Optical Parameter Extraction using Differential Evolution Rendering in the Loop

Mauricio Olguin Carbajal, Ricardo Barrón Fernández, and José Luis Oropeza Rodríguez

Abstract— **Image synthesis is highly dependent on rendering algorithm and optical properties of scenario objects. The goal of this work is to develop a methodology to obtain some illumination parameters of a real scenario represented by an acquired image, and use these parameters for a virtual scenario rendering with the same objects as the original. The proposed methodology consists, first, in acquiring an image of the working scenario, and by using a DE (Differential Evolution) algorithm to render images that gradually approximate to the real acquired image, by some virtual scenario parameter modification based on the DE optimization. We call it "ED Rendering in the loop". Finally we use the obtained parameters to render an image to compare it with similar methods.**

Index terms—**Differential evolution, rendering, images, loop, illumination, model.**

I. Introduction

THE The goal of this study is the development and test for a methodology that can make the parameter extraction needed to render photorealistic images by using DE - rendering algorithm. The illumination model used greatly depends on virtual scenario optical parameters for final image rendering. Some of these parameters are: object color, reflection index, refraction index, texture, etc. Also we must define light position, intensity, color, attenuation index, etc. The illumination model uses these and many other parameters for final image rendering. This present study proposes a scenario parameters extraction methodology from a real scenario by using a more objective evaluation and procedure.

II. Related Work

Realistic image synthesis efforts have been focused in accurate and efficient rendering algorithm development, mainly based on Montecarlo [1] and analytic approximations. This algorithm has parameters which directly or indirectly represents optical properties of scenario objects. To achieve the rendered image having a realistic appearance, it is necessary for these parameters to be exact. There are few methods to measure or estimate optical parameters that are of relevance for computer graphics.

Manuscript received May 26, 2011. Manuscript accepted for publication August 24, 2011.

Mauricio Olguin Carbajal is with Postgraduate department, Centro de Innovacion y Desarrollo Tecnológico en Computo (CIDETEC), IPN, Mexico City, Mexico (e-mail: molguinc@ipn.mx).

Ricardo Barrón Fernández and José Luis Oropeza Rodríguez are with with Center for Computing Research (CIC), National Polytechnic Institute (IPN), UP "Adolfo Lopez Mateos", Mexico City, Mexico (e-mail: rbarron@cic.ipn.mx, joropeza@cic.ipn.mx).

A. Parameter Extraction

Existent methods for optical physics parameter measure require the use of very expensive equipment, for example the one used for scattering properties measure for colloidal chemistry [4] which give very little usable data for computer graphics. More recently there has been made efforts to measure or estimate objects optical physical properties for using these parameters in computer graphics with modern rendering algorithms. In accordance with Srinivasa G. Narasimhan et al. [2] there exist two object optical properties estimation methods: direct measurement and indirect estimation. For direct measure some optical methods like Goniphotometry have been used which measure phase function for translucent media scattering [7]. Indirect estimation uses an analytical approach [5] or numerical solutions for light transport [6]. Fukawa et al. [8] uses an acquisition device, based on laser range scanner to obtain a 3D model of objects texture, later this texture 3D model it will be used in its virtual objects. Also Gero Muller et al. [9] use an acquisition parameters system formed by a hemisphere with fixed cameras and lights which use massive parallel processing with no moving parts, this approach obtains an object 3D texture as well as the object color and reflection index. However the system requires expensive equipment with a lot of processing power.

Wojciech Matusik et al. [10] use a 6 cameras and a light array as an acquisition device, putting objects on a spinning table and uses a multi-background technique for alpha channel acquisition and mate environment for multiple viewpoints which achieves tridimensional object appearance reconstruction with color, reflection and refraction index. Later, these parameters could be used for virtual objects; however equipment and processing are still expensive.

G. Müller et al. [11] have designed a device with a single camera and a single light source which is still, while the camera moves in a semicircle to obtain reflection and color parameters of realistic 2d texture materials varying conditions namely Bidirectional Texture Function (BTF). This technique gives amazing results, however it is still an expensive and time consuming process. The great data amount generated by a system like this also requires a compression for acquired data. Non compressed data can require a lot of space (over 1 gigabyte store space for 81 views and 81 lights for 256x256 texture patch in accordance to Wai Kit Addy Ngan [3]).

Simpler approximations to measure optical parameters have been developed, for example Srinivasa G. Narasimhan et al. [2] made a simple device and technique to estimate scattering properties for a broad class of different media. An acquisition device is a small media containing tank with a light source inside, and an isolated chamber under the media where a camera measures scattering parameters effects, as well as a phase and absorption light index. Later the obtained parameters will be used on a Montecarlo rendering algorithm for photorealistic image synthesis, with impressive results. In this paper we present a simple acquisition device and use of a heuristic population technique (DE), for optical parameter estimation. Unlike previous approaches that require complicated setup or devices, our method and setup can be used to estimate the illumination parameters that the user requires using very few resources.

B. Rendering Parameters of Optimization

There are some previous works related to rendering parameters optimization, and here we show the most relevant. The first is a proposal to design the environment lighting by using optimization techniques applied to a rendering system that uses Radiosity based on an image synthesis system, developed by Kawai, Painter and Cohen [12]. This proposal is based on the illumination parameter optimization and works based on targets and constraints that the user defined to modify the environment lighting. This radioptimization system finds the “best” possible sets for: light source emissivity, elements reflexivity and light source directionality. The Kawai et al. proposal already has a pre-designed scenario where the parameters of objects such as color, hardness, specular reflectivity and diffraction are proposed by the user in advance, unlike ours proposed during the extraction of parameters. As an optimization method Kawai worked with a constrained optimization system that uses the BFGS (Broyden, Fletcher, Goldfarb and Shanno) algorithm, which evaluates the objective function and the gradient in the current step for the design of space to calculate a search direction.

Yu, Debeverec, Malik and Hawkins [13] as well as Yu [14], analyzes the reflection models recover problem for realistic scenes from photographs. This method recovers real scene reflectance properties for all the surfaces by using photograph sets, and then rendering a virtual version of the real scene in which textures are mapped to the scene from the real one, but which responds to virtual lighting conditions. The goal is to find the parameters of the BRDF (Bidirectional Reflectance Distribution Function), for use in rendering. This method allows adding arbitrary changes to the structure and lighting, such as extra items. The system input is a geometric model of the scene and a set of high dynamic range photographs taken with known direct illumination.

Once the parameters are obtained and the image is rendered, we proceed to compare the generated images with real scenario images, both in original and in new conditions; the result is that the methodology adequately predicts the resulting image under new lighting. The parameters obtained are the diffuse and specular reflectivity for red, green and blue color components. The problem of inverse lighting has also been approached by Schonenman et al. in their article "Painting with light". Their solution is proposed so the stage designer uses a "light brush" with which the user can specify areas of an image rendered from the scene which they wish to illuminate with a certain level of intensity. The system looks for the best configuration of lights to illuminate the scene by minimizing the difference between the rendered scene and the desired lighting. As a rendering algorithm they use ray trace.

Finally, Elorza and Rudomin [15], develop a proposal which makes lighting design by image rendering in closed environments, based on a solution to the problem of inverse lighting, using a genetic algorithm as optimization technique and an algorithm for radiosity as a rendering technique. The parameters optimized are the number, position and intensity of light sources used in the scene. The goal is to find the parameters that render an image by minimizing energy use and maximizing lighting.

Our proposal, unlike the previous ones, gets a broader set of parameters such as: rates of diffuse reflection, diffraction, roughness, and brightness, and the parameters of the light source such as position and intensity.

III. RENDERING IN THE LOOP

Our proposal is a methodology for optical parameter estimation, based on a simple acquisition image device, a two image comparison function (one acquired and one rendered), and DE algorithm that gradually finds good optical input parameters for our rendered image and compares with our acquired image (Aimage). Thus, we can obtain our particle fitness value depending on its similarity with reference image. As DE generations pass, rendered images reduce its difference values respect to Aimage. At run end we have a similar set of images and a file with the scenario parameters needed to render these images. These parameters could be used to render new images at greater resolutions and then used to render more complicated environments or animations. As we can see, figure 1, receive two inputs, first, a virtual scenario describing the objects, positions and orientations of the real one; second, a reference image (Aimage), which is our target image.

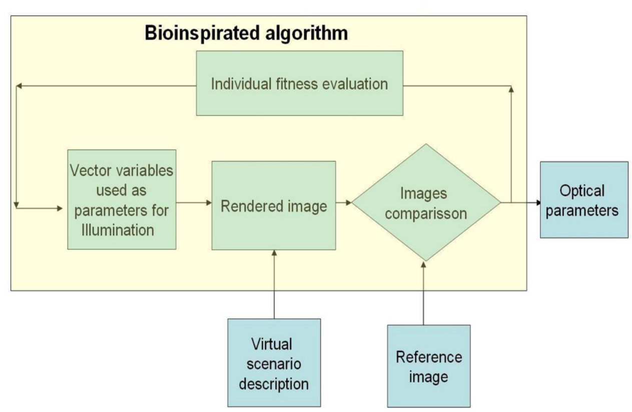

Figure 1. Metodology block diagram.

The normal form of parameter estimation can be conceptualized as an open-loop system, where you have a system that performs the extraction/estimation of parameters, receiving input data to be used, but making the comparison out of the system. However, our proposed methodology can be conceptualized as a closed-loop in which the comparison step is not carried out, but included in the parameters estimation.

A. Image Acquiring System

Image acquiring system, fig.1a, consist of a 15x15x20 cm. box, which has 2 high luminosity led's as a light source, as well as a hole to a camera, for image acquisition. At 6 cm. from the bottom of the box we put a false floor section to sustain objects for parameter extraction.

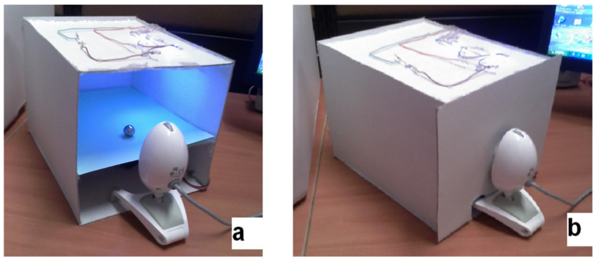

Figure 2. Acquisition image system.

To minimize external reflections we put a front side cover, fig. 1b, in a way that only the lens of the camera is visible, and not all the camera. For comparing both images we propose eq. 1. Comparison function gives one value for each RGB component. With a high component value, the difference between both images is big; with a little component value the difference is small.

$$\text{Dif} = \sum_{i=0}^{i=n} | \text{RGBObj}_i - \text{RGBTraza}_i |$$

where:
n = Xmax * Y max,
RGBObj_i = Each pixel *i* component objective image value,
RGBTraza_i = Each pixel *i* component rendered image value.

B. Proposed Algorithm

We present the pseudo code for the rendering in the loop algorithm as follows:

```
ImageO = Acquisition image function
Initialize individuals
   For each individual
      For each variable
      Select one random value between 0.0 and 1.0
        while (Generation_number, or stop criterion is not
        reached)
           for individual=1 to NumberOfIndividuals
              Calculate Fitness
                 Use individual variables as raytracer virtual
                  world parameters
                 Run raytracer and render imageT
                 Compare ImageO with ImageT
              Calculate Fitness
              If (U_Test_vector_fitness > XiG_vector_Fitness)
                 Assign U_Test_vector_fitness to
                 XiG_vector_Fitness
              Calculate Global Best
           End individual loop
        Next Generation until stop criterion is reached
```

The proposed algorithm receives a reference image that has been obtained by the acquisition system and stored into a 128x96 matrix called "MatrizObj". Then we initialize the DE individuals in order to generate a population to evaluate the aptitude of each individual. The evaluation consists of two steps:

- Using the X vector of one chosen individual, we assign to each variable vector a parameter from a virtual world. The virtual world is used as input for a raytrace algorithm to generate an image of 128 x 96, stored in a Matrix named "MatrizTraza".
- We call comparison function for two images, taking as input both matrixes "MatrizObj" and "MatrizTraza" to generate three numeric values, which is the difference between two images RGB values. These values will be stored as individual fitness. At the same time we store in a BMP file the best individual values from each generation. Only one image will be saved per each 10 generations.

The DE evaluates every individual, then selects the best and compares the aptitude from random and test individuals and selects the best from both. The best individual information will be used by DE to calculate the global best. At the same time we store in a TXT file the global best individual values from each 10 generations.

As a rendering algorithm, we use a raytracing with Phong-Blinn shading. This basic illumination model is used due to the low amount of mathematical calculations needed, compared with more complex models. We know that final image realism could be limited, but it is enough for our methodology test purposes. We must consider the great amount of images generated because every individual will generate one raytracing image. Also we have many individuals (60 to 80), test vectors and five hundred generations per test.

C. Setting Virtual Scenario

The virtual scenario must be constructed by hand; this means that we must put the plane, the sphere and camera so they are in the same place as the original image. Then we render a sample image to compare with the original. If they have distance and size differences, we adjust the virtual scenario until both images are virtually equal.

IV. TESTS AND RESULTS

Our virtual world is composed of a simple flat floor, a yellow plastic sphere and a light source. We capture the objective image as follow, fig 2.

Figure 3. System acquired objective image.

The variables vector of each DE individual has a correspondence with the virtual world light source, so the first three variables will have the X, Y and Z values of the light source position, and the fourth variable will be the intensity of the light source. Others variables are the sphere and plane HSV color component values.

The DE we use has a random offspring selection, with one pair of selected solutions and a binary recombination type. This means, that we use a DE/rand/1/bin. Also DE is a multi-objective algorithm looking to minimize RGB difference values, each corresponding to a function minimizing the generated image from the raytracing. We have a light dominance, with two or three elements. We conducted ten program runs, with a stop criterion of 500 generations, with 80 individuals. It uses a CR of 0.7, and F of 0.6 values. Now the results of one run are presented as follows, Fig. 3.

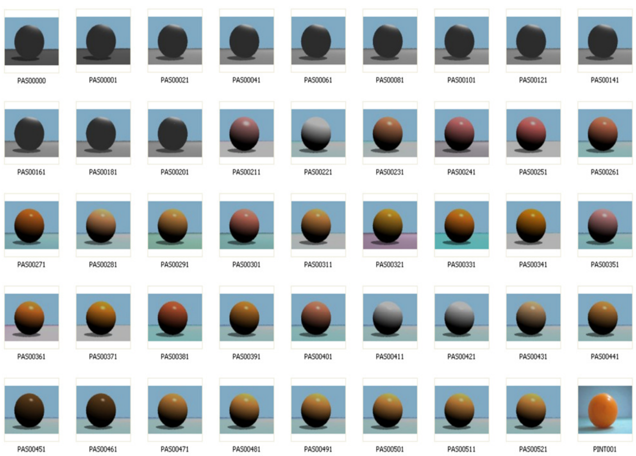

Figure 4. Evolution of generated images, we present the best image of 10 generations, the number of each interaction is shown.

Fig, 3 shows the evolution for parameter estimation; in first place the light source parameters search, running from PAS00000 to PAS00201, two hundred generations to find good position and intensity for the light source. In second place, from PAS00211 to PAS00521 we search for the object optical parameters like color, reflection index, refraction index, roughness, and shininess, for both the sphere and the plane.

The algorithm searches the more fitting positions for the light source as well as intensity; this can be seen in the relative position of the sphere's shadow in the last generation. Finally the sphere's and floor color fitting takes more time but gives good results considering the shading model used, as seen in Fig.4.

The set of experiments includes different sphere colors, in fig. 5. We use red, green and blue spheres to test the primary colors; afterward we use another four spheres to test other common colors such as white, yellow and orange.

Every experiment gives us a final image that we use to compare with the acquired reference image, Table I.

These results show a best average of 5.57 for the red sphere and worst (10.66) for the orange. These are good results considering the shading model. If we use a better model we can obtain better results. Also we made a set of experiments including a rendered scene with the same shading model and used the methodology to estimate the optical parameters user defined. Here are the target objects, Fig. 7.

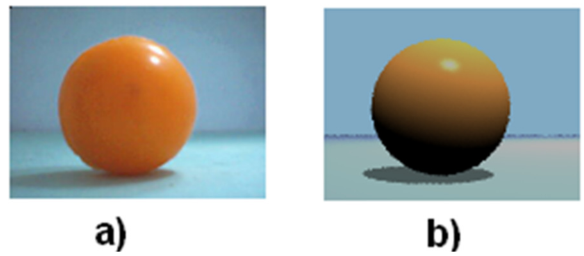

Figure 5. a) Real yellow sphere, b) Virtual rendered sphere.

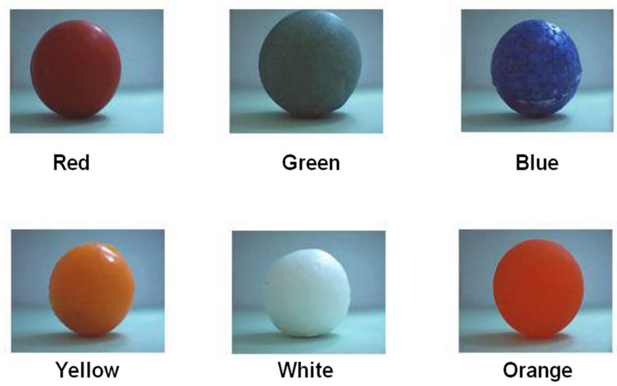

Figure 6. Target real scenarios used to test the metodology.

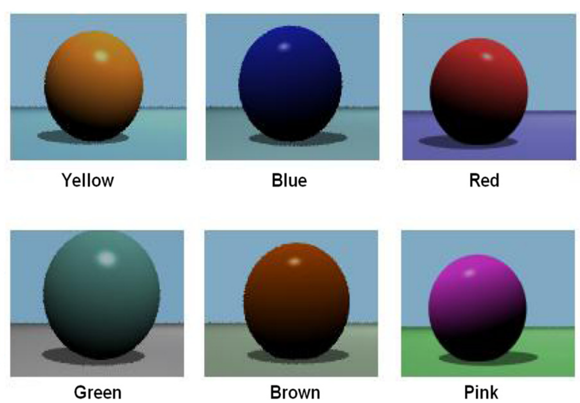

Figure 7. Target virtual scenarios used to test the metodology.

TABLE I
COMPONENT AND AVERAGE DEVIATION FROM ORIGINAL IMAGE COMPARED TO THE BEST GENERATED CORRESPONDING IMAGE

	Deviation			
Sphere	%R	%G	%B	Average
Red	6.25	5.25	5.20	5.57
Green	6.93	6.18	4.39	5.83
Blue	10.84	9.52	7.43	9.26
Yellow	7.29	6.95	10.17	8.14
White	12.61	10.55	6.29	9.82
Orange	8.79	7.23	15.96	10.66

The results for this set of experiments give us lower deviations with respect to the previous set, Table II. The methodology obtains a best average of 2.63 for the blue sphere and a worst 6.79 for the pink.

When we compare the original with the final image the differences are minimal, Figure 8.

TABLE II
COMPONENT AND AVERAGE DEVIATION FROM RENDERED ORIGINAL, COMPARED TO THE BEST GENERATED CORRESPONDING IMAGE

	Deviation			
Sphere	%R	%G	%B	Average
Red	6.25	5.25	5.20	6.21
Green	6.93	6.18	4.39	5.30
Blue	10.84	9.52	7.43	2.63
Yellow	7.29	6.95	10.17	5.77
Brown	12.61	10.55	6.29	3.54
Pink	8.79	7.23	15.96	6.79

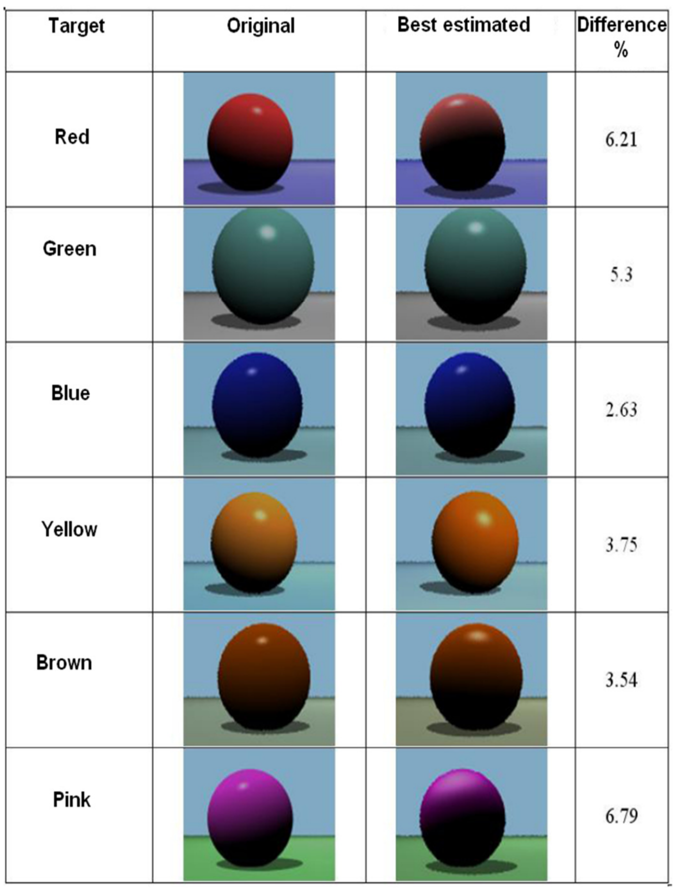

Figure 8. Original image and best individual generated image comparisson.

TABLE III
COMPARISSON, OUR BEST AND WORST VERSUS OTHER EXTRACTION METHODS

	Deviation			
Method	%R	%G	%B	Average
Wai Kit Addy 1	8.92	9.55	8.54	8.97
Wai Kit Addy 2	12.41	9.67	8.58	10.22
Ward	3.30	3.48	3.14	3.30
Ward-Durn	2.20	2.32	2.25	2.25
Our worst	**7.43**	**5.41**	**7.54**	**6.79**
Our best	**1.83**	**2.10**	**3.96**	**2.63**

We made a comparison with other extracting parameter methods. Comparing the original image and the generated one we obtained good results, Table III.

V. CONCLUSIONS

Our methodology searches for two relevant optical parameters, the light source parameters, and the object optical parameters, and obtains good results. The floor's and Sphere's colors is near to the original, nevertheless the position of the light source and its intensity.

The deviation showed by our methodology is close to the best results obtained for other methods (our best 2.63, global best 2.25), but it must be considered that other methods do not search for light source parameters. Our method can obtain more global parameters than any other without expensive equipment, obtaining competitive results. This encourages us to continue with more experiments making adjustments to the algorithm. We think it's possible to include other object parameters like a variable for bump mapping. Also is possible to use a meta-heuristic, and could be good to use an image preprocessing searching for edges, curves and image orientation.

REFERENCES

[1] Y. Willems P. Dutre, F. Suykens, *Optimized Monte Carlo path generation using genetic algorithms*, Report CW267, May 1998.

[2] Srinivasa G. Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K. Nayar, HenrikWann Jensen, "Acquiring Scattering Properties of Participating Media by Dilution," *ACM Transactions on Graphics* (SIGGRAPH'2006), 2006.

[3] Wai Kit Addy Ngan, *Acquisition and Modeling of Material Appearance*, S.M., Massachusetts Institute of Technology, Phd. Thesis. 2004.

[4] E.G. Finsy, and J. Joosten, "Maximum entropy inversion of static light scattering data for the particle size distribution by number and volume," *Advances in measurements and control of colloidal processes*, Butterworth-Heineman, Ch. 30, 1991.

[5] H. Jensen, S. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proc. SIGGRAPH 01*, 2001, pp. 511–518.

[6] B. Sun, R. Ramamorthi, S.G. Narasimhan, and S.K. Nayar, "A practical analytic single scattering model for real time rendering," *ACM Trans. on Graphics (SIGGRAPH)* 24, 3, 1040–1049, 2005.

[7] Fuch E. and Jaffe J. S. "Thin laser light sheet microscope for microbial oceanography," *OPTICS EXPRESS* 10 (2), 145–154, 2002.

[8] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance based object modeling using texture database: acquisition, compression and rendering," in *Proceedings of the 13th Eurographics workshop on Rendering*, Eurographics Association, 2002, pp. 257–266.

[9] Gero Müller, Gerhard H. Bendels, Reinhard Klein, "Rapid Synchronous Acquisition of Geometry and Appearance of Cultural Heritage Artefacts," in *6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*, 2005.

[10] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan, "Acquisition and Rendering of Transparent and Refractive Objects," in *Thirteenth Eurographics Workshop on Rendering*, 2002.

[11] G. Müller, J. Meseth, M. Sattler, R. Sarlette and R. Klein, *Acquisition, Synthesis and Rendering of Bidirectional Texture Functions*, EUROGRAPHICS 2004 State of The Art Report, 2004.

[12] James S. Painter, John K. Kawai, and Michael F. Cohen, *Radioptimization – goal based rendering*, 1993.

[13] Jitendra Malik, Yizhou Yu, Paul Debevec and Tim Hawkins, *Inverse global illumination, Recovering reflectance models of real scenes from photographs*, 1999.

[14] Yizhou Yu. *Modeling and editing real scenes with image-based techniques,* 2000.

[15] Isaac Rudomin Goldberg, Joaquin Elorza Tena, *An interactive system for solving inverse illumination problems using genetic algorithms*, 1997.

[16] Brian Smits, James Arvo, Chris Schoeneman, Julie Dorsey, and Donald Greenberg, "Painting with light," in *SIGGRAPH '93 Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, July 1993.