EG UK Theory and Practice of Computer Graphics (2006)
M. McDerby, L. Lever (Editors)

Perception-based Lighting Design

H. N. HA and P. Olivier

Informatics Research Insitute, University of Newcastle Upon Tyne, UK.

Abstract

Perception-based lighting design approaches model image quality using a cognitively grounded objective function which is in turn optimised through manipulation of the lighting parameters of a scene. We present, and demonstrate, a detailed implementation of perception-based lighting design, including the application and evaluation of stochastic optimisation using genetic algorithms.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

The perceptual quality of a rendered 3D scene is highly dependent on the set-up of the lighting parameters. Finding optimal lighting parameters – positions, directions, colours, and intensities of light sources – is referred to as the *lighting design* process. Automated lighting design is a pressing requirement in application domains such as scientific visualisation, in which the components of a scene and relative configuration with respect to the viewers, cannot be anticipated. In such domains the automatic generation of images with recognisable visual characteristics is highly desirable. Likewise, automatic lighting design can play an indispensable role in computer-generated image synthesis and film production.

Much existing research, motivated by studies of visual cognition, has been conducted with the aim of increasing the perceptual realism of computer-generated images [McN00] [FP04]. Perceptual realism in a computer-generated image is a matter of perceptual equivalence of the image to the corresponding real world scene. A number of investigations have been conducted into the creation of efficient perceptually motivated image quality metrics [McN00] [PN03] [FP04] [FAP] that are critical to the quantification of the perceptual quality of an image. Many perception-based problems must be taken into consideration to achieve perceptual realism in computer generated imagery and the problem remains a significant challenge.

In this paper, we present and demonstrate, a detailed implementation of perception-based lighting design, including the application and evaluation of stochastic optimisation using genetic algorithms. After a review of existing work in lighting design we extend Shacked and Lischinski's [SL01] method (see section 3) by applying a powerful stochastic optimisation framework, and we present a detailed description of the implementation of this approach.

2. Previous work

Direct design methods are the traditional approach to lighting design for image synthesis. Users interactively modify lighting parameters and reflect on the impact of their design choices on the resulting image. This process is repeated until desired visual properties of the rendered scene are achieved. Few attempts have been made either to automate or assist the process of lighting design despite the fact that traditional direct design methods are inherently tedious and time-consuming.

Schoeneman et al [SDS*93] proposed an approach in which lighting design can be considered as an inverse problem. Their approach optimises an image through the manipulation of light intensities and colours alone. Users set up desired properties for the final image through an interactive interface, and the system searches for an optimal solution whose properties best match those specified. Similarly, Kawai et al. [KPC] reported a technique in which the illumination for an environment rendered using radiosity-based techniques is achieved in which lighting emission, direction and surface reflectance, are optimised. Once again, in this

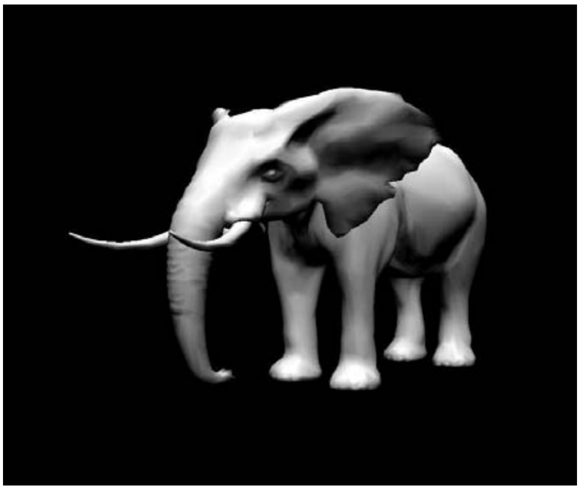

Figure 1: *First sample solution for the Shacked and Lischinski base algorithm [SL01].*

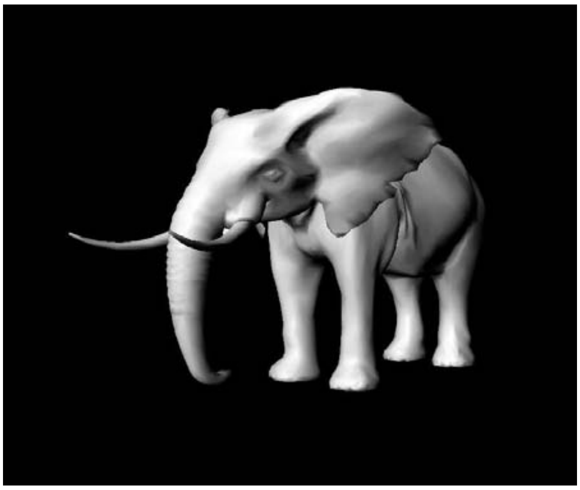

Figure 3: *Third sample solution for the Shacked and Lischinski base algorithm [SL01].*

approach users are required to specify the illumination expected in the final image.

Poulin and Fournier [PF92] proposed an inverse method for setting light positions with respect to the specification of shadows and highlights in a 3D scene. An interactive sketch-based interface enabled users to draw desired shadow areas with a mouse pointer [PRJ97]. An objective function was defined such that the shadow region for a computed point light (and also some extended light geometries) bounds the sketched regions as tightly as possible. Jolivet et al [VJ02] presented an approach to optimising light positions in direct lighting using Monte-Carlo method. A declarative paradigm was utilised in order to help users specify the lighting goal in an intuitive manner.

In Design Galleries [MAP*97] Marks et al map an input vector containing light position, light type, and light direction to an output vector containing a set of values that summarises the perceptual qualities of the final image. A finite set of light positions are predefined and lights are moved from one predefined position to another during the optimisation process. An image can be derived at each light position by rendering the scene with a light type selected from a set of light types. A simple metric is used to measure the distance between images; and final set of images are presented

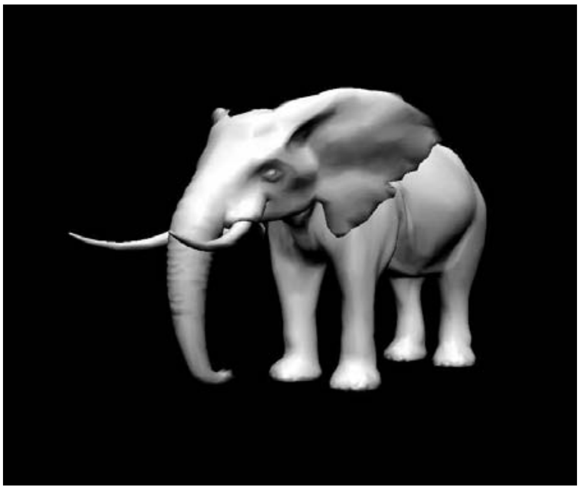

Figure 2: *Second sample solution for the Shacked and Lischinski base algorithm [SL01].*

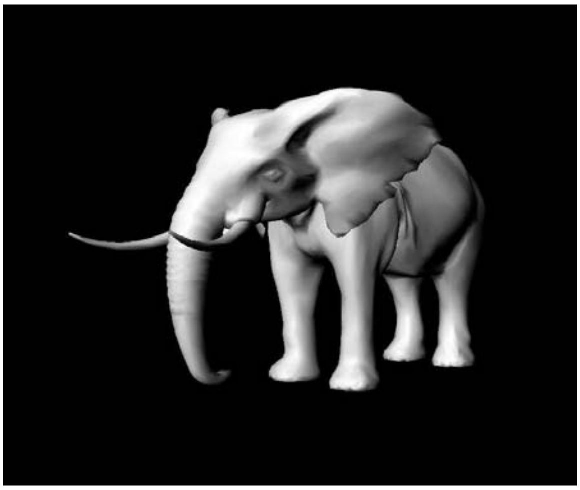

Figure 4: *Fourth sample solution for the Shacked and Lischinski base algorithm [SL01].*

to users in the form of a number of subsets of image thumbnails clustered on the basis of the distance metric.

2.1. Perception-based lighting

There has been significant effort in recent years in the development of approaches to computer graphics based upon explicit models of a viewer's perception of graphical renderings. Perceptually adaptive approaches have ranged across whole scope of graphics algorithm and interaction development, from schemes for polygon simplification and global illumination that take account of limits on visual attention and acuity, to the design of anthropomorphic animations and gaze-contingent displays [PN03] [FP04].

Perception-based lighting design has included implicit approaches that aim to maximise illumination entropy for a fixed viewpoint. Gumhold [Gum] describes a perceptual illumination entropy approach in which he uses limited user studies to model user preferences in relation to brightness and curvature. In [LHV] a more explicit model of perceptual preferences is used in the Light Collages framework in which lights are optimised such that the diffuse illumination is proportional to the local curvature and specular highlights are used only for regions of particularly high curvature.

In our work we take as a starting point an approach that maintains an explicit model of object perception due to Shacked and Lischinski [SL01]. In their perception-based lighting design scheme the position and intensity (of specular and diffuse components of a local illumination model) of light sources are optimised using an evaluation function that characterises separate aspects of low-level processing in the segmentation and recognition of objects. At the heart of this approach is an objective function that is the linear combination of five distinct measures of image quality: edge distinctness (F_{edge}); mean brightness (F_{mean}); mean shading gradient (F_{grad}); intensity range (F_{var}); and evenness of the distribution of intensity in the image (F_{hist}).

$$F(\theta_k,\phi_k,I_{dk},I_{sk},R_k) = w_e F_{edge} + w_m F_{mean} + w_g F_{grad} + w_v F_{var} + w_h F_{hist} \quad (1)$$

Where θ_k is the elevation angle of k^{th} light; ϕ_k is the azimuth angle of the k^{th} light; I_{dk} is the diffuse intensity of k^{th} light; I_{sk} is the specular intensity of k^{th} light; R_k is the distance of k^{th} light (fixed for directional lights); and w_e, w_m, w_g, w_v and w_h are weights for different components of the objective function.

We omit a sixth component of the image quality function used by Shacked and Lischinski which biases the optimisation of a key light to a particular elevation and orientation above, and in front of, the object (relative to the viewpoint). The implementation of this component involves the direction of the light being used as an image quality function is unjustifiably ad hoc and we omit it from our treatment.

Although it is standard practice in photography, and might be explained in terms of evolutionary psychology [Gro94] [Mil05] (that our perceptual system evolved for scenes lit by the sun or moon), constraining the light position to a quarter sphere, in front of, and above, the centre of the scene is sufficient. Shacked and Lischinski's framework is formulated such that the lower values of $F(\theta_k,\phi_k,I_{dk},I_{sk},R_k)$ correspond to lighting configurations with better visual characteristics and a greedy gradient descent minimisation algorithm is utilised in the discovery of appropriate lighting configurations.

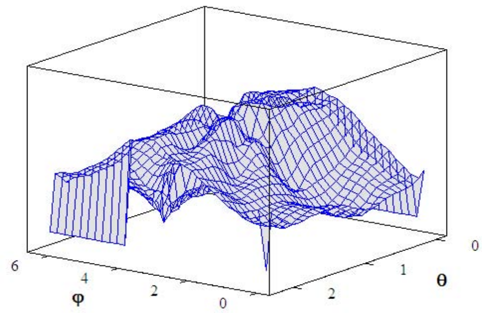

Figure 5: *Visualisation of the objective function $F(\theta,\phi)$ for a single object (elepahant) in figure 7.*

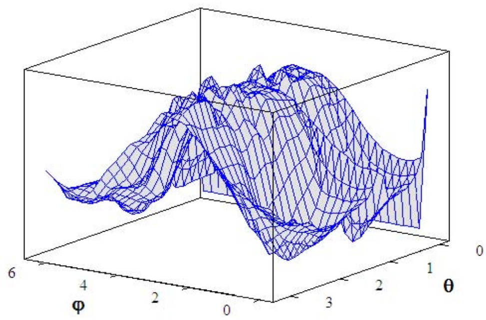

Figure 6: *Visualisation of the objective function $F(\theta,\phi)$ for the two object scene in figure 7.*

3. Optimisation of lighting configurations

Shacked and Lischinski make no attempt to characterise the nature of their objective function and the suitability of the greedy search employed. From equation (1) it is clear that the multi-objective optimisation problem incorporates significant non-linearity and as the geometric complexity of the scene increases the likelihood of local minima will increase. With a greedy search of a space such as this the solution is

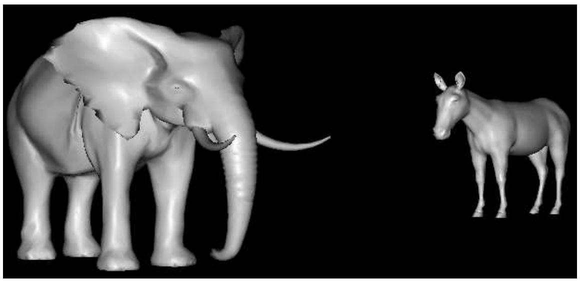

Figure 7: *Two object scene use for the visualisation of the objective function $F(\theta,\phi)$ in figures 5 and 6.*

highly dependent on the starting condition and figures 1-4 illustrates the resulting images for four searches conducted from different starting configurations.

Figures 5, 6 depicts a plot of $F(\theta,\phi)$ for one light source, at constant intensity, for a scene containing one and two objects (shown in figure 7). The presence of substantial numbers of local minima supports our observation that greedy search techniques are insufficient. The nature of the optimisation problem suggests a requirement for a more general optimisation strategy, and a stochastic approach, such as a genetic algorithm (GA), is fitting candidate. The encoding of the lighting problem in a GA is straightforward. The free variables θ_k, ϕ_k, I_d, I_s, R are encoded directly in the chromosome as real numbered alleles. We have evaluated the optimisation problem with varying population sizes and configurations for the GA. From our experiments we established that a population size of 30, 10% elitism and crossover and mutation rates of 80% and 20% respectively were sufficient.

Note that the process of optimising a single (key) light takes place under the constraint that the light position is limited in the values of elevation and azimuth angles (e.g. the position of a light is limited to a quarter sphere in front and above the center of the scene). This constraint must be respected at all times during optimisation, but is particularly important at initialisation and during the generation of random mutations (members of the population whose values do not reside in constrained ranges are rejected). Note that as the constrained region is convex, crossover cannot yield offspring that violate the constraint.

Although the selection of population size, mutation and crossover are problem specific, our particular configuration was established through experimentation on a variety of scenes. Evidence of satisfactory convergence of the GA is illustrated in figure 8 which shows a plot of $F(\theta_k,\phi_k,I_{dk},I_{sk},R_k)$ for the best members of the population (averaged over ten runs) at each generation where the population size was 30. The GA exhibited consistently better results than the greedy search both in terms of the optimisation result and the visual quality of the solution. Figure 12 shows a direct comparison of the greedy and GA optimisation results, although care must be taken in generalising from specific examples such as these.

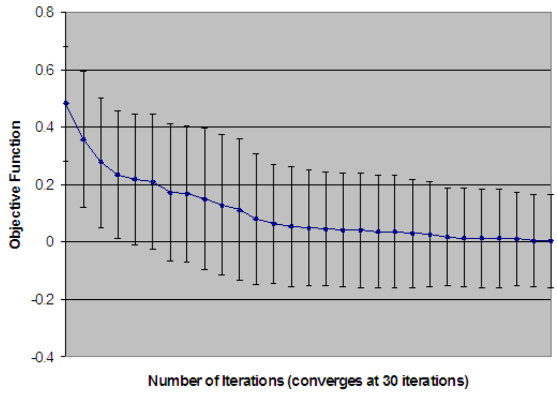

Figure 8: *Convergence of the genetic algorithm (averages over 10 runs).*

4. Implementation

Visible surface detection is achieved useing colour coding and a pixel type map. Objects are rendered without lighting, and each polygon of an object is rendered with a unique colour. To obtain a unique colour for each polygon, a colour is generated from a incremental integer variable `Ic`. Pseudo code for rendering objects with this colour-code method and the creation of a colour-code array is given in figure 9. Firstly, we describe the data structure, to maintain colour-code information associated with each polygon of an object in the scene as follows:

```
Struct ColorCode
{
 Integer ObjectID;
 Integer PolygonID;
 Byte Red;
 Byte Green;
 Byte Blue;
}
```

`ObjectID`: identification of an object.
`PolygonID`: identification of a polygon of an object.
`Red`: red component of a color of a polygon.
`Green`: green component of a color of a polygon.
`Blued`: blue component of a color of a polygon

An array of colour code structures, `CCArray`, is used to store information about colour-codes for all objects. Figure 10 shows an example of an object rendered with this colour-code technique. In our system, a colour-code image is saved in a matrix for later processing and can be simply obtained by directly accessing the OpenGL frame buffer. Each point in the matrix contains three color components of a corresponding pixel in the rendered image. The data structure for `CCArray` is as follows:

```
Struct ColorCodeImage
{
 Byte Red;
 Byte Green;
 Byte Blue;
}
```

```
Procedure RenderWithColorCode
Integer k;
Integer Ic, r, g, b
Ic = 0;
Nj: Number of polygons of the object j;
No: Number of objects;
CCArray Array[0.. No-1] of ColorCode;
For j=0 to No do
 For k=0 to Nj do
   red = Ic&0xff0000;
   red = r »16;
   green = Ic &0x00ff00;
   green  = g » 8;
   blue = Ic &0x0000ff;
   CCArray[Ic].ObjectID = j;
   CCArray[Ic].PolygonID = k;
   CCArray[Ic].Red = r;
   CCArray[Ic].Green = g;
   CCArray[Ic].Blue = b;
   Render-Polygon k of object j (r,g,b);
   Ic = Ic +1;
 End;
End.
```

Figure 9: *Creation of a colour-code array.*

4.1. Visible surface detection

The visibility of a polygon can be easily calcualted on the basis of information saved in the colour-code array and a colour-code image. In our system, the visibility of polygons are used to calculate a normal map of visible polygons that is utilised in initialising light positions in the optimisation process. Pseudo code for detecting visibility of a polygon of an object is given in figure 11.

4.2. Pixel type map

A pixel type map indicates the type of a pixel in 2D rendered image. An edge map of objects in a 3D scene is extracted by applying an edge detection operator to the depth buffer. Sobel and Laplace edge detection operators are combined to enhance the accuracy of edge detection. Figure 10 illustrates a result of edge detection using the depth buffer. A pixel type map is derived by combining an edge map and a colour-code image matrix. With the colour-code we know which points belong to which objects. Points belonging to objects are either edges or surface points. With an edge map we know which points are edge points. Finally, a pixel type map contains three types of pixels:

EDGE Pixel belonging to an edge of an object.
SURFACE Pixels belonging to a surface of an object.
BACKGROUND Pixels that are not associated with any object in the scene, note that background properties are not considered during optimisation process.

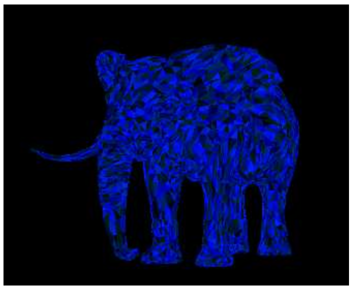

Figure 10: *Example of a colour-code rendering.*

```
CCArray  Array[0.. No-1] of ColorCode
Calculated in previous step;
CCodeImg Array[0..iHeight][0..iWidth]
  of ColorCodeImage
iHeight: Height of color-code image matrix.
iWidth: Width of color-code image matrix;
Boolean IsVisibleFace(
Integer objID;
Integer faceID)
{
p1 Array[0..2] of Byte ;
 ColorCode sCC;
 Integer np;
 Integer i,j;
np: sum of polygons of all ob-
jects has id less than ObjID  ;
np = np + faceID;
  cCC = CCArray[np];
for i=0 to iHeight
for j=0 to iWidth
Begin
If((sCC.Red equal CCodedImg(i,j).Red)AND
   (sCC.Green equal CCodedImg(i,j).Green)AND
   (sCC.Blue equal CCodedImg(i,j).Blue))
 IsVisibleFace = true;
End;
  IsVisibleFace = false;
End.
```

Figure 11: *Color-code based visible surface detection.*

4.3. Normal map and light initialisation

Normals are represented using a data structure containing information about normals of visible polygons. Normal calculation of visible polygons is easily achieved, and depending on the number of lights used in the optimisation process, a set of normals of visible polygons is subjected to a cluster analysis (a standard KMean clustering method). Centroids of clusters are used as initial positions of lights. A normal is represented as a normalised 3D vector rather than using spherical representation. The KMean method takes an array of normalised 3D vectors and a specified number of clusters as inputs and returns cluster centroids.

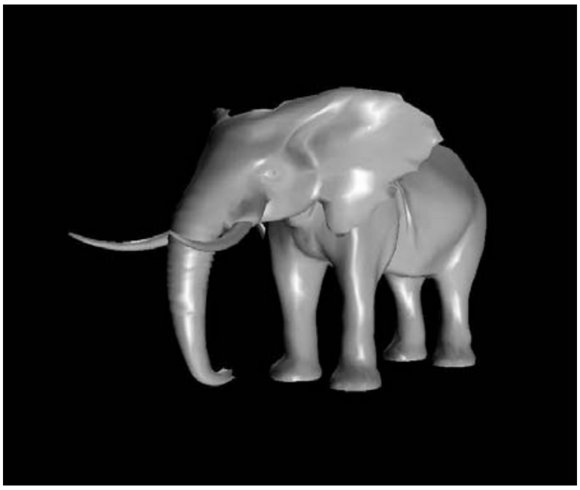

Figure 12: *Genetic algorithm optimisation of the Shacked and Lischinski objective [SL01].*

5. Discussion

We have presented a detailed implementation to the perception-based lighting design approach proposed by Shacked and Lischinski [SL01] and extended approach through the use of a stochastic optimisation framework in the form of a genetic algorithm. However, we note a number of shortcomings, both in the framework and the philosophy of lighting design through objective optimisation. With respect to the framework, many aspects of visual perception are not accounted for and the approach lacks ecological validity. For example, none of the objects are textured and human judgments as to texture differences (and even colour difference in the presence of surface textures only) are poorly understood.

More fundamentally, the approach of lighting design through the optimisation of an objective that characterises an *ideally lit scene* can be criticised as being simplistic with regard to the role that lighting design plays in visual media. Indeed, different lighting is used to convey different moods, emotion, and factors other than the geometrical properties of the elements of a scene.

Consequently, the likely outcome of a program of research into the design of lighting will not be a accurate (and empirically verified) objective function for ideally illuminated scenes, but a framework for the specification of lighting using example 3D scenes and photographs, and tools to allow artists to interactively modify scene lighting through inverse design. Such approaches presume the ability to model target scenes in the form of a perceptually meaningful objective function, and to optimise source scenes using these objectives. We see the declarative approach to lighting design presented here as necessary component of this wider enterprise to build new tools for the specification and design of lighting.

References

[FAP] FARUGIA J.-P., ALBIN S., PEROCHE B.: A perceptual adaptive image metric for computer graphics. In *Proc. EUROGRAPHICS'04*, vol. 2.

[FP04] FARUGIA J.-P., PEROCHE B.: A progressive rendering algorithm using an adaptive perceptually based image metric. *Computer Graphics Forum 23*, 3 (2004). (Proc. Eurographics'04).

[Gro94] GROSS M.: *Visual Computing*. Springer-Verlag, 1994.

[Gum] GUMHOLD S.: Maximum entropy light source placement. In *Proc. IEEE Visualization 2002*, pp. 275–282.

[KPC] KAWAI J., PAINTER J., COHEN M.: Radioptimization – goal based rendering. In *Proc. SIGGRAPH'93*, pp. 147–154.

[LHV] LEE C. H., HAO X., VARSHNEY A.: Light collages: Lighting design for effective visualization. In *Proc. IEEE Visualization 2004*, pp. 281–288.

[MAP*97] MARKS J., ANDALMAN B., P. A. B., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH '97* (1997), pp. 389–400.

[McN00] MCNAMARA A.: *Comparing real and synthetic scenes using human judgments of lightness*. Doctoral dissertation, University of Bristol, 2000.

[Mil05] MILLERSON G.: *Lighting for Television and Film*. Focal Press, 2005.

[PF92] POULIN P., FOURNIER A.: Lights from highlights and shadows. In *Symposium on Interactive 3D Graphics (SI3D'92)* (1992), pp. 31–38.

[PN03] P. S. A. R., N. S. P.: Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM TOG 22*, 3 (2003), 537–542.

[PRJ97] POULIN P., RATIB K., JACQUES M.: Sketching shadows and highlights to position lights. In *Computer Graphics International* (1997).

[SDS*93] SCHOENEMAN C., DORSEY J., SMITS B., ARVO J., GREENBURG D.: Painting with light. In *Proc. SIGGRAPH '93* (1993), pp. 143–146.

[SL01] SHACKED R., LISCHINSKI D.: Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum 20*, 3 (2001), 215–226.

[VJ02] V. JOLIVET D. PLEMENOS P. P.: Inverse direct lighting with a monte carlo method and declarative modelling. In *Proc. Computer Graphics and Geomtric Modelling* (2002).