Genetic Algorithms for Estimation of Reflectance Parameters

Belen Masia* Adolfo Munoz† Alfonso Tolosa Oscar Anson‡ Jorge Lopez-Moreno§ Jorge Jimenez¶
Diego Gutierrez‖
Universidad de Zaragoza, Spain

Figure 1: Evolution of the result of the genetic algorithm as the number of generations increases. *Left:* original image from which the Buddha's reflectance properties are meant to be captured. *Rest (from left to right):* partial results every ten generations, showing convergence to the solution.

Abstract

Most of the current appearance acquisition methods require the use of specialized equipment and/or involved capture sessions. We propose a single-image approach based on genetic algorithms which greatly simplifies the process, and allows to capture reflectance properties of both opaque and translucent objects. Given the under-constrained nature of an image-based approach, we leverage two well-known illumination models, Phong and the diffuse approximation, to reduce the high-dimensional parameter space. We additionally explore this reduced parameter space to analyze the resulting behaviour of the algorithm.

CR Categories: I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Reflectance; I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shading;

Keywords: appearance acquisition, inverse rendering, BSSRDF, reflectance, genetic algorithm

1 Introduction

Realistic image synthesis requires precise reflection and scattering models of real-world materials. As rendering algorithms become more sophisticated, efficiently simulating all aspects of light transport, a new area of research has gained importance over the last few years: appearance acquisition. Capturing the appearance of an object implies obtaining its BRDF or BSSRDF (for translucent objects), in order to be able to model the interaction of light with that object.

In a traditional direct rendering approach the lights, material, camera position and geometry of a 3D scene are known parameters used in the generation of the final 2D image. However, in many cases it is useful to obtain unknown information of the 3D scene from a rendered image, a problem known as *inverse* rendering. This includes inverse lighting (i.e. estimating the position and characteristics of the light sources of the scene), estimation of the camera position and orientation, obtention of the geometry of the scene, and appearance acquisition. As the complete problem of inverse rendering is highly under-constrained, previous knowledge of any of these (lights, geometry, camera position or appearance) is usually leveraged to determine the rest of them.

In this paper we will explore the specific problem of *appearance acquisition*, assuming that the rest of the information of the 3D scene is known, and starting with a single image as input. To do so, our formulation is that of an optimization problem. Starting from the initial image, successive images are rendered and compared with the original, until the objective function, defined as the error between both images, falls below a certain value (or alternatively until a maximum execution time is exceeded). We therefore do not present a method for real acquisition of BSSRDF parameters, but a way to estimate those parameters using an image of the object and rendering a virtual scene.

The method proposed to solve this optimization problem is based on genetic algorithms. Genetic algorithms are inspired in biological evolution: each string of parameters to optimize is analogous to a chromosome, and the way in which these strings are generated and evaluated when searching for the best solution applies the concepts of natural selection, reproduction and mutation. Whenever solving an optimization problem, falling into local minima of the objective function is always a concern, and in our case the objective function, i.e. difference between the original image and our rendered image, has a large number of them. Statistically, genetic algorithms have been demonstrated to be less prone to this problem than other well-known optimization methods, as mutation favours diversity, increasing the probability of overcoming local minima. We show how genetic algorithms can be used to capture the appearance of an object in an image. We also provide insight into how to configure the parameters of genetic algorithms when applying them to the

*e-mail:belenmasia@gmail.com
†e-mail:adolfo@unizar.es
‡e-mail:oanson@unizar.es
§e-mail:jlm@unizar.es
¶e-mail:jim@unizar.es
‖e-mail:diegog@unizar.es

specific problem of appearance acquisition, and their influence on the final result.

2 Previous Work

An obvious choice to measure general reflection properties is using a gonioreflectometer [Li et al. 2006]. However, a complete characterization of a spectral, anisotropic BRDF may require up to 10^5 samples, so several optimization strategies have been introduced. By using a camera instead of a single photoreceptor, lots of samples can be obtained simultaneously [Ward 1992]. However, calibration issues need to be considered, which make measurements less precise.

More general solutions that include sub-surface scattering capture typically use complex measuring equipment [Matusik et al. 2002; Debevec et al. 2000; Goesele et al. 2004; Peers et al. 2006; Tariq et al. 2006]. Image-based approaches, while simpler in conception, usually require large sets of data acquired from different angles and/or lighting conditions [Yu et al. 1999; Lensch et al. 2003; Shen and Takemura 2006; Ghosh et al. 2008; Donner et al. 2008]. Reduction of these sets can be achieved by adding some knowledge of the geometry of the object whose optical properties are being captured [Boivin and Gagalowicz 2001].

Wu and Tang [Wu and Tang 2006] separate the sub-surface scattering component of a BSSRDF, starting from a single image together with a set of diffuse priors. Other methods to capture a generalized BSSRDF from single images impose constraints on the positions of the camera and light sources [Wang et al. 2008]. We refer the reader to the excellent work by Weyrich and colleagues [Weyrich et al. 2008] for a more comprehensive overview of appearance acquisition techniques.

In our work, we are interested in exploring the feasibility of appearance acquisition of complex materials by using genetic algorithms. This approach has been successfully used before in the field of computer graphics for texture synthesis, analysis and parameterization [Sims 1991; Salek et al. 1999; Qin and Yang 2002], image-based simulation of facial ageing [Hubball et al. 2008], image recognition [Katz and Thrift 1994; Koljonen and Alander 2006], or extraction of geometric primitives [Roth and Levine 1994].

3 Genetic algorithms

Genetic algorithms, first introduced by John Holland in 1975, are probabilistic heuristic algorithms for search and optimization which apply the concepts of biological evolution: natural selection, reproduction and mutation. As any other optimization method, the algorithm tries to find a set of variables, $(x_1, x_2, ..., x_n)$, so that the objective function, $F(x_1, x_2, ..., x_n)$, reaches its maximum (or minimum). This section gives an overview of how these algorithms work, but we refer the reader to Winter and colleagues' work [Winter et al. 1995] for a more comprehensive explanation on genetic algorithms and their application.

Given that each possible set of input variables $(x_1, x_2, ..., x_n)$ is equivalent to a chromosome (i.e. an individual) and each parameter x_i is denominated $gene_i$, the analogy with the theory of evolution is immediate: starting from a population of n chromosomes, each of them delivers a solution to the problem, and only the chromosomes yielding the better solutions survive to produce the next generations and perpetuate their *genetic material*. Genetic diversity is completed by sexual reproduction and random mutations.

The algorithm consists of four steps: initialization, selection, reproduction and termination. Selection and reproduction are iterated until the condition for termination is reached.

Initialization. The first step implies the creation of an initial population of individuals (or sets of variables corresponding to the parameters we want to estimate). The genes of these individuals are generated randomly within the search space. Both the size of the initial population and the limits of the search space are input parameters to the algorithm and have a great influence on its performance, as analyzed in Section 4.3.

Selection. In order to apply the principle of natural selection, it is necessary to evaluate the performance of each generated individual. To do this, each individual is assigned a rating, called *fitness*, representing the proximity of that individual to the solution. Chromosomes are then ordered according to their *fitness* and the ones with the lowest *fitness* values are eliminated and substituted by the descendants of the surviving chromosomes (*based-on-rank selection*). This way only the genetic material delivering the best results is perpetuated.

Reproduction. This step entails the creation of the next generation using two genetic operators: *crossover* and *mutation* (see Figure 2). Crossover is a genetic operator used for exchange of genetic material, in which two chromosomes are randomly selected and an exchange of genes between them is performed. Mutation, on the other hand, ensures genetic diversity from one generation of individuals to the next by randomly modifying the value of some genes.

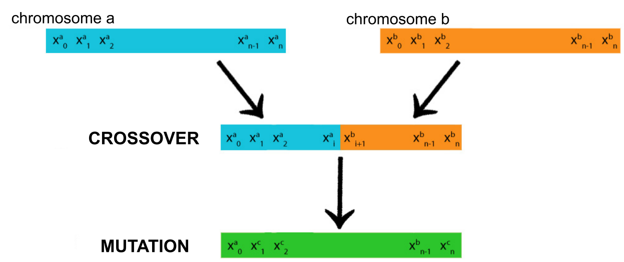

Figure 2: Crossover and mutation.

Termination. Typical termination conditions of the iterative process are a solution being found which satisfies a certain minimum criterion, the specified maximum number of generations being reached or the solution found not being able to be improved any further.

In the following section, we present our adaptation of the genetic algorithms approach to the problem of appearance acquisition, and comment on some implementation details.

4 Appearance acquisition

To be able to run genetic algorithms for appearance acquisition, we first need to define the variables and the objective function. In our work, the variables are the parameters of the rendering model to be used, whilst the objective function is the difference between the input image and the result of each iteration. In order to reduce the dimensionality of the problem, we assume that other parameters such as lighting or the geometry of the object are known.

4.1 Variables and objective function

Our method works both for opaque and translucent materials, but the parameters differ in each case. For opaque materials, interaction of light with the surface of the object is rendered using the *Phong*

η	Relative index of refraction
σ_s	Scattering coefficient
σ_a	Absorption coefficient
p	Normalized phase function
$\sigma_t = \sigma_a + \sigma_s$	Extinction coefficient
$\sigma_s' = (1-g)\sigma_s$	Reduced scattering coefficient
$\sigma_t' = \sigma_a + \sigma_s'$	Reduced extinction coefficient
$\alpha' = \sigma_s' + \sigma_t'$	Reduced albedo
$\sigma_{tr} = \sqrt{3\sigma_a \sigma_t'}$	Effective extinction coefficient
$F_{dr} = -1.440/\eta^2 + 0.710/\eta + 0.668 + 0.0636\eta$	
$A = (1+F_{dr})/(1-F_{dr})$	
$z_r = 1/\sigma_t', z_v(1+4A/3)$	
$r = \|x_i + x_o\|$	
$d_r = \sqrt{r^2+z_r^2}, d_v = \sqrt{r^2+z_v^2}$	

Table 1: Symbols used in the formulation of the dipole model.

model [Phong 1973]; the illumination on a certain point p on the surface is obatined as the sum of the ambient, diffuse and specular components as:

$$I_p = k_a I_a + \sum_{\text{lights}} (k_d(\vec{L} \cdot N) I_d + k_s(\vec{R} \cdot \vec{V})^{\alpha} I_s) \tag{1}$$

$\vec{L}$ indicates the direction of the rays of light from a light source to a point of the surface, N is the normal to the surface, $\vec{R}$ indicates the specular direction and $\vec{V}$ the direction joining the point and the camera. I_a, I_d and I_s are the specular, diffuse and ambient intensities, respectively. The parameters which need to be estimated by the algorithm are the ambient, diffuse and specular reflection constants k_a, k_d and k_s plus the Phong exponent α.

On the other hand, when working with translucent materials the presence of subsurface scattering requires a more complex illumination model. We have used the diffuse approximation model described by Jensen et al. [2001] to render translucent materials, which decouples single and multiple scattering. Single scattering is obtained in a precise way, whereas multiple scattering is approximated by means of dipole diffusion. The complete BSSRDF describing the outgoing radiance at point x_o in direction $\vec{w}_o$ is thus the sum of both components:

$$S(x_o, \vec{w}_o) = S_d(x_o, \vec{w}_o) + S^{(1)}(x_o, \vec{w}_o) \tag{2}$$

where S_d and $S^{(1)}$ represent multiple and single scattering respectively. These terms are given by:

$$S_d(x_o, \vec{w}_o) = \frac{1}{\pi} F_t(\eta, \vec{w}_o) \int_A R_d(\|x_o - x_i\|) E(x_i) dA(x_i) \tag{3}$$

$$S_o^{(1)}(x_o, \vec{w}_o) = \sigma_s \int_{2\pi} \int_0^{\infty} F p(\theta) \Psi(s_i' + s) S(x_i, \vec{w}_i) ds d\vec{w}_i \tag{4}$$

where $E(x_i)$ represents incoming irradiance, $F = F_t(\eta, \vec{w}_o) F_t(\eta, \vec{w}_i)$ is the product of two Fresnel transmittances, s_i' y s indicate scattering paths and Ψ is an exponential attenuation function. R_d is the diffuse reflectance function, which is computed as (see Table 1 for a definition of all the symbols):

$$R_d(r) = \frac{\alpha'}{4\pi}[z_r(\sigma_{tr} + \frac{1}{d_r})\frac{e^{-\sigma_{tr} d_r}}{d_r^2} + z_v(\sigma_{tr} + \frac{1}{d_v})\frac{e^{-\sigma_{tr} d_v}}{d_v^2}] \tag{5}$$

Using this formulation (for the complete details and derivation of these equations, consult Jensen et al. [2001]), it can be shown that the dipole model depends on only four parameters [Xu et al. 2007]: σ_s (scattering coefficient), σ_a (absorption coefficient), η (relative index of refraction) and p (normalized phase function), which may additionally show spectral dependencies.

Finally, for both opaque and translucent materials, we define the objective function simply as the error between the generated image at each iteration and the original image.

4.2 Implementation of the algorithm

We provide here some insight on how the genetic algorithms framework maps to our appearance acquisition problem. A discussion of the influence of the specific parameters is provided in subsection 4.3.

The first step of any genetic algorithm is *initialization*. A set of chromosomes consisting of strings of reflectance parameters (eight in the case of translucent and four in the case of opaque materials, as explained before) are set. In this first generation the parameters take random values within the search space, which, in the absence of previous knowledge of the material, is [0,1]. The number of chromosomes created is a configuration parameter of the algorithm.

An image is then rendered for each of the chromosomes created in each generation to calculate the *fitness* value of each chromosome and thus perform the *selection* step. This *fitness* value is calculated with a per-pixel least squares function measuring the difference between the individual channels in the original and the rendered images. The set of parameters delivering the most approximate solution are used to create the next generation. The number of chromosomes being replaced conforms another configuration parameter of the algorithm.

Once the best chromosomes have been selected, *reproduction*, involving crossover and mutation, takes place. In our implementation crossover is performed at only one point of the chromosome (as in the case shown in Figure 2), which has proven enough for our objectives, but more complex crossover procedures are also possible. During mutation gene values vary between ±0-30% of their original value. Our research shows that greater variations introduce a too random behaviour and control over the evolution of the algorithm is easily lost, whereas very small variations need many generations for the algorithm to reach a valid solution.

The processes of selection and reproduction continue iteratively until the termination condition is met. Given that our goal is to study the effectiveness of the algorithm and the influence of its configuration parameters on the final result, we simply define our termination condition as a fixed number of generations. This suffices in our context, although changing the termination condition to an error threshold is straightforward.

4.3 Parameter space

Genetic algorithms have a series of input parameters (initial number of individuals, crossover and mutation probabilities, etc), whose correct configuration is vital in reaching a consistent solution within a reasonable execution time. In order to select the most adequate values for these parameters, we have performed a series of tests, taking into account both the accuracy of the final solution and the computation time required. The results of these tests for the most relevant configuration parameters are discussed here, and can be seen on Figure 3 for the case of the *Phong* model. The accuracy was measured as percentage of error between the real ground truth values and the values obtained by the algorithm.

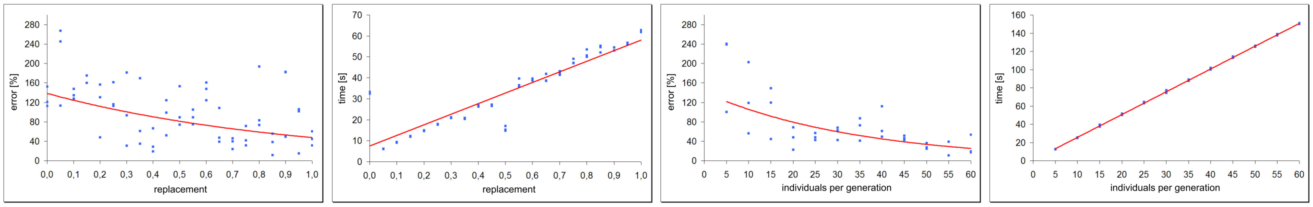

Figure 3: *Left and center left:* Percentage of error and execution time as a function of the probability of replacement. *Center right and right:* Percentage of error and execution time as a function of the number of individuals in each generation. Data obtained for the *Phong* model.

Probability of replacement. The probability of replacement accounts for the percentage of individuals which are eliminated in each selection process. Following the evolution simile, the higher this probability is, the faster the population evolves. However, running times also increase significantly, as all the chromosomes and their corresponding image need to be created for each generation. The greatly significant influence of this parameter, both in execution time and in the accuracy of the result, can be seen in Figure 3 (*left*).

Probability of crossover. As explained in Section 3, crossover represents sexual reproduction and takes place after the selection and replacement process. The probability of crossover represents the percentage of individuals which are the result of combining the genes of two survivor chromosomes. An increase in *sexual reproduction* (hence favouring genetic diversity) causes the percentage of error to decrease slightly. As the resulting images of more combinations of genes need to be calculated, execution time increases slowly.

Probability of mutation. Representing the percentage of genes which mutate from one generation to the next, this probability is critical when working with a small number of individuals per generation. Variations in the initial genes are crucial to progressively reach the optimal solution, and the higher this probability, the lower the percentage of error with minimum time penalty.

Number of generations. The number of generations is, together with the number of individuals per generation discussed below, the parameter with the greatest influence. It indicates the number of generations which are created before the algorithm terminates and delivers a solution (alternatively, an error threshold can be trivially set as termination parameter). Figure 1 shows how the solution progressively evolves along generations. With an infinite number of generations, the solution would perfectly match the original. In practice, a compromise has to be found between execution time and accuracy of the solution, determined by the number of generations.

Number of individuals per generation. The effect of the number of individuals of each generation in the performance of the algorithm is straightforward: the more individuals, the least the percentage of error, as more possibilities are evaluated. However, there is a substantial increase in the execution time, as shown in Figure 3 (*right*).

Reduced search space. Being able to reduce the size of the parameter search space considerably helps the algorithm to converge faster. We carried out a test varying the size of the search space to analyze its effect on the final solution. For this test the probabilities of replacement, crossover and mutation were all fixed to 0.8, the number of generations was 50 and the number of individuals in each generation 40. The search space was reduced to values between $\pm 100\%$ and $\pm 150\%$ of the original value. Results are shown in Figure 4 for a translucent object (thus using the diffuse approximation as the illumination model). It can be seen how reducing the search space yielded a better solution for the same number of iterations.

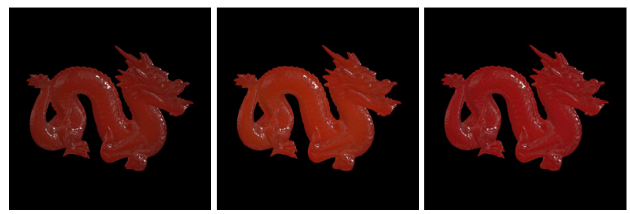

Figure 4: *From left to right:* Original image, image rendered using a global search space (parameters between 0 and 1) and image rendered using a reduced search space (parameters between $\pm 100\%$ and $\pm 150\%$ of the original value). The second one approximates the ground truth image more closely for the same number of iterations. Rendering time = 15 minutes.

5 Results and Conclusion

We have presented a method based on genetic algorithms suitable for capturing the appearance of opaque and translucent materials depicted in single images. The algorithm converges to an approximate solution in reasonable times with little user interaction.

Additionally, we have studied the influence of the different configuration parameters of the genetic algorithm in both the accuracy of the result and the execution time of the algorithm, thus providing a guidance for future implementations. Genetic algorithms have already proved useful in solving non-structured or inverse problems in many other fields; however, configuring their input parameters remains a fundamental task which varies across applications, and thus has to be individually studied for each specific optimization problem. The correct election of these parameters is a key aspect for achieving a valid solution in a reasonable time, as expectedly more accuracy in determining the solution implies higher execution times. Special attention must be paid to the number of individuals per generation and to the percentage of individuals being replaced in each iteration, given their stronger influence in both computational cost and accuracy of the result. When possible, reducing the search space of the parameters significantly increases the efficiency of the algorithm.

Additional results are shown in Figure 5. The probabilities of replacement, crossover and mutation were all fixed to 0.8, the number of generations was 50 and the number of individuals in each generation was set to 40. All images in this paper have been rendered on a AMD Opteron Quad-core machine @$3GHz$ and $4GB$ of RAM, and took between 15 and 20 minutes in the case of translucent materials and around one minute in the case of opaque objects. For the diffusion approximation, we have used the fast hierarchical rendering technique of Jensen and Buhler [2002].

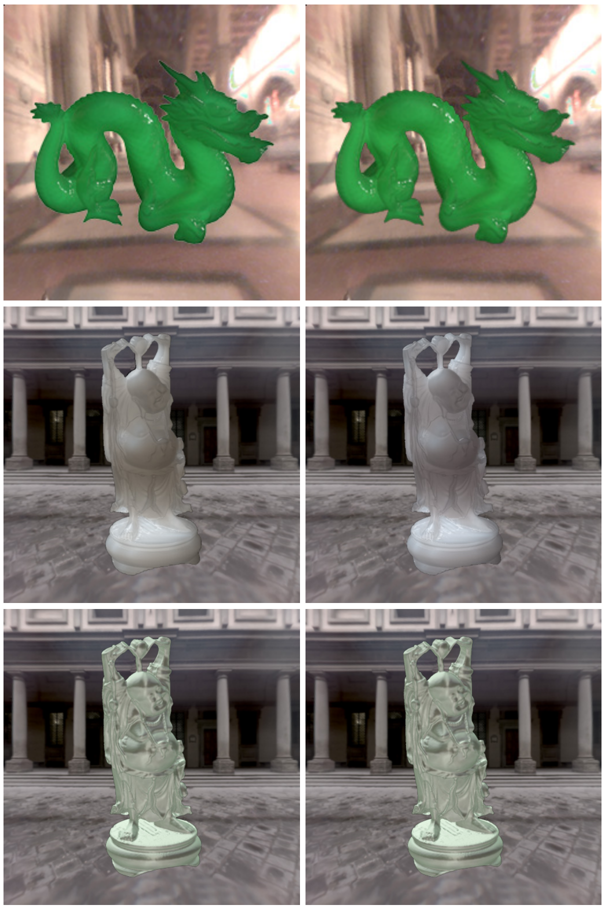

Figure 5: *Left column:* Original image. *Right column:* Image rendered with our algorithm. In the first two rows the material is translucent, and thus modeled with the *dipole* model. The last row shows an opaque material, modeled with the *Phong* model. Rendering times are around 20 minutes for the translucent ones and 1 minute for the opaque one.

6 Future Work

One of the main lines for future research is exploring the possibilities that mutation techniques can offer with the objective of accelerating convergence to the solution and of overcoming local minima. Besides, the operation with the highest cost is rendering the scene with each set of parameters for evaluation by comparison with the original image, so creating the chromosomes of possible solutions intelligently instead of relying on brute force is vital, and more sophisticated mutation functions could also help in this direction.

To reduce the parameter space, we have assumed that information of light sources, geometry and camera position was known, and only the reflectance characteristics of an object in the image were unknown. It would be interesting to stress our approach further and see how genetic algorithms perform as the problem becomes even more ill-posed.

Further strategies which can improve the implementation include parallelization and improvement of the efficiency of GPUs. As mentioned, the bottleneck of the implementation lies in generating an image for evaluation for each string of parameters; given that these strings are completely independent between them, several evaluations could be performed in parallel to reduce the execution time.

7 Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This research has been funded by the University of Zaragoza (project UZ2007-TEC06) and the Spanish Ministry of Science and Technology (TIN2007-63025).

References

BOIVIN, S., AND GAGALOWICZ, A. 2001. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *SIGGRAPH*, ACM, Los Angeles, California, 107–116.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., , AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. In *SIGGRAPH*, ACM.

DONNER, C., WEYRICH, T., D'EON, E., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2008. A layered, heterogeneous reflectance model for acquiring and rendering human skin. ACM, New York, NY, USA, vol. 27, 1–12.

GHOSH, A., HAWKINS, T., PEERS, P., FREDERIKSEN, S., AND DEBEVEC, P. 2008. Practical modeling and acquisition of layered facial reflectance. ACM, New York, NY, USA, vol. 27, 1–10.

GOESELE, M., LENSCH, H. P. A., LANG, J., FUCHS, C., AND SEIDEL, H.-P. 2004. Disco: acquisition of translucent objects. In *SIGGRAPH*, ACM, New York, NY, USA, 835–844.

HUBBALL, D., CHEN, M., AND GRANT, P. 2008. Image-based aging using evolutionary computing. *Computer Graphics Forum, 27*, 2, 607–616.

JENSEN, H. W., AND BUHLER, J. 2002. A rapid hierarchical rendering technique for translucent materials. In *SIGGRAPH*, ACM, San Antonio, Texas, 576–581.

JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In *SIGGRAPH*, ACM, Los Angeles, California, 511–518.

KATZ, A., AND THRIFT, P. 1994. Generating image filters for target recognition by genetic learning. *PAMI 16*, 9 (Septiembre), 906–910.

KOLJONEN, J., AND ALANDER, J. 2006. Genetic algorithm for optimizing fuzzy image pattern. *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)* (Octubre), 46–53.

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics (TOG) 22*, 2, 234–257.

LI, H., FOO, S., TORRANCE, K., AND WESTIN, S. 2006. Automated three-axis gonioreflectometer for computer graphics applications. *Optical Engineering 45*, 4, 1–11.

MATUSIK, W., PFISTER, H., ZIEGLER, R., NGAN, A., AND MCMILLAN, L. 2002. Acquisition and rendering of transparent and refractive objects. In *Thirteenth Eurographics Workshop on Rendering*, EG, 267–277.

PEERS, P., VOM BERGE, K., MATUSIK, W., RAMAMOORTHI, R., LAWRENCE, J., RUSINKIEWICZ, S., AND DUTRÉ, P. 2006. A compact factored representation of heterogeneous subsurface scattering. In *SIGGRAPH*, ACM, New York, NY, USA, 746–753.

PHONG, B. 1973. *Illumination of computer-generated images*. PhD thesis, University of Utah.

QIN, X., AND YANG, Y. 2002. Estimating parameters for procedural texturing by genetic algorithms. *Graphical Models 64*, 1, 19–39.

ROTH, G., AND LEVINE, M. 1994. Geometric primitive extraction using a genetic algorithm. *PAMI 16*, 9 (Septiembre), 901–905.

SALEK, P., TARASIUK, J., AND WIERZBANOWSKI, K. 1999. Application of genetic algorithms to texture analysis. *Crystal Research and Technology 34*, 1073–1079.

SHEN, L., AND TAKEMURA, H. 2006. Spatial reflectance recovery under complex illumination from sparse images. In *Computer Vision and Pattern Recognition*, IEEE, 1833–1838.

SIMS, K. 1991. Artificial evolution for computer graphics. In *SIGGRAPH '91*, ACM, New York, NY, USA, 319–328.

TARIQ, S., GARDNER, A., LLAMAS, I., JONES, A., DEBEVEC, P., AND TURK, G. 2006. Efficient estimation of spatially varying subsurface scattering parameters. In *Workshop on Vision, Modeling, and Visualzation (VMW)*.

WANG, J., ZHAO, S., TONG, X., LIN, S., LIN, Z., DONG, Y., GUO, B., AND SHUM, H.-Y. 2008. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics (TOG) 27*, 1, 1–18.

WARD, G. J. 1992. Measuring and modeling anisotropic reflection. In *SIGGRAPH*, ACM, Chicago, Illanois, 265–272.

WEYRICH, T., LAWRENCE, J., LENSCH, H., RUSINKIEWICZ, S., AND ZICKLER, T. 2008. Principles of appearance acquisition and representation. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, ACM, New York, NY, USA, 1–119.

WINTER, G., PRIAUX, J., GALN, M., AND CUESTA, P. 1995. *Genetic algorithms in engineering and computer science*. John Wiley and sons.

WU, T.-P., AND TANG, C.-K. 2006. Separating subsurface scattering from photometric image. In *International Conference on Pattern Recognition (ICPR)*, IEEE Computer Society, 207–210.

XU, K., GAO, Y., LI, Y., JU, T., AND HU, S.-M. 2007. Real-time homogenous translucent material editing. *Computer Graphics Forum 26*, 3, 545–552.

YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. 1999. Inverse global illumination: Recovering reflectance models of real ccenes from photographs. In *SIGGRAPH*, ACM, Los Angeles, California, 215–224.