

Automatic Light Source Placement for Maximum Visual Information Recovery

Pere-Pau Vázquez

Departament de LSI, Technical University of Catalonia, Barcelona, Spain

Abstract

The automatic selection of good viewing parameters is a very complex problem. In most cases, the notion of good strongly depends on the concrete application. Moreover, when an intuitive definition of good view is available, it is often difficult to establish a measure that brings it to the practice. Commonly, two kinds of viewing parameters must be set: camera parameters (position and orientation) and lighting parameters (number of light sources, its position and eventually the orientation of the spot). The first ones will determine how much of the geometry can be captured and the latter will influence on how much of it is revealed (i. e. illuminated) to the user. Unfortunately, ensuring that certain parts of a scene are lit does not make sure that the details will be communicated to the user, as the amount of illumination might be too small or too high. In this paper we define a metric to calculate the amount of information relative to an object that is effectively communicated to the user given a fixed camera position. This measure is based on an information-based concept, the Shannon entropy, and will be applied to the problem of automatic selection of light positions in order to adequately illuminate an object. In order to validate the results, we have carried out an experiment on users, this experiment helped us to explore other related measures.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three Dimensional Graphics and Realism]: Color, shading, shadowing, and texture

1. Introduction

The selection of the adequate viewing parameters is a very complicated problem that is usually solved through a large process of test and error that requires longtime human dedication. Obviously, the necessity of human intervention becomes impractical for large collections of models. Moreover,

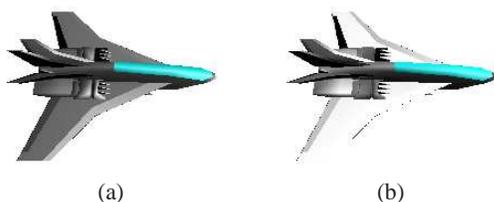


Figure 1: Good and bad lighting of an airplane. In (a) we can see both the shape and the details of the plane while the illumination of image (b) has a large specular region that avoids us to distinguish the shape of the object.

many applications from scientific visualization are often employed by users with little or null experience in Computer Graphics. It is desirable then to find a way to automatically setting adequate viewing parameters. Recently, research on the automatic placement of cameras has attracted the attention of the Computer Graphics community, but only few papers focus on the important problem of correctly illuminating a scene. See for instance Figure 1, where a good and a bad illuminations of an airplane model are shown. Our aim in this paper is to present an information theory-based metric that measures the contributions of regions of the same (perceptually) CIE LUV color in order to evaluate the amount of visual information generated by illumination contained in a view. We apply this measurement for the automatic selection of light positions that adequately illuminate an object or scene. Finally, we have tested the quality of our measure using photographs that were evaluated with our method and by users.

The rest of the paper is organized as follows. In Section 2 we review previous work on parameter tweaking and

the related work of automatic camera placement, Section 3 presents our perception-based measure of visual information revealed by illumination. In Section 4 a method for the automatic positioning of a single light source is developed. We also show how the extension to several light sources is straightforward. In Section 5 we have made a study that compares the results of our measure with the opinion of more than thirty people for a certain set of digital photos of real objects. In Section 6 we present a new approach based on Multi Scale Entropy. Finally, Section 7 concludes our work and points out some lines for future work.

2. Previous Work

The selection of viewing parameters for scene rendering is a complex and tedious process. Parameter tweaking can be divided into two phases: a) Camera position and direction setting, and b) Light source selection and positioning. The problem of good camera positioning has become an active field of research mainly due to the emergence of the so-called Image-Based Rendering methods. Light source selection has usually been treated from the point of view of inverse lighting. Now we proceed to review the previous work on these two areas of research.

2.1. Camera Placement

Colin [Col90] presents a method to select a good view to observe a scene modeled with an octree. Kamada and Kawai [KK88] define a criterion for the quality of a view for orthogonal projections. Plemenos and Benayada [PB96] extend Kamada's definition, considering the amount of detail shown in a view as the number of visible faces. Barral *et al.* [BDP00] present a method for the automatic exploration of objects or scenes. In this case, the quality of a view is computed by defining a new importance function that depends on the visible pixels of each polygon.

Vázquez *et al.* [VFSH01] have presented a new measure based on the Shannon entropy [Bla87], *viewpoint entropy*, to evaluate the amount of *geometric* information seen from a point. It has been successfully applied to some Computer Graphics problems such as automatic selection of good views of molecular models [VFSL02], Image-Based Modeling [VFSH03], or automatic path computation through complex models [AVF04].

In the robotics literature, the goal of selecting a small set of cameras which allow to observe all object surfaces has also been studied. Usually this problem is stated as: *Determine where to place the next camera position given N previous camera locations, for $N \geq 0$* . The two main approaches are: search-based and silhouette-based. Search-based methods use optimization criteria to search a group of potential viewpoints of the next best view (see for instance Wong *et al.* [WDA99] or Massios and Fisher [MF98]). Silhouette-based methods use the visible silhouette of objects to create

heuristics that predict where the next best view is placed (see for instance Abidi [Abi95]). Other methods worth mentioning focus on the complete coverage in 3D modelling (as in Klein and Sequeira [KS00], Roberts and Marshall [RM98], and Tarbox and Gottschlich [TG95]).

2.2. Light Source Selection

Adequate lighting selection research can be divided in two subfields: inverse lighting, and maximum visual information communication. Here we review recent work on these fields.

In inverse lighting, the user specifies how the scene should look like and the adequate parameters are searched. Therefore it is assumed that the user has a knowledge on the object shape and material properties. Although there is a broad bibliography in inverse lighting (for a survey see [PP03]), we only cite here some examples. Schoeneman *et al.* [SDS*93] describe an interactive system that, given a set of lights with fixed positions, determines their colors and intensities in order to match a target image painted by the designer. Kawai *et al.* [KPC93] control light emissions and directions, as well as surface reflectances for designing the illumination in an environment rendered with a radiosity based method. As in the former case, the user has to specify how the final image should be illuminated. None of these methods automatically sets the light source positions. Costa *et al.* [CdSF99] have implemented an automatic method of light placement and intensity selection. Their objective is to obtain a configuration that determines a given radiance distribution. Although it is a powerful approach, the objective function needs to be specified by the user with a scripting language, and therefore is not easily applicable when the user is not expert. Poulin and Fournier [PF92] and Poulin *et al.* [PRJ97] manipulate highlights and shadows in order to define a resulting illumination. These modifications are translated to the corresponding changes in light sources positions. Jolivet *et al.* [JPP02] present a Monte-Carlo method for the selection of light positions in direct lighting. They use a declarative paradigm in order to help the users to describe the lighting goal in a more intuitive way.

Opposite to inverse lighting problems, other methods seek the adequate light sources configuration that reveals the maximum of information to the user, by means of adequately placing light sources, no matter which object or scene is inspected. The Design GalleriesTM (DG) system is a method to automatically set parameters for computer graphics and animation. They automatically compute and organize sets of views or animations which are perceptually different from each other. The resulting images are presented to the user to choose among them. Apart from some parameters concerning to material properties, they also study light selection and placement [MAB*97].

Gumhold has also explored the problem of automatic parameter setting [Gum02]. He has presented a method for the

automatic light source placement which uses an entropy-based function, the *lighting entropy*. He defines the unit of information $-\log p_i$ as a function of the measured brightness of the visible pixels. The brightness of a pixel is computed as the Y tristimulus value of the CIE 1931 standardized color model. It is calculated with the following formula: $Y = 0.21262 \cdot R + 0.71514 \cdot G + 0.07215 \cdot B$. Then, the lighting entropy is defined as $H(X) = -\sum_{i=1}^m p_i \log p_i$, where the probability p_i is defined as the number of pixels whose brightness falls into interval i (the logarithms are taken in base 2 and $0 \log 0 = 0$ for continuity). The author defines the unit of information $-\log p_i$ as the number of pixels that fall into interval i where index i is computed as $i = \lceil m \cdot (Y + \frac{1}{2}) \rceil$. Therefore, their measure is maximum when the number of different brightness values in an interval is uniform across the scene. The number m of intervals chosen by the author is 30. As the author says, tests with users revealed that most of the best views as selected by his method were discarded by them due to the fact that they presented too large specular regions. Thus, he improved his method taking into account the comments of the users. Some fast methods for light positioning are also presented.

Halle and Meng present a system for lighting three dimensional scenes [HM03]. Although their system is not fully automatic and must be parameterized with a total of six parameters, they provide a default light sources configuration that is claimed to produce good results for a variety of scenes. In this case, instead of defining a quality metric, they use photographic, cinematographic, and theatric experience to adapt the three-point lighting model by Birn [Bir00] to interactive visualization. A similar approach, also based on cinematography and theater is presented by El-Nasr and Horwill [ENH04].

Shacked and Lischinski [SL01] propose a quality function formed by six terms that are weighted by the user. Their system optimizes these parameters based on a perceptual quality metric. Their objective is to effectively communicating information on the scene: shapes, materials, and their relationships. The quality metric they build is composed by six factors. Each of them is devoted to a different kind of information (such as edge detection or variance reduction), and some of them may have contradictory effects such as the histogram equalization term that, when applied, tends to increase the variance (which is controlled by the variance reduction term). Therefore, manual calibration is necessary for every scene whereas they have empirically found some weights that perform well for a certain number of models.

2.3. Image Information Measurement

There has been an extensive research in image information measurement with different objectives, such as image compression or noise reduction to name a couple. Typical approaches measure the amount of information contained in an image by an entropy-based measure, the Shannon entropy:

$$H(X) = -\sum_{i=1}^N p_i \log p_i,$$

where $X = \{X_1, X_2, \dots, X_N\}$ is an image containing integer values and N is the number of different values of a pixel and p_i are the values obtained from the histogram of X :

$$p_i = \frac{\#X_j = i}{N},$$

the logarithms are taken in base 2 and $0 \log 0 = 0$ for continuity. As $-\log p_i$ represents the *information* associated with the result X_i , the entropy gives the average *information* or the *uncertainty* of a random variable. The unit of information is called a *bit*. In general, the process of measuring information consists in creating a histogram of colors of the image and each entry of the histogram forms a probability in the entropy. Shannon entropy is null when the channel contains no information and maximum when all probabilities are equal. In our case the channel is the image, thus, this means that entropy will be 0 if all the pixels are the same color and maximum when all pixels have a different color.

However, this is not the only approach. For instance, for image restoration other entropies are used because Shannon entropy leads to minimization functions that are difficult to handle (see [SFPA98]). Consequently, several other measures have been proposed, the main ones are:

- Burg [Bur78]:

$$H_b(X) = -\sum_{i=1}^N \ln X_i,$$

where X_i is the number of pixels that have value i .

- Frieden [Fri78]:

$$H_f(X) = -\sum_{i=1}^N X_i \ln X_i$$

- or Gull and Skilling [GS84]:

$$H_g(X) = -\sum_{i=1}^N X_i - M_i - X_i \ln \frac{X_i}{M_i},$$

where M_i is the number of pixels of value i of a given model, usually a flat image.

In all cases N is the number of pixels and i represents the index of a pixel. All these entropies correspond to different probability distributions that one can associate with an image. The key difference is that the initial proposal by Shannon is null for a flat signal and increases when the data contains information. On the contrary, the proposed measures are maximum when the image is flat and decrease with information [SFPA98, SFR98]. In the concrete case of the Gull-Skilling approach, the entropy is null when the image is equal to the model M .

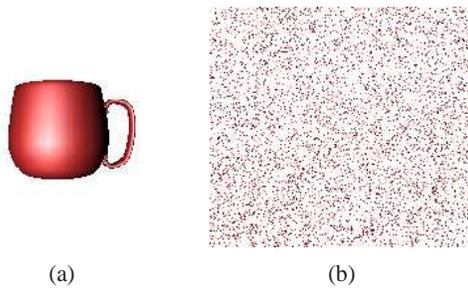


Figure 2: The same pixels differently distributed in two images. In (a) we can see a mug while in (b) the pixels of the left image have been randomly distributed. Both images would yield the same color histogram.

Unfortunately, all these entropy definitions lead to measures that are insensitive to pixel correlation or dependent on the background [SFR98]. This is illustrated in Figure 2 where the same pixels appearing on the left image appear randomly distributed on the right image. All these measures would yield the same entropy value for both images whereas, depending on the application in mind both can be differently evaluated. For instance, for someone who is involved in image processing, the image on the left contains more information than the image on the right, while for someone working on image transmission, it is clear that the second image will require higher bandwidth for lossless transmission. To overcome those problems one can use a multiresolution scheme that has been developed under the name of multiscale entropy. It basically consists in analyzing the entropy of different instances of the same image at different resolutions. In image restoration some techniques which involve wavelet coding have been used successfully. These techniques are based on the Gull-Skilling information measurement, and they completely depend on the chosen noise model [SFR98]. However, we will show in Section 5 that a variation of this approach, that uses Shannon entropy instead of the proposed measure, does also work for automatic light source placement.

3. Measurement of Visual Information due to Illumination

In this section we present our measure of visual information revealed by illumination based on Shannon entropy.

3.1. Introduction

From the previous work, only Gumhold [Gum02] has defined a fully automatic method based on Shannon entropy for lighting parameter tweaking which needs no user intervention. The probability distribution of the entropy measure presented is based on the normalized number of pixels whose

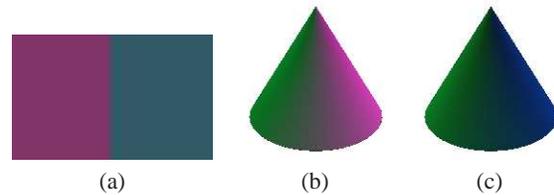


Figure 3: In (a) we can see two different colors which have the same Y value. (b) and (c) show two different illuminations of a cone. The first one, with a pink and a green light source, and the second with a green and a blue light source. The former provides a higher amount of information because those colors are more easily distinguishable for us than blue and green in the right image. However, as in the right cone blue and green have similar brightness values and therefore some regions that appear different to us, are summed together, this one would be chosen if we do not take into account perceptible differences.

brightness falls in an interval. Unfortunately, such a distribution function may cause some problems. First, the use of brightness values does hide the information concerning to the color that is perceived by the human visual system. Consequently, some colors that appear different to us are considered to be the same with this method when computing the amount of illumination information (see Figure 3a). If only a single material and a single light source (or several light sources with the same emission color) is present in the scene, this would not be important. However, scenes with many different materials or with two differently colored light sources might yield contradictory results, as an illumination which is best according with this method may be communicating a lower amount of information than another one. This is exactly what happens in Figures 3b and 3c. The left cone is illuminated with a pink and a green light sources while the right one is illuminated with a blue and a green light sources. When displayed, our visual system distinguishes easier between green and pink than between blue and green, and therefore the left cone shows a higher range of perceptible colors and therefore communicates a higher amount of information. As blue and green on the right cone have similar brightness values, with Gumhold's method the right cone would be chosen. Second, the normalization by the number of projected pixels does produce a lateral effect: a scaling of a model, under the same lighting conditions does yield the same entropy with this method.

In this paper we propose a new metric for illumination information measurement. It differs from previous approaches in that we measure the perceptual-based spectrum of LUV-colors. Moreover, we take into account spatial issues, that is, all the pixels with the same LUV color will accumulate if they form a single continuous region (see Section 3.2), as otherwise they are perceived as separate sources of informa-

tion by an observer. We proposed this measure in a previous work [VS03a]. Here, we show how it can be used for evaluating visual information due to illumination from real images and we propose other metrics for illumination information measurement (see Section 6).

3.2. An Information Theory-based measure of the Illumination Information

An accurate positioning of a light source must reveal as much information as possible of a scene. The problem is how to define a formula that indicates the quantity of information present in an image. Vázquez *et al.* [VFSH01] have developed an entropy based measure that can be used to determine the amount of geometric information captured from a point. This value is calculated by projecting the scene onto a bounding sphere centered in the viewpoint and using as distribution function the relative projected areas of each face. Therefore, its value is maximum when all the faces are seen with the same relative projected area. Although this definition effectively solves the problem of measuring geometric information, when we render a scene the current shading may hide some faces of the object. Thus, to measure the illumination information that arrives to a point, it is necessary to build a distribution function based on the visual stimulus that arrives to this point. Moreover, as the human visual system is limited, the measure must be sensitive to this and only take into account perceptible information. With these conditions we can build an entropy by using as unit of information the relative area of each region *whose color is different from its surrounding*. The expression of this measure will be:

$$H_L(X) = - \sum_{i=0}^m A_i \log A_i,$$

where m is the number of different regions, A_i is the relative area of the region i . In this case, the number of regions is not determined by any user-defined parameter but these regions are automatically detected from an analysis of the image, as explained later in this section. The background is represented by region index 0. Remark that it is important to take the background into account because the probability distribution must add up to 1. Some approaches do not use the background information and instead they normalize the probabilities dividing by the number of valid pixels (see Gumhold [Gum02]). But this normalization may hide some information, as a zooming-in of a certain view (provided that the object still remains inside the viewing frustum) will give the same value despite we are really seeing the object better and therefore this fact should be detected (see Figure 4).

Observe that the background plays another important role. In most cases, the background is chosen black or white to have a different color than the rest of the object, nevertheless, if the scene does not contain an ambient term, it might happen that the object is completely black under a certain light

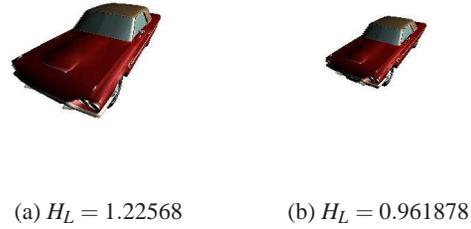


Figure 4: In (a) we can see a zoom-in of (b). If the background was not used, both would yield roughly the same quality value. With illumination entropy we have a noticeable difference.

position. If the background is black, the information present is zero, as we are not seeing anything. On the other hand, if the background is not black, it will help us to see the silhouette of the object, and this must be somehow taken into account. The same problem may arise when the background is white, as specular regions may be interpreted as background, specially if these regions are near (or at) the bounds of an object. In our examples we used a background color of white and therefore the silhouette of a completely dark object is not zero. In any case, the background is counted when measuring the illumination entropy, as this will reflect in some manner whether the silhouette is visible or not. Another important problem is the color regions measurement, which is treated in the following subsection.

3.3. Region Selection

As stated, most of the previously presented measures do not take into account the distribution of the colors or brightness values across the scene [Gum02, Bur78, Fri78, GS84], while this may also be informative to the user ([SFR98]). So, in order to take this into account we take as information unit the relative area of each isolated region of the same color. We do this instead of accumulating the relative area of all the pixels of "the same" color (in the sense that they are not distinguishable by the human visual system), as a larger number of regions implicitly provides a higher amount of information. This is easily demonstrated in Figure 5 where two different distributions of the same number of black and light grey pixels are shown. The left image contains a higher amount of information than the image on the right. With most of the previously proposed measures both images would yield the same value. Thus, we need some kind of segmentation algorithm that creates a different region for every different perceptible color. Note that this does not mean that we are simply *counting* the number of different regions. This simple strategy does not work as an infor-

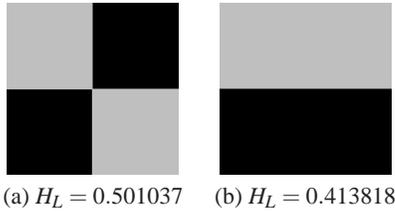


Figure 5: (a) shows a checkerboard of four squares. Note that in (a) we intuitively identify four squares, while in (b) only two regions are perceived. Therefore, (a) is more informative than (b).

mation measurement in this context (see [VFSH03] for the details).

In our approach the colors are transformed to CIE LUV format. In order to detect if two neighboring pixels have the same color, we use the CIE LUV color difference formula: $\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2}$. It is known that colors that have a CIE LUV difference of less than 1 appear to be the same [NMP98], so this is the criterion we have used to distinguish between regions. For a real complex environment, we could consider differences of more than one and we would probably obtain similar results. The region detection is created through a so called seed algorithm. To avoid that a region labeled with the same color could have some pixels that compared to each other yield a difference value of more than one (this could happen if the belonging criterion depends only on the closer neighbors of the pixel), we always compare the pixels with the initial (seed) one. This prevents a very smooth color gradient traversing a big scene to be labeled as a single region. Moreover, this avoids the problem of mixing perceptible different colors in a same measure, as it happens in Figure 3. If we were using brightness values, we would select the right cone (with a lighting entropy of value 0.8058 versus a lighting entropy of value 0.7860 for the left cone) while our method yields a smaller value for the right cone (illumination entropy of 0.9562 versus a value of 0.9598 for the left one).

The criterion used to determine that two pixels belong to the same connected region is: their color difference (in LUV space) is below one and they share at least one edge. With this method a typical checkerboard texture will not be considered as two different regions but 64 (we consider here a situation of constant shading along the texture, otherwise, the illumination will also introduce a higher number of different perceptible regions). For large images, where two pixels sharing only a single edge may be difficult to detect to us, we can raise to a larger number of edges of contact. In Figure 5 we can see this with an example. Figure 5a has a higher illumination entropy than Figure 5b, which corresponds to our perception of four regions in the first case and two in the second.

4. Automatic Light Source Placement

4.1. Single light source

We have implemented our method in two stages:

1. A non-optimized version that uses OpenGL default's lighting.
2. An optimized version which renders soft shadows.

In a first stage, the selection of the best position for a single light source has been implemented as a brute-force algorithm. We place the light source at a set of different positions on a bounding sphere of the object and measure its illumination entropy. These positions are placed regularly on the bounding sphere. The position with the highest quality is selected. Note that this method is general and can be applied whichever the shading algorithm is used (that is, we could use the results of a radiosity simulation, for instance), as the calculations are performed on the resulting image. In this case we have used OpenGL's default lighting mode.

In a second stage, we have improved the rendering tool in order to generate soft shadows based on the algorithm by [UA04]. We have also built an optimization method similar to the one by Gumhold [Gum02].

4.1.1. Optimized version

Our initial algorithm only uses OpenGL default shading and is not optimized for speed. Its purpose is to determine the best light position by inspecting a dense set of light positions regularly placed on a bounding sphere of an object. The results are compared with Gumhold's method [Gum02], which also works with OpenGL simple shading. In the next section we present an optimization method suitable for OpenGL rendering.

The highest cost of this method is incurred by the rendering tool and the capture of the data. Each time a new illumination has to be analyzed, the scene must be rendered and the image must be read back to main memory. In order to accelerate the computation we reduce the size of the window read back to main memory by reading the depth buffer at the beginning and inspecting it in order to obtain the bounding box of the object projection. This allows to obtain a proper light position in one to four seconds for scenes of several thousands of polygons. Some other accelerations can be added, but they are strongly dependent on the rendering tool.

As seen in [SL01], it is difficult to visually evaluate the quality of a rendering. One of the keys that can help to decide if the method is good or not is the amount of details that are revealed with the selected illumination parameters. On the other hand, the optimized images with the method by Shacked and Lischinski [SL01] also yield the highest values of illumination entropy with our system, with the advantage that we do not need a preprocess of calibration and different tuning for each model or view. In Figure 6 we compare the results of our algorithm with Gumhold's method. Note that,

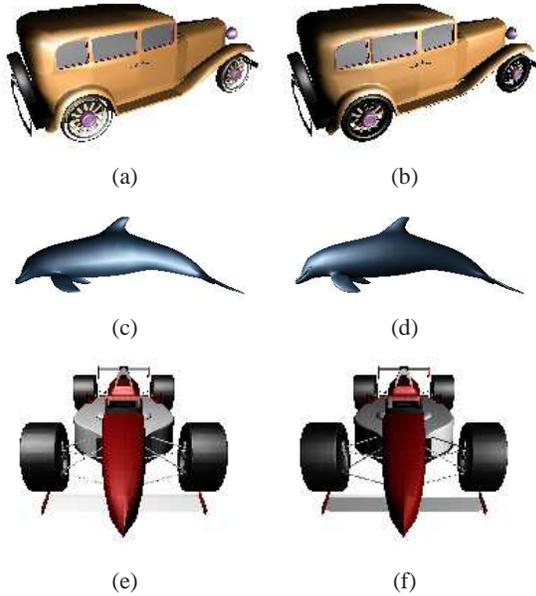


Figure 6: Comparison of our method with Gumhold’s system. (a), (c), and (e) were generated with Gumhold’s method, and (b), (d), and (f) with our measure. We can see some details that are not revealed in (a) (the wheels of the car) and (c) (the eye and face of the dolphin) while they are visible in (b) and (d) respectively. Furthermore, note how some parts of the silhouette may be hardly distinguishable in Figure (e) (the front flaps of the Formula 1 car) because the background information is not taken into account and is shown in (f).

although in some examples the illumination results are similar, in others some detail may be hidden with Gumhold’s method due to the fact that the intensities are weighted together even though they may come from different regions of the scene (compare the wheels of Figure 6a and 6b or the front flaps of the Formula 1 car in Figure 6e and 6f, for instance). In order to validate the quality of our method we have asked to several people to choose between each pair, in all cases there is more people that prefers the best images selected with our method than the ones generated using Gumhold’s system (see Table 1 in Section 5 for the results).

4.1.2. Optimization

We have improved the rendering tool in order to simulate soft shadows using the algorithm proposed by Uralsky and Ahmad [UA04], also published in [Ura05]. This algorithm is based on percentage-closer filtering method [RSC87] and exploits the capabilities of graphics hardware to *blur* the pixels near the border of the hard shadow. As this method is more costly than the simple rendering using OpenGL illumination (and eventually hard shadows), we have also developed a global optimization algorithm that looks for the best light source position adaptively.

The optimization presented is similar to the algorithm by Gumhold [Gum02] and the algorithm presented in [VS03b]. We start with a set of points placed on a bounding sphere of the object of interest. We analyze the illumination information generated when placing light sources at these points and start an adaptive subdivision that uses the information captured in order to refine the search for the best light position. As the entropy function is not continuous, it is difficult to determine the maximum reachable entropy in the neighborhood of a certain point whose entropy is known. To solve this problem, Gumhold assumes that the entropy measurements are Lipschitz continuous and uses the assumption to estimate the maximum reachable entropy in the neighborhood of a point and subdivides the triangles of the initial mesh accordingly.

We use a similar strategy than the presented in [VFSH01]. First, an initial mesh is created by placing a set of points around the object. Then we evaluate the entropy that yields a light source placed at these points. Then, we start an iterative process that has three steps:

1. First, we calculate the maximum entropy growing rate in all edges.
2. We estimate the maximum reachable entropy in all edges and the point of maximum entropy.
3. We insert new points (calculate entropy and add the new edges) at the positions where estimated entropy is higher than the current maximum evaluated entropy.

The maximum growing rate allows us to infer how much can entropy grow if we insert a new point in an edge. We do it the following way: Let G be the maximum growing rate. We approximate the behavior of the entropy function between two connected points by a piecewise linear function composed by two pieces of slope G and $-G$ respectively. Let H_1 and H_2 be the entropies of the two endpoints of the edge, and d the distance between them, the point x of maximum entropy should be placed at: $x = (H_2 - H_1 + G * d) / (G * 2)$.

Then we estimate the entropy in point x and this point will be a candidate to insert a new light position if its estimated entropy is higher than the already computed. Then, we insert new points at these positions, we evaluate the illumination entropy that generate, and insert the required new edges.

This process is repeated several times. Our experiments found that three to four iterations are enough to find a good light source position, usually better than the found with the brute force method (of course, unless we use an extremely dense set of points to evaluate with the naive algorithm). In most cases the algorithm takes less than half a second. Some examples can be found in Figure 7. In Figure 8 we show the inspected positions for the models shown in 7a and 7c respectively. Note how the adaptive subdivision scheme avoids the regions where not high entropy can be expected (in these cases, the positions where the light would be placed right under the object, that usually gives poor visual results). For the sake of comprehension, we have scaled the mesh according

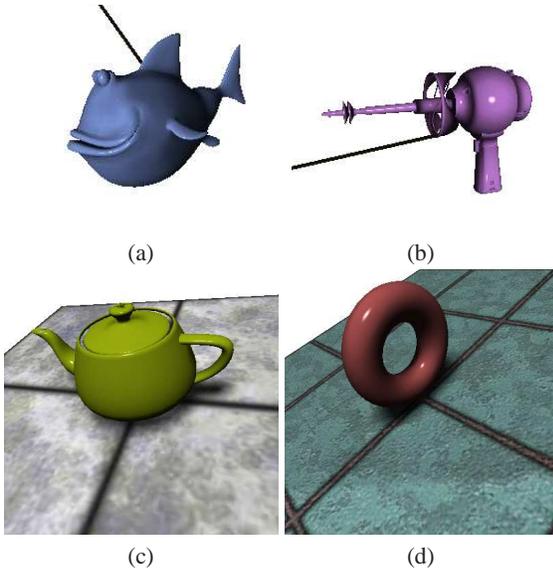


Figure 7: In (a) and (b) we can see two examples of objects adequately illuminated with our optimized method. Figures (c) and (d) shows two more images where the optimal illumination also shows the soft shadows cast by the light source.

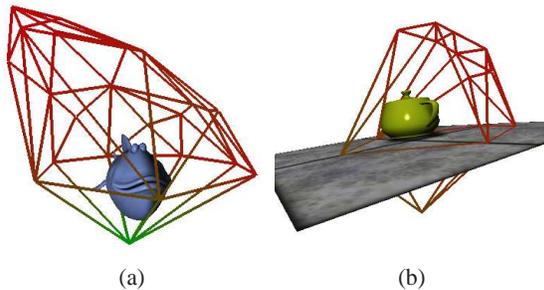


Figure 8: The inspected meshes. In (a) the point of minimum entropy (green) is on the other side of the camera location, while in (b) also the point under the floor is green because the light source did not illuminate the scene.

to the entropy at the vertices, that is, the vertices with lower entropy are closer to the object and the vertices with higher entropy are further. Moreover, the vertices of higher entropy are colored red and the ones of lower entropy are colored green.

4.2. Several Light Sources

Given the presented measure, it is straightforward to extend the algorithm to cope for more than one light source. Although it has the disadvantage that it scales badly with the number of light sources, it is interesting to see the behaviour of our method. In this case we also use simple OpenGL light-



Figure 9: Optimal configuration of two light sources for a pear and a car models.

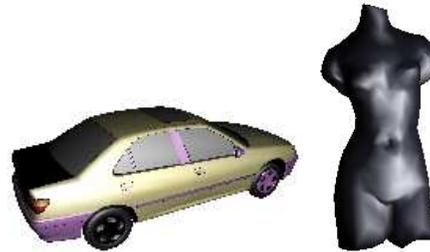


Figure 10: Automatic selection of light intensity and position.

ing and no optimization has been made, although the introduction of a second light source to our optimized method does not seem difficult.

In Figure 9 we can see two examples of objects optimally illuminated by two light sources, a pear and a car.

4.3. Other parameters

Apart from adding light sources, it is also easy to extend the implementation in order to cope for other parameters such as the optical properties of the light source, such as the intensity, the attenuation factor, or the lighting direction in case of directional light sources. In the example of Figure 10 we also look for the best intensity color in a range going from 0 to 1. We did not test with colors different from grey levels because it is not intuitive that it can hide material properties if they are complementary to the light emission color, and because the user usually expects something close to a *white* light source, but the implementation is trivial.

5. Validation

In this Section we present the experiment conducted to compare the results of our measure with the opinion of users.

5.1. Test conditions

In order to validate the usefulness of our measure, we have carried out a series of experiments with users. First, we have



Figure 11: Test images.

Model	% Votes
Car 6a	29.2%
Car 6b	70.8%
Dolphin 6c	25.0%
Dolphin 6d	75.0%
Car 6e	29.1%
Car 6f	70.9%

Table 1: Synthetic images test. We have compared the best view as selected with our method with Gumhold's best view. Rows Car 6a, Dolphin 6c, and Car 6e were chosen by Gumhold's Lighting Entropy while the rest were selected by our method.

compared the results with synthetic images, and then with real pictures. In the first case, we compare the results shown in Figure 6, and ask the users to choose between each pair. In the second case, we took several pictures with a digital camera under different illumination conditions and asked the users to select the one that provided the highest amount of information. This set of images is shown in Figure 11.

All the tests were done on the same computer to maintain the same optical conditions of the screen. The users were all asked the same question "Which is the image that provides a highest amount of information?", and they were shown five sets of four pictures at full resolution (all images are larger than 360×360). Each of the four pictures was taken under different illumination conditions in a dark room and with a lamp. At each picture, the lamp was changed in order to provide significant different illumination conditions.

We also asked some of the people which was the reason for selecting one or another picture and the answers were not very coincident. The only noticeable comment was on the picture *Eraser 1* which was chosen by most of the users because, as they said, the text was easier to read.

Images in Figure 6 were analyzed by 24 people and the real ones (Figure 11) were shown to a total of 33 people. Some of the people were skilled in Computer Graphics and some were not. The results show the same preferences in all the image sets for both groups. For the synthetic images, we compare the best views as selected by Gumhold's method, and the ones selected with our method. The percentage of votes appear in Table 1. Note that users prefer the images selected with our method over Gumhold's approach.

The results concerning the the users' votes on real images appear in the second and third columns of Table 2.

5.2. Illumination entropy

Once the people was asked, we ran our algorithm on the images and compared with the results of the tests. The results

Model	Votes	Rank	Illumination Entropy	Rank
Apple 1	3.03%	4	2.8111	4
Apple 2	24.24%	2	2.86844	3
Apple 3	66.66%	1	2.99963	1
Apple 4	6.06%	3	2.90306	2
Phone 1	15.15%	3	3.18279	3
Phone 2	3.03%	4	3.0847	4
Phone 3	30.30%	2	3.29065	1
Phone 4	51.51%	1	3.27789	2
Bottle 1	18.18%	3	3.47712	3
Bottle 2	45.45%	1	3.41176	4
Bottle 3	33.33%	2	3.52244	2
Bottle 4	3.03%	4	3.62498	1
Eraser 1	24.24%	2	2.25908	2
Eraser 2	0.00%	3 – 4	2.173378	4
Eraser 3	0.00%	3 – 4	2.18667	3
Eraser 4	75.75%	1	2.28459	1
Mouse 1	27.27%	2	2.81646	2
Mouse 2	0.00%	4	2.75692	4
Mouse 3	15.15%	3	2.82946	1
Mouse 4	57.57%	1	2.78751	3

Table 2: Illumination entropy results. The tests show that our measure selects the same best picture than the users 2 out of the 5 tests and in another case (the mobile phone) there is almost coincidence (first and second rank are changed).

with our measure are shown in the last two columns of Table 2.

Although there is not a total success, note that there is notable coincidence in 3 out of the 5 models. In two cases there is a total match of the ranking and in the other case the first and the second positions are changed. For the remaining two sets our measure does fail. In the following Section we introduce another measure that yields a better results for this experiment.

6. Multi Scale Entropy

In this Section we introduce a measure based on Multi Scale Entropy and show how it performs even better than illumination entropy for our test images.

6.1. Introduction

As commented in Section 2.3, apart from the Shannon entropy, some other entropies have been used in order to determine the amount of information of an image. We have

also tested with Burg's approach [Bur78] and Frieden's method [Fri78]. However, as expected, both methods do not handle pixels correlation properly, this leading to measure both the images in Figure 2 as equally informative. We also tested with the Gull and Skilling's entropy [GS84]. For these experiments, it is necessary to have a flat image in order to subtract it in the formula (see Section 2.3). In our case we used as flat image the background of our images.

Unfortunately, in all these cases, the entropy does not behave as expected for images such as the checker board in Figure 5, if we consider it to have four regions instead of one, as when applying the formula, the entropy (in this case in absolute value) does not grow.

To overcome these problems, Starck introduces the concept of multiresolution into the entropy [SFPA98]. He considers that the information in some dataset is the sum of the information at different resolution levels l . The wavelet transform W of a signal by a fast algorithm contains a set of coefficients $w_{l,k}$, where l is the level, and a set of c_k coefficients representing the signal at a very low resolution. If the number of levels is high enough, we can assume that the coefficients c_k furnish information only about the background, and not about the signal of interest.

6.2. A Multiresolution Entropy for Illumination Evaluation

As we have seen, given a wavelet decomposition of an image, we can measure the information on the wavelet coefficients. Note that this is useful in the cases where we are illuminating objects, as most of the image will contain background, but this also leads to good measurements for scenes.

In order to measure the amount of information contained in an image, we define the illumination contents as:

$$H_W(X) = - \sum_{l=1}^L \sum_{k=0}^{N_l} H(w_{l,k}),$$

where L is the number of levels of the multiresolution, and $H(w_{l,k})$ the Shannon entropy of the wavelet coefficients of level l (that is $-\sum_{i=1}^n p(w_{(l,k)_i}) \log p(w_{(l,k)_i})$).

The Multi Scale Entropy method by Starck is based on the Gull and Skilling measure because it is used for image restoration of astronomical pictures. As we only want to account for the information present in an image, we use the Shannon entropy of the wavelet coefficients. Taking this approach, we take advantage of the wavelet transform in order to reduce the effects of noise that has been introduced in our digital images when capturing. Thus, for the analysis of the test series, our measure will theoretically account for the object represented and the background noise will be somewhat alleviated. In our case we have implemented the Haar wavelet transform over the RGB image (the three channels are considered as integer values ranging from 0 to 255 as

usual). We do not use the LUV transformed image because in this case we do not need the notion of color distance for perception.

It is important to note that in all of the intuitive cases appearing in this paper the illumination entropy from the wavelet transform also behaves as expected, that is, it also distinguishes better the colors we are able to see best (Figure 3), it is sensitive to a change of size (Figure 4), and it is sensitive to different distribution of the pixels across the image (Figure 5). We compare the results we obtain with our wavelets scheme with the users votes in Table 3. Notice the high level of coincidence in the apple, the bottle, and eraser (first and last ranks match, and in some cases there is a total match on the ordering of the pictures). In the other two sets the image ranked as the best one by our algorithm corresponds to the one which had the second number of votes (moreover, in the phone case, the positions 3 and 4 match). It is important to note that the numbers shown here were computed using only 4 levels of wavelet decomposition. With a higher number of levels, the distances between the results increase, but the ranks do not change. In general, 3 to 5 levels are enough to rank the images.

6.3. Discussion

We have proposed two measures of the amount of information present in an image that seek to maximize the information revealed to the user given a certain illumination of the scene. This leads to the design of a fully automatic method. In order to validate the results and explore its application for the illumination measurement of real images we have carried out an experiment on more of thirty people. The results show that both of our measures are likely to be used with real images. In particular, the measure based on the wavelet transforms behaves slightly better, probably due to the fact that the transform somewhat alleviates the noise that is present on images mainly due to the background that was chosen for the capture process. On the other hand, the initial approach is good enough and is faster because the measurements do not need to transform the image.

Furthermore, both of the presented measures have several advantages. They are compact and general and thus, they must not be manually calibrated for every scene, as they give a quantity that can be compared with successive renderings of the same scene. As they measure the amount of information present in an image, they can also be used to compare between similar images. See for instance Figure 12. It shows four images of a mandrill, the one on the top left is the original and the rest have been blurred with a kernel of 3×3 , 5×5 , and 7×7 pixels respectively. Intuitively, we can see that the amount of information present on the second one is lower than in the former, but how different, we do not know. With our method we can measure the amount of information of both maps and prove that the result corresponds with intuition.

Model	Votes	Rank	Multi Scale Ill. Entropy	Rank
Apple 1	3.03%	4	82.1669	4
Apple 2	24.24%	2	83.0565	3
Apple 3	66.66%	1	83.5652	1
Apple 4	6.06%	3	83.2745	2
Phone 1	15.15%	3	86.3995	3
Phone 2	3.03%	4	85.7774	4
Phone 3	30.30%	2	86.5848	1
Phone 4	51.51%	1	86.5808	2
Bottle 1	18.18%	3	86.7166	2
Bottle 2	45.45%	1	87.0839	1
Bottle 3	33.33%	2	86.1157	3
Bottle 4	3.03%	4	85.8089	4
Eraser 1	24.24%	2	84.2826	2
Eraser 2	0.00%	3-4	84.0394	4
Eraser 3	0.00%	3-4	84.1415	3
Eraser 4	75.75%	1	84.6501	1
Mouse 1	27.27%	2	87.4908	1
Mouse 2	0.00%	4	87.2889	2
Mouse 3	15.15%	3	86.4938	4
Mouse 4	57.57%	1	87.1582	3

Table 3: Multi scale entropy results. In this case, the results are very good, and the measurements are very similar to the opinion of the test people. In three out of the five cases, the images ranked as first and last match the results of the interviewed people, and another case (the phone) the first and the second positions are changed. Finally, in the other two sets (mouse and phone) the chosen by the users is the one we rank second best (note that in the case of the mouse when capturing the last image the camera was slightly moved accidentally, but as most of the users did not notice, we kept the image on the test).

The six quality terms employed by Shacked and Lischinski focus similar goals [SL01]. For instance, the histogram term seeks to equalize the amount of quantities of each luminance value appearing on the scene, although without taking into consideration the part of the scene where they appear. Hence, we made some tests with the images provided by the authors and we found that the images selected as the best ones with their optimized method are also selected as the best ones with both of our methods.

7. Conclusions and Future Work

In this paper two measures of the visual information provided by the illumination have been presented. The first one takes as information unit the area of each region with the same perceptible color. In order to do this, we first trans-

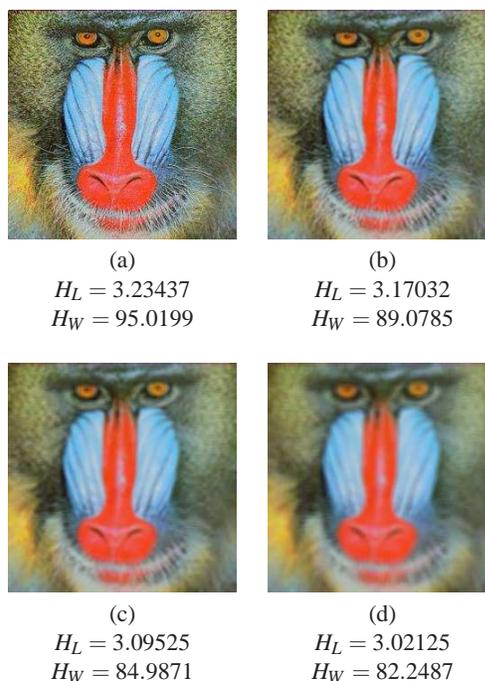


Figure 12: Filtering an image loses information. This sequence of images is composed by the original one and subsequent filterings with a blur kernel of size 3×3 , 5×5 , and 7×7 respectively. As expected, the amount of measured information decreases.

form the RGB image into the CIE LUV space. The second approach iteratively transforms the input RGB image using the Haar wavelet transform. An algorithm for the automatic selection of the best light position that uses those measures has been implemented. Moreover, to validate the results, we have conducted an experiment with more than thirty users. Several sets of four pictures of the same object, each one under different illumination conditions were created and showed to users. Their opinions are very similar to the results obtained with our measures, in particular, the method that works on the wavelet transforms yields very good results, with a high degree of coincidence with interviewed people.

The first metric is faster to evaluate and it is therefore suitable for the automatic selection of light positions if speed is important. We have built an optimized method that adaptively samples positions using the computed entropies to infer the probably best position. We have also extended our initial method to cope with two light sources.

In the future we will explore other measures based on the wavelet transform. Some applications are envisioned, such as for placing light sources in architectural models (in this case probably together with the help of a lighting design technique) or automatically lighting environments in games.

Our methods can also be used in order to find interesting exploration paths in complex models, as given a certain illumination we can measure the amount of information that can be revealed given a certain camera path.

Acknowledgments

The author wants to thank Mateu Sbert for proof reading the paper and his useful comments. Moreover, the author thanks the collaboration from all the people that helped to do the test both from the University of Girona and the Technical University of Catalonia. Many thanks to the anonymous reviewers for their useful comments that helped to improve this paper. This project has been partially supported by TIN2004-07451-C03-01 and TIN2004-08065-C02-01 of the Spanish government and the GAMETOOLS (IST-Work Programme) from the EU Commission.

References

- [Abi95] ABIDI B.: Automatic sensor placement. In *Proc. Intelligent Robots and Computer Vision: Algorithms, Techniques, Active Vision, and Materials Handling* (1995), pp. 387–398. 2
- [AVF04] ANDÚJAR C., VÁZQUEZ P.-P., FAIRÉN M.: Way-finder: guided tours through complex walkthrough models. *Computer Graphics Forum* 23, 3 (Sep 2004), 499–508. 2
- [BDP00] BARRAL P., DORME G., PLEMENOS D.: Scene understanding techniques using a virtual camera. In *Proc. Eurographics'00, short presentations* (2000), de Sousa A., Torres J., (Eds.). 2
- [Bir00] BIRN J.: *Digital Lighting and Rendering*. New Riders Press, Thousand Oaks, CA, 2000. 3
- [Bla87] BLAHUT R. E.: *Principles and Practice of Information Theory*. Addison-Wesley, Cambridge, MA, 1987. 2
- [Bur78] BURG J.: Modern spectral analysis. In *Annual meeting international society exploratory geophysics* (1978), Childers D., (Ed.), IEEE, pp. 34–41. 3, 5, 11
- [CdSF99] COSTA A. C., DE SOUSA A. A., FERREIRA F. N.: Lighting design: A goal based approach using optimization. In *Rendering Techniques '99* (New York, NY, 1999), Springer Wien, pp. 317–328. 2
- [Col90] COLIN C.: Automatic computation of a scene's good views. In *Proc. MICAD* (Feb. 1990). 2
- [ENH04] EL-NASR M. S., HORSWILL I.: Automating lighting design for interactive entertainment. *ACM Computers in Entertainment* 2, 2 (2004), 15–15. 3
- [Fri78] FRIEDEN B.: *Image Enhancement and Restoration*. Springer-Verlag, Berlin, Germany, 1978. 3, 5, 11
- [GS84] GULL S., SKILLING J.: Maximum entropy method in image processing. In *IEE Proceedings 131, Part F* (1984), IEE, pp. 646–659. 3, 5, 11
- [Gum02] GUMHOLD S.: Maximum entropy light source placement. In *Proc. of the Visualization 2002 Conference* (Oct. 2002), IEEE Computer Society Press, pp. 275–282. 2, 4, 5, 6, 7
- [HM03] HALLE M., MENG J.: Lightkit: a lighting system for effective visualization. In *IEEE Visualization* (2003), IEEE, (Ed.), Harvard Med. Sch., Harvard Univ., USA, pp. 363–370. 3
- [JPP02] JOLIVET V., PLEMENOS D., POULINGEAS P.: Inverse direct lighting with a Monte Carlo method and declarative modelling. *Lecture notes in Computer Science* 2330 (2002), 3–12. 2
- [KK88] KAMADA T., KAWAI S.: A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing* 41, 1 (Jan. 1988), 43–56. 2
- [KPC93] KAWAI J. K., PAINTER J. S., COHEN M. F.: Radioptimization - goal based rendering. In *Proc. of SIGGRAPH-93: Computer Graphics* (Anaheim, CA, 1993), pp. 147–154. 2
- [KS00] KLEIN K., SEQUEIRA V.: The view-cube: An efficient method of view planning for 3d modelling from range data. In *Proc. of the Workshop on the Application of Computer Vision* (Palm Springs, CA, December 2000), pp. 186–191. 2
- [MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUMML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: A general approach to setting parameters for computer graphics and animation. In *SIGGRAPH 97 Conference Proceedings* (Aug. 1997), Whitted T., (Ed.), Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 389–400. ISBN 0-89791-896-7. 2
- [MF98] MASSIOS N. A., FISHER R. B.: A best next view selection algorithm incorporating a quality criterion. In *Proc. of the British Machine Vision Conference* (1998), pp. 780–789. 2
- [NMP98] NEUMANN L., MATKOVIC K., PURGATHOFER W.: Perception based color image difference. *Computer Graphics Forum* 17, 3 (Sept. 1998), 233–241. 6

- [PB96] PLEMENOS D., BENAYADA M.: Intelligent display in scene modeling. new techniques to automatically compute good views. In *Proc. International Conference GRAPHICON'96* (July 1996). 2
- [PF92] POULIN P., FOURNIER A.: Lights from highlights and shadows. *Computer Graphics* 25, 2 (Mar. 1992), 31–38. 2
- [PP03] PATOW G., PUEYO X.: A survey on inverse rendering problems. *Computer Graphics Forum* 22, 4 (2003), 663–687. 2
- [PRJ97] POULIN P., RATIB K., JACQUES M.: Sketching shadows and highlights to position lights. In *Proceedings of Computer Graphics International 97* (June 1997), IEEE Computer Society, pp. 56–63. 2
- [RM98] ROBERTS D., MARSHALL A.: Viewpoint selection for complete surface coverage of three dimensional objects. In *Proc. of the British Machine Vision Conference* (1998). 2
- [RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM Press, pp. 283–291. 7
- [SDS*93] SCHOENEMAN C., DORSEY J., SMITS B., ARVO J., GREENBERG D.: Painting with light. In *SIGGRAPH 93 Conference Proceedings* (Aug. 1993), Kajiya J. T., (Ed.), Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, pp. 143–146. 2
- [SFPA98] STARCK J., F.MURTAGH, PIRENNE B., ALBRECHT M.: Astronomical image compression based on noise suppression. *Publications of the Astronomical Society of the Pacific* 108 (1998), 446–455. 3, 11
- [SFR98] STARCK J., F.MURTAGH, R.GASTAUD: A new entropy measure based on the wavelet transform and noise modeling. *IEEE Transactions on CAS II* 45, 8 (1998). 3, 4, 5
- [SL01] SHACKED R., LISCHINSKI D.: Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum (Proceedings of Eurographics 2001)* 20, 3 (Sept. 2001), C–215–C–226. 3, 6, 12
- [TG95] TARBOX G., GOTTSCHLICH S.: Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding* 61, 1 (may 1995), 84–111. 2
- [UA04] URALSKY Y., AHMAD A.: *White Paper: Soft Shadows*. Tech. rep., NVidia Corporation, July 2004. Available in <http://developer.nvidia.com>. 6, 7
- [Ura05] URALSKY Y.: *GPU Gems 2: Programming techniques for high-performance graphics and general-purpose computation*. edited by Mat Parr; series editor Randima Fernando, Addison-Wesley, Upper Saddle River, NJ, 2005, ch. 17 (Efficient Soft-Edged Shadows Using Pixel Shader Branching). 7
- [VFSH01] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference (VMV-01)* (Stuttgart, Nov. 21–23 2001), Ertl T., Girod B., Niemann G. G. H., Seidel H.-P., (Eds.), IOS Press, Amsterdam, pp. 273–280. 2, 5, 7
- [VFSH03] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Automatic view selection using viewpoint entropy and its application to image-based modeling. *Computer Graphics Forum* 22, 4 (Dec 2003), 689–700. 2, 6
- [VFSL02] VÁZQUEZ P.-P., FEIXAS M., SBERT M., LLOBET A.: Viewpoint entropy: A new tool for obtaining good views for molecules. In *D. Ebert, P. Brunet, and I. Navazo editors, Data Visualisation 2002 (Eurographics/IEEE TCVG Symposium Proceedings)* (May 27-29 2002), Eurographics/IEEE, pp. 183–188. 2
- [VS03a] VÁZQUEZ P.-P., SBERT M.: Perception-based illumination information measurement and light source placement. *Lecture Notes in Computer Science* 2669 (May 2003), 306–316. 5
- [VS03b] VÁZQUEZ P., SBERT M.: Fast adaptive selection of best views. In *Lecture Notes in Computer Science, number 2669, (Proc. of ICCSA'2003)* (2003), Springer-Verlag GmbH, pp. 295–305. 7
- [WDA99] WONG L., DUMONT C., ABIDI M.: Next best view system in a 3-d object modeling task. In *Proc. International Symposium on Computational Intelligence in Robotics and Automation (CIRA)* (1999), pp. 306–311. 2