

Maximum Entropy Light Source Placement

Stefan Gumhold

WSI/GRIS, University of Tübingen

ABSTRACT

Finding the “best” viewing parameters for a scene is quite difficult but a very important problem. Fully automatic procedures seem to be impossible as the notion of “best” strongly depends on the human judgment as well as on the application. In this paper a solution to the sub-problem of placing light sources for given camera parameters is proposed. A light position is defined to be optimal, when the resulting illumination reveals more about the scene as the illuminations from all other light positions, i.e. the light position maximizes the information that is added to the image through the illumination. With the help of an experiment with several subjects we could adapt the information measure to the actually perceived information content. We present fast global optimization procedures and solutions for two and more light sources.

CR Categories: I.3.8 [Computer Graphics]: Miscellaneous—Lighting Design; G.1.6 [Numerical Analysis]: Optimization—Global Optimization;

Keywords: Lighting Design, Visualization, Illumination, Maximum Entropy, Optimization, User Study

1 INTRODUCTION

Lighting design approaches are seldom used in scientific visualization. Especially, in the exploration stage the scientist does not know much about the data set that he is looking at. With lighting design tools based on the inverse paradigm the user has to specify objectives and constraints that the illumination should satisfy. These are not applicable, as they require that the user can express his wishes. Instead most visualization tools provide a standard illumination or let the user interactively explore different lighting parameters. The latter forces the user to explore the data set as well as the lighting parameters what unnecessarily elongates the exploration process.

In this paper we present a tool to automatically design the illumination especially tweaked for the visualization process. The basic idea is to maximize the information introduced by the illumination. Information maximization is a common goal in scientific visualization and the most effective visualization techniques follow this idea. For example in vector field visualization the line integral convolution method proofed to be remarkably efficient. The technique initializes the image with a random noise texture, which has maximum information content. In the second phase the random texture is averaged along stream-lines. This can be interpreted as the annihilation of the information that conceals the vector field. The information portraying the vector field is maximized. Another example is the design of color or material palettes by the use of histogram methods. Here a scientific quantity of interest is monotonically mapped onto a visual attribute such that over the complete data set the probability of each possible value for the visual attribute

is equally likely. The histogram approaches maximizes the information content of the visual attribute.

Section 3 describes how to place light sources in a way that maximizes the illumination information. A conservative global maximization procedure is presented as well as a fast modification that finds the global maximum in most cases. Figure 3 shows examples of maximum information illuminations. They are not as helpful as expected. The first drawback is that a lot of information is found on flat surfaces, which typically do not compose the most important parts of the scene. The second drawback is that highlights are exaggerated, what perturbs more than it helps. In order to resolve the drawback we designed an experiment as described in section 4, in which several potential or already trained users of visualization tools had to adjust the light source positions until they could perceive the scene from a fixed view point as well as possible. Section 5 describes how we used the experimental results to modify the definition of illumination information to the *perceptual illumination information*. Section 6 details the placement of several light sources before we conclude in section 7.

2 RELATED WORK

Maximum entropy solutions are widely spread in all types of sciences [2]. For view-point selection maximum entropy solutions have been investigated in computer vision [1] and more recently in computer graphics [14].

For the more general problem of choosing viewing parameters and light source parameters different approaches exist in the literature. In the interactive evolutionary approach [7, 13, 5] the user is himself the optimality criterion and has to judge suggestions proposed by an evolutionary process until optimal parameters are found. The second type of approaches [15, 10, 6, 8, 3] define the viewing parameters by solving inverse problems, i.e. the user somehow specifies how the scene should look like by defining some locations of highlights or shadows and the computer tries to find viewing parameters matching the imagination of the user. Marks et al. [9] use a perceptual based difference measure for images in order to generate a gallery of maximally perceptually different images by variation of the viewing parameters. The user selects the viewing parameters by the selection of one of the images from the gallery. The lighting design tool of Shacked and Lischinski [12] optimizes the lighting parameters according to a perceptual based quality metric. They define six quality terms, for which the user has to specify weights. No user experiments have been performed to justify their quality metric.

To accelerate the illumination calculations we used similar techniques as described in [11]. As a result our implementation does not support shadow computations. The more sophisticated relighting engine as proposed by Gershbein and Hanrahan [4] could be used to support shadows.

3 MAXIMIZING ILLUMINATION ENTROPY

This section covers the basic approach for the maximization of illumination information. First we define the illumination information

for given lighting parameters and describe how to compute it. The most costly part of the computation is the lighting of the scene. In the second subsection we detail how to accelerate multiple illumination computations for a fixed view-point. The last subsections work out two optimization strategies that look for light positions of directional light sources that maximize the illumination information.

3.1 Illumination Information and Entropy

From Shannon's source coding theorem we know that the information content of a series of n random variables is equal to n times the entropy of the series. Suppose the random variables can assume the finite set of values $\{v_1, \dots, v_m\}$ and that in the given series the probability of a variable to have the value v_i is p_i . Then the entropy of the series is defined as

$$H = \sum_{i=1}^m p_i \log_2 \frac{1}{p_i}. \quad (1)$$

The series of n symbols can be encoded with $n \cdot H$ bits for example with an adaptive arithmetic coder [16]. Let us now define the illumination entropy in accordance with our needs. In scientific visualization one of the most often exploited visual attributes is color. For illuminated datasets only the hue and saturation components of the color can be effectively used as they are influenced less by the illumination as the brightness. In order to not misrepresent the color attribute in the final image all light sources should be white and, therefore, there is no degree of freedom for the lighting design in the choice of light source color.

The illumination entropy is based only on brightness information, which is sufficient as the information content of the hue and saturation components are not influenced by the lighting design. We used the Y tristimulus value of the CIE 1931 standardized color model and converted it from RGB via the standard default color space sRGB ($Y = 0.21262 \cdot R + 0.71514 \cdot G + 0.07215 \cdot B$). For a given illumination of the scene the Y values of all covered pixels are interpreted as random variable series. The typically supported set $\{0, \dots, 255\}$ of different brightness values is too large for our purposes. More appropriate is the number of distinguishable brightness values, which is between 20 and 50. The actual choice is not important and influences the illumination design only negligibly. We chose to use $m = 30$ different bins b_i of Y values, i.e. if normalized to $[0, 1]$ a value Y falls into bin b_i with index $i = \lceil m \cdot (Y + \frac{1}{2}) \rceil$. For a given illumination of a scene view with n covered pixels the probabilities p_i are computed from the counts c_i of Y values falling into bin b_i by $p_i = \frac{c_i}{n}$. The illumination entropy is finally computed via equation 1.

One inherent difficulty with the illumination entropy is that it is not even continuous no matter what illumination model we use. This can be easily seen by defining the characteristic functions $\mathbf{1}_i : [0, 1] \mapsto \{0, 1\}$ for each bin, which are one inside the bin and zero outside. With the characteristic bin functions, the counts compute to $c_i = \sum_j \mathbf{1}_i(Y_j)$, where j runs over all n covered pixels. Thus even if the brightness values Y_j depend continuously on the illumination parameters, the counts, the probabilities and, therefore, the illumination entropy function are discontinuous because of the characteristic functions. Figure 1 b) shows the illumination entropy of the scene in a) for different positions of a white directional light source drawn over a sphere with the entropy values as radial offsets and additionally color coded. In c) only the largest thirty percent of the entropy values were used for radial displacement, which was raised to the power of thirty in order to emphasize the discontinuous characteristic of the entropy function. We can see that on a large scale corresponding to some degrees the entropy is smooth and only on a very fine scale corresponding to fractional degrees it

is rough. The amplitude of the roughness itself is only in the order of fractional per mills.

3.2 Fast Lighting

In order to optimize the light position a lot of lighting computations have to be performed for the same view-point. The standard approach is to use the graphics hardware and render the scene again and again with different lighting parameters. This has the advantage that all illumination features of nowadays graphics accelerators are available. But the big disadvantage is that the lighting calculations are of scene complexity. For the visualization of iso-surfaces of volumetric data scenes with a complexity of up to a million of triangles arise. Even new graphics accelerators allow only four frames per second if the triangles are not organized into strips and no more than six fps if they are. Reading out the frame buffer costs additional time.

As the scene and the view point do not change when the lighting parameters are varied, for each rendering of the scene the same pixels are covered and lighted with the same surface properties. The lighting calculation can be easily done in software with a complexity of the number of covered pixels when for each covered pixel the surface properties are known. Our acceleration technique uses the Phong-Blinn illumination model. In the following we derive the necessary parameters and formula to relight the same scene with different light source positions.

We assume that all calculations are performed in a global coordinate system originating in the pinhole of the camera. The color components $C_\alpha, \alpha \in \{R, G, B\}$ of each covered pixel are computed from the surface normal \hat{n} , the light location l and the *viewing vector* v pointing from the surface location in the pixel to the pinhole

$$\begin{aligned} C_\alpha &= A_\alpha \cdot a_\alpha + D_\alpha \cdot d_\alpha \cdot \langle \hat{n}, \hat{u} \rangle_+ + S_\alpha \cdot s_\alpha \cdot \langle \hat{n}, \hat{h} \rangle_+^E \\ u &= l + v \\ h &= \hat{u} + \hat{v}, \end{aligned} \quad (2)$$

where " $\langle \dots, \dots \rangle_+$ " denotes the scalar product of two vectors, if it is greater than zero, and zero otherwise and " $\hat{\cdot}$ " denotes the direction of a vector. The parameters A_α, D_α and S_α are the material properties of the surface specifying the corresponding ambient, diffuse and specular color components, whereas E defines the surface shininess. The light source l has corresponding ambient, diffuse and specular color components a_α, d_α and s_α , which we assume to be defined by the user.

For each covered image pixels j the software lighting algorithm needs to know the surface properties $\hat{n}_j, v_j, A_{\alpha,j}, D_{\alpha,j}, S_{\alpha,j}$ and E_j . As the entropy only depends on the brightness values, the material properties can be reduced to the brightness components $A_{Y,j}, D_{Y,j}, S_{Y,j}$. To collect the surface properties the basic rendering algorithms have to support four rendering modes: *standard mode*, *normal mode*, *brightness mode* and *parameter mode*. The standard mode supports the standard approach of rendering with the graphics accelerator. The normal mode provides the lighting design tool with the surface normals. OpenGL lighting is turned off and the surface normals are rendered as colors. A normal vector \hat{n} is coded via the RGB color $C = (R, G, B) \in [0, 1]^3$

$$C = \frac{1}{2} [\hat{n} + (1, 1, 1)^T].$$

The surface normals are automatically interpolated over the triangles by the graphics accelerator providing approximated Phong shading. When the lighting design tool reconstructs the surface normals from the colors obtained via `glReadPixels`, the normal vectors need to be normalized after the back transformation. A

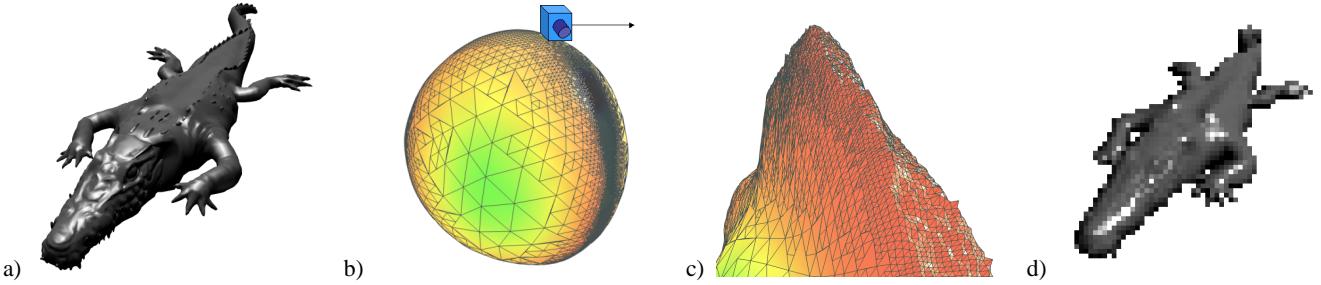


Figure 1: a) example scene b) entropy function of directional light c) view from camera location illustrated in b) with the upper 20 percent of the entropy values taken to the power of 30 d) simulation of level three of hierarchical illumination

background color of $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ was used to mark invalid pixels with normals of length zero. An uniform eight bit quantization of the normal components hardly influences the lighting calculations.

The viewing direction \hat{v} of the viewing vector v is completely defined by the pixel coordinates, when the modelview matrix and the camera parameters are given. The length of v can be computed from the depth value of the pixel, which can also be obtained with `glReadPixels`. To ensure a maximal resolution of the depth value, the clipping planes need to be chosen as tight as possible to the scene limits.

The brightness mode renders the surface parameters $A_{Y,j}$, $D_{Y,j}$ and $S_{Y,j}$ quantized to eight bit again without any OpenGL lighting. Finally, the parameter mode renders E and any further parameters as for example the importance which is introduced in section 5. After rendering the scene three times in normal, brightness and parameter mode, the lighting design tool has collected for each covered pixel the necessary lighting parameters. It can now light the image with different light source placements in time proportional to the number of valid pixels. This allowed for an image resolution of 400 times 300 and 50 thousand covered pixels forty scene lightings per second.

In order to accelerate the lighting computations further we reduced the number of covered pixels by grouping together neighboring pixels. We built an *image hierarchy* in a recursive fashion. Groups of four pixels were combined to one pixel of double side lengths. Let the original image, i.e. the base level of the image hierarchy, be the *zero-th* level. A larger pixel on the *first* image hierarchy level is called *parent pixel* and its children the *sub-pixels*. The surface properties of the parent pixel are averaged from its children. As some of the sub-pixels can be invalid, each parent pixel obtains an additional field: the *count* c_j of valid pixels on the finest level. From the first image hierarchy level we can construct the *second* level by again combining four pixels of level one. The surface properties of the sub-pixels are again averaged and the counts are summed, such that the coarsest level consists of one pixel with its count equal to the number of valid pixels. Building the image hierarchy consumes about $\frac{4}{3}$ of the time spent for one lighting calculation. With the image hierarchy all computations can be performed in different resolutions. Figure 1 d) visualizes the third image hierarchy level of the scene in a). Stepping one level deeper in the image hierarchy reduces the time complexity for lighting by a factor of four but also introduces more and more error.

3.3 Global Maximization

Even if the entropy function is not continuous on a fine scale, for the resolution, at which we want to find light positions, it makes sense to at least assume that the entropy function is Lipschitz continuous, i.e.

$$\forall \hat{l}_1, \hat{l}_2 \in S^2 : \|\hat{l}_1 - \hat{l}_2\| \leq \delta \implies \|H(\hat{l}_1) - H(\hat{l}_2)\| \leq \delta \cdot L \quad (3)$$

with a Lipschitz-constant L , that will be estimated from the entropy function as described below. If we choose $\hat{l}_1, \hat{l}_2 \in S^2$ such that $\|\hat{l}_1 - \hat{l}_2\| = \delta$, equation 3 tells us, that the absolute value of the gradient of the entropy is limited by L . Let us exploit the Lipschitz-continuity to avoid unnecessary lighting computations during the search for the maximum entropy.

We propose the following adaptive search algorithm, that estimates L and determines the maximum entropy value up to a user defined perceptual error ϵ . We implemented the approach for the placement of one directional light. It can be generalized to a light in general position as well.

1. Evaluate the entropy function on a coarse sampling over S^2 and form a triangular mesh.
2. Determine the minimum H_{\min} and maximum H_{\max} of the sampled entropy values.
3. Estimate the Lipschitz-constant $L = k_{\text{safety}} \cdot L_{\text{app}}$ from the current piecewise linear approximation.
4. For each triangle t determine the maximum entropy value H_t that an L -Lipschitz continuous function can reach inside the triangle. If $H_t \geq H_{\max}$, mark triangle t to be split.
5. Let $H_{t,\max}$ be the maximum value of the reachable entropy values H_t . If $\frac{H_{t,\max} - H_{\max}}{H_{\max} - H_{\min}} \leq \epsilon$ terminate.
6. Split all marked triangles into four, evaluate the entropy function on the newly introduced vertices and go back to step 2.

For step 1 the spherical domain was sampled on the vertices of an octahedron. Another choice would be the icosahedron. In step 2 the minimum and maximum values H_{\min} and H_{\max} of the entropy are determined in order to decide, when a triangle has to be subdivided in step 4 and when to terminate in step 5.

To estimate the Lipschitz constant L of the entropy function in step 3 we computed the Lipschitz constant L_{app} of the current piecewise linear approximation and multiplied it with a safety factor k_{safety} . A value of 2 for the safety factor worked well for the views we investigated. L_{app} is equal to the maximum norm of the entropy gradient over the whole domain. As the entropy gradient is constant on each triangle of the approximation, we simply computed the gradient over each triangle and determined the maximum norm resulting in L_{app} .

Step 4 is the crux of the algorithm. It determines for each triangle $t = (p_0, p_1, p_2)$ the maximum entropy value H_t , that an arbitrary L -Lipschitz continuous function h can reach on the triangle t . The only constraint on h is that $h(p_0), h(p_1)$ and $h(p_2)$ must be equal to the measured entropy values H_0, H_1 and H_2 . Let h_{wc} be the worst case function over t , that grows with the maximum gradient L on all rays originating in one of the triangle vertices p_i . In Figure 2 a)

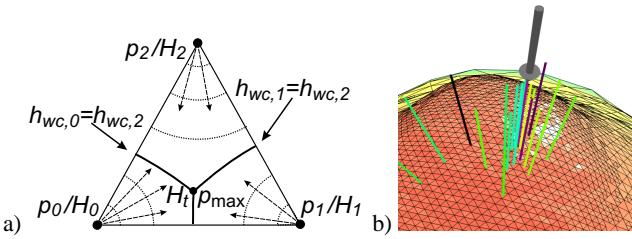


Figure 2: a) finding the maximum reachable entropy value over a triangle, b) steepest ascent of seed (black) from coarse approximation to local maxima (bright magenta), green/cyan lines are the additional samples needed for gradient estimation and the grey arrow shows global maximum

some of these rays are drawn in dashed style. h_{wc} is the point wise minimum of the three functions $h_{wc,i \in \{0,1,2\}} = H_i + \|p - p_i\| \cdot L$. The circles around the triangle vertices in Figure 2 a) illustrate isolines of the functions $h_{wc,i}$. The maximum of h_{wc} is found at the point p_{max} , where all $h_{wc,i}$ have the same value. In the special case, where all H_i are the same, this point is the intersection of the three bisectors between pairs of the triangle vertices. In the general case p_{max} is the solution to the following two equations

$$h_{wc,0}(p_{max}) = h_{wc,1}(p_{max}) = h_{wc,2}(p_{max}). \quad (4)$$

The quartic nature of these equations avoids an explicit solution. A steepest descent solver was used instead, that starts in the triangle center. Whenever p_{max} left the triangle during optimization, it was placed on the edge (p_i, p_j) , through which it left the triangle, by plugging $p_{max} = (1 - \tau)p_i + \tau p_j$ into equation 4 and solving for τ . Compared to the time spent for lighting computations, the time consumed by the iterative solver was negligible, i.e. less than a percent of the total computation time. Figures 1 b) and c) show the triangular subdivision of the unit sphere resulting from the adaptive entropy maximization.

3.4 Fast Maximization Method

Even with the adaptive global maximization method several thousands of lighting calculations are necessary. If we use for example level three of the image hierarchy the maximum could be found in less than a second, but its location is slightly off the location found on the original image resolution.

In this section we combine the speed of the low-resolution maximization with the accuracy of the original resolution. In order to receive a coarse sampling of the entropy function we perform the global maximization procedure to a coarse level of the image hierarchy. Level three proved to be valuable for this purpose. As the maximum found with the coarse image hierarchy level is displaced from the original maximum we limited the spatial resolution by restricting the subdivision process to a maximum number of three iterations. From the coarse sampling we used the local maxima as seed points for an iterative maximization procedure.

A simple steepest ascent method with adaptive step size λ was used. In order to compute the entropy gradient $\nabla H(l)$ we sampled the entropy function in tangential x and y direction at a distance of ϵ from the current location. Because of the roughness of the entropy function, λ and ϵ have to be chosen carefully. We used an initial λ corresponding to half the edge length of the spatial resolution of the coarse sampling. For each seed we performed the following local maximization algorithm:

1. given seed location l with $H(l)$ and initial λ
2. construct tangential direction x and y and set $\epsilon = \frac{\lambda}{2}$

3. sample $H(l + \epsilon \cdot x)$, $H(l + \epsilon \cdot y)$ and estimate $\nabla H(l)$
4. step to $l' = l + \lambda \cdot \hat{\nabla} H(l)$, if $H(l') > H(l)$ set $l = l'$ and goto 2
5. set $\lambda = \frac{\lambda}{2}$, if $\lambda > 0.5$ degrees goto 2

Figure 2 b) shows an example of the adaptive steepest ascent method. The light directions corresponding to the selected seed locations are the black lines. With decreasing λ the brightness of the lines increase to magenta. The green and cyan lines show the additional samples needed for gradient estimation. The grey arrow shows the maximum found with the global optimization procedure and it points nearly exactly to the same location as the brightest magenta line.

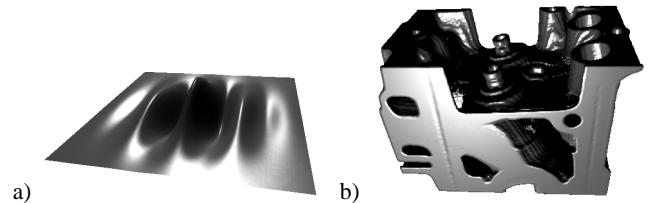


Figure 3: Two examples of maximum entropy light source placement, that exaggerate highlights.

4 THE USER EXPERIMENT

Figure 3 shows two bad examples of maximum entropy illuminations, where huge highlights are placed on flat surfaces. This disturbs more than it is helpful. To learn more about the question of how to design the illumination in order to allow the spectator to optimally perceive the three dimensional shape of the surface we designed the following experiment.

4.1 Experiment Design

The experiment was kept as simple as possible. The five test scenes in figure 7 a) with fixed view points were presented to fifteen subjects, some of them skilled in visualization, some of them not. The lighting was restricted to one directional light source that could be adjusted either with two sliders or by mouse tracking. The subjects were asked to adjust the light such that they could perceive the three dimensional shape of the scene best. Every subject started with the same initial light position and was requested to first investigate the different illuminations before accepting a light. No color was used in the scenes and the background was set to red in order to avoid that the subject concentrates on the visibility of the silhouette.

The subjects easily familiarized with the test environment. The adjustment of the lights in the five scenes took from three minutes to fifteen minutes.

4.2 Results

Figure 7 c) shows the illuminations selected in the user experiments averaged over all subjects, where we averaged the brightness values and not the light positions. In b) the entropy function together with the selected light directions is shown. Again the thirty highest percent of the entropy value are displayed after taken to the power of ten. The large variation in the adjusted positions can be led back by additional questioning onto two factors. Firstly, the subjects could not definitely select the best light position from several good ones.

And secondly, the subjects had different preferences. In the interrogation of the subjects we could find out two common preferences. The highlights need to be chosen carefully at areas with high curvature and should not be too large in extent. And secondly, the scene should not contain dark parts.

5 PERCEPTUAL ILLUMINATION ENTROPY

In this section we want to incorporate the perceptual preferences found in the experiments. For this we refine the illumination entropy definition in two ways. First we change the bins used for classification of the brightness values in order to put penalties on dark and very bright illuminations. In a second step we put more emphasis onto high curvature regions. The extensions will offer three parameters to adjust the perceptual entropy definition. Finally, we select the parameters by fitting the model to the results from the user experiment.

5.1 Perceptual Binning

To prevent the lighting design tool from producing dark regions or too bright regions, we cannot just neglect the contributions to the entropy from probabilities p with small and large indices i , because then for different illuminations a different number of counts are considered, such that the resulting entropies cannot be compared anymore. Instead we chose larger intervals for the bins of low and high brightness values than for bins corresponding to moderate brightness values.

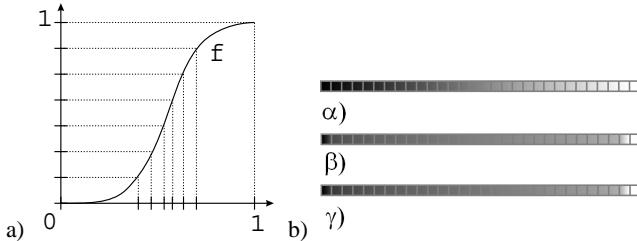


Figure 4: a) example of a brightness preference function, b) brightness bins α identity, β f_0 and γ from model fit

The simplest way to change the bin intervals is to define a strictly monotonic function $f : [0, 1] \mapsto [0, 1]$ with $f(0) = 0$, $f(1) = 1$ and $f(Y_1) > f(Y_2) \Leftrightarrow Y_1 > Y_2$. The bin intervals are defined equal sized in the range of f . The bin index of a brightness value Y computes to $i = \lceil m \cdot (f(Y) + \frac{1}{2}) \rceil$. The function f is called the (brightness) *preference function*. Figure 4 shows an example of a preference function and illustrates how it stretches the bin intervals.

In order to independently discriminate dark and bright regions, we defined an array of curves dependent on two parameters one for dark discrimination: $\omega_0 \in [0, 1]$ and one for bright discrimination: $\omega_1 \in [0, 1]$, where small weights ω correspond to high discrimination and a value of one to no discrimination. We started with a polynomial of degree three, defined as

$$f_0 = -(2 + 3\omega_0 + \omega_1) \cdot Y^3 + (3 - 2\omega_0 - \omega_1) \cdot Y^2 + \omega_0 \cdot Y. \quad (5)$$

The preference function in figure 4 shows f_0 for maximum discriminations $\omega_0 = \omega_1 = 0$. No discrimination ($\omega_0 = \omega_1 = 1$) yields the identity function. The amount of discrimination potential of f_0 is not enough. To increase the potential without destroying the monotonicity we applied f_0 five times resulting in the brightness preference function $f_{\text{perc}}(\omega_0, \omega_1) = f_0(f_0(f_0(f_0(f_0(Y)))))$ for our perceptual model.

5.2 Curvature Importance

The second aspect the subjects took into consideration was the surface curvature. In regions with high surface curvature there should be more information than in flat regions. This can be modeled with the concept of *importance*, which can also be used to emphasize the information content of other important features.

If we multiply the entropy by the total number n of covered pixels we receive the total information of the illumination. As $n \cdot p_i$ is the number of brightness values Y_j falling into bin b_i , each brightness value Y_j contributes $\log_2 \frac{1}{p_i(Y_j)}$ bits to the information. In order to weight the information contributed by the brightness value Y_j of a covered image pixel j with the importance ι_j we define the *importance weighted* illumination information I as

$$I = \sum_{j=1}^n \iota_j \cdot \log_2 \frac{1}{p_i(Y_j)}. \quad (6)$$

The per pixel importance parameter can seamlessly be worked into the presented fast lighting approach by adding an importance accumulator, whenever a counter was used to count grouped pixels. In addition to count the pixels, the importance accumulator adds up the importance values of the grouped pixels.

Given a view of a scene, we have to specify for each covered pixel j an importance value ι_j based on the surface curvature. As we already collected for each covered pixel the surface normals, we can exploit them to estimate the surface curvature. For this we compute for each pixel the length of the normal gradient in radiant from normals of adjacent pixels. This angular measurement is not only high in regions of high curvature, but also at internal silhouettes and sharp creases, what are important features as well. In a second step we sort all pixels by the radiant values and assign the importance based on the position $k(j) \in \{1, \dots, n\}$ in the radiant order

$$\iota_j = \left(\frac{k(j)}{n} \right)^{\omega_\kappa}. \quad (7)$$

The curvature exponent ω_κ is the third parameter in the perceptual entropy model. A value of zero will weight all pixels equally and with increasing exponent the importance of the information on high curvature regions is increased. Figure 7 d) illustrates the importance of the covered pixels for the five test scenes with $\omega_\kappa = 4.2$.

5.3 Fitting the Model

It is left to adjust the parameters ω_0 , ω_1 and ω_κ to the results of the experiment. As the subjects preferred significantly different light source positions, we could not deduce an optimal light source position from the experiment that could be compared to the perceptual entropy based optimal light source position. Instead we interpreted all light source positions from the user experiment as good but not optimal. Thus for a given scene the perceptual entropy function $I_{\omega_0, \omega_1, \omega_\kappa}(l)$ mapping the light source position l to the perceptual illumination entropy resamples the user experiment if it reproduces high entropy values $I_{\omega_0, \omega_1, \omega_\kappa}(l_i=1 \dots n)$ for all selected light source positions $\{l_1, \dots, l_n\}$, where high should be measured relative to the maximum achievable entropy. These considerations lead to the following measure M for the wellness of the fit between a perceptual entropy function and the user experiment.

$$M(\omega_0, \omega_1, \omega_\kappa) = \sum_{i=1}^n \left(\frac{I_{\omega_0, \omega_1, \omega_\kappa}(l_i)}{\max_{l \in S^2} I_{\omega_0, \omega_1, \omega_\kappa}(l)} \right)^2$$

The ω 's were derived for each scene as the parameters that maximize M . To do this in reasonable time we first sampled M on a

coarse grid over the ω 's. We chose the grid

$$(\omega_0, \omega_1, \omega_\kappa) \in \{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\}^2 \times \{0, 1, 2, 4, 8\}.$$

In a second phase we applied a similar local optimization method

	gabor	engine	lobster	crocodile	galleon
n	77874	47003	94755	53762	92990
t_{glb}	241	505	138	164	117
$\#\text{glb}$	3666	6290	3416	3582	1522
t_{app}	14.2	10.8	5.2	4.5	9
$\#\text{app}$	149+154	137+123	145+127	162+66	129+106
seeds	7	5	4	3	4
ω_0	0.96	0.94	0.94	0.95	0.93
ω_1	0.085	0.084	0.082	0.057	0.068
ω_κ	4.7	4.3	3.9	4.0	4.0

Table 1: Results for the test scenes: number of covered pixels n , optimization time for global optimization t_{glb} in seconds, number of illuminations $\#\text{glb}$, for fast approximate optimization time t_{app} and number of coarse level illuminations plus number of high resolution illuminations and the number of *seed*, finally the fitted ω 's.

as the described steepest ascent procedure. The last three columns of table 1 show the resulting ω -values. All five scenes yielded similar values justifying the presented approach. There is one outlier: the ω_1 value discriminating the highlights is smaller in the case of the crocodile. This is probably due to the fact that the bumpy surface of the crocodile always produces a lot of highlights. The averaged ω 's were $\omega_0 = 0.94$, $\omega_1 = 0.075$ and $\omega_\kappa = 4.2$. Figure 7 b) shows the perceptual entropy functions for these ω -values together with the light locations adjusted by the subjects. In e) the optimal illuminations are shown.

6 MULTIPLE LIGHT SOURCES

In most cases a single light source does not result in a satisfactory illumination. Therefore, we have to generalize our approach to an arbitrary number of light sources. We followed two approaches. For the case of two directional light sources we considered the enlarged search space and again globally optimized the entropy function with a trade-off parameter between information maximization and light source independence.

As every further directional light source will introduce two more free parameters what makes the optimization procedure more and more difficult, we also implemented a procedure that adds a further light to a given illumination maximizing the perceptual entropy of the combined illumination. This allows for fast placement of an arbitrary number of light sources.

6.1 Independence vs. Maximum Information

The independence of two illuminations can be expressed with conditional entropies. Suppose two illuminations $Y_{1,j}$ and $Y_{2,j}$ of n covered pixels are given. The combined illumination Y_j with both light sources turned on computes to $Y_{12,j} = \min(1, Y_{1,j} + Y_{2,j})$, from which the perceptual illumination entropy $I(Y_1, Y_2) := I(Y_{12,j})$ can be computed. Let us define the conditional probabilities $p_{i_2|i_1}(Y_1, Y_2)$ that a brightness value $Y_{2,j}$ of a pixel j falls into bin b_{i_2} under the condition that $Y_{1,j}$ of the same pixel already fell into bin b_{i_1} and the probabilities $p_{i_1,i_2}(Y_1, Y_2)$ that the brightness

values $(Y_{1,j}, Y_{2,j})$ fall into the bins i_1 and i_2 . Then the conditional entropy computes to

$$H(Y_2|Y_1) = \sum_{i_1=1, i_2=1}^m p_{i_1, i_2} \log_2 \frac{1}{p_{i_2|i_1}}.$$

$n \cdot H(Y_2|Y_1)$ measures the information added by illumination Y_2 under the precondition that the scene is lighted with Y_1 . With the same method as in the case of the unconditioned entropy, we can compute the importance weighted conditional probabilities via

$$I(Y_2|Y_1) = \sum_{j_1=1, j_2=1}^n \iota_{j_1} \cdot \iota_{j_2} \cdot \log_2 \frac{1}{p_{i(Y_{2,j})|i(Y_{1,j})}}.$$

An optimal placement of two light sources should maximize on the one hand $I(Y_1, Y_2)$ and the total conditional information $I(Y_2|Y_1) + I(Y_1|Y_2)$ and on the other hand it should ensure that each light source adds a similar amount of information to the illumination, i.e. the difference $I(Y_2|Y_1) - I(Y_1|Y_2)$ should be minimal. This leads to the following definition of an *optimality measure* for two light sources

$$M_\tau(Y_1, Y_2) = (1 - \tau) \cdot I(Y_1, Y_2) + \tau \cdot (I(Y_2|Y_1) + I(Y_1|Y_2)) \quad (8)$$

$$|I(Y_2|Y_1) + I(Y_1|Y_2)|. \quad (9)$$

The parameter $\tau \in [0, 1]$ allows to trade off between maximum

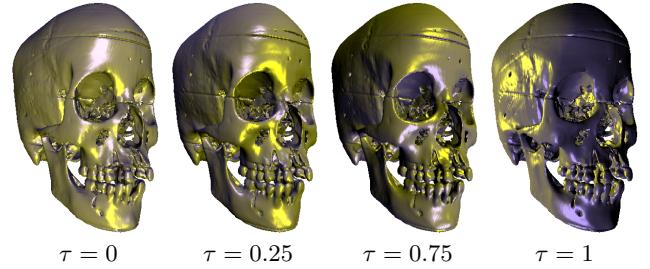


Figure 5: illumination of a skull with a yellow and a blue light sources and different independence parameters τ

entropy ($\tau = 0$) and independence of the light sources ($\tau = 1$). Figure 5 shows illuminations of a skull with different values of τ for the placement of a yellow and a blue light source. The smaller τ the higher is the information content but also the higher is the overlap of the illuminations, such that for $\tau = 0$ the blue light source cannot be distinguished. For $\tau = 1$ the illuminations are maximally independent, which can be easily seen from the good perception of the blue and yellow regions on the skull.

We used the same optimization strategies as in the case of single light source placement and used the same sampling over S^2 for both light source. On each subdivision with light positions $\mathcal{L} = \{\hat{l}_1, \dots, \hat{l}_k\}$ the optimality measure $M_\tau(\hat{l}_1, \hat{l}_2)$ is computed for all pairs $(\hat{l}_1, \hat{l}_2) \in \mathcal{L}^2$. To avoid multiple lighting computations, the illuminations were cached. The k^2 optimality measures are combined by maximization to the function \mathcal{M} over S^2

$$\mathcal{M}_\tau(\hat{l}_1) = \max_{\hat{l}_2 \in L} M_\tau(\hat{l}_1, \hat{l}_2),$$

which is used for further adaptive subdivision of the sampling function. As the computational complexity is already in the order of k^2 in the number k of sample locations, this approach is not suitable for generalization to more than two light sources. Instead we designed the following incremental approach.

6.2 More Than Two Light Sources

Given a set of light sources, the addition of a further light source is quite simple. For this we lighted the scene with the known light sources and store the resulting illumination in a color image. Then we optimize the position of a further light source with the techniques described in section 3. For each tested light location we combine the buffered illumination with the illumination of the new light source into a second color image, from which we extract the brightness values and compute the perceptual entropy.

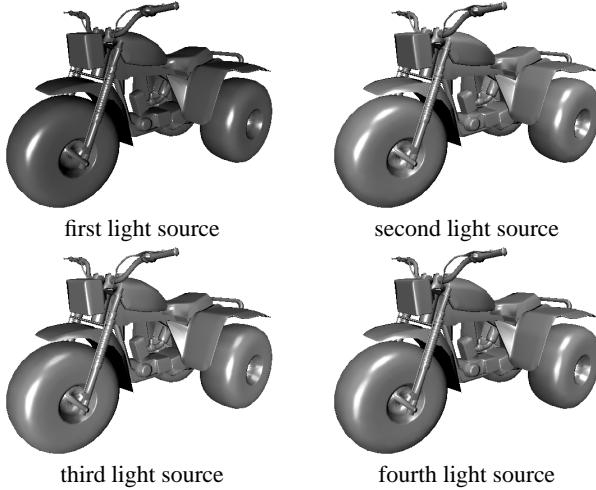


Figure 6: incremental adding up to four white light sources

If additionally the intensities of the different light sources are optimized, even higher maximum entropy values can be reached. When we compute the perceptual entropy for a given light position of the further light source, we first lighten the scene with the new light source. In the combination step we vary the intensity of the new light source between zero and one as well as the intensity of the combined illumination of all previous light sources and select the weights that achieve the maximum entropy. For the case of only white light sources the process of adding new lights soon converges in the sense that the entropy is not increased anymore. For colored light sources the process converges slower as the brightness range of colors significantly darker than white is smaller and the entropy maximum cannot be reached as fast. Figure 6 shows the incremental addition of up to four white light sources. In the final image the intensities of the lights sources were 0.46, 0.35, 0.46 and 0.5. The bottom row of figure 7 shows the test scenes lighted with differently colored light sources.

7 CONCLUSION

We started off with an intuitive motivation of how to place light sources by maximizing the illumination information. A safe and a fast global optimization procedure were designed that work well with entropy functions. We also proposed a fast and hierarchical method for the lighting calculations. From user experiments we learned how to quantify the perceived information content. For this we adapted the binning needed for entropy computations and we introduced an importance weighting based on surface curvature measured in the image. The resulting perceptual entropy function reproduced the results of the experiment.

Furthermore, we generalized the approach to more than one light source with the notion of independence applicable for two light sources and the incremental addition of a light source for an arbitrary number of light sources.

In future work we want to investigate light sources in general positions. For this the same optimization strategies can be exploited. Basically, we have to implement a tetrahedral grid that allows for adaptive subdivision. A further very important point for future work is the incorporation of shadows as can be done with the relighting engine described by Gershbein and Hanrahan [4].

REFERENCES

- [1] T. Arbel and F. Ferrie. Viewpoint selection by navigation through entropy maps. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, volume I, pages 248–254, Los Alamitos, CA, September 20–27 1999. IEEE.
- [2] Robert M. Bevensee. *Maximum entropy solutions to scientific problems*. Prentice-Hall, 1993.
- [3] Antonio Cardoso Costa, Antonio Augusto Sousa, and Fernando Nunes Ferreira. Lighting design: A goal based approach using optimisation. In Dani Lischinski and Greg Ward Larson, editors, *Rendering Techniques '99*, Eurographics, pages 317–328. Springer-Verlag Wien New York, 1999.
- [4] Reid Gershbein and Pat Hanrahan. A fast relighting engine for interactive cinematic lighting design. In Sheila Hoffmeyer, editor, *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)*, pages 353–358, New York, July 23–28 2000. ACMPress.
- [5] G. Joblove, W. Latham, K. Sims, and S. Todd. The applications of evolutionary and biological processes to computer art and animation. *Computer Graphics*, 27(Annual Conference Series):389–390, 1993.
- [6] John K. Kawai, James S. Painter, and Michael F. Cohen. Radioptimization - goal based rendering. In *Computer Graphics Proceedings, Annual Conference Series*, 1993, pages 147–154, 1993.
- [7] Sandeep Kochhar. A prototype system for design automation via the browsing paradigm. In *Proceedings of Graphics Interface '90*, pages 156–166, May 1990.
- [8] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [9] J. Marks, B. Andelman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 389–400. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [10] Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo, and Donald Greenberg. Painting with light. In James T. Kajiya, editor, *SIGGRAPH 93 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 143–146. ACM SIGGRAPH, ACM Press, August 1993.
- [11] Carlo H. Sequin and Eliot K. Smyrl. Parameterized ray tracing. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, pages 307–314, July 1989.
- [12] Ram Shacked and Dani Lischinski. Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.
- [13] Karl Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25(4):319–328, July 1991. SIGGRAPH '91 Proceedings.
- [14] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Viewpoint selection using viewpoint entropy. In Thomas Ertl, editor, *Proceedings of Vision, Modeling, and Visualization '01*, Annual Conference Series. Addison Wesley, November 2001.
- [15] A Witkin and M. Kass. Spacetime constraints. *Computer Graphics*, August 1988, 22(4):159–168, 1988.
- [16] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

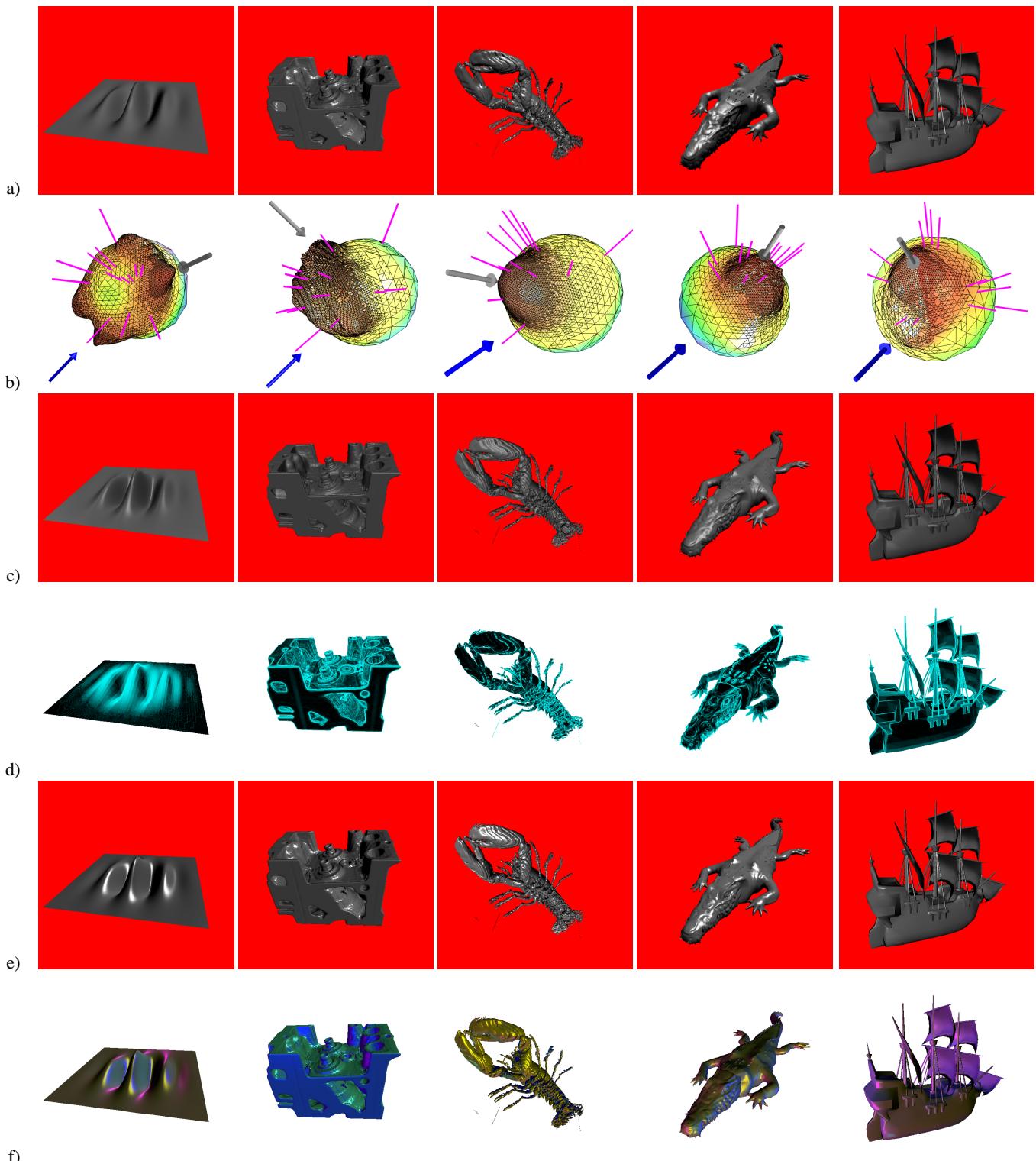


Figure 7: a) the five test scenes, b) fitted perceptual entropy function and selected light source directions, c) average of brightness values over the adjusted illuminations, d) curvature importance, e) optimal illumination maximizing perceptual entropy, f) scenes illuminated with multiple colored light sources