Rendering Iridescent Colors Appearing on Natural Objects

H. Hirayama[†], Y. Yamaji[†], K. Kaneda[†], H. Yamashita[†], and Y. Monden[‡]
[†]Faculty of Engineering
Hiroshima University
{hira, san, kin, yama}@eml.hiroshima-u.ac.jp
[‡]Interdisciplinary Faculty of Science and Engineering
Shimane University
monden@cis.shimane-u.ac.jp

Abstract

Iridescent colors appearing on natural objects, such as the feathers of a hummingbird, surfaces inside seashells, and shells of beetles, are beautiful and impressive. The cause of such iridescent colors is interference of light inside multilayer film structures covering the natural objects. This paper proposes a method for realistically rendering iridescent colors appearing on natural surfaces, taking into account interference of light inside multilayer structures. To render the iridescent colors, we have developed an illumination model for multilayer structures fluctuated through the use of a hypertexture. We also have improved the hypertexture to be able to represent fluctuated parametric curved surfaces. Several examples of iridescent colors on the surfaces of a seashell and a beetle demonstrate the usefulness of the proposed method.

keywords: iridescent colors, multilayer films, fluctuated surfaces, hypertextures, Bezier surfaces

1. Introduction

Researchers in the field of computer graphics have been involved for some time in rendering natural objects such as water, clouds, plants, and so on, and thus many methods for rendering natural objects have been developed[1][10][3]. Despite these advances, however, many problems remain unaddressed in rendering optical phenomena appearing on natural objects because of their complexity and multiformity.

One of the unaddressed problems is the rendering of iridescent colors appearing on natural objects, such as the feathers of a hummingbird, the inside of seashells, shells of beetles, and so on. The iridescent colors have a strong impact and are caused by interference of light inside multilayer film structures covering the natural objects. For example, the multilayer structure found on the surfaces of seashells and pearls consists of aragonite crystal layers and protein membranes[9][11] (see Fig. 5(a)), the feathers of a hummingbird consist of keratin and melanin layers[7], and the structure of butterfly wings consists of air and chitin layers[15]. The purpose of this study is to realistically render iridescent colors appearing on the surfaces of such natural objects, taking into account interference of light inside multilayer structures.

Methods for visualizing interference phenomena caused by layered structures, for example, soap bubbles[14][4], film-coated glass[6], pearls[9], windowpanes[8], and so on, have been developed. In these methods, however, layers are restricted to uniform structures, that is, structures with each layer having a constant thickness over the entire surface, meaning that multiple layers with fluctuated thickness and refractive indices cannot be handled. The methods cannot render iridescent colors appearing on natural objects, because the natural objects are comprised of fluctuated multilayer structures, and the color therefore is dispersed. In this paper, we extend a rendering method that can handle the fluctuations of such multilayer structures.

Methods for representing fluctuated objects have been proposed in the field of computer graphics, with the representative methods including bump mapping[5], displacement mapping[2], and hypertexture[12]. However, bump mapping cannot change the shape of objects, displacement mapping cannot change the attributes of surfaces such as refractive indices, and hypertexture cannot handle parametric free form surfaces.

To render iridescent colors appearing on natural objects, we extend an illumination model of uniform multilayer films[8] in order to handle fluctuated multilayer films based on hypertexture. Furthermore, we also improve hypertexture to be able to represent fluctuated parametric free form surfaces.

In this paper, we first discuss interference of light inside multilayer films, and introduce a method for calculating reflectances and transmittances of the multilayer films. Then, in Sec. 3, a new method for realistically rendering iridescent colors on natural objects is proposed, and in Sec. 4, several examples of the proposed method are demonstrated.

2. Illumination Model of Multilayer films

Iridescent colors appearing on natural objects are caused by interference of light inside multilayer films covering the object. In this section, a method for rendering iridescent colors caused by multilayer films[8] is described.

2.1. Uniform multilayer models

Incident light upon multilayer films is reflected and refracted at each boundary of the layers, and interference and absorption of light occur inside the multilayers (see Fig. 1). As a result, iridescent colors appear on the surface of a natural object with multilayer structures.

In computer graphics, several methods have been developed to render optical phenomena caused by films coated on an object. The most accurate method for rendering optical phenomena takes into account multiple reflections and refractions, interference, and absorption of light inside multilayers[8].

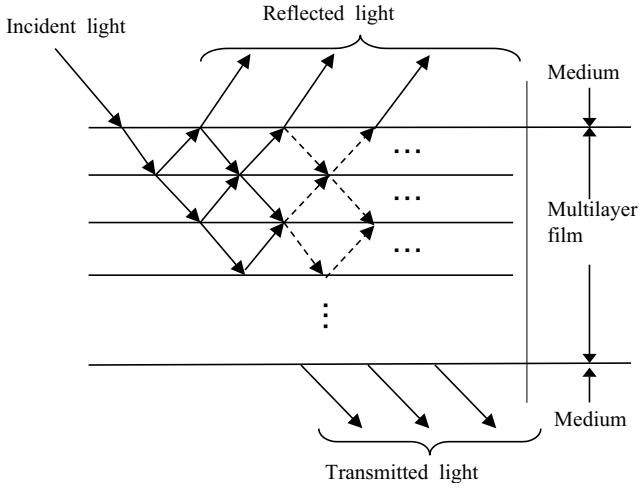


Figure 1. Light propagation inside a multilayer film

2.2. Calculating reflectances and transmittances of multilayer structures

To calculate reflectance and transmittance of $N+2$ layered films shown in Fig. 2, both a refractive index and the thickness of each layer are specified as input data. Reflectivities and transmissivities of each boundary and phase differences of each layer are derived by Fresnel formulae and Snell's law. Reflectance and transmittance of the multilayer film are calculated by the following seed and recurrence equations[8].

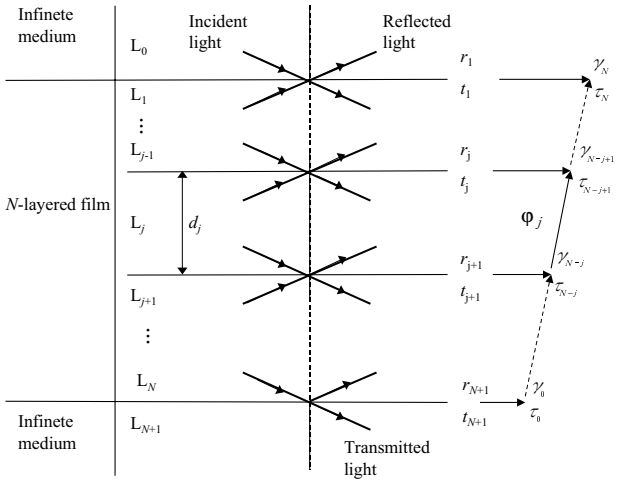


Figure 2. $N+2$ layered films

The seed equations $(j = N+1)$:

$$\gamma_0 = r_{N+1}, \tag{1}$$

$$\tau_0 = t_{N+1}, \tag{2}$$

The recurrence equations $(j = N, \cdots, 1)$:

$$\gamma_{N-j+1} = \frac{r_j + \gamma_{N-j}\, e^{2i\varphi_j}}{1 + r_j\, \gamma_{N-j}\, e^{2i\varphi_j}}, \tag{3}$$

$$\tau_{N-j+1} = \frac{t_j \tau_{N-j}\, e^{i\varphi_j}}{1 + r_j\, \gamma_{N-j}\, e^{2i\varphi_j}}, \tag{4}$$

where i denotes an imaginary number, and r_j and t_j are a reflectivity and a transmissivity at a boundary between $(j-1)$th and j-th layers, respectively. φ_j is a phase difference in j-th layer. γ_{N-j+1} and τ_{N-j+1} are a composite reflectivity and a composite transmissivity between $(N+1)$th and j-th layers, respectively, taking into account multiple reflections and refractions, interference, and absorption of light. Finally, composite reflectances and transmittances of the multilayer film are obtained by calculating the square of the absolute value of the composite reflectivities and transmissivities, respectively[8].

2.3. Iridescent colors caused by uniform multilayer films

An example of rendering iridescent colors caused by uniform multilayer films is shown in Fig. 6(a). The center and left plates are coated with the multilayer structures of a seashell and a buprestid beetle, respectively (see Figs. 5(a) and (b)). Each layer covering the right plate becomes 20 nm thicker than that of the left plate.

Iridescent colors appearing on the surfaces in Fig. 6(a) are too regular compared with those of natural objects, because the multilayer films are of uniform thickness. To render iridescent colors appearing on natural objects, multilayer films with fluctuated thickness should be taken into account.

3. Modeling Natural Object Surfaces

For realistic rendering of iridescent colors appearing on natural objects, it is necessary to represent not only geometrically smooth surfaces but also fluctuated surfaces such as the inside of seashells, shells of beetles, and so on. For modeling the surfaces of natural objects, a hypertexture[12] defined on a Bezier surface is newly developed in this paper.

3.1. Hypertexture

In 1989, a hypertexture with the following features was introduced to represent the complex shapes of objects[12].

- Each object has an object density function, $\mathrm{D}(\mathbf{x})$, and the output of the function varies from 0 to 1, depending on the density of the object. The output range is called a soft region. The object density function $\mathrm{D}(\mathbf{x})$ is expressed as follows.

$$\mathrm{D}(\mathbf{x}) = \begin{cases} 0 & (d \geq s) \\ \dfrac{d^2}{s^2} & (0 < d < s) \\ 1 & (0 \leq d) \end{cases}, \tag{5}$$

  where s and d denote a softness and a distance between the object surface and position $\mathbf{x}$, respectively.

- The shape of an object is modulated by a density modulation function (DMF), f_i, as the function modulates the density distribution inside the soft regions. A hypertexture, H, with object density function D at position $\mathbf{x}$ is given by

$$\mathrm{H}(\mathrm{D}(\mathbf{x}), \mathbf{x}) = \mathrm{f}_n(\cdots(\mathrm{f}_1(\mathrm{f}_0(\mathrm{D}(\mathbf{x}))))), \tag{6}$$

  where $\mathrm{f}_i(i = 0, \cdots n)$ are DMFs.

A ray marching algorithm[16] is used to render objects with hypertextures. In the algorithm, a ray progresses step by step over a short distance, and hypertextures at each position, $\mathbf{P}$, are calculated. If the hypertexture density $\mathrm{H}(\mathrm{D}(\mathbf{P}), \mathbf{P})$ is greater than 0, the normal vector $\mathbf{N}$ of the object at the position $\mathbf{P}$ is calculated using the densities at three sampled points near the position $\mathbf{P}$.

$$\mathbf{N}(\mathbf{P}) = \begin{bmatrix} \mathrm{H}(\mathbf{P}) - \mathrm{H}(\mathbf{P} - \Delta\mathbf{P}_x) \\ \mathrm{H}(\mathbf{P}) - \mathrm{H}(\mathbf{P} - \Delta\mathbf{P}_y) \\ \mathrm{H}(\mathbf{P}) - \mathrm{H}(\mathbf{P} - \Delta\mathbf{P}_z) \end{bmatrix}^T, \tag{7}$$

where $\Delta\mathbf{P}_x$, $\Delta\mathbf{P}_y$, $\Delta\mathbf{P}_z$ denote the neighbor positions to the position $\mathbf{P}$ toward the directions x, y, and z, respectively. Reflected and transmitted rays are also cast recursively from the position $\mathbf{P}$.

3.2. Extension of hypertexture

In the proposed method, a hypertexture is used for modeling fluctuated multilayer structures covering the natural objects whose surfaces show iridescent colors. However, a hypertexture defined on parametric curved surfaces has not been developed yet. Parametric surfaces such as Bezier surfaces are indispensable for modeling natural objects, and we have therefore extend a hypertexture defined on Bezier surfaces.

3.2.1. Fluctuated multilayer structures

To render iridescent colors appearing on a natural object, multilayer structures covering an object are modified through the use of the hypertexture. That is, the thickness and/or refractive index of each layer are fluctuated by the object density function used in the hypertexture.

$$d'_j(d_j, \mathbf{x}) = \mathrm{f}_n(\cdots(\mathrm{f}_1(\mathrm{f}_0(d_j, \mathbf{x})))), \tag{8}$$

$$n'_j(n_j, \mathbf{x}) = \mathrm{g}_n(\cdots(\mathrm{g}_1(\mathrm{g}_0(n_j, \mathbf{x})))), \tag{9}$$

where d'_j and d_j are a modulated layer and a layer of a given thicknesses of j-th, respectively, and n'_j and n_j are a modulated index and a given refractive indices of j-th layer, respectively. $\mathbf{x}$ is a position on the surface of multilayer structures, and f_k and $\mathrm{g}_k(k = 0, \cdots, n)$ are thickness and refractive index modulation functions, respectively. Reflectances and transmittances of fluctuated multilayer structures are then calculated using Eqs. 1 through 4[8].

3.2.2. Improved ray marching

A ray marching algorithm is expensive in computational cost, because the density of objects is evaluated at every sampled point of a ray. In the proposed method, the ray marching process is skipped ahead to the intersection between a cast ray and bounding volumes of objects with the hypertexture. The bounding volume may contain a single object or a group of objects. If a ray intersects one of the bounding volumes, the ray marching process is invoked.

3.2.3. Calculating distance between a point and a Bezier suface

The hypertexture is extended for application to objects modeled by Bezier surfaces. To obtain the density of the hypertexture defined on a Bezier surface, it is necessary to

calculate the distance between a certain point and the Bezier surface. In this section, after explaining a distance calculation in 2D space, we propose a method for calculating the distance from a certain point to a Bezier surface in 3D space.

Let's consider a degree N Bezier curve, $\mathbf{C}(u)$,

$$\mathbf{C}(u) = \sum_{i=0}^{N} \mathbf{P}_i \mathrm{B}_i^n(u), \tag{10}$$

where u is a parameter, $\mathbf{P}_i (i = 0, \cdots, N)$ are control points, and B_i^N is a Bernstein function. To calculate the distance from point $\mathbf{Q}$ to Bezier curve $\mathbf{C}(u)$, the orthogonal relationship between a line passing through the point and the curve is used (see Fig. 3).

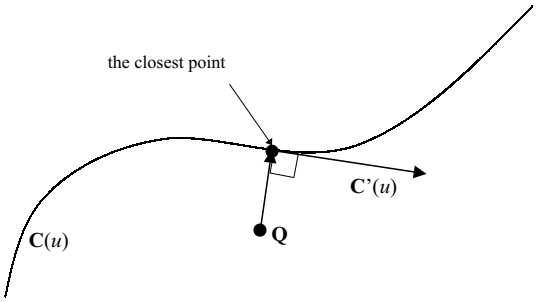


Figure 3. The orthogonal relationship

$$g(u) = \frac{\partial \mathbf{C}(u)}{\partial u} \cdot [\mathbf{C}(u) - \mathbf{Q}]. \tag{11}$$

Substituting Eq. 10 into Eq. 11, the following equations are derived.

$$g(u) = \sum_{i=0}^{2N-1} \mathrm{A}_i \mathrm{B}_i^{2N-1}(u), \tag{12}$$

$$\mathrm{A}_i = \sum_{k=0}^{N-1} \sum_{l=0}^{N} (\mathbf{P}_{k+1} - \mathbf{P}_k) \cdot (\mathbf{P}_l - \mathbf{Q}) \cdot \frac{\binom{N-1}{k}\binom{N}{k}}{\binom{2N-1}{i}} f_i(k,l), \tag{13}$$

$$f_i(k,l) = \begin{cases} 1 & (i = k+l) \\ 0 & (\text{otherwise}) \end{cases}. \tag{14}$$

Equation 12 is also a Bezier curve with degree $(2N-1)$. Intersection points with the perpendicular lines and the curve are calculated by solving equation $g(u) = 0$, and a Bezier clipping algorithm[13] is employed to solve the equation. Among the intersection points, the closest point to point $\mathbf{Q}$ is selected to determine the distance from the point to the curve.

The method for calculating a 2D distance described above is extended to the method for a 3D distance. That is, a method for calculating the distance between a point and a Bezier surface is developed.

Let's consider a Bezier surface with $N \times M$ degree,

$$\mathbf{S}(u,v) = \sum_{i=0}^{N} \sum_{j=0}^{M} \mathbf{P}_{ij} \mathrm{B}_i^N(u) \mathrm{B}_j^M(v). \tag{15}$$

To calculate intersection points between a Bezier surface, $\mathbf{S}(u,v)$, and its normal passing-through-point, $\mathbf{Q}$, an extended version of Eq. 11 with respect to u-direction is derived.

$$g_u(u,v) = \frac{\partial}{\partial u}\mathbf{S}(u,v) \cdot [\mathbf{S}(u,v) - \mathbf{Q}], \tag{16}$$

Substituting Eq. 15 into Eq. 16, the following equations are derived.

$$g_u(u,v) = \sum_{i=0}^{2N-1} \sum_{j=0}^{2M} \mathrm{A}_{ij} \mathrm{B}_i^{2N-1}(u) \mathrm{B}_j^{2M}(u), \tag{17}$$

$$\mathrm{A}_{ij} = \sum_{k=0}^{N-1} \sum_{l=0}^{M} \sum_{n=0}^{N} \sum_{m=0}^{M} \frac{\binom{N-1}{k}\binom{N}{n}}{\binom{2N-1}{i}} \frac{\binom{M}{l}\binom{M}{m}}{\binom{2M}{j}} \cdot (\mathbf{P}_{k+1,l} - \mathbf{P}_{kl}) \cdot (\mathbf{P}_{nm} - \mathbf{Q}) f_i(k,l) f_j(n,m), \tag{18}$$

$$f_i(k,l) = \begin{cases} 1 & (i = k+l) \\ 0 & (\text{otherwise}) \end{cases}. \tag{19}$$

Equation 17 shows that $g_u(u,v)$ is a Bezier surface with $(2N-1) \times 2M$ degree. In the same way, a function, $g_v(u,v)$, with respect to v-direction can be derived. To calculate intersection points, we find the parameters (u,v) that satisfy both equations, $g_u(u,v) = 0$ and $g_v(u,v) = 0$, at the same time. A Bezier clipping algorithm can be also employed to solve the problem (see Fig. 4). The clipping process is carried out for u and v directions, alternately. As a result of the clipping, intersection points between a Bezier surface and its normal are obtained. The distance between the point and the Bezier surface is determined by the closest intersection point to the calculation point.

It is possible that the closest point to the calculation point is located on the edges or vertices of the Bezier surface. To address that problem, the closest point is selected among the intersection points, with both a Bezier surface and Bezier curves defined on the surface edges and four vertices of the surface.

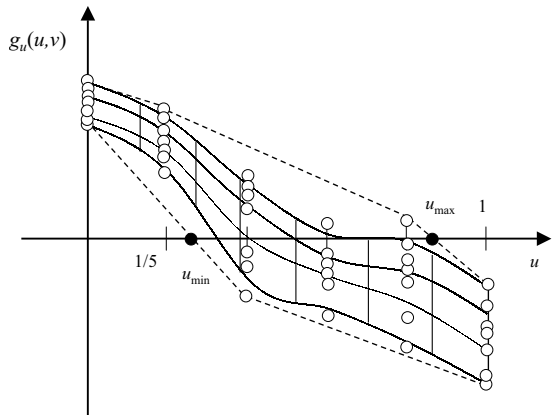
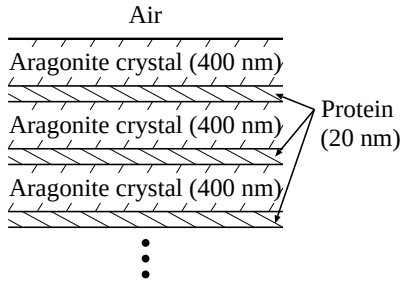


Figure 4. A Bezier clipping algorithm

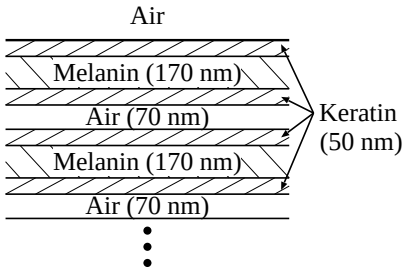
4. Examples

4.1. Multilayer structures of natural objects

Figure 5 shows multilayer structures of a surface inside a seashell and inside the shell of a buprestid beetle (*Chrysochroa fulgicissima*). In the examples used in this section, the multilayer structure of the seashell consists of fourteen sets of aragonite crystal and protein layers[9][11], and that of the beetle consists of six sets of keratin, melanin, keratin, and air layers.



(a) a seashell



(b) a buprestid beetle

Figure 5. Multilayer structures

4.2. Comparison of fluctuated multilayer structures and uniform structures

Figure 6 shows a comparison of iridescent colors appearing on fluctuated multilayer structures and those on uniform structures. Each plate in Fig. 6(a) has a uniform multilayer structure on the surface, while in Fig. 6(b) each plate has a fluctuated multilayer structure. The center and left plates have the same multilayer structures as those of the seashell and the buprestid beetle shown in Figs. 5(a) and (b), respectively. Each layer covering the right plate in Fig. 6 becomes 20 nm thicker than that of Fig. 5(b).

The multilayer structures in Fig. 6(b) are modulated by using the following DMF.

$$d'_j = d_j + 100 \times f_{\text{noise}}(\mathbf{x}), \tag{20}$$

where j denotes the number of layers, $\mathbf{x}$ denotes a position on the surfaces of plates, and the position $\mathbf{x}$ is a sampled point on a ray. d'_j and d_j are a modulated thickness and a given thickness, respectively. f_{noise} is a noise function given in the article[12], and the range is $[-1, 1]$.

Using the extended hypertexture, iridescent colors are dispersed over the plates as shown in Fig. 6(b).

4.3. A seashell

Iridescent colors appearing on a surface inside a seashell are rendered by the proposed method in Fig. 7. The shape of the seashell is modeled using twelve Bezier patches, and the multilayer structure of the seashell shown in Fig. 5(a) is used. The multilayer structure is modulated by the extended hypertexture, that is, the thickness of aragonite crystal layers is modulated in the same way as used in Fig. 6(b) (see Eq. 20).

Refractive indices, n_j, of both the aragonite crystal layers and the protein layers are also modulated.

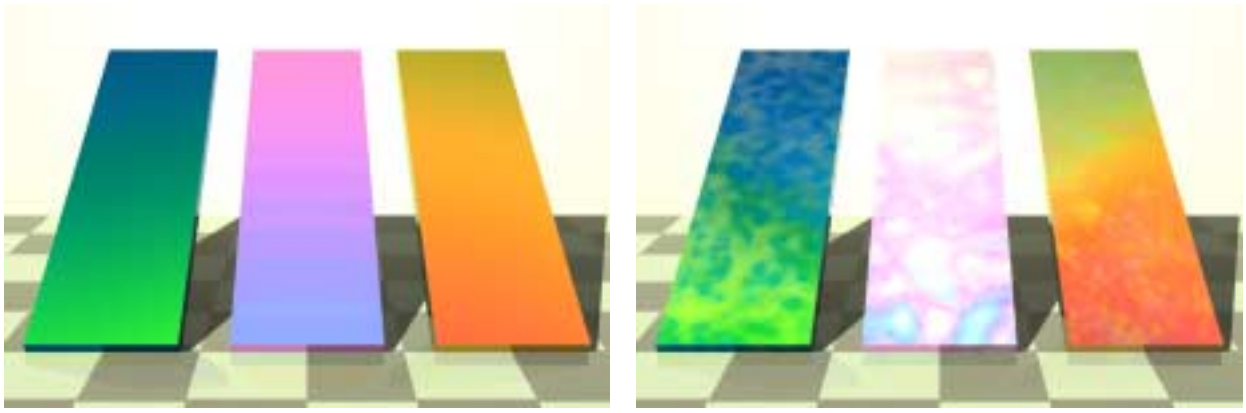
$$n'_j = n_j + 0.01 \times f_{\text{noise}}(\mathbf{x}), \tag{21}$$

where n'_j and n_j are a modulated index and a given refractive index, respectively.

Observing the rendered seashell, iridescent colors are subtly dispersed due to the fluctuation of the multilayer structure.

4.4. A buprestid beetle

Iridescent colors appearing on the shell of a buprestid beetle are rendered in Fig. 10. The shape of the beetle is modeled using 176 Bezier patches, and the multilayer structure of the beetle is used as shown in Fig. 5(b). Both the multilayer structure and the object shape are modulated by



(a) Uniform multilayer structures (b) Fluctuated multilayer structures

Figure 6. Comparison of fluctuated multilayer structures and uniform structures

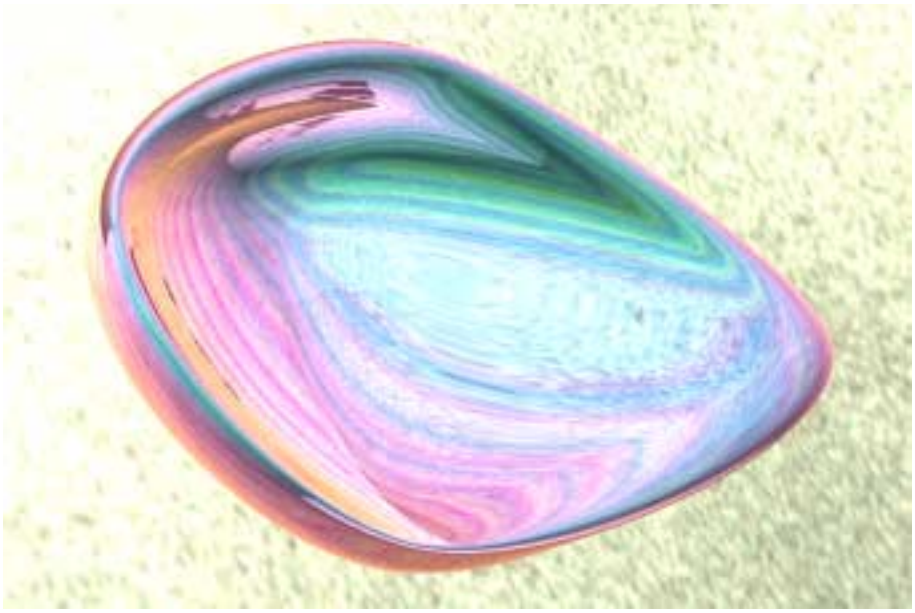


Figure 7. A seashell

the extended hypertexture. The thickness of each layer is modulated using the following DMF.

$$d'_j = d_j + g(u), \tag{22}$$

$$g(u) = \begin{cases} 0 & (u < 0.6, u > 0.8) \\ 10 & (u > 0.65, u < 0.75) \\ \dfrac{10 \times (0.1 - |0.5 - u|)}{0.05} & (\text{otherwise}) \end{cases}, \tag{23}$$

where u is a parameter of Bezier surfaces, $g(u)$ is a DMF shown in Fig. 8, and d'_j and d_j are a modulated thickness and a given thickness, respectively. The refractive indices of multilayers on the shell of the buprestid beetle are fixed.

The shape of the buprestid beetle is also modulated using the following equations with respect to the parameter u-direction of Bezier surfaces.

$$\begin{aligned} \mathrm{H}(\mathrm{D}(\mathbf{x}), \mathbf{x}) = \mathrm{D}(\mathbf{x}) &- f_1(f_0(5 \times u), f_0(0.2), f_0(0.1)) \\ &- f_1(f_0(10 \times u), f_0(0.05), f_0(0.01)), \end{aligned} \tag{24}$$

$$f_0(t) = [1 + f_{\text{noise}}(t)]\, t, \tag{25}$$

where f_1 (see Fig. 9) and f_0 are DMFs. For v direction of Bezier surfaces, the modulation is carried out in the same way.

Fluctuated shells of beetles are rendered by the proposed method, and the metallic colors found in actual buprestid beetles are represented.

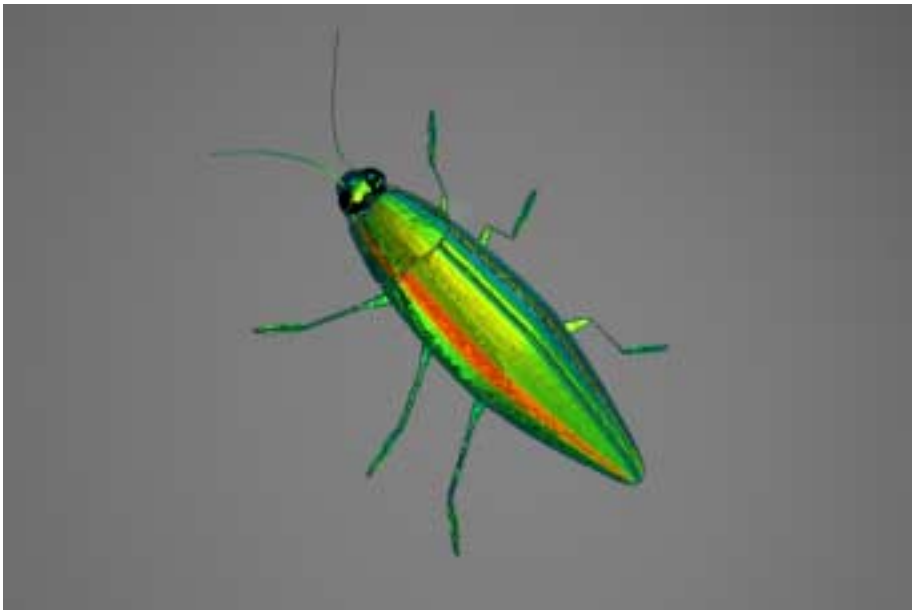


Figure 10. A buprestid beetle

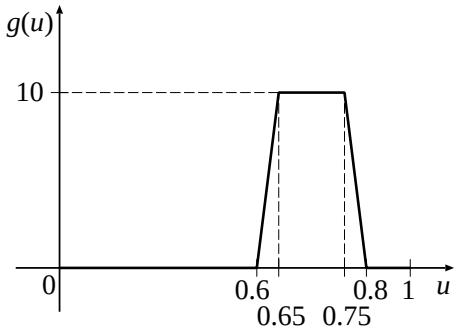


Figure 8. DMF, $g(u)$

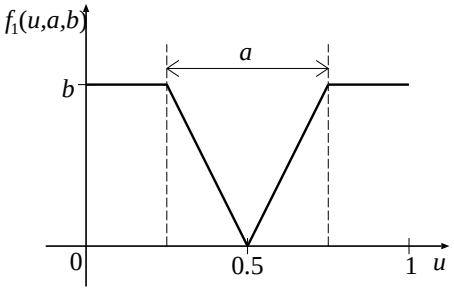


Figure 9. DMF, $f_1(u)$

5. Conclusions

We have proposed a method for realistically rendering iridescent colors appearing on surfaces of natural objects, taking into account interference of light inside multilayer structures. The proposed method fluctuates multilayer structures based on a hypertexture that have been extended to handle parametric free form surfaces. Several examples of iridescent colors of natural objects demonstrate the usefulness of the proposed method.

Further research should be carried out to determine suitable parameters of DMFs and noise functions to render a realistic appearance of objects modulated by the hypertexture. Another challenging task ahead includes a rendering method for taking into account the effects of suspended particles contained in multilayer films and directional diffuse reflection of a base material covered with multilayer films.

References

[1] N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara, K. Muraoka, and N. Saito. Visual simulation of water currents using a particle-based behavioural model. *Journal of Visualization and Computer Animation*, 6(3):155–172, 1995.

[2] R. L. Cook. Shade trees. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):223–231, July 1984. Held in Minneapolis, Minnesota.

[3] O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. *Proceedings of SIGGRAPH 98*, pages 275–286, 1998.

[4] M. L. Dias. Ray tracing interference color. *IEEE Computer Graphics and Applications*, 11(2):54–60, 1991.

[5] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics, Principles and Practice, Second Edition*. Addison-Wesley, 1990.

[6] J. S. Gondek, G. W. Meyer, and J. G. Newman. Wavelength Dependent Reflectance Functions. *Proc. SIGGRAPH 94*, pages 213–220, 1994.

[7] G. H. Greenwalt. Iridescent color of hummingbird feathers. *Journal of the Optical Society of America*, 50(10):1005–1013, 1960.

[8] H. Hirayama, K. Kaneda, H. Yamashita, Y. Monden, and Y. Yamaji. Visualization of optical phenomena caused by multilayer films with complex refractive indices. *Proc. Pacific Graphics '99*, pages 128–137, 1999.

[9] N. Nagata, T. Dobashi, Y. Manabe, T. Usami, and S. Inokuchi. Modeling and visualization for a pearl-quality evaluation simulator. *IEEE Trans. Visualization and Computer Graphics*, 3(4):307–315, 1997.

[10] T. Nishita, E. Nakamae, and Y. Dobashi. Display of clouds and snow taking into account multiple anisotropic scattering and sky light. *Proceedings of SIGGRAPH 96*, pages 379–386, 1996.

[11] S. Ogura. Grating and multilayer structures observed in nature —microscopic observations for pearls, scales of butterflies, moths, and fish—. *A monthly publication of The Japan Society of Applied Physics*, 66(12):1330–1334, 1997. (in Japanese).

[12] K. Perlin and E. M. Hoffert. Hypertexture. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3):253–262, 1989.

[13] T. Sederberg and T. Nishita. Curve intersection using bezier clipping. *Computer-aided Design*, 22(9):538–549, 1990.

[14] B. E. Smits and G. W. Meyer. Newton's colors: Simulating interference phenomena in realistic image synthesis. *Proc. Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 185–194, 1990.

[15] H. Tada, S. E. Mann, I. N. Miaoulis, and P. Y. Wong. Effects of a butterfly scale microstructure on the iridescent color observed at different angles. *Applied Optics*, 37(9):1579–1584, 1998.

[16] H. K. Tuy and L. T. Tuy. Direct 2-d display of 3-d objects. *IEEE Computer Graphics & Applications*, 4(10):29–33, 1984.