User-Assisted Acquisition, Processing and Rendering of Material from Images

Peter Sikachev, Andrey Ilyin, Alexey Ignatenko
Computational Mathematics and Cybernetics Faculty
Moscow State University, Moscow, Russia
peter.sikachev@gmail.com, ailyin@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

Abstract

The acquisition of complex realistic models of reflective properties of materials and there photorealistic interactive rendering are still significant problems of computer graphics. The complexity of these problems is connected with the structural and reflectance properties of materials, which determine the human perception of this material. Nowadays there are a lot of methods for solving this problem [1].

In this paper we propose an image-based method which uses intuitively obvious user interaction and can robustly determine different material models of the surface of the object. For each material the proposed algorithm can construct the determinative Phong model that best fits reflective properties for this material. High-quality material models of real objects can be acquired using relatively little amount of input data. The presented system accurately reproduces reflective properties of objects and afterwards allows interactive rendering of objects with applied constructed materials. Note that the geometry shape reconstruction of the objects is not considered in this work.

Keywords: Image-Based Rendering, Material Reconstruction, BRDF, Phong's Model, Clusterization.

1. INTRODUCTION

The use of realistic models at all stages of the image synthesis is an essential condition of the photorealistic rendering.

It is impossible to generate these models manually along with the ever growing need for more visual complexity. Therefore automatic and semi-automatic acquisition of real world models techniques becomes more relevant.

In this paper we focus on the acquisition of realistic materials. In particular, we propose an acquisition of spatially-variant Phong model process that is efficient, reliable and requires minimum manual assistance. Other methods described in literature (see Section 2 for details) are either used for spatial-invariant material acquisition, or assume certain type of material (for example, clothes [16]), or need preliminary calculated values [2]. In our work Phong models are constructed with no assumption of the material type using intuitively obvious user interaction. Our main contributions are:

- robust and effective material reflective properties reconstruction process (Phong model reconstruction) based on photographs acquired from convenient digital camera;
- effective algorithm of "material" distribution among the object's surface;
- constructed material rendering at interactive frame rates.

The proposed algorithm needs relatively few photographs (about 15-25 for an object), which accelerates the acquisition process.

As a result of the data acquisition, reconstruction and clusterization processes we get a compact representation of the spatially varying material which suits well interactive rendering (for example, see Figure 8).

In the next section there is a brief overview of the existing methods. The description of acquisition and preprocessing data is given in Section 3. In Section 4 the reconstruction process is described. The obtained results and rendering methods are given in Section 5. The concluding remarks are in Section 6. Finally, future research plans on this topic are described in Section 7.

2. RELATED WORK

Nowadays image-based material representation receives growing attention in the computer graphics [1].

For the description of reflective properties of material several lighting models can be used: Phong [3], Blinn-Phong [3], cosine lobe [3], Torrance-Sparrow (and its modifications [4]), Lafortune [2, 5] and arbitrary BRDF [6]. At this stage of research we limit ourselves to Phong's model, but in the future (see Section 7 for details) we are going to use more complicated models.

Some methods assume the usage of complex devices during the acquisition step [4, 7], others need only the photographs and the calibration information of the camera (orientation, position) [1-3]. In our work we use methods similar to [2].

The hardware acceleration is also widely used for rendering and also for the acquisition process. Most of methods are optimized for modern GPUs [1, 8].

Methods which are working with arbitrary BRDFs often use parameterization and factorization [10-12].

3. DATA ACQUISITION

3.1 Data Obtaining

The material reconstruction system works with the following data types:

- geometry shape;
- set of photographs;
- light sources and cameras parameters by which photographs were taken.

As mention before, geometry reconstruction is not a problem of this work. For the shape acquisition of test objects the 3D scanner was used.

The photographs can be obtained from the conventional digital camera. Each image should be accompanied with light sources information and camera parameters (position, orientation etc) by which the photograph was taken. Image sequences can be divided in the following categories:

- fixed camera position
- varying camera position

When using the first type of photograph sequence the camera and light source parameters must be specified manually. See an example of such a sequence in Figure 1.

Figure 1: Photograph sequence with fixed camera parameters.

In case of using the second type of photograph sequences, camera and light source parameters can be found automatically. For automatic calibration special object is needed (a paper list with a 'check board' image). The determination of the camera parameters (camera calibration) using photograph sequence is done by using GML Camera Calibration Toolbox [15]. To determine the position of the light sources we make a photograph from the light source position and then calibrate the camera using this image. See Figure 2 for the second type of image sequence.

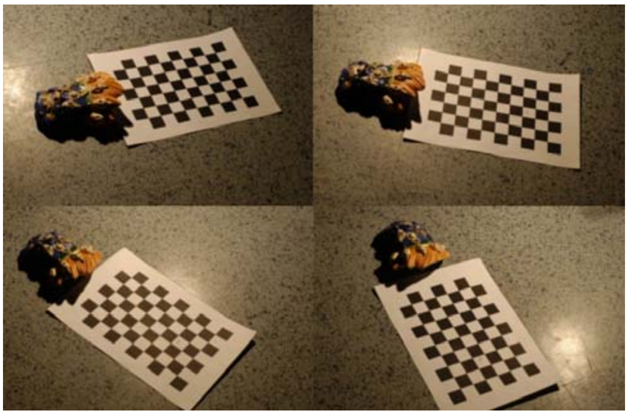

Figure 2: Photograph sequence with calibration object.

3.2 Normal Map Generation

The proposed material reconstruction algorithm needs an access to local geometrical properties of the object surface. This information can be encoded with normal maps. For each photograph from input sequence we generate a corresponding normal map. The color of each pixel in RGB color space is defined in Equation 1. See normal map example at Figure 3.

$$c_r = 128(n_x + 1),\ c_g = 128(n_y + 1),\ c_b = 128(n_z + 1) \quad (1)$$

$c_r,\ c_g,\ c_b \in [0,255]$ - red, green and blue components of the pixel color; $n_x,\ n_y,\ n_z$ - components of a unit normal vector to the surface of an object.

Figure 3: The normal map for a test object.

3.3 Initial User Mark-Up

One of the main features of our system is the user intuitive interface of initial mark-up. Mark-up is performed in the following way. First, the user defines the number of materials presented in the object. Each material is associated with the identifier (color that would be used to mark-up regions with it). Then the user marks regions that describe best the properties of material. 15-25 points is sufficient for most of materials. See Figure 4 for user mark-up example.

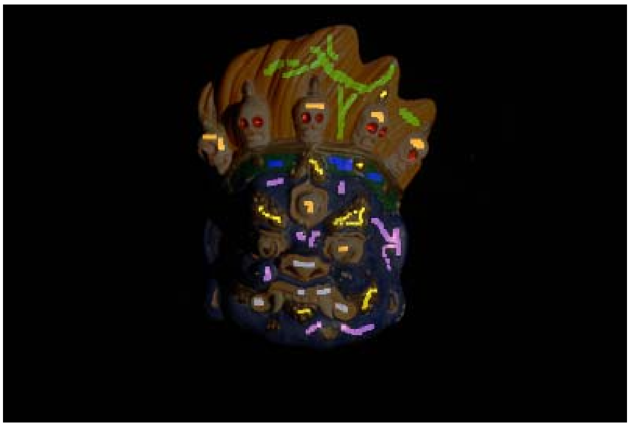

Figure 4: Initial marl-up (for better recognition the number of points is increased tenfold)

4. MATERIAL RECONSTRUCTION METHOD

In Section 3 we described the acquisition of the input data and its initial preprocessing. Let us describe our method of material reconstruction. At this stage of research, we limit ourselves to Phong's model of material (see Equation 2). The reconstruction step is divided into two steps. At first step for each user selected material we create the basic Phong's models (basic materials are created). At the second step the best material is associated with each region of the model (i. e. we clusterize object points amongst the materials).

Let us discuss in details proposed algorithms.

4.1 Basic Materials Creation

For each selected by user set of separate material points we should create a Phong's model, that describes best the reflectance properties of this material:

$$I_{pt} = k_a + k_d I_l(\vec{n},\vec{l}) + k_s I_l(\vec{r},\vec{c})^{k_{sh}} \quad (2)$$

I_{pt} - theoretical pixel intensity, I_l - light source intensity, k_a, k_d, k_s, k_{sh} - Phong's model factors, $\vec{n}$ - normal vector, $\vec{l}$ - light source vector, $\vec{r}$ - reflected vector, $\vec{c}$ - camera vector.

Phong's model is defined by calculating its factors k_a, k_d, k_s, k_{sh}. During this process optimization problem is solved where the following error functional is minimized:

$$E(m,p) = \sum_{\forall p \in m} \left(I_{pt}(p) - I_{pr}\right)^2 \quad (3)$$

$E(m,p)$ - the difference between real intensity and theoretical model, I_{pr} - real pixel intensity (acquired from photograph), m - set of pixels, which correspond to the reconstructed material (acquired from user mark-up).

This minimization problem is solved by means of Levenberg-Marquardt optimization [14].

At this step, for each set of pixels selected by user we create basic Phong's models or define basic materials. The problem of clusterization of other pixels amongst these materials described in next section.

4.2 Clusterization Amongst Basic Materials

At this step each region of the object's surface should be associated with one of the basic materials. In other words, object's points should be clusterized amongst basic materials. Points, selected by user are also reclusterized because little amount of them could have been selected improperly. Thus, during clusterization all points of model are examined.

For each point of material (pixel of photograph) the best basic material model (i.e. with minimum absolute difference) is selected. During this process attributes of each point (normal, color position etc) inputs in Phong's model and absolute difference between this (theoretical) model and real pixel color (on photograph) is calculated.

$$E(m,p) = (I_{pt}(m) - I_{pr})^2$$
$$m' = m : E(m,p) = \min_{\forall m \in M}(E(m,p)) \quad (4)$$

$E(m,p)$ - error for basic model M in point p.

5. RESULTS

5.1 Results of Reconstruction Algorithm Work

After clusterization of points amongst basic materials, for each input photograph, the mask of clusterized materials and corresponding error mask are created by the algorithm. During the material mask mark-up for each material its marker is used. See Figures 5 and 6 for example.

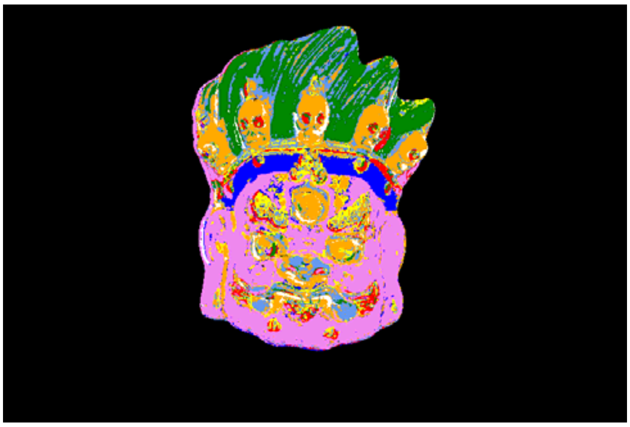

Figure 5: Material mask (result of material clusterization on the object's surface).

.

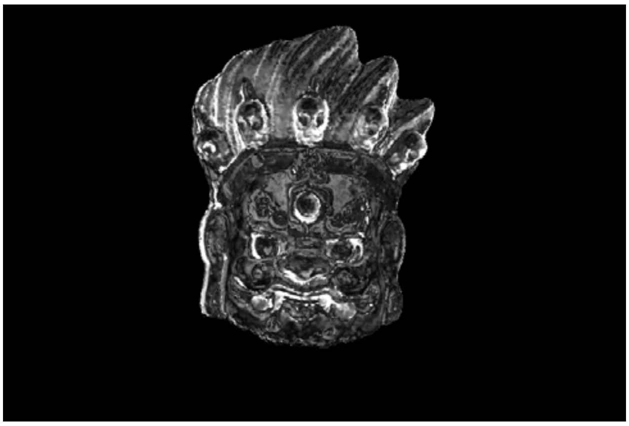

Figure 6: Error mask. More bright color corresponds to larger error between theoretical model and pixel on photograph. Colors are scaled to be visible.

5.2 Rendering

After we had created materials and clusterized them on the object's surface we can render the results. At this step we cycle through all triangles of the object and every triangle we associate with a single material. The material is selected form the material masks using the corresponding error masks. Triangle is assigned with a material with the least value in error mask.

Two images are given below: the first is camera made. The second is the result of rendering. Note that aliasing is due to the fact that the present rendering algorithm assigns to each triangle only one material.

6. CONCLUSION

A research of many existing material reconstruction systems was made and new methods of reconstruction were developed and implemented. During research we developed a software suite for reconstruction and rendering of materials which demonstrates that proposed algorithms work. In Section 7 we describe some possible trends of future work.

Figure 7: Object photograph.

Figure 8: Result of material reconstruction (rendering from the same viewpoint).

7. FUTURE WORK

One of the main limitations, greatly affecting results demonstration, is per-triangle mapping of materials. Blending of materials and texture mapping is currently unsupported. Implementing per-pixel material mapping is our primary task.

Below more resource-intensive and long-term tasks are listed.

We shall try to minimize run-time of reconstruction algorithm for achieving more interactivity. For this purpose part of specific calculations can be moved from CPU to GPU. Due to GPU fast handling of parallel processes ability, we can expect interactive representation of user changes (during material mark-up).

The reconstruction algorithm works with a photograph. The idea is to use powerful segmentation algorithms during initial mark-up. Brute-force applying of segmentation is useless (for example, specular glares would be selected as a 'new' material). Therefore this algorithm should also take in consideration the following parameters: camera and light sources positions, local surface properties (normal maps) of the object. Developing of such an algorithm is the matter of our future research.

Nowadays Phong's model does not provide the best definition of the material reflectance properties. It is also physically incorrect. For achieving more photorealistic results we are going to use more realistic and physically well-defined models. Then we shall reject simpler models and research acquisition and visualization of materials, defined with arbitrary BRDF.

8. REFERENCES

[1] *G. Müller, J. Meseth, M. Sattler, R. Sarlette, R. Klein "Acquisition, Synthesis and Rendering of Bidirectional Texture Functions" in Computer Graphics forum, Vol. 24, No. 1, March 2005, pp. 83-109.*

[2] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-Based Reconstruction of Spatially Varying Materials. In Rendering Techniques '01, London, Great Britain, 2001, pages 104-115.

[3] Hartmut Schirmacher, Wolfgang Heidrich, Martin Rubick, Detlef Schiron, and Hans-Peter Seidel. *Image-Based BRDF Reconstruction. In B. Girod, H. Niemann, H.-P. Seidel, Proc. Vision, Modeling and Visualization '99.*

[4] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. *Object Shape and Reflectance Modeling from Observation. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997.*

[5] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, Donald P. Greenberg. *Non-Linear Approximation of Reflectance Functions. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997.*

[6] Chris Wynn. *An Introduction to BRDF-Based Lighting, NVIDIA Corporation.* http://developer.nvidia.com/attach/6568

[7] Stephen Robert Marschner. *Inverse Rendering for Computer Graphics. Doctoral Thesis, 1998.*

[8] Chris Wynn. *Real-Time BRDF-based Lighting using Cube-Maps, NVIDIA Corporation.* http://developer.nvidia.com/attach/6567

[9] Jan Kautz, Michael McCool. *Hardware Rendering with Bidirectional Reflectances. University of Waterloo Technical Report CS-99-02, 1999.*

[10] Rafal Jaroszkiewicz. *Fast Extraction of BRDFs and Material Maps from Images. Master's Thesis, 2003.*

[11] Jason Lawrence, Szymon Rusinkiewicz, Ravi Ramamoorthi. *Efficient BRDF Importance Sampling Using A Factored Representation. ACM SIGGRAPH 2004 Papers. 496 – 505, 2004.*

[12] Kautz, J., McCool, M. D. *Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In Tenth Eurographics Workshop on Rendering, 281–292, 1999.*

[13] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidriech, Hans-Peter Seidel. *Image-Based Reconstruction of Spatial Appearance and Geometric Detail. ACM Transactions on Graphics, Pages 1–27, 2003.*

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.).* Cambridge University Press, 1992.

[15] *GML C++ Camera Calibration Toolbox* http://research.graphicon.ru/calibration/gml-c-camera-calibration-toolbox-5.html

[16] Kautz, J. *Approximate Bidirerectional Texture Functions.In GPU Gems 2, edited by Matt Pharr, pp. 177-187. Addison-Wesley.*