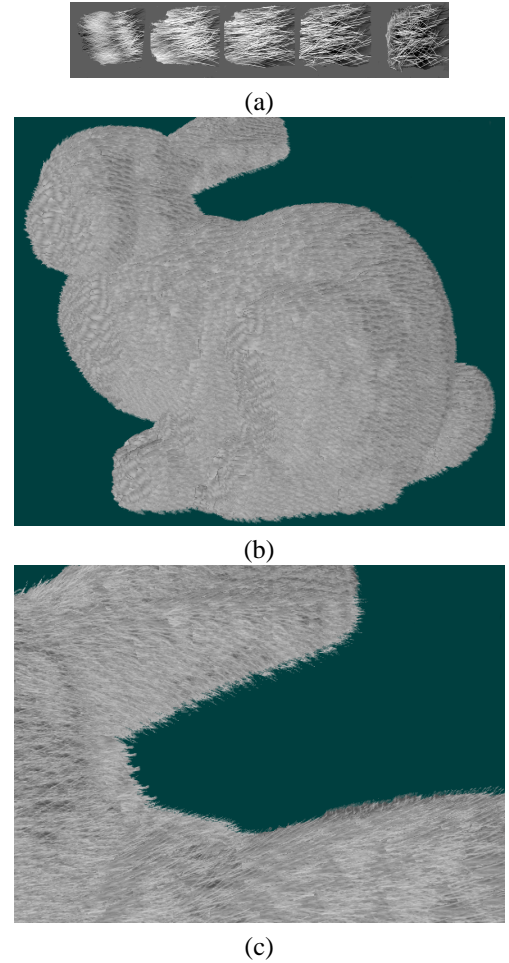


Figure 5: Synthesized images using micropolygons(fur): (a) sampled images of a material unit, (b),(c) synthesized images of visual patterns

4.5. Image composition of BTF and effects caused by intricate geometry

The synthesized visual patterns of intricate geometries are composed with the rendering results of low-pass filtered BTF, which represents shadings and speculars. An example of rendering result by the low-pass filtered BTF is shown in figure 6.

As the composition methods of the rendering results of low-pass filtered BTF and generated visual patterns, we tried two methods. The first method is simple, per-pixel multiplication. The other method is to per-image multiplication, in which multiplication is performed between images of material units and the value of BTF at their sampling points (which are placed on the surface).

Examples of the results by the first method, per-pixel multiplication, are shown in figure 7 Depending on the tex-

tures used for micropolygons, the silhouette synthesized using micropolygons is expanded from the original positions of the border of the surface. In this case, rendering using BTF with offset surface produces better results.

The second method, per-image multiplication, is performed as follows: the rendering result using BTF is synthesized first, and the image is used as an input texture to the rendering process of the micropolygons. The pixel shader look up the BTF synthesized image at the sampling point of the micropolygon (not at the position of the rasterized pixel), and multiply the pixel values from the image synthesized with BTF and from the image of the material unit. Examples of the results by the second method are shown in figure 8.

The two methods has both advantages and disadvantages. Since the per-image multiplication uses the value of the BTF at the sampling point on the surface, it is not necessary to offset the surface for BTF-based rendering using per-image multiplication. Moreover, per-image multiplication can process occluding boundaries of the surface without problem, whereas per-pixel multiplication may not correctly process the occluding boundaries. However, since the multiplication is done for each images composing the visual patterns, the shadings and speculars in the result of per-image multiplication look less smooth than those of per-pixel multiplication.

5. Results

Rendering experiments were performed for the proposed method. For the experiments, a material unit was taken from a bunny's fur. The low-pass filtered BTF was also sampled from the same material.

Figure 5 shows the synthesized images of a bunny using micropolygons. For the material unit for the synthesis, we used a strand of fur. 5 sample images were used as the samples of the material unit. The angles between the tilt direction and the view directions of the samples were 160° , 130° , 90° , 40° and 10° . The number of micropolygons were 10230, which are uniformly sampled. The sizes of the micropolygons were about 0.04 times the full length of the bunny model. The tilt-directions of the sampling points were directed toward the tail of the bunny. And the tilt-directions were elevated by 30° from the surface of the model. Anti-aliasing was performed with thresholds of alpha being 0.4, 0.5 and 0.6.

From figure 5 (b), we can see that the visual patterns caused by strands of fur are appropriately rendered. The silhouette of the fur was also rendered naturally. Figure 6 shows the rendering result of low-pass filtered BTF.

Figure 7 shows the result of composition by per-pixel multiplication, using the BTF values from the images shown in 6. The effect of shading rendered by BTF and the



Figure 6: Synthesized images using low-pass filtered BTF.

visual effect synthesized by micropolygons were combined into the images. Figure 8 shows the result of composition by per-image multiplication.

The CPU used for the rendering was Xeon 3.40GHz with 1.00GB of memory. The video hardware was Quadro FX 1300 with 128MB of video memory. The overall rendering was performed by 2 steps. Rendering based on low-pass filtered BTF was processed separately. Then, the synthesis of the visual patterns and the composition of the final results was processed at the same time. The rendering performance of the visual patterns including the composition of the final results was about 2 fps.

6. Conclusion

We proposed a method to synthesize objects with small-scaled geometries such as furs, carpets and towels. Since small-scaled geometries produce a complex silhouette and unique visual effects on its surface, it is difficult to synthesize them by common rendering techniques.

To compensate the problem, we apply two different types of rendering method for different scales of geometries. Final results are produced by compositing them. The first method is for small-scaled geometries using actually sampled images from real objects and a large number of micropolygons to produce visual patterns of small geometries. The second method is for large scale geometries, which utilizes low-pass filtered BTFs to synthesize shading, specular and other view dependent effects.

We also proposed a hardware implementation of our method using vertex and pixel shader, which achieves real-time rendering.

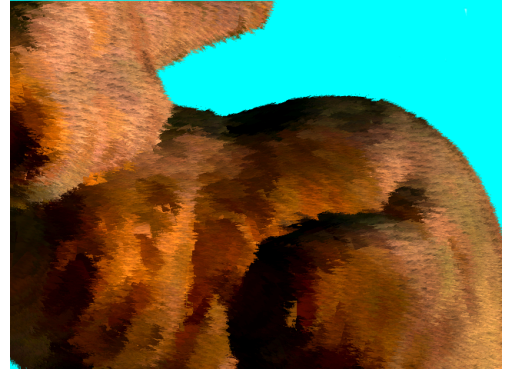


(a)

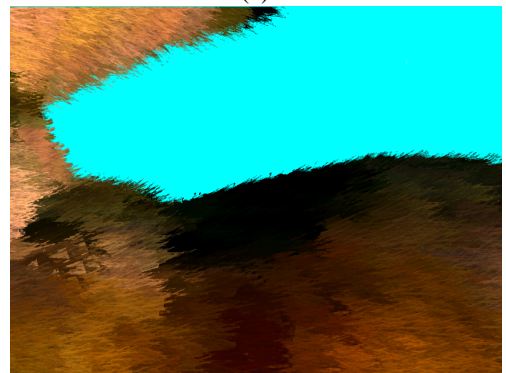


(b)

Figure 7: Results of composition by per-pixel multiplication.



(a)



(b)

Figure 8: Results of composition by per-image multiplication.