

Plausible and Realtime Rendering of Scratched Metal by Deforming MDF of Normal Mapped Anisotropic Surface

Young-Min Kang
Tongmyong University
ymkang@tu.ac.kr

Hwan-Gue Cho
Pusan National University
hgcho@pusan.ac.kr

Sung-Soo Kim
ETRI
sungsoo@etri.re.kr

ABSTRACT

An effective method to render realistic metallic surface in realtime application is proposed. The proposed method perturbs the normal vectors on the metallic surface to represent small scratches. General approach to the normal vector perturbation is to use bump map or normal map. However, the bumps generated with those methods do not show plausible reflectance when the surface is modeled with a microfacet-based anisotropic BRDF. Because the microfacet-based anisotropic BRDFs are generally employed in order to express metallic surface, the limitation of the simple normal mapping or other normal vector perturbation techniques make it difficult to render realistic metallic object with various scratches. The proposed method employs not only normal perturbation but also deformation of the microfacet distribution function (MDF) that determines the reflectance properties on the surface. The MDF deformation enables more realistic rendering of metallic surface. The proposed method can be easily implemented with GPU programs, and works well in realtime environments.

Keywords: Realtime rendering, anisotropic reflectance, metal rendering, MDF deformation

1 INTRODUCTION

In this paper, we propose a procedural method that efficiently renders plausible metallic surfaces as shown in Fig.1. Anisotropic reflectance models have been widely employed to represent the metallic surface. However, realistic representation of small scratches shown in Fig.1 were not main concern of those methods.

Torrance and Sparrow proposed microfacet-based rendering model where the surface to be rendered was assumed as a collection of very small facets[12]. Each facet has its own orientation and reflects like a mirror. The reflectance property of this surface model is determined by microfacet distribution function(MDF).

Many researchers improved the microfacet-based rendering model to represent various materials. Methods that can control the roughness of the surface were introduced[4, 3], and those methods were also improved by Cook and Torrence[5].

A smooth metallic surface reflects the environments like a mirror. However, the most metal objects have brushed scratches or random scratches. These scratches make the reflectance on an actual metallic surface different from that on the perfect mirror surface. The peculiar reflectance on metallic surface is determined by the direction of the scratches, and



Figure 1: Realtime rendering with proposed method.

in most cases, has anisotropic appearance. There have been various techniques for representing the anisotropic reflectance[8, 14, 11].

Ashikhmin and Shirley proposed an anisotropic reflection model with intuitive control parameters[1, 2]. Their model is successfully utilized to express the surface with brushed scratches.

Wang *et al.* proposed a method that approximates the measured BRDF(bidirectional reflectance distribution function) with multiple spherical lobes[13]. Although this method is capable of reproduce various materials including metallic surface, it has a serious disadvantage in that expensive measured BRDF is required. Moreover, it is still impossible to accurately render small scratches and light scattering with camera close up to the surface.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Although there have been many approaches to representation of metallic surface [15], relatively little attention has been given to the representation of the small scratches on the surface and the reflectance disturbance caused by the scratches. In most cases, only the reflectance anisotropy caused by the scratches was modeled. An efficient and accurate computation of specular reflection has been also introduced for realtime applications[9]. However, it cannot be applied to normal mapped surface because the method is based on vertex geometry.

In this paper, we propose a procedural method that does not require any measured data. The proposed method efficiently and plausibly renders the small scratches and its light scattering on anisotropic reflectance surfaces.

2 REALISTIC METAL RENDERING

In this section, a procedural approach to metallic surface rendering is proposed. The proposed method is based on microfacet model, and the small scratches on the surface are represented with normal vector perturbation. In order to increase the realism, we also deform the MDF according to the perturbation of the normal vector.

2.1 MDF for Anisotropic Reflectance

The reflectance property of microfacet-based surface model is determined by the microfacet distribution function(MDF) $D(\omega_h)$ which gives the probability that a microfacet is oriented to the direction ω_h . Ashikhmin *et al.* proposed an anisotropic reflectance model with the following MDF:

$$D(\omega_h) = \frac{\sqrt{(e_x + 1)(e_y + 1)}}{2\pi} (\omega_h \cdot \mathbf{n})^{e_x \cos^2 \phi + e_y \sin^2 \phi} \quad (1)$$

, where \mathbf{n} is the normal vector at the point to be rendered. The actual parameter ω_h in the MDF is the half way vector between the incident light direction and outgoing viewing direction. e_x and e_y are parameters that control the anisotropy of the reflection, and ϕ is the azimuthal angle. ω_h is a unit vector which is sufficiently represented with only two components as $(\omega_h.x, \omega_h.y, \sqrt{1 - \omega_h.x^2 - \omega_h.y^2})$. Therefore, the MDF is also defined in 2D space as shown in Fig.2.

Fig.2 shows an example of anisotropic MDF using Eq.1 with different e_x and e_y . As shown in Fig.2, the incoming light energy is scattered differently in x (tangent) and y (binormal) axes of tangent space. Such anisotropic reflectance is appropriate for metal rendering. In this paper, we assume that metallic surfaces reflect light energy according to the anisotropic model described in Eq.1

Fig.3 shows the rendering results by changing the parameters e_x and e_y of Eq.1. As shown in the figure, the

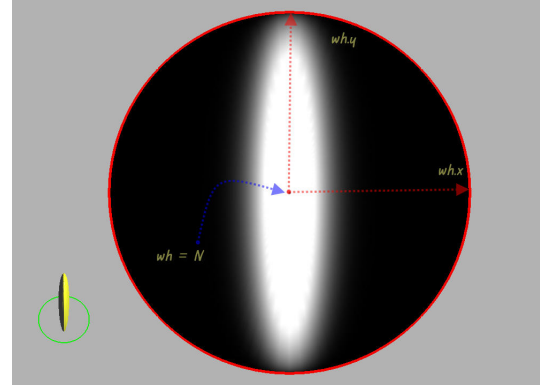


Figure 2: MDF in 2D space



(a) $e_x, e_y : 20, 20$ (b) $e_x, e_y : 200, 10$ (c) $e_x, e_y : 10, 200$

Figure 3: Surfaces rendered with Eq.1: (a) isotropic, (b)&(c) anisotropic reflectance.

anisotropic reflectance on metallic surface can be easily controlled. However, this method is not capable of capturing the small scratches and the light scattering in details when the camera is moved close to the surface. A simple approach to this problem is to perturb the normal vectors on the surface, but the perturbed normal vectors on anisotropic reflection surface may introduce another problem. The limitation of simple normal perturbation is described in the next subsection.

2.2 Limitation of Normal Perturbation

There have been continuous efforts to represent higher geometric complexity with simple mesh by perturbing the normal vectors[10, 6, 7]. Bump mapping is well known in graphics literature, normal mapping is an improved method which does not compute normal vectors during the rendering phase[10].

In this paper, we are interested in representing the light scattering by the small scratches on the anisotropic reflection surface. In order to represent the scratches we employed the well-known normal map approach. Fig.4 shows the scratch maps (essentially normal maps), and the expected rendering results. The scratch maps are seamless textures and procedurally generated.

Heidrich and Seidel applied Blinn-Phong shading to the normal mapped geometry[6]. Their method is successful only when the reflection is isotropic. However, the normal mapping on anisotropic reflection surface, unfortunately, cannot reproduce the original anisotropic reflectance on the distorted surface. Other normal perturbation methods such as displacement mapping also suffer from the same problem. Fig.5 shows the un-

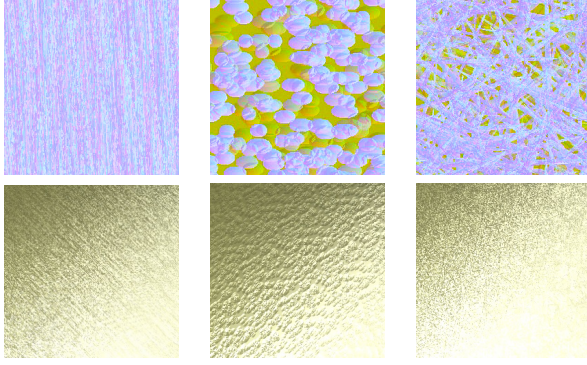


Figure 4: Scratch maps and expected rendering results: (top row) scratch maps and (bottom row) expected results.

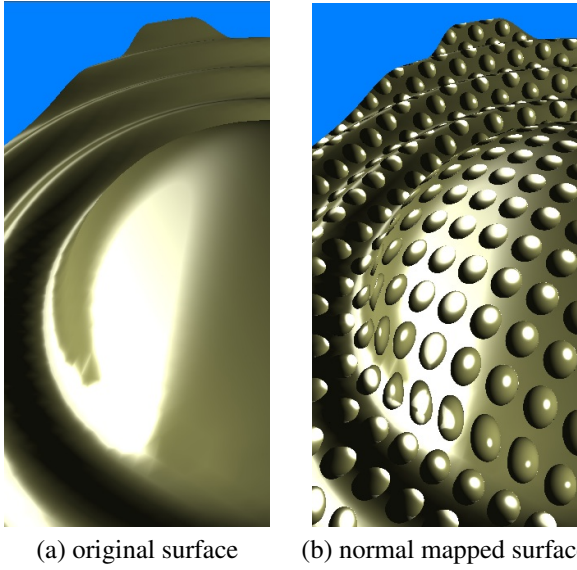


Figure 5: Normal vector perturbation on an anisotropic reflection surface: (a) original surface and (b) normal mapped surface.

satisfactory rendering results when the simple normal mapping is applied to an anisotropic reflection surface with MDF function shown in Eq.1. As shown in the figure, the anisotropic reflectance on the original surface (a) is not preserved in the normal mapped surface (b). The reflectance on the area where normal vectors are perturbed is rather isotropic. Moreover we can observe some artifacts that specular reflection is severely distorted at the left lower region.

The problem shown in Fig.5 is because the normal mapping or other normal vector perturbation methods only change the normal vector \mathbf{n} . However, the MDF $D(\omega_h)$ is dependent not only on \mathbf{n} but also on ω_h . In Eq.1, the only argument was ω_h because the normal vector is constant in tangent space. However, the normal vector should be another argument when normal perturbation is applied. Let us denote the perturbed normal vector as $\tilde{\mathbf{n}}$. The MDF can then be rewritten as follows:

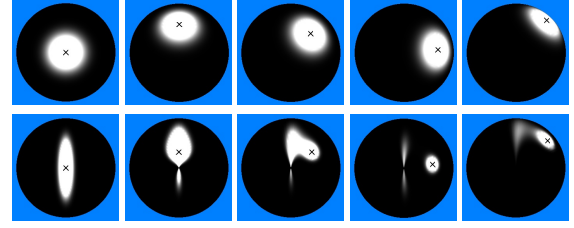


Figure 6: MDF with perturbed normal vectors: (top row) perturbation with isotropic MDF and (bottom row) perturbation with anisotropic MDF.

$$D(\omega_h, \tilde{\mathbf{n}}) = \frac{\sqrt{(e_x + 1)(e_y + 1)}}{2\pi} (\omega_h \cdot \tilde{\mathbf{n}})^{e_x \cos^2 \phi + e_y \sin^2 \phi} \quad (2)$$

Heidrich and Seidel computed the dot product of half way vector and the perturbed normal vector to calculate the specular reflection on the normal mapped surface. Eq.2 also computes the dot product. However, this method does not work well for anisotropic reflection surface. Fig.6 shows the MDF computed with Eq.2 and perturbed normal vectors. The cross mark in the figure indicates the perturbed normal. The top row of Fig.6 shows isotropic MDF when the normal vector is perturbed. As shown in the figure, Eq.2 produces reasonable deformed MDF for the isotropic MDF. However, the simple normal perturbation is not successful with anisotropic MDFs. The bottom row of fig.6 shows the results when we employed an anisotropic MDF. The results show that simple normal perturbation approach is hopelessly unsuccessful to preserve the original reflection property.

2.3 MDF Deformation

In order to overcome the limitation of the simple normal mapping on anisotropic reflection surface, the MDF should be properly deformed with the original anisotropic property maintained. Fig.7 shows the MDF deformation concept. Fig.7 (a) shows an example of anisotropic MDF, and (c) shows the deformed MDF in accordance with the normal vector perturbation amount of $(\Delta x, \Delta y)$ in tangent space. Let us denote the deformed MDF as $D'(\omega_h)$. We can easily derive $D'(\omega_h)$ with the deformation concept shown in Fig.7 (b). A certain point \mathbf{p} in the domain of the original MDF $D(\omega_h)$ must move to another location \mathbf{p}' in the domain of the deformed MDF $D'(\omega_h)$. The direction and magnitude of the movement are determined by the movement from the center of the original MDF space (\mathbf{C}) to that of the deformed MDF space (\mathbf{C}'). The movement of the center is in fact the perturbation of the normal vector, and can be denoted as $(\Delta x, \Delta y)$. Let us denote the transformation that move a point from \mathbf{p} to \mathbf{p}' in accordance with the normal perturbation $(\Delta x, \Delta y)$ as $\mathcal{T}(\mathbf{p}, \Delta x, \Delta y)$. The transformation $\mathcal{T}(\mathbf{p}, \Delta x, \Delta y)$ can be easily derived with \mathbf{R} , the intersection of the

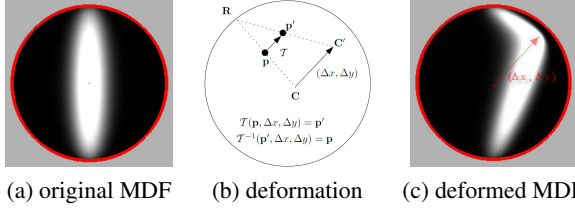


Figure 7: MDF deformation concept and corresponding points.

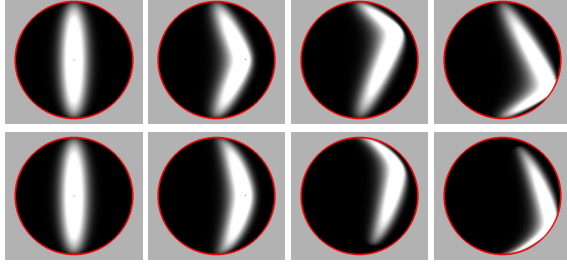


Figure 8: MDF deformation examples: (top row) linear interpolation results and (bottom row) smooth interpolation results.

circumference of the MDF space and the ray from the center through the point \mathbf{p} .

The simple approach shown in Fig.7 move the point \mathbf{p} in the same direction with the center movement, and the magnitude of the movement is linearly interpolated. Therefore, the transformation can be expressed as follows:

$$\mathcal{T}(\mathbf{p}, \Delta x, \Delta y) = \mathbf{p} + \frac{|\vec{\mathbf{R}}\mathbf{p}|}{|\vec{\mathbf{R}}\mathbf{C}|}(\Delta x, \Delta y) \quad (3)$$

Although the transformation shown in Eq.3 deforms the MDF in accordance with the normal vector perturbation, the bending of the deformed anisotropic reflectance is excessive at the moved center as shown in Fig.7 (c). In order to obtain more smooth interpolation, we used the following transformation:

$$\mathcal{T}(\mathbf{p}, \Delta x, \Delta y) = \mathbf{p} + \sqrt{\frac{|\vec{\mathbf{R}}\mathbf{p}|}{|\vec{\mathbf{R}}\mathbf{C}|}}(\Delta x, \Delta y) \quad (4)$$

Fig.8 compares the MDF deformation results with the linear (Eq.3) and the smooth (Eq.4) interpolations. The top row shows the linear version while the bottom row shows the smooth version. As shown in the figure, the smooth interpolation version looks more natural.

It is obvious that computing the deformed MDF at each sampling point on the surface is extremely inefficient. Explicit deformation of the MDF is only conceptual process. In the actual rendering process, we never compute $D'(\omega_h)$. Only the original MDF $D(\omega_h)$ is used with the inverse transformation $\mathcal{T}^{-1}(\mathbf{p}', \Delta x, \Delta y)$. In other words, we conceptually

employ $D'(\omega_h)$ for the normal mapped surface, but actually use $D(\mathcal{T}^{-1}(\omega_h, \Delta x, \Delta y))$ which has the equivalent value.

The inverse transformation of Eq.4 can be easily obtained as follows:

$$\mathcal{T}^{-1}(\mathbf{p}', \Delta x, \Delta y) = \mathbf{p}' - \sqrt{\frac{|\vec{\mathbf{R}}\mathbf{p}'|}{|\vec{\mathbf{R}}\mathbf{C}'|}}(\Delta x, \Delta y) \quad (5)$$

Now we can simply calculate $D(\mathcal{T}^{-1}(\omega_h, \Delta x, \Delta y))$ to compute the MDF at the point where the normal vector is perturbed with $(\Delta x, \Delta y)$. Because Δx and Δy are the x and y components of the perturbed normal vector, $D(\mathcal{T}^{-1}(\omega_h, \Delta x, \Delta y))$ can be also rewritten as $D(\mathcal{T}^{-1}(\omega_h, \tilde{\mathbf{n}}))$.

It should be noted that the MDF with the inverse transformation, i.e., $D(\mathcal{T}^{-1}(\omega_h, \tilde{\mathbf{n}}))$, still remain in the original MDF space. The normal vector is always $(0,0,1)$ in tangent space. Therefore, the dot product of any vector \mathbf{v} and the normal vector \mathbf{n} (i.e., $\mathbf{v} \cdot \mathbf{n}$) is simply the z component of the vector, $\mathbf{v} \cdot z$, and the actual MDF we used is as follows:

$$D'(\omega_h, \tilde{\mathbf{n}}) = \quad (6)$$

$$D(\mathcal{T}^{-1}(\omega_h, \tilde{\mathbf{n}}), \mathbf{n}) = \frac{\sqrt{(e_x+1)(e_y+1)}}{2\pi} \mathcal{T}^{-1}(\omega_h, \tilde{\mathbf{n}}) \cdot z^\epsilon$$

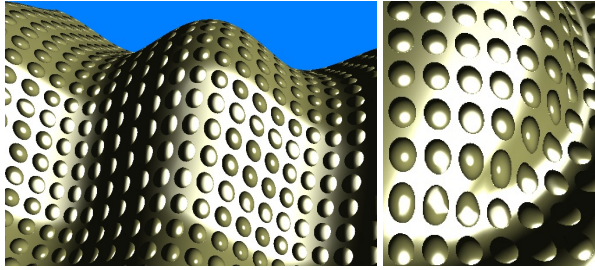
,where the exponent ϵ is $e_x \cos^2 \phi + e_y \sin^2 \phi$.

Fig.9 shows the effect of the MDF deformation by comparing the specular reflections on the illusory bumps. The bumpy illusion on the surface shown in Fig.9 (a) is generated only with normal mapping method while the result shown in Fig.9 (b) is generated with MDF deformation techniques. The original surface has anisotropic reflection property. However, as shown in the figure, the original MDF does not reproduce the anisotropic reflectance on the bumps. Even worse, the shapes of the specular reflection areas are weirdly distorted on some bumps. The deformed MDF removes such disadvantages as shown in Fig.9 (b). The anisotropic reflectance is well preserved on each illusory bump, and no weird shapes are found.

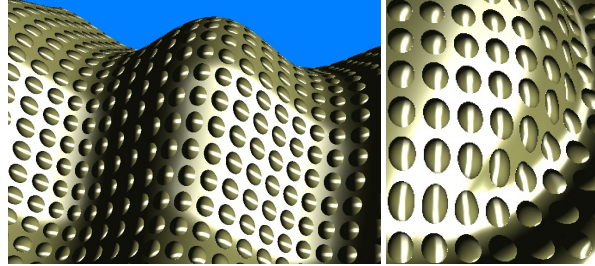
2.4 Scratch Map Generation

As mentioned earlier, we represent the natural metallic appearance by engraving small scratches on the surface. Those scratches are expressed with perturbed normal vectors, and some example normal maps were already shown in Fig.4.

The scratch maps can be generated with various techniques, but it can be easily and efficiently created in a procedural manner. In order to devise a scratch map generation method, we employed engraving a hemisphere as a basic operation. The normal vectors on the engraved hemispherical surface can be easily computed

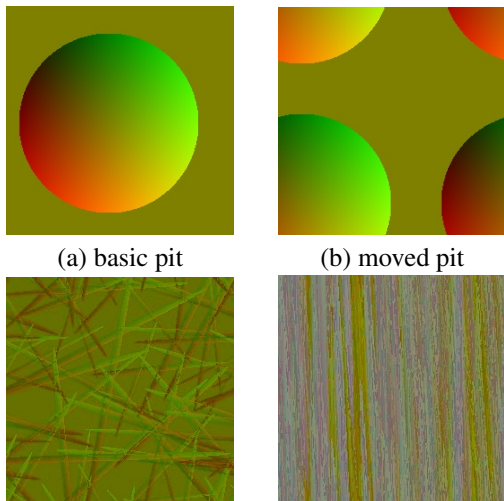


(a) Normal mapped surface without MDF deformation



(b) Normal mapped surface with deformed MDF

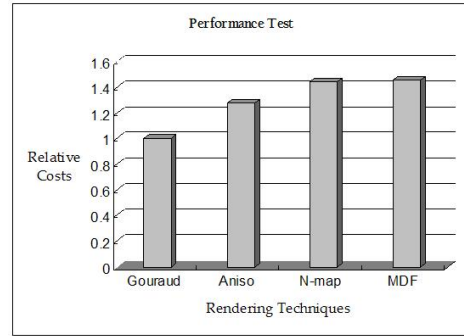
Figure 9: Effect of MDF deformation on anisotropic reflection surface: normal mapping (a) without MDF deformation and (b) with additional MDF deformation applied.



(c) random direction (d) directional tendency

Figure 10: Concept of scratch map generation

in tangent space. Fig.10 (a) shows the basic scratch texture with one engraved hemisphere. The center of the hemisphere can freely move within the texture space. We made our texture seamless as shown in Fig.10 (b). We can also scale the hemisphere and stretch in any direction, and arbitrarily increase the number of engraved pits. The depth of the engraved scratch can be also arbitrarily changed. Fig.10 (c) and (d) show the scratch maps generated by stretching the engraved pits in random direction and in a certain range of directions respectively.



Tech	Gouraud	Aniso	N-Map	MDF
Cost	1	1.28	1.44	1.46

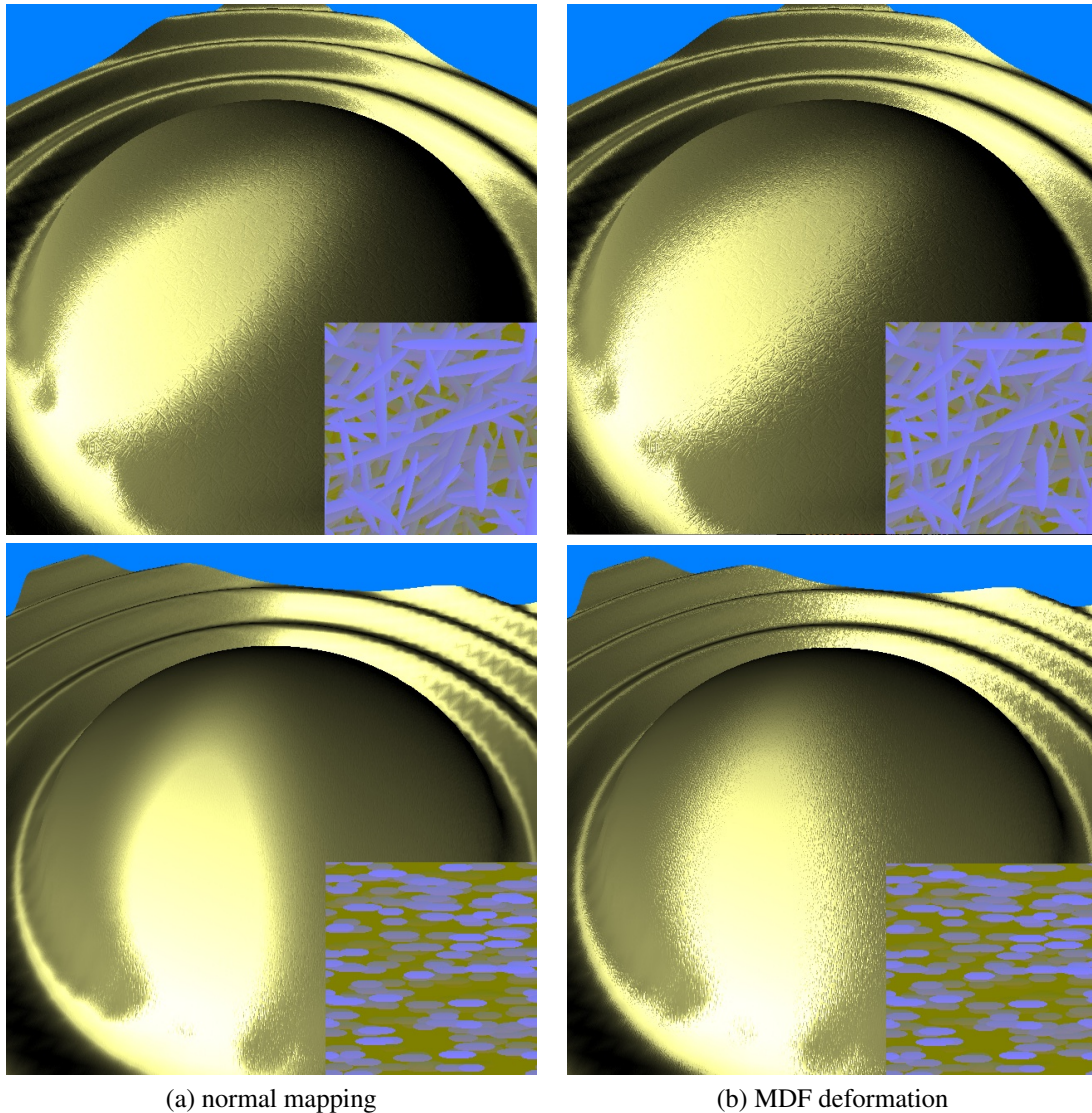
Figure 11: Rendering performance of the proposed method compared with other realtime methods.

3 EXPERIMENTS

The techniques proposed in this paper was implemented with OpenGL shading language, and the computing environments were Mac OS X operating system with 2.26 GHz Intel core 2 CPU, 2 G DDR3 RAM and NVIDIA 256M VRAM GeForce 9400M. Fig.11 is the performance analysis of the proposed method compared with previous traditional approaches. The label 'Aniso' means Ashikhmin-Shirley anisotropic reflection model, 'N-map' represents normal mapping, and 'MDF' indicates the proposed MDF deformation techniques. The computational cost of Gouraud shading is taken as a unit cost, and other rendering techniques were compared with the unit cost. As shown in the figure, the proposed method with deformed MDF is just slightly more expensive than usual normal mapping (labeled as N-Map in the figure) which works very well in realtime environments.

Fig.12 compares the light scattering on normal mapped anisotropic reflection surface. Fig.12 (a) shows the rendering results where normal mapping is applied without deforming the MDF while (b) shows results rendered with additional MDF deformation. The normal map image in the right bottom corner is the scratch map applied. As shown in the figure, the scratches represented by simple normal mapping do not plausibly scatter the light. However, the results with the proposed method in (b) show realistic light scattering along the rim of the specular reflection area.

Fig.13 shows the effect of the MDF deformation when environments are mapped on the surface. The reflection on the surface is modeled with Ashikhmin and Shirley BRDF model. The left column of the Fig.13 shows the result without the environment mapping while the right column shows the rendering results with environment mapping. The first row in the figure shows the original anisotropic reflection surface of Ashikhmin and Shirley's model with the scratch map texture in the right bottom corner. The middle row



(a) normal mapping

(b) MDF deformation

Figure 12: Comparison of light scattering on (a) simple normal mapped surface and (b) normal mapped surface with additional MDF deformation.

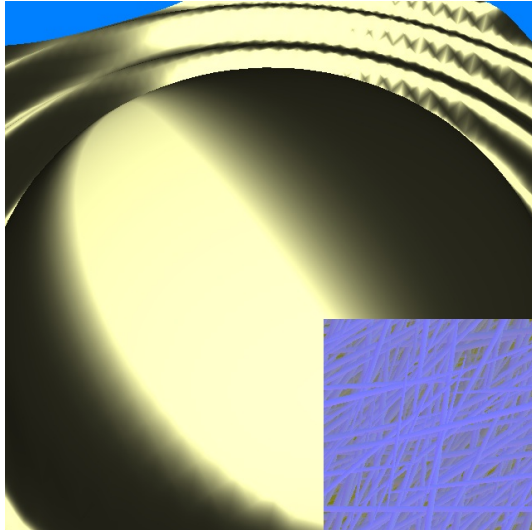
shows the results only with the simple normal mapping, and the bottom row shows the result when the proposed MDF deformation is additionally applied. As shown in the figure, the additional MDF deformation increases the rendering quality, and reproduces the light scattering by the scratches.

Although, in this paper, we employed Ashikhmin and Shirley BRDF for modeling the anisotropic reflection surface, the proposed method works with any anisotropic reflection surface. For example, our method works better with Ward BRDF model. The Ward BRDF is also an anisotropic reflection model[14].

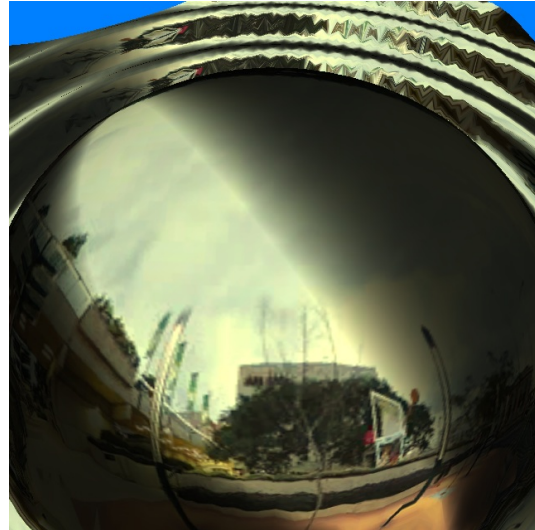
Fig.14 shows the effect of the proposed method when the surface is model with Ward anisotropic BRDF. The reflection on the surface is modeled with Ward anisotropic BRDF model. The left column of the Fig.14 shows the result without the environments mapping while the right column shows the rendering

results with environments mapping. The first row in the figure shows the original anisotropic reflection surface of Ward BRDF model. The middle row shows the results only with the simple normal mapping, and the bottom row shows the results when the proposed MDF deformation is additionally applied. As shown in the figure, the simple normal mapping on Ward BRDF surface does not provide plausible light scattering. In fact, the effect of the perturbed normal vector can be hardly observed without environment mapping. Only when the proposed method is applied, we can obtain plausible light scattering on the scratched surface as shown in the bottom row.

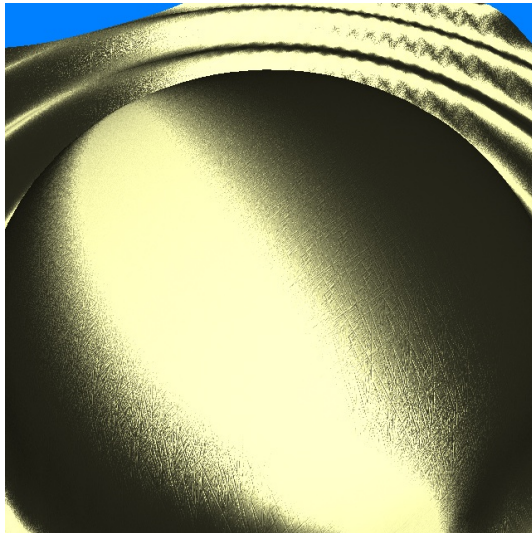
Fig.15 shows the close-up comparison of light scattering effects of simple normal mapping and the proposed method. The results shown in (a) and (b) were rendered with Ward BRDF for anisotropic reflection on the surface while Ashikhmin and Shirley BRDF model



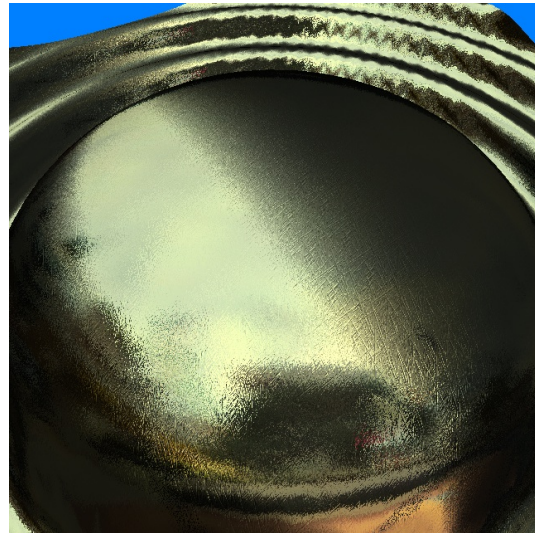
(a) Anisotropic reflection (Ashikhmin-Shirley model)



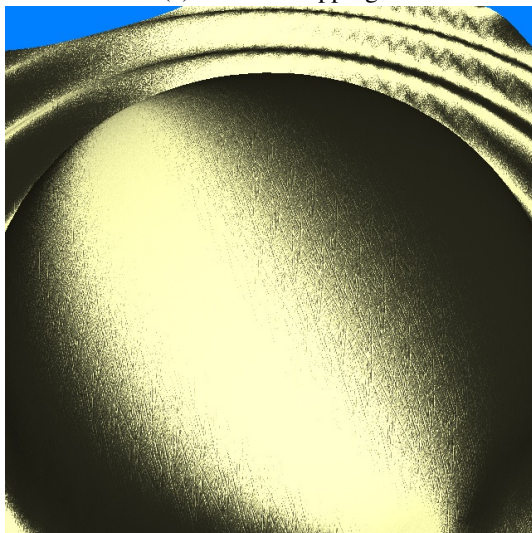
(b) Anisotropic reflection with environments



(c) Normal mapping



(d) Normal mapping with environments

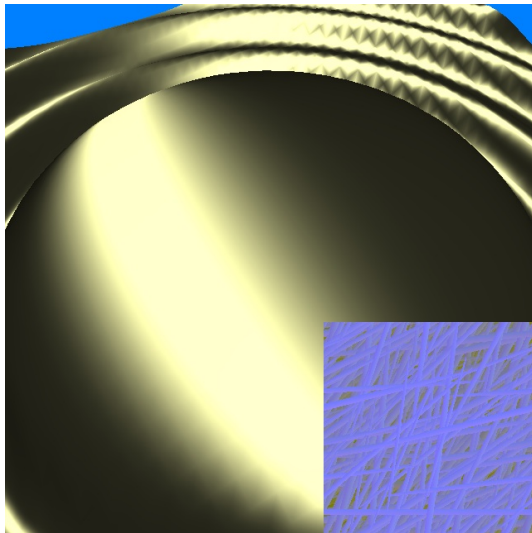


(e) MDF deformation

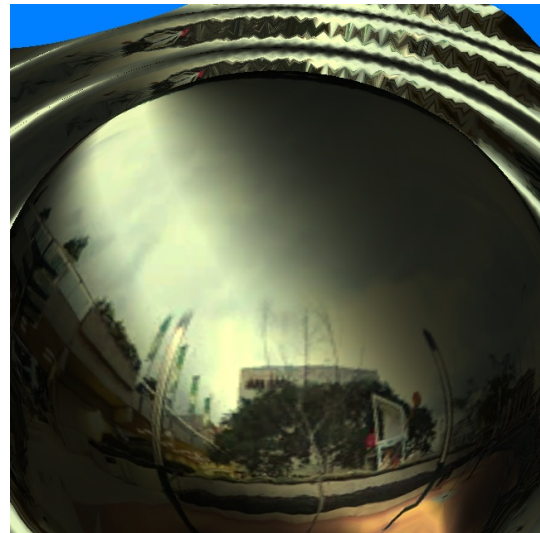


(f) MDF deformation with environments

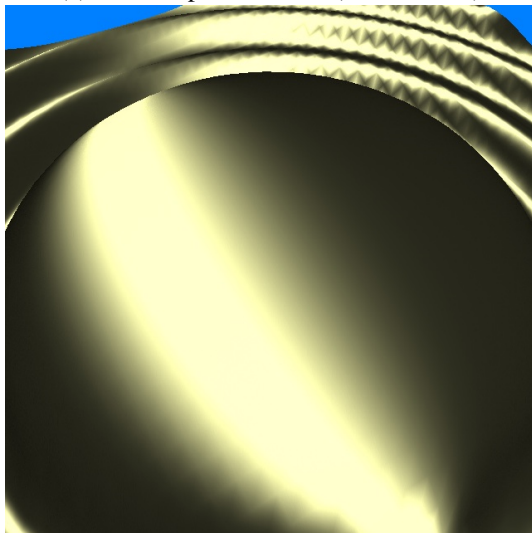
Figure 13: The effect of the propose method on Ashikhmin and Shirley model: (left column) no environment mapping, (right column) environment mapping, (top row) original anisotropic reflection surface, (b) normal mapping, and (c) normal mapping with MDF deformation.



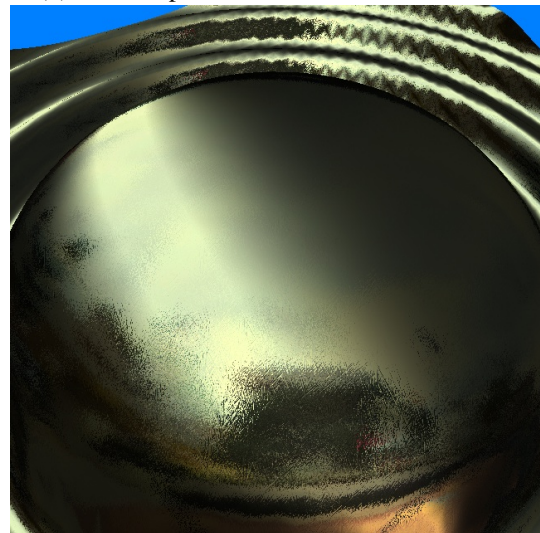
(a) Anisotropic reflection (Ward model)



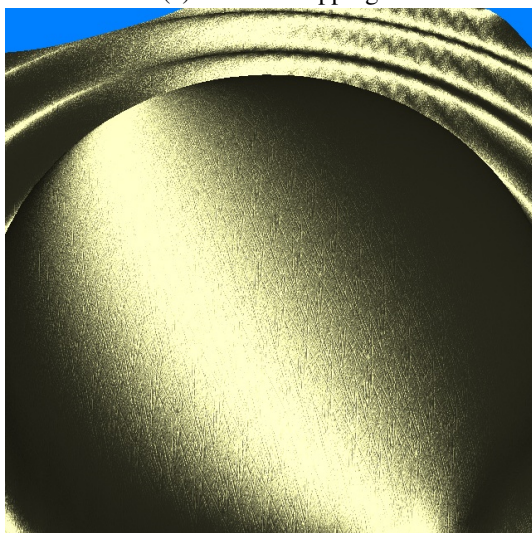
(b) Anisotropic reflection with environments



(c) Normal mapping



(d) Normal mapping with environments

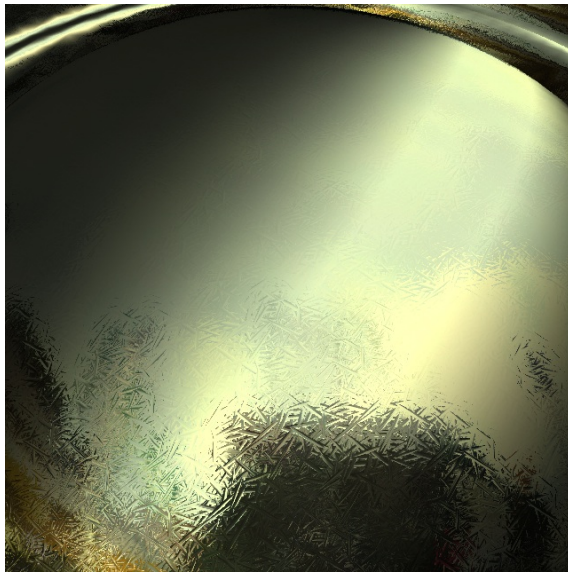


(e) MDF deformation



(f) MDF deformation with environments

Figure 14: The effect of the propose method on Ward's model: (left column) no environment mapping, (right column) environment mapping, (top row) original anisotropic reflection surface, (b) normal mapping, and (c) normal mapping with MDF deformation.



(a) Normal mapping on Ward BRDF surface



(b) MDF deformation on the Ward surface



(c) Normal mapping on Ashikhmin-Shirley BRDF surface



(d) MDF deformation on the Ashikhmin-Shirley surface

Figure 15: Close-up comparison of light scattering: (a) simple normal mapping on a surface with Ward anisotropic reflection model, (b) additional MDF deformation applied on the Ward model, (c) simple normal mapping on Ashikhmin-Shirley BRDF surface, and (d) MDF deformation effect on the Ashikhmin-Shirley surface.

is employed for those shown in (c) and (d). Fig.15 (a) and (c) show the results only with the normal mapping while (b) and (d) are results generated with the proposed MDF deformation method. As shown in the figure, normal mapping with deformed MDF shows superior rendering quality to the simple normal mapping approach.

4 CONCLUSION

In this paper, we proposed an effective and efficient method that improves the normal mapping to be successfully applied to anisotropic reflection surfaces. The proposed method is appropriate for rendering metallic surfaces with small scratches in realtime. We have

shown in this paper that the simple normal mapping or other normal perturbation techniques cannot be applied to anisotropic reflection surfaces. In order to enable normal perturbation to better illusory bumps on surface, we introduced MDF deformation concept. The experimental results show that the proposed method achieves far better rendering quality than simple normal mapping method does. Moreover, the computational cost additionally required for MDF deformation is small enough for realtime environments. The only difference between the proposed method and the traditional anisotropic BRDF models is that ω_h given to the MDF is adjusted. Therefore, the proposed method is easily implemented as GPU program and works well in

realtime environments. The proposed method can be successfully utilized in games or virtual reality systems for rendering high-quality metallic surfaces.

ACKNOWLEDGEMENTS

This work was supported in part by the SW computing R&D program of MKE/KEIT [10035184], "Game Service Technology Based on Realtime Streaming".

REFERENCES

- [1] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based brdf generator. *In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 65–74, 2000.
- [2] M. Ashikhmin and P. Shirley. An anisotropic phong brdf model. *Journal of Graphics Tools*, 5(2):25–32, 2002.
- [3] J. Blinn. Models of light reflection for computer synthesized pictures. *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.
- [4] J. Blinn and M. Newell. Texture and reflection in computer generated images. *Communication of ACM*, 19(10):542–547, 1976.
- [5] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *Computer Graphics (ACM Siggraph '81 Conference Proceedings)*, 15(3):307–316, 1981.
- [6] W. Heidrich and H.-P. Seidel. Realistic, hardware-accelerated shading and lighting. *In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 171–178, 1999.
- [7] M. Pharr and G. Humphreys. *In Physically-based Rendering*. Elsevier (Morgan Kaufman Publishers), 2004.
- [8] M. Poulin and A. Fournier. A model for anisotropic reflection. *Computer Graphics (ACM Siggraph '90 Conference Proceedings)*, 23(4):273–282, 1990.
- [9] D. Roger and N. Holzschuh. Accurate specular reflections in real-time. *Computer Graphics Forum*, 25(3):293–302, 2006.
- [10] H. Rushmeier, G. Taubin, and A. Gueziec. Applying shapes from lighting variation to bump map capture. *In Proceedings of Eurographics Rendering Workshop '97*, pages 35–44, 1997.
- [11] C. Schlick. A customizable reflectance model for everyday rendering. *In Proceedings of the 4th Eurographics Workshop on Rendering*, pages 73–84, 1993.
- [12] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9), 1967.
- [13] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *In Proceedings of ACM Siggraph Asia 2009*, pages 1–10, 2009.
- [14] G. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics (ACM Siggraph '92 Conference Proceedings)*, 26(2):265–272, 1992.
- [15] L. Zirmay Kalos, T. Umenhoffer, Gustavo Patow, L. Szecsi, and M. Sbert. Specular effects on the gpu: State of the art. *Computer Graphics Forum*, 28(6):1586–1617, 2009.