Modeling of the light-scattering properties of the metallic coating

Kristina Zipa, Andrey Ilyin, Aleksey Ignatenko
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
kzipa@graphics.cs.msu.ru, ailyin@graphics.cs.msu.ru, aignatenko@graphics.cs.msu.ru

Abstract

The process of the manual creation of the realistic models is labor-intensive. In order to facilitate the modeling process we propose a fully automatic method for determining the light-scattering properties of objects from photographs taken with a digital camera. In this work we also propose new reflectance model of the textured and metallic coating surfaces.

Keywords: material, texture, metallic coating, reflectance properties, light-scattering, reconstruction.

1. INTRODUCTION

Rendering of the photorealistic images of three-dimensional scenes remains a significant problem in computer graphics nowadays. This problem is significant in the process of the creation of computer games, 3D movies and virtual reality systems. Designing of the virtual scenes consists of the creation of the complex objects – models. At the rendering process reflective and light-scattering properties (textures) are assigned to virtual objects for physical and visual similarity with the real world scenes. The modeling of the metallic effect is often used for photorealistic synthesis of the paint-and-lacquer coating surfaces. Nowadays there are many approaches in the modeling of the materials of the objects. In some cases tabular BRDF [1] are used. It requires large storage, but ensures a high quality results. The acquisition of such models requires special photometric devices.

Therefore, in many cases, preference is given to the parametric models of materials for which exists a lot of effective ways of visualization. Using of this kind of models significantly limits a class of described materials. One of the additional means to enhance the realism is the texturing. When rendering, texture is represented by a bitmap image imposed on the surface of the 3D-model to give it color or the illusion of relief. A wide range of surfaces can be described with textures (soil, plants, minerals, fur and leather, etc.).

Metallic is a lamellar shaped particles of metal (such as aluminum, copper, zinc, brass or bronze) is commonly used in multilayered paints. The surface covered with such paint looks sparkling (see Figure 1). Paint with a metallic effect is used in the automotive industry and in the manufacture of different kind of appliances and cosmetics.

Tabular BRDF can be used for local description of metallic effect. In the case of analytical models additional problem accrues: the micro relief reconstruction or modeling of the sparkles visual effects. The main purpose of this paper is to develop algorithms and methods for determining the light-scattering properties of textured surfaces by photographs.

The input data in the proposed approach are the photographic images of a flat surface, made with various lighting conditions, corresponding normal maps to the surface, light sources and camera positions at the moment of shooting. At each input image the presence of a glare from light source is supposed. The presence of metallic effect is also allowed. The investigated material is assumed to be isotropic, not fluorescent.

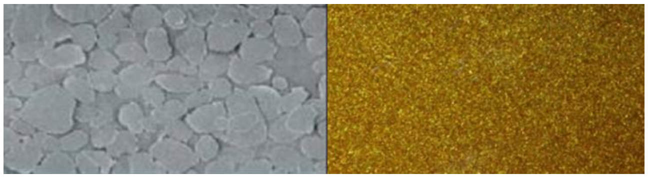

Figure 1: Set of particles (left); Metallic coating (right).

In our work we propose the material model which determines the light-scattering properties of surfaces and metallic effect. Model must satisfy the following requirements:

- Small number of photos for the material model construction;
- Interactive visualization;
- Small amount of memory for storage.

Algorithm for constructing of the proposed model from photographs taken under various lighting conditions was developed as a part of this work. The verification of the model and algorithms was made by comparison of the original photos with the results of visualization under corresponding lighting conditions.

In the following section there is an overview of existing methods of the modeling of the light scattering and reflective properties of surfaces. In **Section 3** the proposed model of a material is described. **Section 4** describes how to find model's parameters from photos. In **Section 5** the basic results of work of algorithm are presented. Finally, main results are provided in **Section 6**.

2. RELATED WORK

To obtain the material models manual modeling with some applications (BRDF Shop) can be used. Parameters of reflective properties of a material are adjusted by the designer. Manual modeling of real world materials can take a lot of time.

For automatic reconstruction of light-scattering properties often high-precision photometric devices or special acquisition systems are used [1]. The sample of the surface covered with the investigated material is illuminated from different directions with the ray of light and the reflected light for various observation angles is measured. The obtained data are usually presented in tabular form. Data storage requires large amounts of memory. For the modern real-time graphic applications the storage problem is still actual. In addition, the material is often assumed to be homogeneous, which eliminates the possibility of texture measuring.

A variety of methods has been developed to determine the textures of materials. Some of them are aimed at texture construction directly during rendering process, for example *BTF* [2], view-dependent textures [3]. Some use preliminary determining, for example texture weaving [4] and image-based texturing [5]. Bidirectional texture function (BTF) is a group of methods, implying the accumulation of large databases on types of surfaces. View-dependent textures method implies high-quality reconstruction of the texture in the rendering process. Input images are projected onto preliminary reconstructed scene. Methods of preliminary texture reconstruction closely connected with geometry reconstruction. Typically, these approaches are applied to use expensive systems of 3D-scanning. Texture weaving divides a texture into patches and for each patch links it to one of the original images. Texture weaving has a high speed because it uses hardware acceleration, and hash maps.

There are some systems for manual modeling of metallic effect (3dsMax, Photoshop, the system, described in [6]). A part of them don't assume interactive visualization, while others only simulate effect, without allowing the achievement of continuity when moving the light source or camera. In the paper [7] metallic parameters in the form of probability model are determined from the given BRDF table. But the approach does not solve the problem of reconstruction of the metallic parameters from photos. The main drawbacks of existing methods:

- Often special equipment is needed;
- Large amounts of data for rendering;
- Texture or only a homogeneous material reconstruction;
- Occurrence of artifacts related with direct illumination;
- Lack of continuity between rendering of the neighbor frames.

The proposed algorithm is designed to eliminate these shortcomings.

3. PROPOSED MODEL

This section presents the proposed model of light-scattering properties of a material containing metallic.

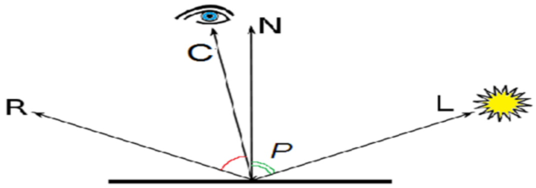

Figure 2: Vectors in proposed model.

Let Ω be a set of points. All examined points are assumed to belong to one of two classes. Either they belong to the investigated material, or to the metallic. That means:

$$\Omega = Material \cup Metallic,\ Material \cap Metallic = \emptyset$$

Let *p* be an examined point (Figure 2). $\mathcal{M}(p)$ is the model, that describe metallic. Then proposed parametric material model is:

$$M(p, \bar{L}, \bar{C}) = \mathcal{M}(p)\ \ /{+}/\ Phong(p)$$

/+/ - is a special addition operation, i.e. either *p* is point of metallic presence, or it contains only the texture and the specular component. To describe the light-scattering properties of the surface we use Phong reflection model:

$$Phong(p)=d(p)\cdot(\bar{n}(p),\bar{l}(p)) + s(\bar{r}(p),\bar{v}(p))^{\alpha}$$

Specular reflection parameters *s* and *α* are the same for all points. Parameter of the scattering power *d(p)* is different for every point and onwards will be named *texture*.

Consider the metallic model $\mathcal{M}$ in details. On the one hand, the metallic is a set of specifications, on the other - a set of particles. Our model combines these two views.

Each particle is presented in the form of an ellipse. It has geometric parameters and color parameters. Color is presented in the format L*a*b. The geometrical parameters of particles are: the area, the size of the major and minor axis, inclination angle to the horizontal axis. Color parameters are luminosity and color.

The probability metallic model $\mathcal{M}$ depends on the distribution of area, shape and color of the particles.

$\mathcal{M}$ $(Size(m,l,k), Shape(b_1,k_1,b_2,k_2), Color(R_E, G_E, B_E, R_D, G_D, B_D))$

$Size(m,l,k)$ - area distribution (m,l,k are parameters of approximated histogram function);

$Shape(b_1,k_1,b_2,k_2)$ - shape distribution (b_1,k_1,b_2,k_2 are parameters of limitation lines for semi-major axis);

$Color(R_E, G_E, B_E, R_D, G_D, B_D)$ - color distribution, where *E* means Expectation, and *D* - Dispersion of RGB.

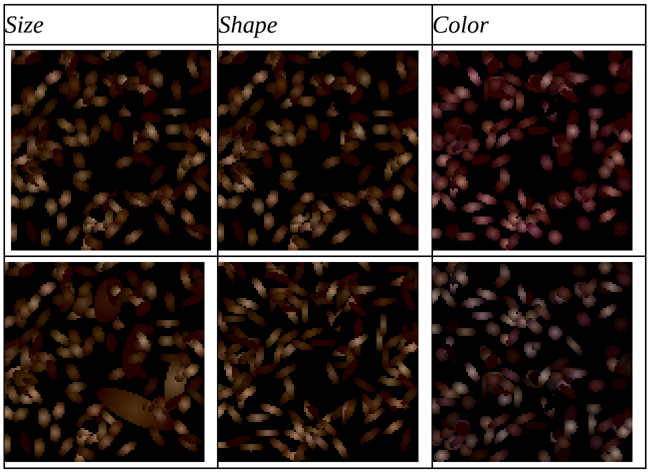

Figure 3: Metallic effect's dependencies from $\mathcal{M}$ parameters.

The proposed model $M(p, \bar{L}, \bar{C})$ takes a small amount of storage in memory, needs only one photo for construction an can be interactively rendered.

4. PROPOSED METHOD

Now we will consider the algorithm for constructing the proposed model from images. Algorithm receives *N* photos of flat surface which contains the texture and metallic. These photos should be taken with different light conditions and camera positions. The output consists of parameters of the proposed model (*T, s, n, Size, Shape, Color*).

Since the model is divided into two independent parts the algorithm of its construction, is also divided into two steps. Thus it is necessary to determine parameters of the Phong and *Metallic* models.

4.1 Determining of Phong Model

At first, consider the algorithm for texture and reflection coefficients of *s, n*. Since the original data contain some errors due to imperfection of the shooting conditions, surface roughness, digital noise, errors while constructing a scene, a bad approximation of the Phong model, the direct solution of linear algebraic equations is impossible. Therefore it is necessary to solve the minimization problem. It's required to find *d (p), s, n*, which minimizes the error function:

$$F(d(p),s,n) = \sum_{k=1}^{N}\left(I_k\,(p) - d(p)\cdot NL_k\,(p) + s\cdot NL_k(p)^{\alpha}\right)^2, \quad p \in Material.$$

We use an iterative algorithm. The overall process is shown in Figure 4.

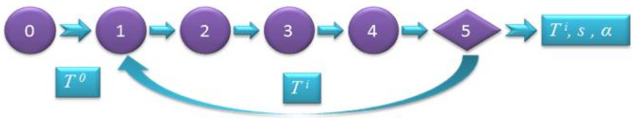

Figure 4: Determining Phong model parameters.

The algorithm iteration consists of five internal steps and of the choice of initial approximation. Parameters of specular component of model are being optimized within every iteration. Globally the process optimizes a texture as long as the quality of the constructed model can be improved. Now we postpone the question of choosing the initial approximation and consider the steps 1-5 in more detail.

4.1.1 Getting specular component

Denote the current iteration i. At the previous iteration there was obtained texture $T^{i-1}(p)$. Knowing it we can express the specular component as the difference between the original image and the diffuse component. Diffuse and specular components:

$$Diffuse(p) = T^{i-1}(p) \cdot (\bar{n}(p),\bar{l}(p)).$$

$$Specular(p) = Image(p) - Diffuse(p) = s(\bar{r}(p),\bar{v}(p))^{\alpha}$$

4.1.2 Selection of control points

Specular contain various inaccuracies associated with the approximate texture of $T^{i-1}(p)$ in addition source images also contain noise and various distortions. By optimization of the artifacts have a significant impact on the resulting parameters s^i, α^i. To get rid of these inaccuracies and reduce the number of data for further optimization process, we apply the special filter.

We divide the whole set of points into subsets. Each subset includes only points for which $(\bar{r}(p),\bar{v}(p))$ is the same. In each set we average all included color values. From each set we choose an arbitrary point and put it in a set of control points *Material'*. Each control point corresponds to the average color c_i.

4.1.3 Optimization of specular parameters

Now we solve the problem of optimizing the parameters s^i, α^i over the set of *Material'*. In order to optimize the desired parameters we use Levenberg-Marquardt method. Minimization is performed for each color channel separately. The functional to minimize is:

$$F(s,n) = \sum_{p \in Material'} \left(s \cdot (\bar{r}(p),\bar{v}(p))^{\alpha} - c(p)\right)^2,$$

c is the color of pixel p.

4.1.4 Recalculation of texture

Subtract from the original image reflectance component.

$$Diffuse(p) = Image(p) - Specular(p).$$

Recalculate the texture as an average:

$$T^i(p) = \frac{1}{N}\sum_{k=1}^{n}\frac{Diffuse_k(p)}{(\bar{n}(p),\bar{l}(p))}$$

Now we can visualize the images with obtained parameters:

4.1.5 The decision of completion

We calculate the error of recovering light-scattering properties of the material in the iteration i.

$$Err_i = \sum_{p \in Material}\sum_{k=1}^{N}\left(Image_k(p) - Image'_k(p)\right)^2$$

If the difference $|Err_{i-1} - Err_i|$ does not exceed the preassigned ε, then the algorithm stops, otherwise the process moves to the next iteration.

4.1.6 Choice of initial approximation

We put an initial approximation to zero at any point. This is equivalent to the recovering s and α initially from the original image. In this case, the convergence will be slow. In addition, the end result in this case is highly dependent on the type of input data. Requirement for the initial approximation are elimination of the primary defect and reduction in the number of iterations.

Divide the original images at $(\bar{n}(p),\bar{l}(p))$. For each point we take the minimum among all images. So we get the texture T_{min}.

$$T_{min}(p) = T^0(p) + X(p); \qquad s^0 \cdot \frac{(\bar{r}(p),\bar{v}(p))^{\alpha^0}{}_{min}}{(\bar{n}(p),\bar{l}(p))_{min}} \overset{\text{def}}{=} X(p)$$

We use the same considerations as in the steps of the basic algorithm but now we have another function:

$$F = s^0 \cdot (\bar{r}(p),\bar{v}(p))^{\alpha^0} - s^0 \cdot \frac{(\bar{r}(p),\bar{v}(p))^{\alpha^0}{}_{min}}{(\bar{n}(p),\bar{l}(p))_{min}} \cdot (\bar{n}(p),\bar{l}(p))$$

After s^0, α^0 have been found $X(p)$ and $T^0(p)$ can be calculated.

Now we can use $T^0(p)$ as an initial approximation for iteration algorithm. The result at the first iteration is significantly better than with zero initial approximation (Figure 5).

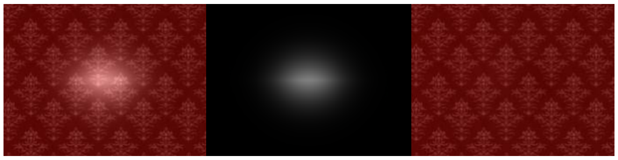

Figure 5: left- T_{min}, *middle- X, right -* T^0.

4.2 Determining of $\mathcal{M}$ Model

The input to the algorithm receives N images of a plane surface with a metallic and a synthesized image with the glare and texture without metallic. In this case, the synthesized image should be obtained under the same parameters as the original picture. The resulting algorithm images are synthesized based on the initial approximation to construct a model of Phong.

The whole process of the algorithm can be divided into 3 phases.

In the first phase, the input image is used to construct the image containing only metallic. The difference of input and synthesized images appears as such images.

In the second - from intermediate images a set of particles with different characteristics are determined, and also *Metallic* set. To construct a set of *Metallic* images obtained in the previous step, are adaptively binarized on the metallic and regular areas.

Analyzed images are converted into a format L*a*b. With the help of sequential scanning color images are searched for homogeneous areas. Each found region is considered as a particle. For particle main parameters are counted.

On the third - the parameters of inclusions are used to build model $\mathcal{M}$. To do this we construct histograms and compute probability characteristic for *Size, Shape* and *Color*.

To verify the determination of the metallic effect PSNR metric is not applicable, since the noise with the same parameters can modify the different pixels of the image. Consider example with two chessboards as a justification.

Figure 6: Chessboards.

This two pictures have PSNR = 1.7. However, from the observer's point of view, there is no structural difference between them. That is why another metric is proposed. It's named *MD* (*Metallic Difference)*. It has a single parameter R (window size). Its basic steps are following:

- Obtain high-frequency images
- Divided image into blocks of R * R pixels.
- Calculate a set of characteristics for each block;
- Summarize the difference between all blocks of the image;
- Normalize result value in bounds from 0 to 100.

The proposed metric has different from PSNR nature. The lower it is the more similar images are. On Figure 7 there are examples of the metallic reconstruction for metal plate with orange and blue metallic coating.

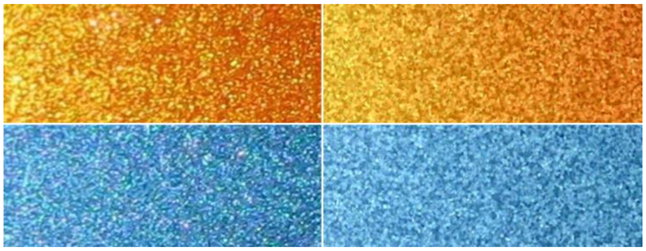

Figure 7: Left- photos of the metallic paint; right- synthesized images (*MD* = 4.05 and 5.40 correspondingly).

However, PSNR still can be used for comparison in case of no metallic presence.

5. EXPERIMENTAL RESULTS

Algorithms were verified on synthetic and real world materials. During tests on synthetic data we generally gains PSNR values between 48-60 dB (while comparing the metallic free initial and rendered images). Example of the input image and reconstructed material is shown on the Figure 9.
The results of the work of the algorithm on real data are illustrated in Figure 8.

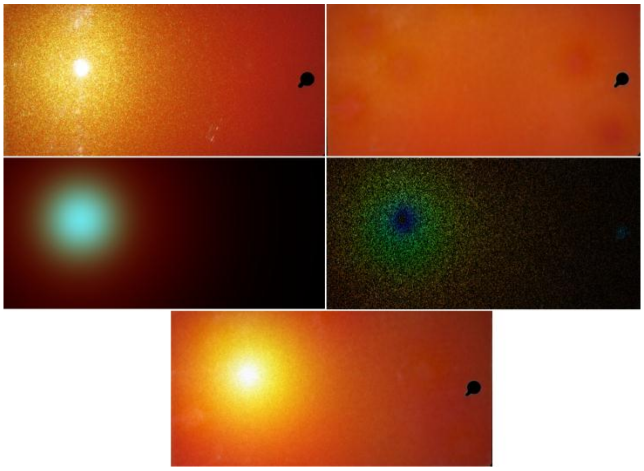

Figure 8: Left to right order**:** original image, generated texture, specular component, metallic component (is multiplied by 10), and the result image with the same light position (*PSNR=19.8, MD = 9.8).*

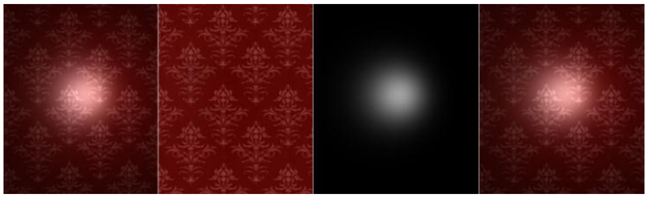

Figure 9: Original image, generated texture and specular component, synthesized image (PSNR=54dB).

6. CONCLUSION

As a result of work we proposed a model for representation of light-scattering properties for textured surfaces with metallic coating. Also we developed an algorithm for constructing the proposed model. The algorithm requires the input of a small number of photographs. However, it recovers with reasonable accuracy the texture and the corresponding parametric model lighting, that allows us to construct a simple probabilistic model of metallic and use it in the future for interactive visualization.

Nevertheless, calibration errors and irregularities of the surfaces are still affecting on the final results and measurement errors. A significant drawback is the requirement for pre-reduction of geometry. However, test results showed that despite the shortcomings, implemented approach has acceptable accuracy for realistic visualization of the constructed model.

7. REFERENCES

[1] Волобой А.Г., Галактионов В.А., Ершов С.В., Летунов А.А., Потемин И.С. Аппаратно-программный комплекс для измерения светорассеивающих свойств поверхностей *"Информационные технологии и вычислительные системы", № 4, 2006.*

[2] G. Müller, J. Meseth, M. Sattler, R. Sarlette and R. Klein. "Acquisition, Synthesis and Rendering of Bidirectional Texture Functions". *Eurographics 2004, STAR.* University of Bonn, Institute for Computer Science II, *2004.*

[3] P. E. Debevec, C. J. Taylor, and J. Malik. "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach". In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 11-20, New Orleans, Lousiana, 1996.

[4] M. Callieri, P. Cignoni, and R. Scopigno. "*Reconstructing Textured Meshes from Multiple Range RGB Maps"*, 7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002, Erlangen (D), Nov. 20 - 22, 2002

[5] Bornik A., Karner K., Bauer J., Leberl F., Mayer H. "High-quality texture reconstruction from multiple views". *The Journal of Visualization and Computer Animation 2001*; 12: 263-276

[6] S.Ershov, K.Kolchin, K.Myszkowski "Rendering Pearlescent Appearance Based on Paint-Composition Modelling", *Eurographics 2001 conf. proc., vol.20 (2001),* number 3.

[7] Aydın Öztürk, Murat Kurt and Ahmet Bilgili. "Modeling BRDF by a Probability Distribution", Proc. of Graphicon'2010, pp. 57-63, Izmir, Turkey, 2010.