Automatic Lighting Design

Thesis by

Hai Nam, Ha

In partial fulfilment of the requirements for the degree of Doctor of Philosophy

Newcastle University

Newcastle upon Tyne, UK

2008

(Submitted March, 2008)

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institution of learning.

Publication

Elements of this work have already been published in:

1. Hai Nam HA, Patrick Olivier. (2006). Perception-based lighting design. *Theory and Practice of Computer Graphics (TPCG)*, pp. 63-69.
   Introduction to the implementation details of the extension to the Shacked's framework, including shadow computations and back-lighting.
2. Hai Nam HA, Patrick Olivier. (2006). Explorations in Declarative Lighting Design. *6th International Symposium on Smart Graphics (SG)*, pp. 160-171.
   Description of the declarative approach to the extension to the Shacked's framework with initial results and the use of different optimisation algorithms.
3. Hai Nam HA, Patrick Olivier. (2007). Perception-based lighting-by-example. *Theory and Practice of Computer Graphics (TPCG)*, pp. 61-68.
   Description of the perception-based *lighting-by-example* framework.
4. Hai Nam HA, Patrick Olivier. (2007). Lighting-by-example with Wavelets. *7th International Symposium on Smart Graphics (SG), pp. 110-123*.
   Description of the wavelet-based *lighting-by-example* framework.

Acknowledgements

This research would not have been possible without the help, support and encouragement of many people and organisations. Firstly, I would like to thank my supervisor, Dr. Patrick Olivier, for your guidance, support and pep talks. You've really made me feel at home here in Newcastle. Apart from your supervision, you are more like a close friend who I can share my thoughts and feelings about everything. I am forever grateful. To my Mum and my family, thanks for everything, you are mighty, and I know I am extremely lucky to have such an amazing family who have always been there for me. Dearest, dearest Dung, how am I going to ever say thanks for your patience and all the wonderful things that you do for me. You are my lucky charm and treasure. Thanks to the Vietnamese people (as represented by the government) who funded my studies, without whom I wouldn't have had the chance to start my research career here in the UK. Special thanks also are due to the Department of Computer Science, Post and Telecoms Institute of Technology, for giving me the time off and the flexibility that pursuing a doctoral degree requires, and the School of Computing Science at Newcastle University, for giving me the support when I needed it most. Lastly, much credit is due to my colleagues in Culture Lab, especially Dan Jackson for the interesting discussions and valuable insights.

Abstract

A significant problem in the automatic design of 3D graphics is the configuration of the lighting for a scene. The number of lights included, and the properties of these lights, has an enormous impact on what a viewer can judge about the content (the objects), properties (the geometric characteristics and spatial relations of the objects) and other aesthetic qualities of a scene. The traditional approach to lighting design for image synthesis is based on manual design methods, whereby users interactively specify values of lighting parameters, render the scene, and modify the lighting parameters until the desired visual properties of the scene are achieved. Non-expert users encounter a number of difficulties in selecting the appropriate lighting parameters, as the process requires both a subtle technical and aesthetic understanding of lighting in computer graphics.

In this thesis, perceptual aspects such as contrast and the non-linear characteristics of our perceptual response to colour are combined with practical studio lighting techniques and a novel treatment of shadows, to yield an extension to existing perceptual approaches to lighting design. This so-called *ideal lighting* approach optimises the lighting configuration for a scene with respect to a set of absolute perceptual metrics. An intuitive approach to lighting design, *lighting-by-example*, is also proposed and extensively explored in forms that exploit both the perception-based lighting framework and a new wavelet formulation. User studies are conducted both to configure the perception-based lighting objective function and to evaluate the performance of the proposed lighting design approaches. Finally, we develop an interactive interface for the lighting design process that incorporates both the *ideal lighting* and *lighting-by-example* approaches.

Contents

List of Figures

Chapter 1: Introduction to Lighting Design

1.1 Problem Statement

A significant problem in the automatic design of 3D graphics is the configuration of the lighting for a scene. The number of lights included, and the properties of these lights, has an enormous impact on what a viewer can judge about the content (the objects), properties (the geometric characteristics and spatial relations of the objects) and other aesthetic qualities of a scene. In a number of design scenarios, the viewpoint on a scene is normally known prior to the design of the lighting. Automated approaches to lighting design can exploit the fact that we know the precise location of all the edges of an object – this includes both silhouette edges, edges between object parts, and self-occluding edges. We also have complete information about which pixels in the image correspond to which geometric features of the objects.

The traditional approach to lighting design for image synthesis is based on manual design methods where users interactively specify the values of lighting parameters, render the scene, and modify the lighting parameters until the desired visual properties such as brightness, shading and contrast of the scene are achieved. Non-expert users are likely to encounter difficulties in adjusting lighting parameters to achieve their desired effects. Indeed, adjusting lighting parameters is not a random process but one driven by both a technical and aesthetic appreciation of lighting in computer graphics. Even experienced artists can find that manual lighting design is a tedious and time-consuming process. Despite this, there have been relatively few attempts either to automate or assist the process of lighting design. Just how lighting design in computer graphics can be automated is the main research question of this thesis. In answering this question, we have developed and evaluated a new approach to lighting design that provides graphic designers with declarative tools for configuring lighting parameters.

1.2 Light Qualities

In the natural world, different classes of light source have uniquely recognizable qualities. By *qualities* of a light source we mean those properties that are generally recognised as making a source distinct from other light sources. While the fundamentals

of lighting in 3D graphics are widely understood, the different qualities of light sources and how to realistically simulate them in 3D renderings is less well known. Exploring the range of lighting qualities available is a useful precursor, as it both circumscribes the range of phenomena involved, and introduces the vocabulary by which designers express the differences in the lighting of scenes. We divide these qualities according to the following attributes of a light source: softness, intensity (and attenuation), colour, and throw [1].

1.2.1 Softness

Softness is one of the most important qualities of light and one that has been found to be particularly difficult to simulate using local illumination techniques (the basis of real-time graphics). In practice, both hard light and soft light can be simulated, but the default local illumination light sources produce a hard light with a sharp focus emanating from either a point in space or an infinitely distant source. It is easy to recognise hard light as it casts sharp shadows and creates small highlights. Figures 1.1 and 1.2 show examples of sharp shadows cast by hard light in both a real environment and in a computer-generated scene.

It is much harder to simulate soft light in real-time since this has a tendency to wrap around objects and to cast diffused soft-edged shadows (rather than sharp shadows of a hard light). The softness of the light primarily depends on distance and size of the light source. The further and smaller the light source is, the less soft it becomes. Soft shadows are normally produced by lights with an extended emissive surface. Figure 1.3 shows an example of soft shadows cast by soft light in a computer-generated scene.

Figure 1.1: *An example of sharp shadows cast by hard light in a real environment* [82].

1.2.2 Intensity

Intensity is one of the most significant qualities of light and it is an important factor when determining the role of a light illuminating in a 3D scene. The intensity of a light can be controlled by the brightness of its colour components. In standard photography and cinematography, the brightness of a film or video is partly dependent on exposure settings of the camera. Of course, in computer graphics, the intensity of a light is directly related to brightness of the final output rather than any adjustment to parameters of the camera itself.

Figure 1.2: *An example of sharp shadows cast by hard light in a 3D scene.*

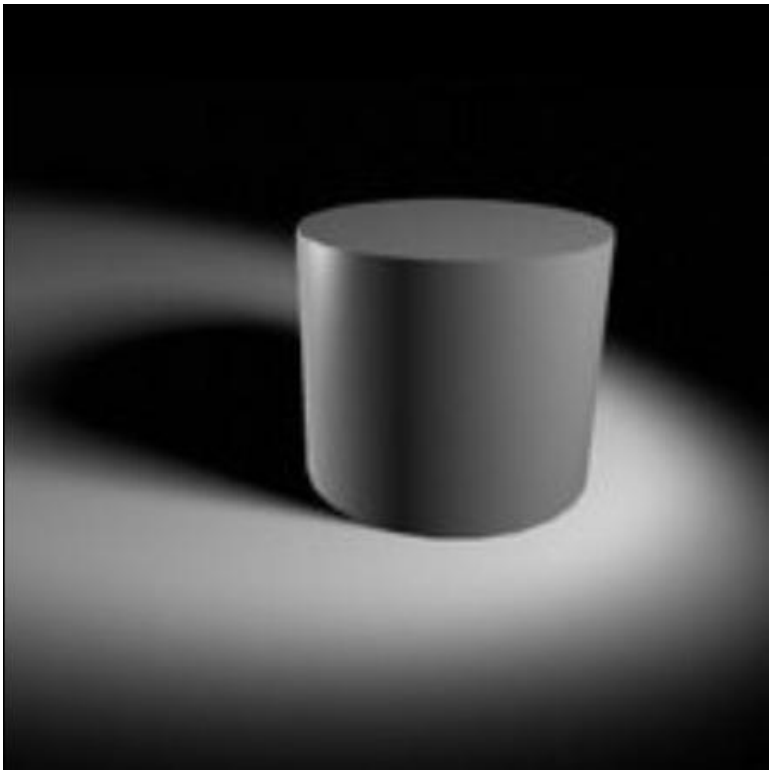


Figure 1.3: *An example of soft shadows cast by soft light in a 3D scene.*

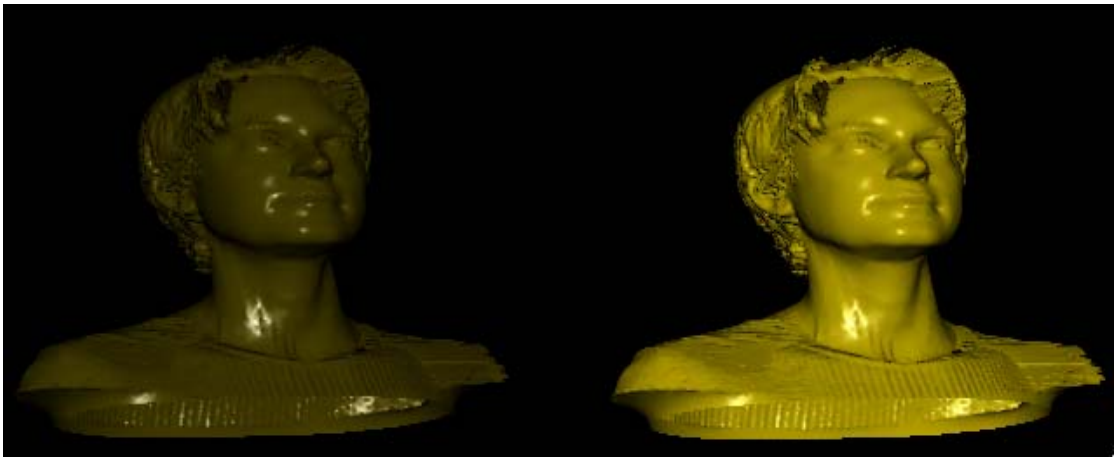


Figure 1.4: *The same model with different intensities.*

1.2.3 Attenuation

The intensity of illumination also varies according to distance between objects and the light source. In reality, the illumination cast on an object decreases as the object is positioned further away from the light – this is referred to as *attenuation*. In the natural world, the amount of light incident on an object is proportional to the inverse of the squared distance between the light source and the object. Attenuation is included in local illumination models in computer graphics, but the *inverse square* relation of reality would create starkly lit scenes (where the light sources are all point-like). Instead, a reduced attenuation function such as a linear function is generally used, although in some cases no attenuation is used at all. A linear attenuation pattern makes illumination decrease evenly from full brightness down to absolute darkness and a cut-off is also often deployed at a specified distance. Having no attenuation is useful for simulating some special natural light sources such as the sun and moon, and in such cases the results are more convincing than those derived by using other attenuation patterns.

1.2.4 Colour

Authentically reproducing natural colours of light can produce more realistic renderings. Research into colour perception has a number of applications in computer graphics. For example, the human visual system adjusts to different dominant colours of light, and the position and direction of a light can be guessed by viewers on the basis of light colour.

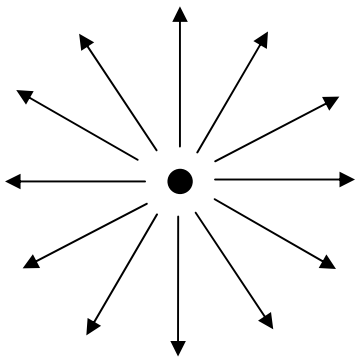
As well as being a crucial quality of light itself, light colour can convey mood and emotion. Creating realistic and impactful renderings requires a thorough understanding of different aspects of colour including colour schemes, colour balance and colour temperature.

1.2.5 Throw

Real light sources often cast patterns of light on subjects rather than illuminating subjects evenly. The quality of how the light is broken up, patterned or shaped, is referred to as the throw or the throw pattern. For instance, whenever you determine the position and direction of a spotlight and adjust the softness and cut-off angle of the spotlight, then you are changing the throw pattern of the light. There are different approaches to creating complex throw patterns in computer graphics, including the use of texture maps to block or limit light transport. Texture map-based approaches are very powerful as we can create many different kinds of throw patterns, such as simulating light passing through certain-shaped windows. Throw is an important quality of light and throw patterns can change a viewer's perception of appearance of the objects and atmosphere for a scene. Realism of the 3D scene can be increased by reproducing various throw patterns that often appear in real world environments.

1.3 Light Types

1.3.1 Point lights



An illustration of a point light: the arrows show the directions of emitted light rays.

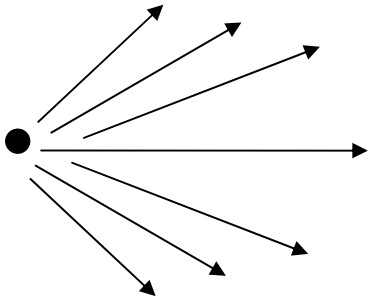
A point light simulates rays emitting from an indefinitely small point in space. Point lights are sometimes referred to as omni-directional light sources due to the fact that rays are emitted uniformly in all directions. In practice, the relative size of a light source over

different spatial scales determines whether it can be reasonably approximated as a point source (i.e. whether its physical extent has any bearing on the actual illumination of a scene). Hence, a point light source is not necessarily physically small and might in reality be a distant star in space or a bare light bulb viewed from several metres.

Figure 1.5: *Example of a point light simulated in computer graphics.*

In the real world, we are unlikely to find a uniformly omni-directional light as most light sources tend to emit more light in some directions than others. Theoretically, in three dimensions, light energy emitted from a point light source is proportional to the inverse square of the distance from point light source with assumption that the distribution is homogenous in all directions, and there is neither absorption nor loss. Although point source lights have this omni-directional characteristic, in computer graphics we can always give a point light a throw pattern that is uneven, like real bulb with cover, to focus more light in some directions than others.

1.3.2 Spotlights



An illustration of a spotlight: the arrows show the directions of emitted light rays.

Spotlights are fundamental tools of lighting design in computer graphics and are preferred by graphic designers as they are easy to direct the focus on specific targets. Much like a point light, a spotlight simulates light emitted from an indefinitely small point. A spotlight, however, differs from a point light in its effective space of illumination, where light emitted from a spotlight is limited to a specified cone which allows a spotlight to produce a beam of light. The beam of light of a spotlight is easily controlled by adjusting the width of the cone.

Figure 1.6: *Example of a spotlight simulated in computer graphics.*

Figure 1.7: *Example of a spotlight in real life [83].*

1.3.3 Directional lights

An illustration of a directional light: the arrows show the directions of emitted light rays

When light is positioned at an infinite distance from the objects being lit, the rays radiating from the light are parallel. Directional light sources can be considered as point lights positioned infinitely far away and the shadows cast by objects are similarly perfectly parallel. In other words, directional lights are infinitely-wide beam light sources. In computer graphics, directional lights are often used to simulate the light sources that are large distances from objects being lit, such as the sun. Light from the sun is virtually parallel because of the extremely small angle subtended, even across the diameter of the Earth. For directional lights, the intensity of light received by objects

being lit is hardly affected by the light position, as the size of the objects are very small compared with the distance from light source to the objects. The only thing that really matters in placing a directional light is what the light is pointing at. In computer graphics, directional lights are useful secondary lighting devices but are rarely used as the main lights for a subject.

Figure 1.8: *Object in the scene is lit by a simple directional light.*

1.4 Lighting Design from the Artist's Viewpoint

Lighting design in film, TV, theatre and computer graphics is primarily the work of artists. Over time, each artist develops their own lighting design process that is appropriate to their own way of working. However, there are usually a number of key steps that most artists will share.

1.4.1 Script analysis

Script analysis is the process of gathering detailed information necessary to successfully complete a lighting design by systematically analyzing the script for story elements, technical requirements and other considerations such as mood and foreshadowing. Understanding the story and how to interpret and re-tell that story using lights is the real key to an attractive, successful and cohesive lighting design. This involves a careful reading of the script in order to gather the technical requirements for each scene, such as

time of day, weather conditions, light sources, placement of light sources like lamps and windows and placement of distinctive objects.

1.4.2 Background research

Background research simply involves gathering further information about relevant properties of scene elements such as optical properties of the materials (including historical details). *Visual research* [84] involves understanding the setting and its surrounding environment. Visual elements will include such aspects as specularity, glossiness, diffuseness and reflectivity. *Technical research* [84] primarily takes place during early phases of a project, and includes employing specific lighting techniques in order to achieve desired lighting effects and determining the rendering power required to get all frames rendered on time and on budget. *Dramatic research* [84] tackles the very heart of the story and characters. Understanding the story and all the characters in their fullness, seeking the emotional truths that lie at its heart, is vital to making decisions about how to light specific scenes. Understanding what the writer is really trying to say will equip the graphic designer to provide a more truthful interpretation and, therefore, more truthful lighting.

1.4.3 Planning

The lighting designer will usually create a lighting plan. Depending on size and scope of the project, a lighting plan can vary significantly in detail and would normally include representational sketches and drawings of their lighting intent. Initially, there is no need to be specific about light types, specific colours, or specific techniques - a plan view of the scene and sketch in general directions and purpose will suffice. A formal lighting plot is not a mandatory part of the lighting design, unless the lighting designers want to convey large amounts of information very precisely. A formal lighting plot contains specific information about light placement and uses standardised sets of symbols and notations.

1.4.4 Implementation

From the perspective of a graphic designer, the process of implementation is one of composing, testing and revising renderings. The lighting workflow is an experimental

endeavour. In computer graphics, lighting design starts with an adjustment to the designer's display such that the whole colour palette is perceivable by the human visual system. A well-lit scene should take advantage of the full range of tones in the colour palette. A poorly adjusted monitor may result in under-lighting or over-lighting scenes, as only a limited range of tones within the available colour palette can be employed during lighting design.

When lighting a computer graphics scene, artists usually start with total darkness, although in practice a small amount of ambient illumination is used. However, the reproduction of ambient light sources in computer graphics is far from the realities of scattered light in real environments, and graphic designers employ different kinds of light sources in order to achieve realistic general illumination, rather than employing standard (simple) ambient light levels.

The next step in lighting workflow is to add light sources to the scene in order to achieve the desired lighting effect. This vital step requires the graphic designer to have a thorough understanding of the characteristics of different light sources, such as point lights, spotlights, directional lights and area lights. Studies [1] have shown that light adjustment takes up to 95% of a lighting designer's time (with only 5% used for the initial configuration). Adjustments require the artist to re-render the scene. Lighting parameter adjustments involve not only the newly added lights but also other lights in the scene. Hence, efficiently managing tests and revisions is an important part of producing high quality renderings. This requires an understanding of what exactly each light contributes to the lighting effects in the scene. In order to enhance the performance and efficiency of the testing and adjustment process, several techniques can be employed, such as light isolation, false colour lights, version flipping, and composition using paint tools.

1.4.5 Evaluation

Evaluation is usually the last step in which the designers revise their work and try to learn from it, to remember the design elements that really worked and to remove or rework elements that are not satisfactory. Computer graphic lighting designers have advantages which are not available to stage, video, or film lighting designers as, on the stage, lighting designs are implemented at immense cost and are hard to adjust.

However, in computer graphics lighting design, usually a single artist implements the lighting design, tweaks and adjusts it, and examines the results within a rendered frame. Feedback from a single rendered frame can provide enough information about whether the lighting design for a scene needs to be adjusted or even to be completely redesigned. It is much, much simpler to relight a CG scene than it is to relight a stage. Once a test frame has been rendered, the lighting designer will probably know right away whether or not adjustments are required. The lighting designer is likely to spend time working in the repeated loop of render-evaluate-adjust before moving on to the next scene.

1.5 Chapter Conclusion

In this chapter, we have reviewed the fundamentals of lighting design, and the impact of different lighting factors on the reproduction of recognizable qualities. Understanding the techniques and principles developed and used by graphic designers in setting up lighting for real environments, as well as for 3D scenes, is important to how we use those techniques and principles in the automation of the lighting design process. Understanding how artists use lighting tools should inform our creation of useful tools that will hopefully reduce the technical workload of lighting design and enable them to concentrate on their artistic goals.

Chapter 2: A Review of Lighting Design

2.1 Inverse Approaches to Lighting Design

Schoeneman *et al*. [2] address lighting design as an inverse problem. Here, users set up a set of desired lighting effects that are expected to appear in the final image; the system tries to find a solution in which the lighting effects are closest to the set of desired lighting effects. The lighting parameters optimized in this approach are light intensities and colours. It is claimed that the reason why light positions themselves are not optimized is that users often know where to put the lighting, but not how to combine them in terms of setting up intensities and colours for lights. Optimal lighting parameters are found at each step on the basis of minimizing an objective function, which is defined as an error function between a set of target functions and a set of approximations of functions.

Such functions can be ray traced images [3] of each scene from the same view point or radiance functions over surfaces in the scene computed by radiosity method [4]. Directly painting on the surfaces of the rendered scene causes a change in the surface radiance functions, and these after-painted surface radiance functions are used as target values in the optimization process that follows. Painted surfaces in the rendered image are given more weight, which biases the optimization towards solutions with properties that best match the painted surfaces. In this approach, painted surfaces can be considered as *examples* affecting the target radiance surface functions. Solutions are limited to direct illumination from lights in order to achieve appropriate interactive time with users. The system employed the method of finding quickly the patches being affected by the brush developed by [5] in order to achieve interactive speeds. This approach has a potential for a lighting design tool which assists users to set up intensities and colours of lights. One of the drawbacks of this “painting with light” approach is that there is always the possibility that a user will paint on surfaces that cannot be illuminated by the preconfigured lights. Although some constraint-based approach was employed, this still appears to be an incomplete solution. An example of the interactive interface proposed is shown in Figure 2.1.

Kawai *et al*. [6] proposed a goal-based approach to lighting design, which can also be considered an inverse approach. The lighting parameters that were optimized in this approach were light source emisivities, spotlight direction, spot light focus, element radiosity and element reflectivity; again the light positions were fixed in this approach. The objective function was based on an energy calculated using the radiosity method (and a set of components loosely motivated by research in visual perception). Perceptual components in the objective function were motivated by Flynn [7, 8] and which try to quantify parameters that elicit a shared human behavioural response and a subjective impression. In particular, the impact of non-uniform, peripheral and bright light, on impressions of clarity, spaciousness, relaxation and privacy, were examined. Once the objective function was defined, the system tried to minimize the objective function in an optimization phase. In this approach, the optimization problem was constrained by goals defined by users that were either physical constraints, design goals or barrier constraints. Physical constraints related light emission and element radiosities as determined by the physical nature of light transport. Design goals were subdivided into equality and inequality constraints on radiosity values. Barrier constraints were bound conditions on optimization variables that must be satisfied.

One of the drawbacks of this approach is that it is not suitable for novice users who would normally not be familiar with terms such as clarity, spaciousness, relaxation and privacy, which themselves should be defined by users via an interactive interface of the system. Another drawback of this approach is that the perceptual components in the objective function seem to be specifically defined for indoor room environments. Thus, this approach lacks generality that is necessary in reality. Figure 2.2 and Figure 2.3 show the interface proposed in the system and some optimized results.

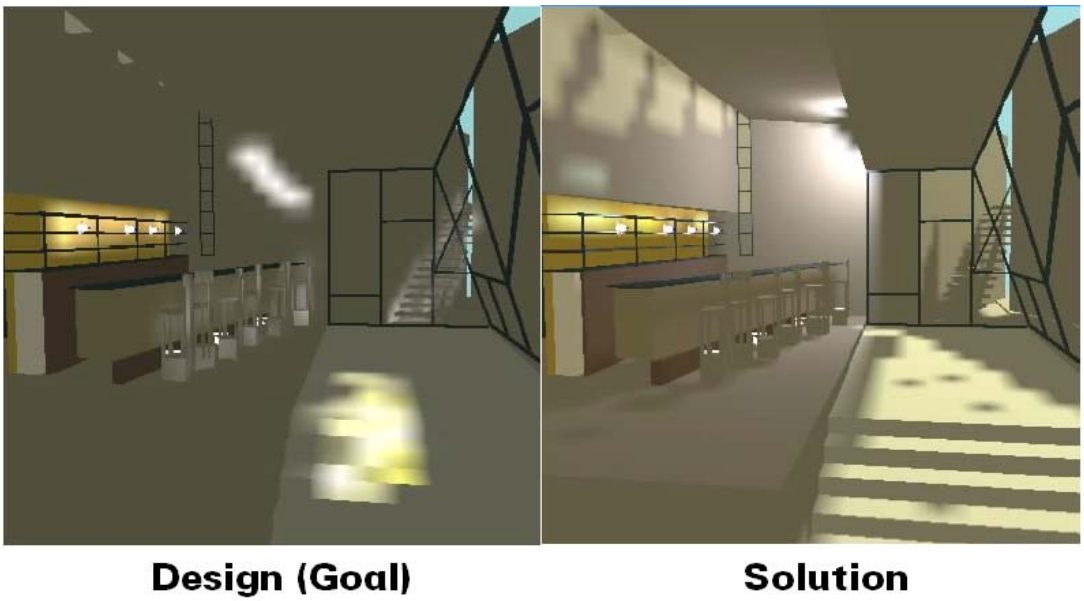


Figure 2.1: *An example of the approach proposed by Schoeneman et al [2].*

Figure 2.2: *Sample interface of the system proposed by Kawai et al[6] which allows users to set the weights of the objective components an specify the constraints.*

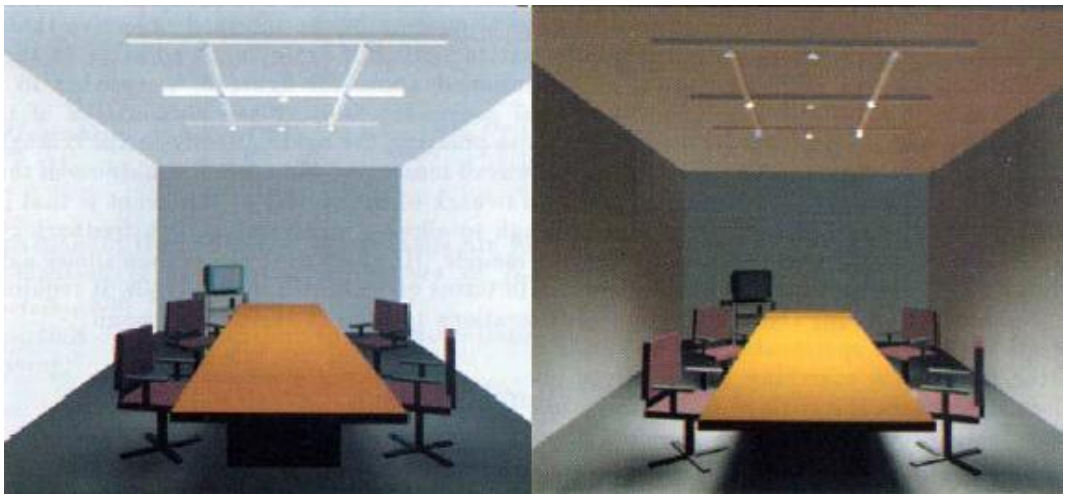


Figure 2.3: *Results of the approach proposed in [6] by Kawai et al. Left image: constrains the table to have small illumination while preserving energy and creating an overall impression of visual clarity. Right image: constrains the table to have small illumination while preserving energy and creating a feeling of privacy.*

Shadows tell us a great deal about the depth and shape of objects and their spatial relations to light sources. Object-light relations can be extracted from the shadows and highlights themselves. Poulin and Fournier [9] developed an inverse method for designing light positions through the specification of shadows and highlights in a 3D scene. In this approach, shadows and highlights were used in the specification of the shape and position of a light source. Highlights are strongly related to the specular element of a reflection model, and a user can define a highlight on the surface of an object by positioning two points that control the roughness coefficients of the surface. The first defines the maximum intensity point of the highlight, while the second determines the boundary of the highlight. When the specular term of Phong's shading model reaches a predefined threshold, the surface roughness coefficient is calculated. These two points are sufficient to determine both the direction of a directional light and the surface roughness coefficient. A boundary fill algorithm was developed to determine the shape of the highlight. Highlights in the scene must be recomputed if camera position is changed, as highlight information is strongly dependant on camera position.

This approach also developed an algorithm for determining light positions and direction from information extracted from shadows, in which a shadow was defined by

manipulating a curve using so-called pivot points. Light positions found by using shadow information are, of course, independent of view point. One of the drawbacks of using highlight information to determining light positions is that highlights just contain information about the direction of light sources. For light types other than directional light sources, more constraints are needed to properly interpret the highlight information.

In an extension to the approach proposed in [9], Poulin et al [10] developed an interactive sketch-based interface that enabled users to sketch desired shadow areas with a mouse pointer. The sketching process starts with selections of an object to cast a shadow and a light. A user sketches strokes on a 2D image plane projected directly from a 3D scene and the light position is recalculated at every stroke. An objective function was defined in such a way that the shadow region for a computed point light (and also some extended light geometries) bounds the sketched regions as tightly as possible. The objective function is based on distance between sketch points and the perspective position of a light, which is maximized in the optimization process (see Figure 2.4 and Figure 2.5 for examples of strokes and actual shadows).

Jolivet et al [11] presented an approach to optimizing light positions in direct lighting using Monte-Carlo ray casting techniques [12]. The core of the approach was to stochastically shoot rays at patches to be illuminated to find out patches that are directly visible. When a ray was shot, the first patch reached by this ray would be added to a candidate list as a candidate for the light position. The ray shooting process stopped when no patch was hit after a predefined number of rays were shot. The form factors were then calculated for every candidate patch in the candidate list on the basis of the ratio of the number of rays hitting the patch to total number of rays cast. The main limitation was the use of just a single quadrilateral emitting patch, built from a candidate list of patches using an algorithm developed in [13]. This approach also proposed a lighting design paradigm based on declarative modelling approach developed in [14] in order to help users specify the lighting goal in intuitive-using linguistic descriptions. The declarative model developed also used fuzzy logic [15] to allow the use of a set of modifiers to enrich the linguistic description of the lighting goal. Again, the main limitation is that other light properties, such as intensities, were not considered, and that it was not clear how to choose properly representative patches from surfaces to be lit.

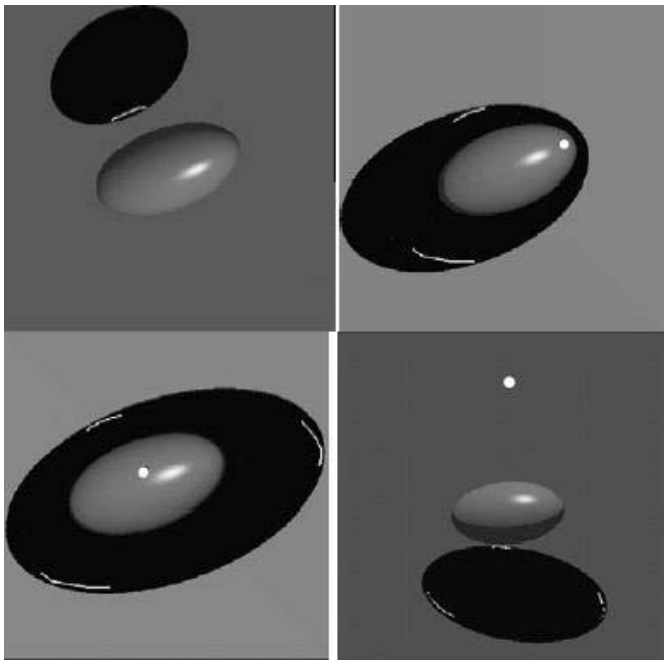


Figure 2.4: *An example of sketches that constrain the position of a point light source proposed by Poulin et al [9].*

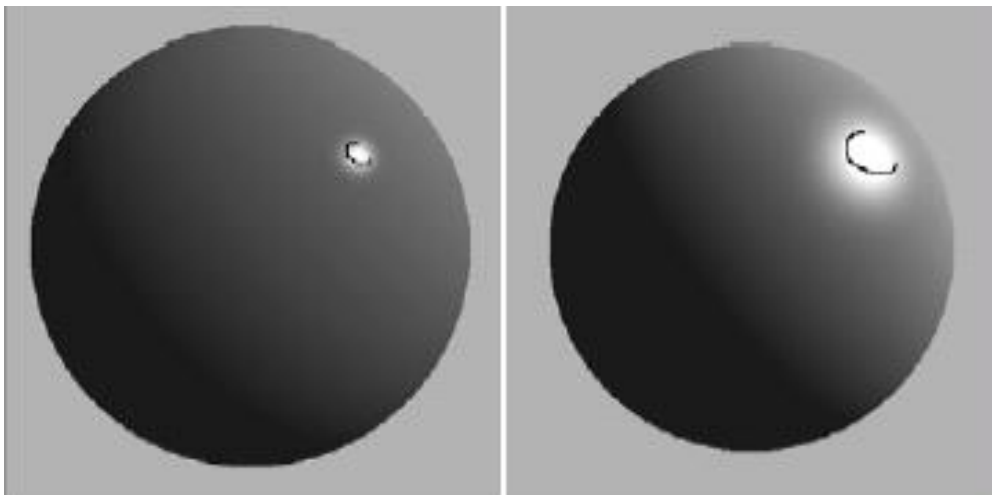


Figure 2.5: *Examples of sketches and their corresponding highlights proposed by Poulin et al [9].*

2.2 Semi-Interactive Approaches

In Design Galleries [16] Marks et al try to build a mapping function between an input vector that contains light position, light type, and light direction and an output vector that contains a set of values that summarizes the perceptual qualities of the final image. Possible light positions are located at so-called *light hook* surfaces defined by the user. During the optimization process lights are moved from one predefined position to another. At each position, a light type is selected from a set of light types, which describes attributes of a light such as its basic class (e.g. point light, area light, or spotlight), characteristics of a shadow cast by that light, its fall-off behaviour and class specific parameters. Corresponding to a light position and a chosen light type, an image is generated and an output vector created. An output vector is calculated on the basis of luminance of images at different low resolutions that are differently weighted. With a large number of light positions and light types, large sets of input and output vectors are created. A dispersion algorithm was developed in order to achieve a set of characteristic input vectors which creates a set of output vectors that covers the pre-dispersed set of output vectors. Dispersion results in a much smaller set of input vectors, which are a drastic reduction in the computational complexity. The dispersion step is based on distances between images, and a luminance-based distance metric was proposed for measuring the distance between two images (see Figure 2.7 and the Design Galleries interface). Design Galleries provide users with a large number of images to select from. Of course, the more images that are presented, the more difficult it is for users to select a particular image, so a graph-based partition scheme based on [17, 18] was applied to group images which have similar illumination effects. The edge costs of the graph used in the partitioning scheme are based on the inverse of the distance metric used in the dispersion step.

Design Galleries can be considered to be in the spirit of an examples-based approach, despite the fact that there is no specific example used as the basis of the objective function in the optimization process. Through its generation of a wide range of clustered images as examples, Design Galleries presents sets of exemplars for users to perform selection on as part of a render-selection loop. Thus, there is no information about what

effects users want to have in the final images, but the user has the opportunity to select likely candidates.

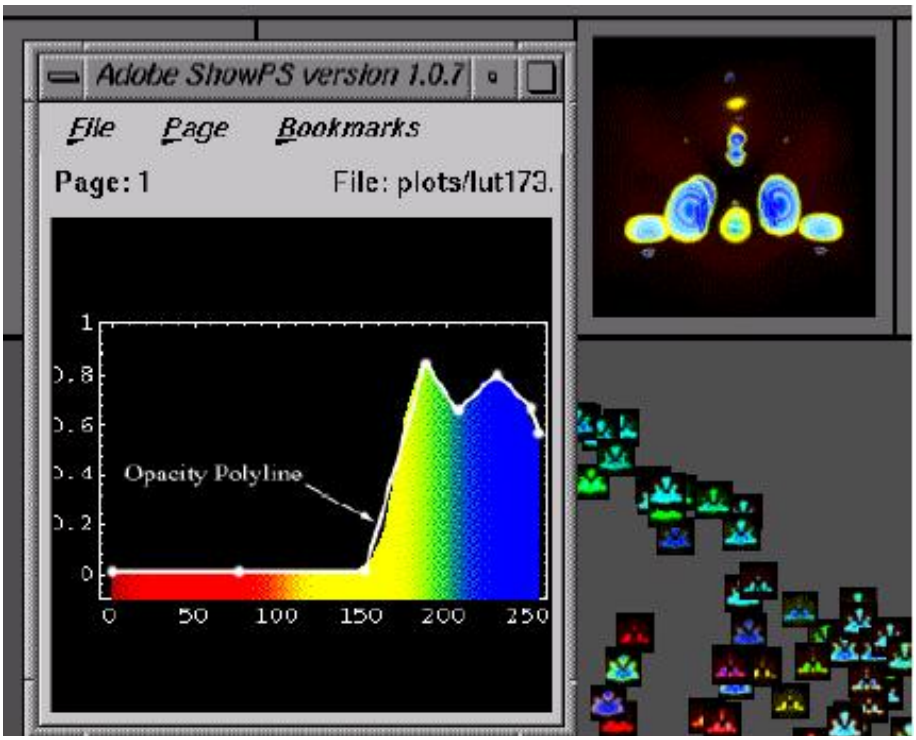


Figure 2.6: *Interface example of the Design Galleries system proposed by Marks et al in [16].*

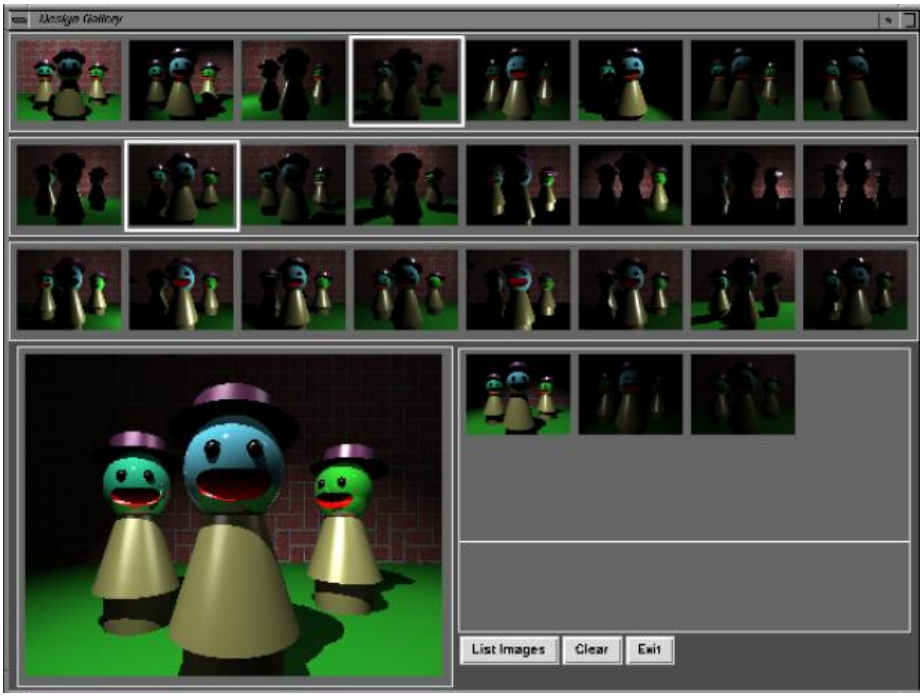


Figure 2.7: *Design Galleries system for light selection and placement proposed by Marks et al in [16].*

One of the limitations of Design Galleries is that the number of images presented to users is large, even with the application of the partition scheme. In addition, the distance metric used in dispersion and partition phases is simply constructed on the basis of the luminance distance between images, which is hard to justify in terms of our visual perception of image difference. Finally, the definition of light hook surfaces as well as light positions on those surfaces appears arbitrary and is not well motivated.

Non-photorealistic lighting is not constrained by physical correctness. Gooch et al [19] proposed a lighting model that uses luminance and changes in hue to convey surface orientation, edges and highlights. By recognising that, in technical illustrations, the communication of shapes and forms is more important than realism, they use shading to convey direction and the shape of the surfaces. A shading technique based on *cool-to-warm* tones was developed to shade surfaces with a limited dynamic range of colours that are visually distinct from colours of edge line. Anderson and Levoy [20] proposed an approach to enhancing visualization of cuneiform tablets using curvature and accessibility-based shading. In this approach, standard illumination was replaced by non-photorealistic rendering. A model of the tablets was first constructed using a high-resolution 3D range scanner. In the unwrapping stage, irregular meshes were partitioned

into rectangular patches with four connected curves. Each of the rectangular patches was then fitted to a grid of springs by iteratively relaxing and subdividing the spring grid in order to create flat rectangles. Finally, curvature and accessibility-based shadings were applied to the unwrapped text. Vicinity shading for volumetric data, which was claimed to enhance the perception of surfaces within the volume, as proposed by Steward [21], has extended the idea of accessibility shading by incorporating uniform diffuse illumination, which arrives equally from all directions at each surface point in the volume, and uses occlusion by local occluders. In simple terms, each surface point is shaded on the basis of uniform diffuse lighting that is blocked only in the vicinity of the surface point. Vicinity shading was claimed to provide better perceptual cues than a regular diffuse–plus–specular illumination model, such as Phong's model [22].

Reid and Pat [87] presented new techniques for interactive lighting design of complex scenes that use procedural shaders, in which geometric and optical information of the visible surfaces of an image is stored in a data structure called deep-frame buffers in order to enhance the performance of the system. The authors focused on the real-time aspects of the interactive lighting design process rather than the automation aspect.

2.3 Perception-Based Lighting Design

In recent years, a number of approaches to computer graphics have been proposed that are based upon explicit models of a viewer's perception of graphical renderings. Perceptually adaptive approaches have ranged across the entire scope of graphics algorithm and interaction development, from schemes for polygon simplification and global illumination that take account of limits on visual attention and acuity, to the design of anthropomorphic animations and gaze-contingent displays [23, 24].

Perception-based lighting design has included implicit approaches that aim to maximize illumination entropy for a fixed viewpoint. Gumhold [25] describes a perceptual illumination entropy approach in which he uses limited user studies to model user preferences in relation to brightness and curvature. The maximum entropy-based approach has been widely used in a variety of fields such as communication theory computer vision, and recently in computer graphics. This approach assumes that all light sources have white colour and that hue and saturation are not important in the lighting

design process – that is, brightness is the only component used for calculating illumination entropy.

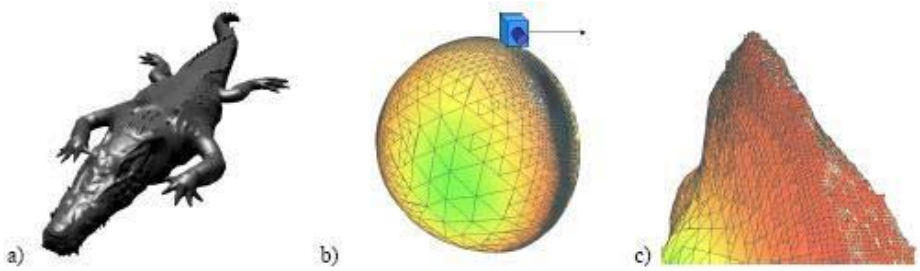


Figure 2.8: *An example scene and entropy illustrated in the approach proposed by Gumhold in [25] (a) an example scene; (b) entropy of a directional light; (c) view from the camera illustrated in (b) with upper 20% of entropy values to the power of 30.*

In this approach, colour values are first converted to brightness values ranging from 0 to 255. Brightness values are then mapped to 30 bins for the calculation of the probabilities that will be used for the illumination entropy computation. A fast lighting approach is also proposed based on the brightness dependent characteristic of the approach, while a hierarchical representation of different resolutions of the image is used for the calculations. The optimization phase, using local and global schemes, tries to maximize the illumination entropy. In a simple experiment, they asked users to set up a directional light source for a fixed view point such that they could best perceive the three dimensional shapes. On the basis of the experiment results, a perceptual illumination entropy computation was proposed. To avoid regions that are too dark or too bright, the bins of brightness values were changed by choosing larger intervals for the bins of low and high brightness than for those of intermediate brightness (using a non-linear function for mapping brightness values to the bins). Another perceptual feature, which was added to the entropy computation, was a measure of curvature importance. It was argued that regions with high curvature carry much more information than those with low curvature. With this assumption, each pixel was given an importance weight. An importance weight of a pixel was estimated on the basis of gradient calculation of the normal at that pixel from normals of adjacent pixels. Although the entropy-based approach is strongly motivated by information theory, it is less well grounded in theories of visual perception.

Indeed, there is no proven relation between information calculated using information theory and the 3D information that the human visual system perceives and process. Many features crucial to the process of visual perception are not considered, such as shading gradient information, the average brightness of the image, and the brightness adaptation mechanism of the human visual system.

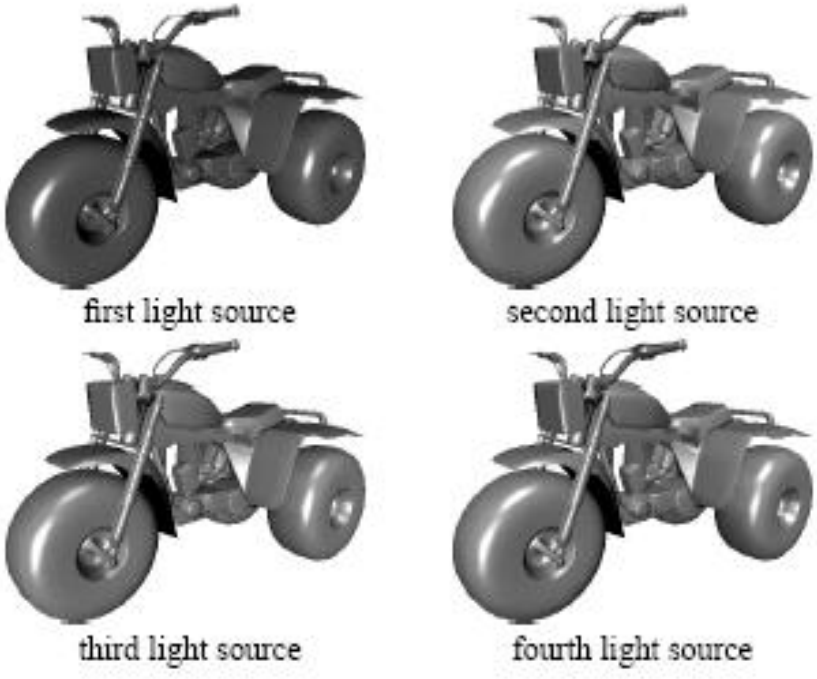


Figure 2.9: *Some results illustrate the effects of incremental adding up to 4 white light sources in approach proposed by Gumhold in [25] .*

Lee et al [26] proposed a more explicit model of perceptual preferences in Light Collages, in which lights were optimized such that the diffuse illumination is proportional to the local curvature and specular highlights were used only for regions of particularly high curvature. The key idea was to map a set of lights to a set of surface patches. Therefore, each light would light a subset of surface patches. In this approach, the three dimensional mesh was segmented into patches on the basis of surface curvature. Segmentation was performed by first calculating the average curvatures at each vertex of the mesh [27] and then applying the *watershed* method [28]. Sharp visual discontinuities across the patch boundaries were addressed by blending illumination from adjacent patches. Blending weights for a vertex were calculated on the basis of the

distance from that vertex to the boundaries of the patches in which the vertex does not reside. These weights were then used to weight the illumination contribution of light sources to that vertex. Only white directional lights were used in this approach and the directions were computed by optimizing a so-called *light placement function*. The objective function was based on two sub-functions, the specular weight function and the diffuse weight function, which formulated on the basis of the mean curvature at the vertices, the view direction, and the light directions. Lights were sequentially added to the scene in the direction that maximizes the objective function. In this approach, Lee et al also explored the use of practical lighting techniques, including the use of backlighting technique to enhance the silhouette visibility of the objects in the scene. One of the drawbacks of *Light Collages* is that the intensities of lights were not included in the optimization framework, and scenes of only one object could be lit.

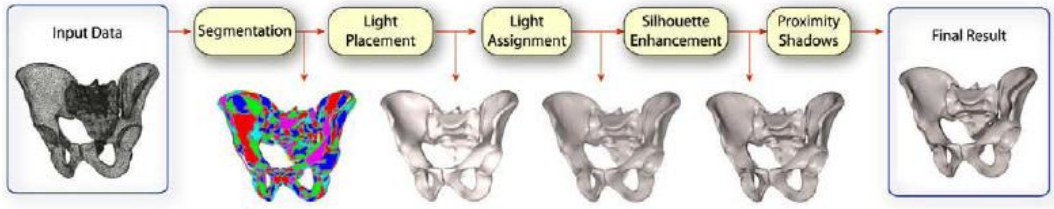


Figure 2.10: *Overview of lighting pipeline proposed by Lee et al in [26].*

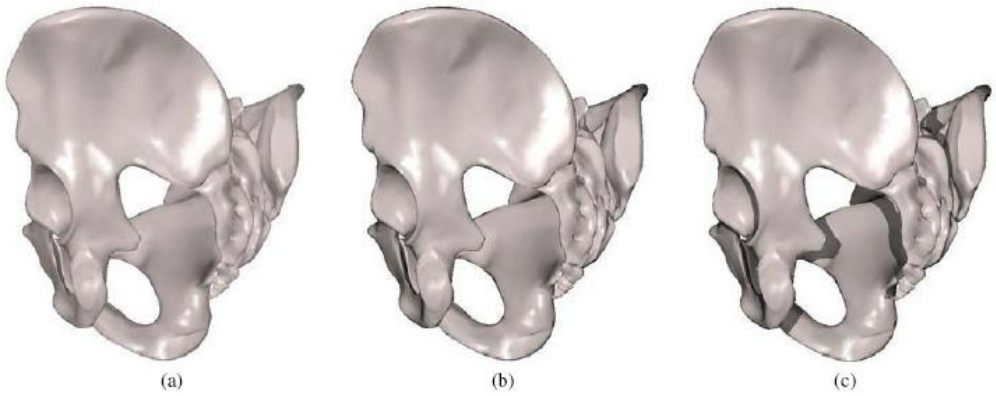


Figure 2.11: *Results from Light Collages proposed by Lee et al in [26] show different effects: (a) Lighting Pelvis with 8 lights; (b) Adding silhouette lighting to the previous scene; (c) Adding proximity shadows to the previous scene.*

Much research has been carried out aimed at narrowing the gap between the perception between real images and computer-generated images. Realism of a computer-generated image is not only a matter of physical correctness, but also of perceptual equivalence of the image to the scene that it is intended to represent. Achieving perceptual equivalence between a computer-generated image and the scene is significantly challenging, since many perception-based problems must be taken into consideration. In particular, how to quantify the perceptual quality of an image? Much research has been carried out to find efficient perception-based image quality metrics. Notably, Shacked & Lischinsky (S&L) [29] proposed an approach to solve lighting design problems using a perceptual quality metric.

At the heart of S&L's research is an objective function that is motivated by the human perception of 3D scenes. The first component taken into consideration in their approach is the magnitude of shading gradient of the rendered image. Psychological research [30,31,32] shows that variations in shading provide important information about shape, depth and the impression of realism. Figure 2.12 shows the effects of shading gradient on different human perceptions of depth. The images on the left and right provide little information about the changes in depth, and the surface of the sphere appears flat (there is a low shading gradient) whereas the middle image has a much higher shading gradient value. S&L use a target value to capture the maximum average shading gradient value that could be potentially measured in an image of that particular 3D scene (with the specific viewing parameters) and seek to minimise the difference between the actual and target gradients.

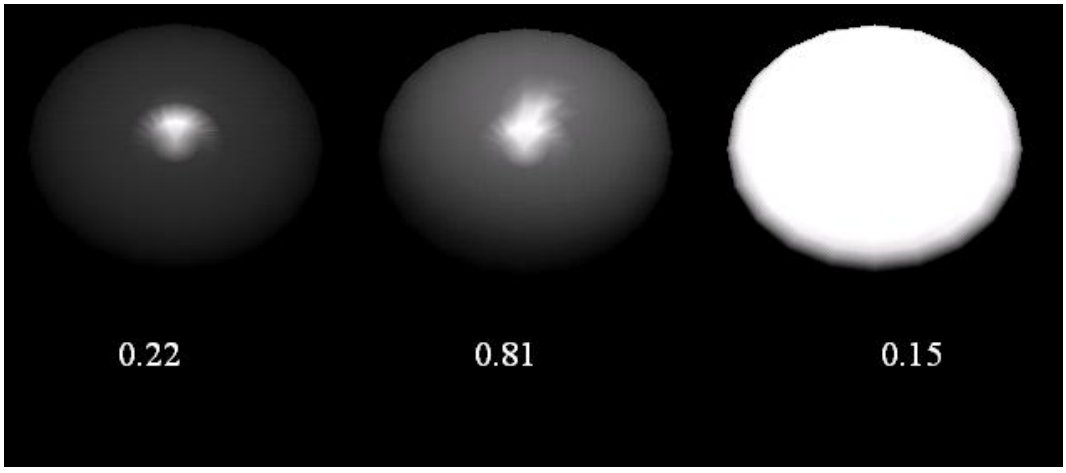


Figure 2.12: *Different gradients give different perception of depth of a sphere. The bottom line numbers are gradient values calculated for a corresponding scene.*

The second component of S&L's objective function is the measure of edge contrast. It is well established that edges convey information about shape [32, 33, 34, 35], and they should be prominently displayed by the image. In the objective function the prominence of edges is determined by the ratio between the number of pixels detected by a pixel-based edge detection operator (applied to the rendered image) and the maximum number of edge pixels that could appear. Visible edges must therefore be determined in order to calculate this edge term.

Figure 2.13 shows the effects of edge visibility component on our perception of a sphere. The image on the right shows the actual front-viewed edge of the sphere. Looking at the middle image, we can hardly recognize the shape of the sphere. On the contrary, we can easily recognize the shape of the sphere on the left image; corresponding to this, the edge visibility value of the left image (0.92) is much bigger than that of the middle image (0.48).

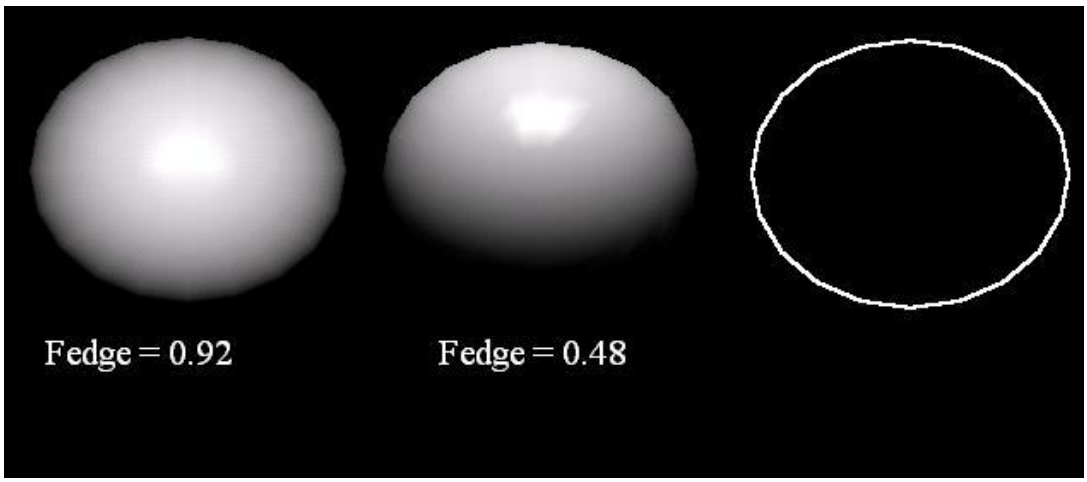


Figure 2.13: *Different edge visibilities give different human perception of shape of a sphere. The bottom line numbers are edge visibility values calculated for corresponding scenes. Right image shows the actual front-viewed edge of the sphere.*

The third component used in the objective function is the variance of the luminance pixels in an image. The variance of the pixel luminance is useful because of its

importance in our brightness adaptation mechanism. In any situation, the human visual system (HVS) adapts to a particular light intensity, which is called *the brightness adaptation level*, and the HVS is most sensitive to intensities around this level. Research on adaptation levels by Tumblin and Rushmeier [36] and Stevens [37] theorized that the adaptation level can be estimated by the log of the luminance visible on the retina. The HVS is not sensitive to intensities at values far below this level. This means that we should not create an image with a large range of intensities, and the variance of the luminance of pixels should be minimized. Whilst this seems to conflict with our goal of maximising the shading gradient component, we can moderate the tension between the average shading gradient component and the variance of luminance component by the selection of a target value.

The mean of all pixel luminance in an image determines the overall brightness of an image, and our perception of images is negatively affected if the brightness is too high or too low, since overly dark or light images tend to weaken the effects of shading and may obscure features in the scene. To address this problem, S&L used an empirically established target brightness. In addition, the histogram of the objected function prevents the production of large areas with uniform shading. Related to the mean luminance is the luminance histogram of an image, which is the distribution of pixels over different grey-scale levels. S&L use an equalized histogram as a target distribution and seek to minimise the difference between the actual histogram and this target equalized histogram (a technique commonly exploited in image processing). Figure 2.14 shows an example of applying histogram equalization to an image. A final component added by S&L to the objective function is the direction of a light source. This component is motivated by psychophysical studies, indicating that it is easier for the HVS to interpret 3D shape when illuminated from above. S&L constrain the direction of light sources to come from above by using a target direction and penalising angular deviations from this.

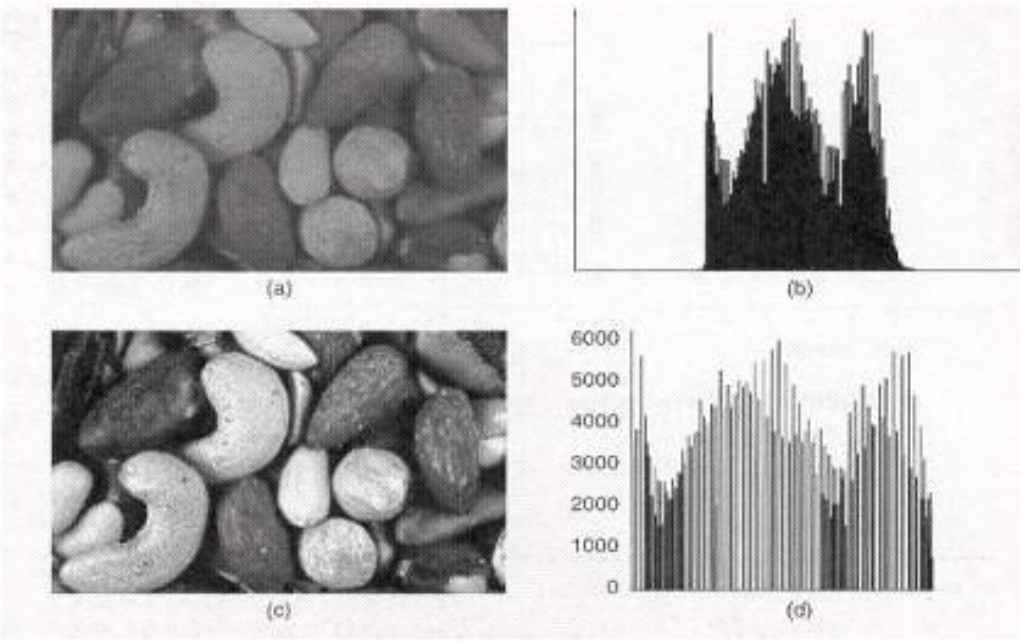


Figure 2.14: *An example of histogram equalization: (a) Original image; (b) The histogram of the original image; (c) Histogram equalized image; (d) The histogram of an equalized image.*

Figure 2.15 shows an example illustrating the sequential effect of modifying lighting parameters for two light sources proposed in [29]. In (a) a random set of lights has been initialized, in (b) the lights have been positioned and their intensities set so as to emphasize the appearance of edges. In (c) the lighting has been adjusted so as to restrict the variance in the luminance of each pixel of the ship, and in (d) an overall value of the mean luminance takes a target value. Finally, in (e) the distribution of intensity has been specified and in (f) the angle of the main light source is restricted to being in front and above.

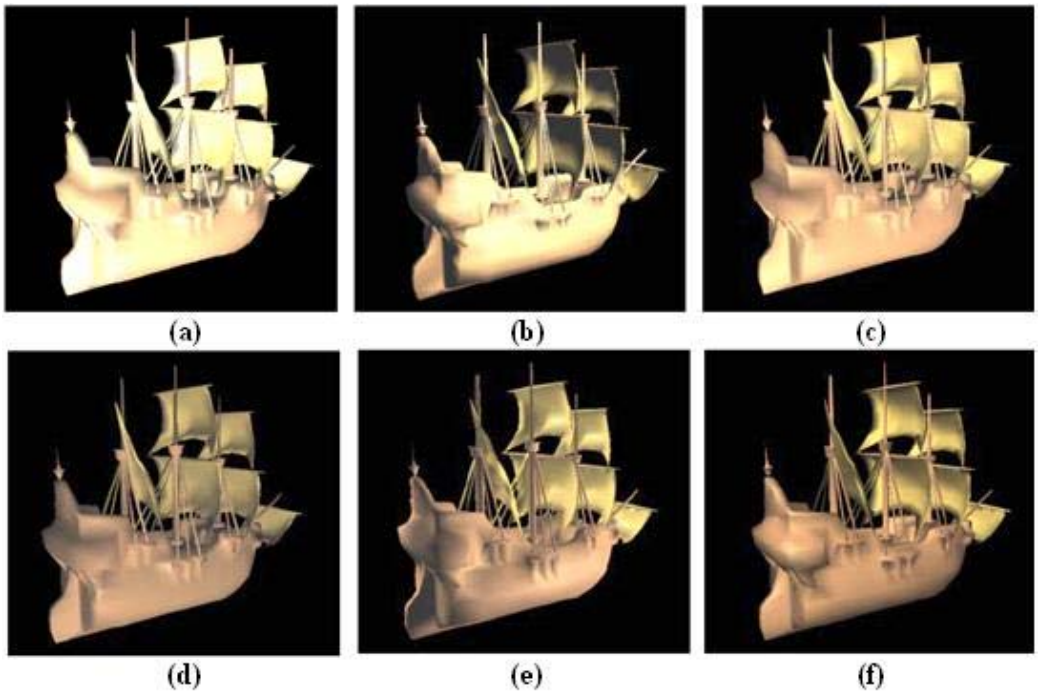


Figure 2.15: *Illustration of the sequential effect of modifying lighting parameters for two light sources proposed by S&L in [29].*

S&L's method is based on local illumination information and pixel luminance statistics, although the main shortcoming is that users have to set up a set of target values for the specific viewing parameters. Important characteristics such as the contrast between objects and curvature of the surfaces are not adequately addressed. The perceptually non-uniform characteristics of different colour systems (on which the metrics were based) are also not addressed.

2.4 Chapter Conclusion

In this chapter, different approaches to automatic lighting design in computer graphics have been reviewed. Various topics that are likely to have a bearing on automatic lighting design have been considered, including image processing techniques, entropy theory and aspects of visual perception. Inevitably, each approach has its advantages as well as its drawbacks, as is typical when different approaches are applicable in different

circumstances. What is apparent is that there is no existing approach which can solve the lighting design problem in its entirety.

Research proposed by Schoeneman et al, which enabled users to directly paint desired lighting effects on the surfaces of 3D geometries, has drawbacks in that a user can paint on surfaces that cannot be illuminated from any light. Although some constraint-based techniques were employed, this is an incomplete solution. Furthermore, the light positions need to be set manually by users, when in fact light placement is one of the most challenging tasks for graphic designers. Work by Kawai et al was intended to equip users with the capability to set lighting goals in some detail; however, the complexity of their approach makes it unsuitable for non-expert users who would not normally be familiar with terms such as clarity, spaciousness, relaxation and privacy, which need to be defined by users using the system. The perceptual components in the objective function proposed in this approach also appear to be specifically defined for indoor environments. Further shortcomings include the local nature of the optimization technique and the fact that the light positions were required to be fixed and are not optimized by the system.

Work proposed by Poulin and Fournier was expected to enhance the performance of the lighting design process by determining the light position from highlights and shadows. The use of highlight information to determine light positions results in the fact that this method was not suitable for various light types, as highlights just contain information about directions. For other light types, other than directional light sources, more constraints were needed to properly interpret the highlight information. Another limitation of this approach was that light positions derived from highlights must be recomputed for different viewpoints, but this is an expensive process that should be avoided as much as possible. With this approach, many attempts must be made to achieve the desired shadow, as it could be an obstacle to transforming a shadow by a series of pivot operations. Apart from light position and direction, no other parameters of lights were to be optimized.

Design Galleries was intended to provide users with an interface to browse the clustered images which were automatically generated by moving lights around and changing light types. One limitation of this approach is that the number of images

presented to users is large, even with the application of the partition scheme approach. Another issue is that the distance metric used in the dispersion and partitioning phases is simply constructed on the basis of the luminance distance between images, which can not be justified in terms of visual perception. Similarly, the first entropy-based approach to lighting design proposed by Gumhold claimed to be motivated by information theory, but many aspects of visual perception were not addressed explicitly. In particular, no account was taken of shading gradient information, the average brightness of the image and the brightness adaptation mechanism of the human visual system.

Light Collage was intended to assign lights to patches in the scene by segmenting the geometries on the basis of curvatures of surfaces. One of the drawbacks of this approach is that intensities of lights were not optimized. In addition, this approach was implemented for scenes with one object. For the scene with multiple objects, the spatial importance should be carefully taken into consideration in order to maximize features at important regions.

Finally, Shacked and Lischinsky proposed an approach to lighting design using perceptual metrics. Their approach is extensive (when compared to earlier work) but there remain a number of aspects related to our perception that it fails to take account of, including contrast between objects, curvature of surfaces, and the perceptual characteristics of different colour systems. Moreover, the optimization scheme employed was basic and unlikely to discover global optima. From our review of the state-of-the-art automatic lighting design we can identify a number of open issues, which we address in subsequent chapters:

- the fully automated determination of light position and intensities;
- further exploitation of perceptual properties, such as contrast between objects which convey depth information, perceptually uniform colour systems, and the relative importance of edges of objects in and outside of shadows;
- more powerful optimization frameworks;
- the application of practical lighting techniques which have been found useful in real lighting design problems in photography and TV studios;
- the methods by which users can state their desired lighting goals.

Chapter 3: Extending Perception-Based Approach

Shacked and Lischinsky's [29] approach to lighting design has been the most successful application of perception research achievements to solving the automatic lighting design problem. However, there are still drawbacks to their approach as already described in the previous chapter. In this chapter, we present a number of extensions that aim to enhance the perceptual quality of the final images.

3.1 Underlying Perception-Based Lighting Design Framework

At the heart of this approach is an objective function that is the linear combination of five distinct measures of image quality: edge distinctness (F_{edge}); mean brightness (F_{mean}); mean shading gradient (F_{grad}); intensity range (F_{var}); and evenness of the distribution of intensity in the image (F_{hist}). We omit the sixth component of the image quality function used by Shacked and Lischinski [29], which biases the optimization of a key light to a particular elevation and orientation above and in front of the object (relative to the viewpoint). This is standard practice in photography and might be explained in terms of evolutionary psychology, in that our perceptual system evolved for scenes lit by the sun or moon [31,38]. However, the implementation of the sixth component of Shacked and Lischinski's approach, in which the direction of the light is used as an image quality function, is unjustifiably ad hoc. Constraining the light position to a quarter-sphere in front of, and above, the centre of the scene is sufficient. The Shacked and Lischinski framework is also constructed such that the lower values of $F(\theta_k,\varphi_k,I^d_k,I^s_k,R_k)$ correspond to lighting configurations with better visual characteristics and a greedy gradient descent minimization algorithm is utilized in the discovery of appropriate lighting configurations. Components in the objective function are differently weighted and standardized in order to avoid the problem in which certain components dominate the objective function. The weights given to components are set experimentally.

The objective function is formulated in Equation (3.1).

$$F\left(\theta_k, \varphi_k, I_k^d, I_k^s, R_k\right) = w_e F_{edge} + w_m F_{mean} + w_g F_{grad} + w_v F_{\text{var}} + w_h F_{hist} \quad (3.1)$$

θ_k is the elevation angle of k^{th} light.

φ is the azimuth angle of k^{th} light.

I_{dk} is the diffuse intensity of k^{th} light.

I_{sk} is the specular intensity of k^{th} light.

R_k is the distance of k^{th} light (fixed for directional lights).

$k = 1, 2, \ldots, K$.

K is the number of lights.

w_e, w_m, w_g, w_v, w_h are weights for different components in the objective function.

3.2 Extending Perception-Based Lighting Design

Components of the perception-based lighting approach are motivated directly from the results of studies of human perception, in particular, object recognition [31]. For example, Shacked and Lischinski's edge enhancement criteria relates directly to theories of object segmentation and neuropsychological findings as to the nature of retinal processes [31]. However, a number of features can be identified that are currently not addressed in perception-based lighting design:

(a) *contrast:* between different surfaces of an object it has been shown to convey significant information about shape and depth of objects [31].

(b) *back-lighting*: is a well-established feature of cinematic and photographic practice aimed at maximizing edge enhancement [1, 26, 38].

(c) *perceptually uniform colour spaces*: standard approaches in lighting design implement metrics over standard RGB (or equivalent) colour spaces, despite the fact that such spaces are highly non-uniform with respect to human judgments of colour [31].

(d) *optimization schemes*: global optimization schemes are likely to lead to better quality solutions. S&L's original proposal used a simple gradient descent-based method.

(e) *Shadow processing*: the visual perception of edges in shadows is not addressed by S&L's scheme and should be thoroughly investigated and developed to yield an accurate objective function.

3.2.1 Contrast enhancement

Contrast enhancement has long been applied to 2D image processing in order to achieve preferable visual cues to human visual system. Indeed, contrast enhancement tools can be found in almost all 2D image processing packages. Psychological research [31] shows that the human visual system (HVS) is more sensitive to small scale contrast than large scale contrast. This can be explained by the brightness adaptation mechanism [36, 37]. Large scale contrast is defined as the difference in intensity between significantly large light and dark areas in the image. Small scale (or local contrast) is the difference in intensity between much smaller adjacent areas in an image. While scanning an image, large scale contrast fails to create a significant impression on the HVS as it quickly adapts to the brightness of the local region in the image. As a result, in 2D image processing much more emphasis is placed on local contrast enhancement. In 2D image contrast enhancement, the contrast of a region is normally calculated on the basis of the contrast at every pixel in the image region, and contrast at a pixel is calculated according to the value of adjacent pixels [39, 40].

In the perception of 3D objects, empirical studies of visual cognition have demonstrated that object perception depends on both the absolute amount of luminance and the difference between the object's luminance and that of its background [41, 42, 43]. The difference in luminance between surfaces of objects has been shown to convey much information about shape and depth of objects. Contrast has been shown to be an important cue to depth [42, 43]. Furthermore, as light scatters off atmospheric particles, contrast is attenuated with distance [42]. Rohaly and Wilson have conducted a range of depth-matching experiments in order to quantify the contrast dependence of perceived depth, and found that perceived depth was a power law function of contrast [43]. This is illustrated in Figure 3.1 - the image on the left gives us no impression about depth and number of objects in the image, whereas the image on the right gives us a clear sense about relative position between surfaces and depth of the object. This difference in perception about the same 3D scene results from the difference in luminance or contrast

between small and big rectangles in the scene. We extend this notion through the provision of a means of evaluating in luminance between adjacent parts of an object and incorporating this in our objective function.

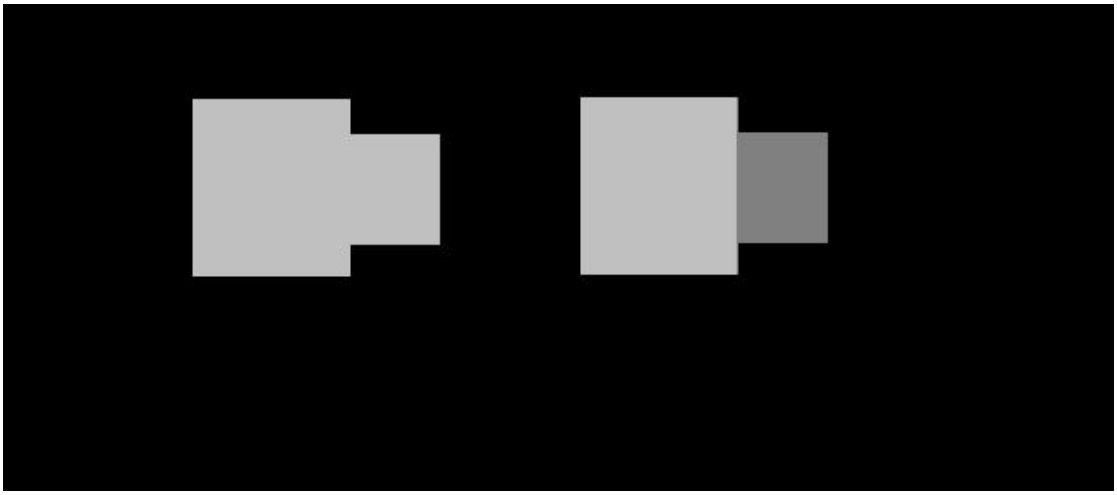


Figure 3.1: *Illustration of the contrast effect on depth perception [rendered using LightOP].*

The formulation of contrast is described in the following equations. The contrast between two parts of an object is given by:

$$C_{ij} = \frac{(Y_i - Y_j)}{Y_j} \tag{3.2}$$

Where:

C_{ij} is the contrast between part i and part j.

Y_i is the mean luminance of part i.

The mean luminance of a part i is calculated as in equation (3.3).

$$Y_i = \frac{1}{N_i} \sum_{I_{x,y} \in P_i} I_{x,y} \tag{3.3}$$

Where:

P_i is the part i of an object.

N_i is the number of pixels of part.

$I_{x,y}$ is the intensity of the pixel at row x and column y.

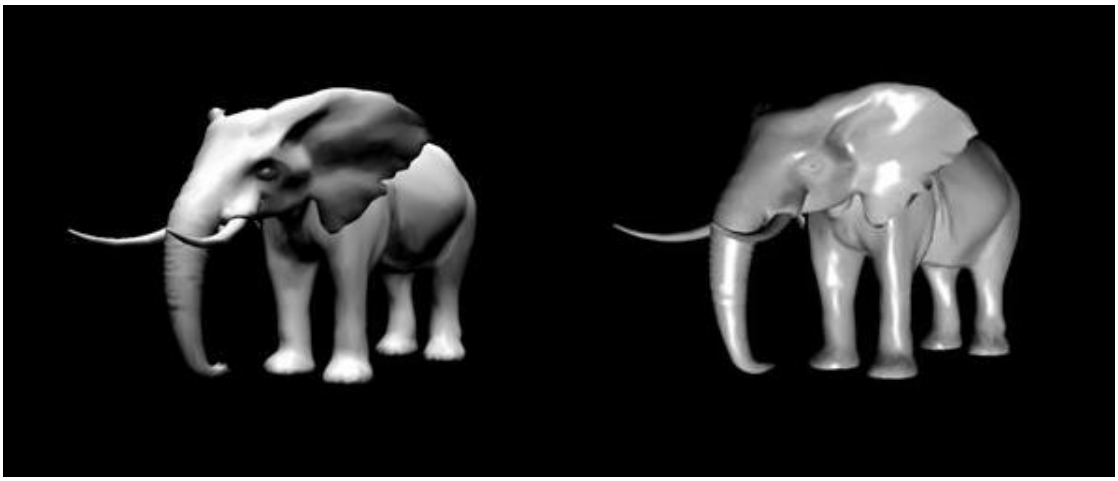
A pixel type map of objects in a 3D scene is extracted by applying an edge detector operator to the depth buffer [44]. Edges in the pixel type map correspond to boundaries between parts of an object. With this assumption, we developed an algorithm to calculate the contrast between adjacent parts of an object using pixel type map. The algorithm can be described in pseudo-code, as in Figure 3.2.

In the pseudo-code, the term of RUN-LENGTH is defined as follows: $p(x_i,y_i)$ is a RUN-LENGTH from $p(x_j,y_j)$ if $p(x_i,y_i)$ is the same pixel type as $p(x_j,y_j)$ AND all the pixels, which belong to a horizontally continuous connection between $p(x_i,y_i)$ and $p(x_j,y_j)$, are the same pixel type as $p(x_j,y_j)$.

```
Procedure ContrastCalculation
Begin
 For each pixel p(x,y) in the pixel type map
  If (p(x,y) is an EDGE pixel)
   Begin
      x_r = x
      Repeat
           x_r = xr +1
           Calculate mean_right_intensity
      Until     (p(x_r,y) is BACKGROUND pixel) OR
           ((p(x_r,y) is EDGE pixel)) AND
       (p(x_r,y)is not a RUN-LENGTH from p(x,y))
      x_l = x
      Repeat
           x_l = x_l -1
           Calculate mean_left_intensity
      Until     (p(x_l,y) is BACKGROUND pixel) OR
           ((p(x_l,y) is an EDGE pixel)) AND
      (p(x_l,y) is not a RUN-LENGTH from p(x,y))
       Calculate mean_contrast at this point from
      mean_right_intensity and mean_left_intensity
      Add mean_contrast to mean_global_contrast
 End;
End
```

Figure 3.2: *Algorithm for calculating the contrast between adjacent parts of an object.*

Figure 3.3(a) is optimized without contrast component in the objective function and Figure 3.3(b) is optimized with contrast component included in the objective function. At the risk of over-generalizing, the contrast component leads to better cues as to the depth and the relative positions of parts of an object.



(a) Greedy optimization of the Shacked & Lischinski [29] base function *(b) Addition of the contrast component*

Figure 3.3: *Effect of the contrast component.*

3.2.2 Edge enhancement and backlighting technique

In practical contexts, such as a photographer's studio or a film set, backlighting is a useful technique that often makes a valuable contribution to more pictorial lighting [38, 45]. Backlighting enhances the visibility of external edges and facilitates a viewer in segmenting objects from the background. This is especially true where the surfaces of the object are dark. Likewise, backlighting catches the sharply-folded contours and gives them shape and solidity.

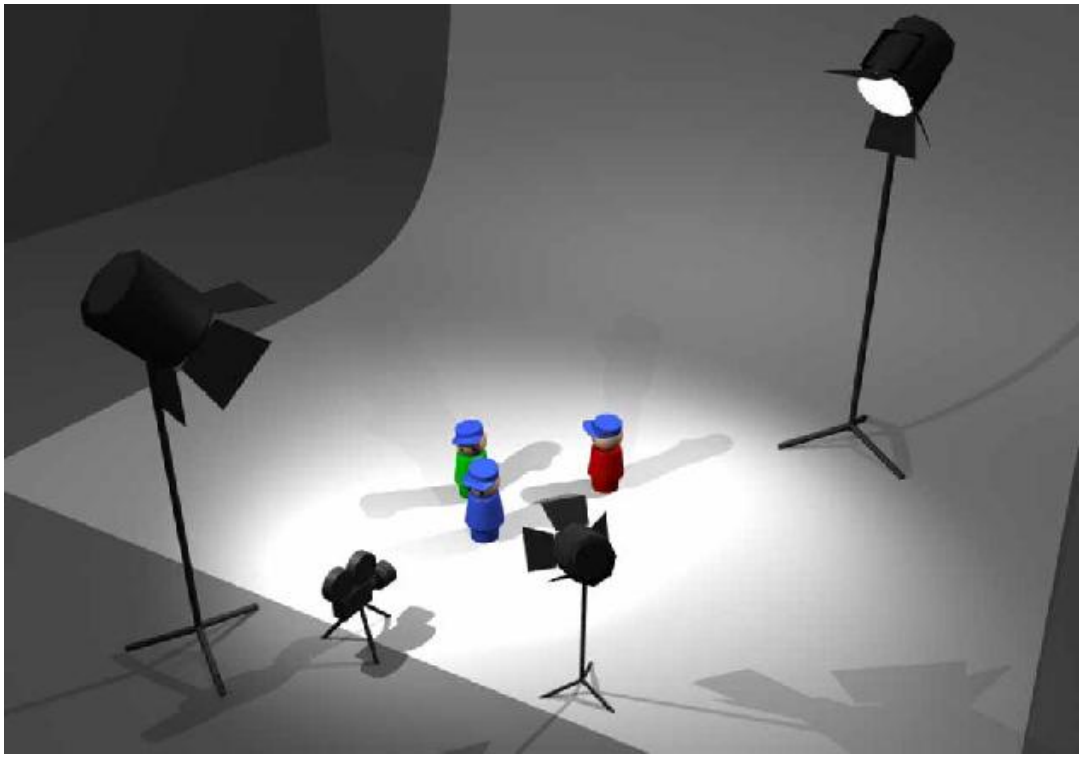


Figure 3.4: *The standard 3 light studio setup: key light, fill light, and backlight* [86].

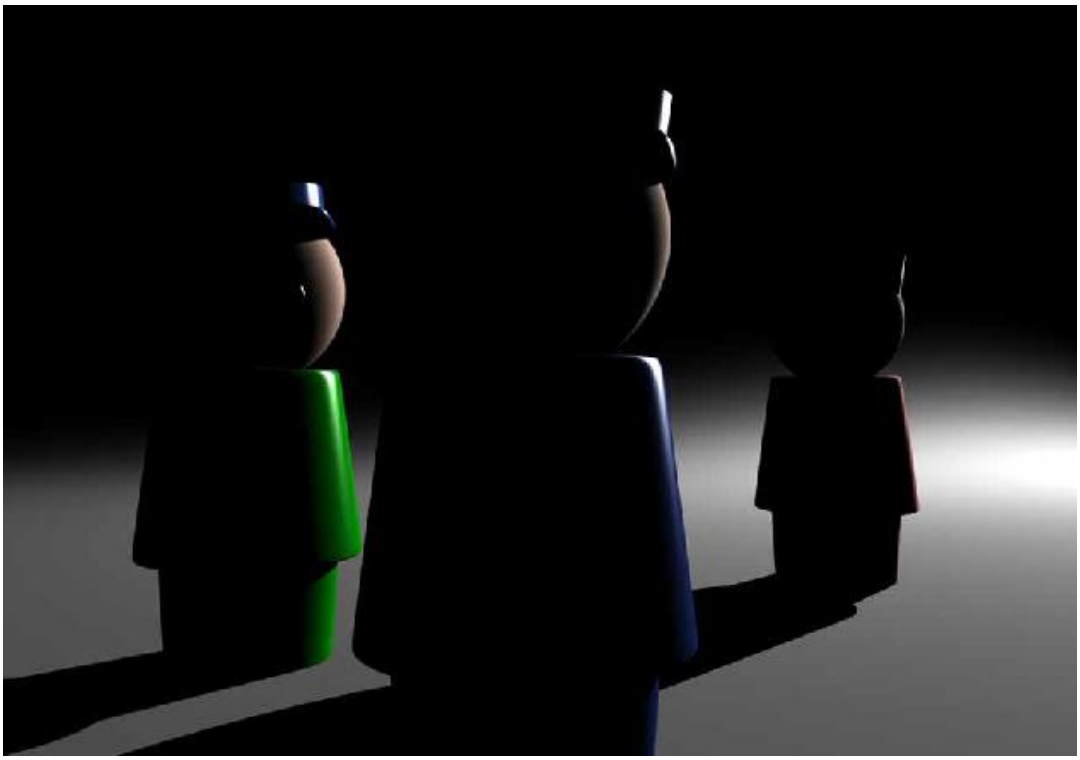


Figure 3.5: *A backlight is normally used to separate the subjects from the background* [86].

The initial position of the backlight is calculated as a centroid of a set of vertices of objects in a 3D scene. During the optimization process, the azimuth angle of the backlight is fixed. Since backlighting primarily affects external edges, we only use the edge component in the objective function when optimizing parameters of the backlight. In reality, the backlight is always positioned so as to light an object from above and behind. Thus the elevation angle of the backlight is confined to a predefined range, and this is a constraint during the optimization process. Back light optimization can be viewed as a distinct process from establishing the position and intensity of the key light. Thus, we implement it as a second optimization stage using a greedy algorithm.

The impact of adding backlighting on the contrast enhanced objective function can be seen in Figure 3.6(a) and Figure 3.6(b). Figure 3.6(a) shows the result due to the back light alone, and Figure 3.6(b) shows the image resulting from the addition of the backlight in which the geometry of the feet is more clearly discernable, as are the top of the head and the top of the right ear.

(a) Contribution of the back light *(b) Result of adding of the backlight*

Figure 3.6: *The effect of backlighting.*

3.2.3 Perceptually uniform colour space

Despite the fact that perception-based graphics algorithms attempt to model and quantify human responses to visual stimuli, they routinely use standard RGB (or equivalent colour spaces) and ignore the fact that such colour spaces are highly non-linear with respect to our judgments of colour difference. The problems with RGB colour spaces are that there is only a small range of potential perceivable colours, it is not easy for humans to say how much RGB to use to make a given colour, two points at a certain distance apart in a colour spectrum of the space may be perceptually different, and two other points at the same distance apart in another colour spectrum of the space may be perceptually the same.

A perceptually uniform colour space is one of which a change of the same amount in a colour value, anywhere in that space, should produce a change of about the same visual importance. When colours are stored as limited precision values, which is the case in computer graphics, perceptually uniform colour spaces also can improve the reproduction of tones. The most commonly used perceptually uniform systems are CIE (L*, u*, v*) and CIE (L*, a*, b*), which were standardized by Commission Internationale de l'Eclairage (CIE). These two colour spaces were derived from the space CIE 1931 XYZ colour space, which can map the colours that are perceptually the same to a spectral power distribution. These colour spaces were intended to be inferred directly from the XYZ space, but are actually more perceptually uniform than XYZ space.

In a work aimed to create a metric, which consisted of sharpness and colourfulness for improvements to visual fidelity and perceived quality, Winkler [46] used the sum of the mean distance of pixel chroma to a neutral grey and the standard deviation of chroma in the image for colourfulness. He found out that chroma computed in the (L*, u*, v*) colour space gave much better results than that computed in perceptually non-uniform colour space. Perceptually uniform colour spaces have been widely used in the area of content-based image and video retrieval, and are proven to give significantly better results compared to perceptually non-uniform colour spaces [47, 48]. A number of attempts have been made to combine Colour Image Fidelity (CSF) modelling with

CIE (L*, a*, b*) in order to combine models of spatial frequency response and colour space non-uniformities [49,50] .

In this approach, RGB colour values are transformed to the CIE (L*, a*, b*) colour spaces – which is approximately uniform [51]. As described in equation (3.1) and equation (3.2), RGB values are firstly converted to abstract primaries labelled as XYZ. In computing components of the objective function in a perceptually uniform colour space the goal is to obtain edge contrasts, gradients and variances that are maximal in their actual visual distinctiveness. Even for target-based components such as F_{mean} the error function in the optimization will be enhanced due to a linear relationship between computed (from the colour space) and perceived differences.

$$\begin{cases} X = 2.36R - 0.515G + 0.005B \\ Y = -0.89R + 1.426G + 0.014B \\ Z = -0.46R + 0.088G + 1.009B \end{cases} \tag{3.1}$$

$$\begin{cases} L^* = 25\left(\dfrac{100Y}{Y_w}\right)^{\frac{1}{3}} - 16 \\ a^* = 500\left(\left(\dfrac{X}{X_w}\right)^{\frac{1}{3}} - \left(\dfrac{Y}{Y_w}\right)^{\frac{1}{3}}\right) \\ b^* = 200\left(\left(\dfrac{Y}{Y_w}\right)^{\frac{1}{3}} - \left(\dfrac{Z}{Z_w}\right)^{\frac{1}{3}}\right) \end{cases} \tag{3.2}$$

X_w, Y_w, Z_w are values of X, Y, Z that are calculated at white colour.

Figure 3.7 illustrates a result of incorporating the perceptually uniform colour space.

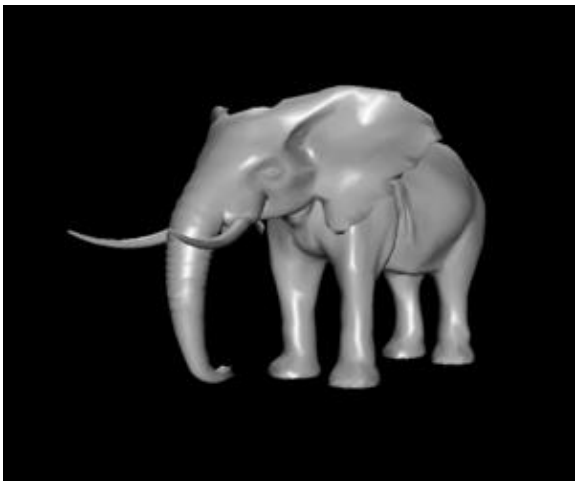


Figure 3.7: *Using a perceptually uniform colour space.*

3.2.4 Optimization of lighting configurations

Shacked & Lischinski make no attempt to characterize the nature of their objective function or justify the suitability of the simple steepest descent optimization scheme employed [52]. This is a multidimensional optimization problem where the optimum of the objective function cannot be computed explicitly from independent variables, but only by evaluating the objective function. Partial derivatives for the free variables were computed in order to approximate the gradient of the quality function. Partial derivatives could be numerically approximated by moving to the direction of each free variable at a differential step, rendering the scene and calculating the value of the objective function. After moving to all directions of the free variables, the optimal direction at a certain point in the optimization loop was chosen according to the values of the objective function calculated at every direction. The optimal direction was then selected as the seed for the new optimization loop. The optimization loop continues until a local minimum was found.

Obviously, the optimization scheme implemented in S&L's approach would readily converge to a local minimum. Thus, the optimization results would strongly depend on the initial values of the free variables of the objective functions that were almost randomly selected. Hence, there is a need for more powerful optimization techniques. For a better optimization framework, an investigation into the characteristics of the

objective function has been carried out. Figure 3.8 and Figure 3.10 show examples of shapes of objective functions for the scenes in which there are one and two objects respectively (the scene that corresponds to the objective function shown in Figure 3.10 is shown in Figure 3.9). From Figure 3.10 it is clear that the multi-objective optimization problem incorporates significant non-linearity, and as the geometric complexity of the scene increases the number of local minima will increase. With a greedy search of a space such as this, the solution is highly dependent on the starting condition.

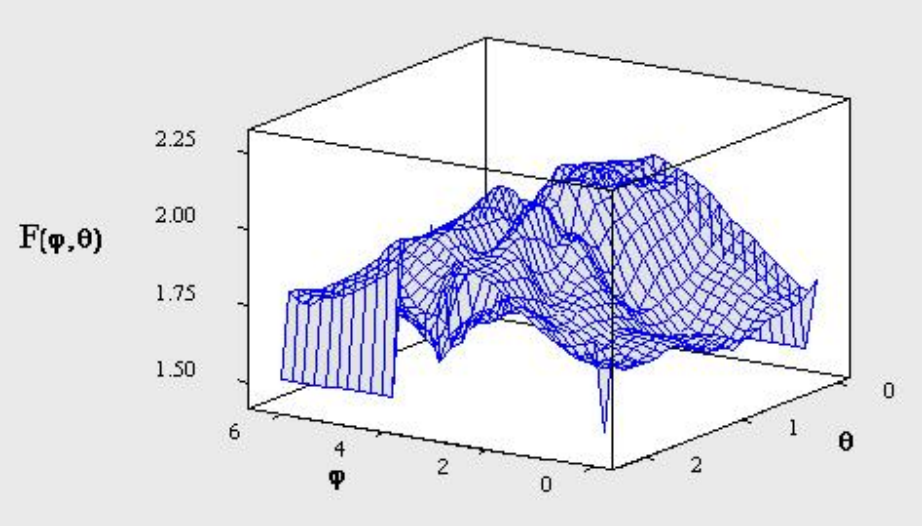


Figure 3.8: *$F(\varphi,\theta)$ for one object (elephant), θ is the elevation angle of the light and φ is azimuth angle of the light.*

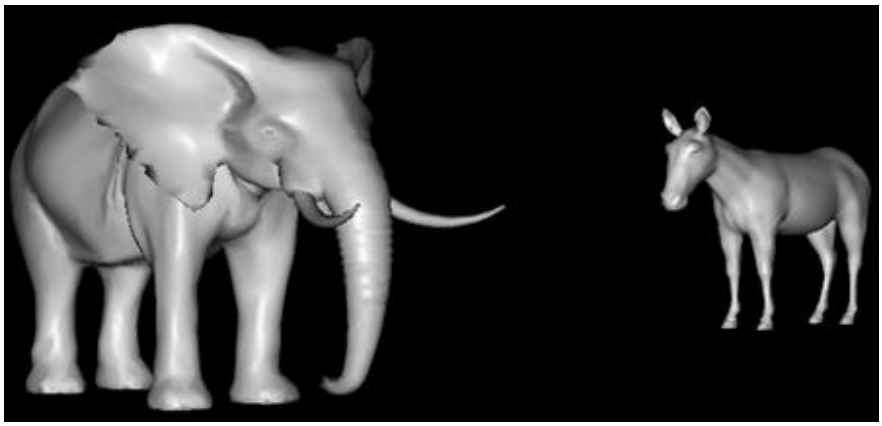


Figure 3.9: *Two object scene.*

Figure 3.8 and Figure 3.10 depict a plot of F(θ,φ) for one light source at constant intensity, and the presence of substantial numbers of local minima undermines the likely utility of a greedy search.

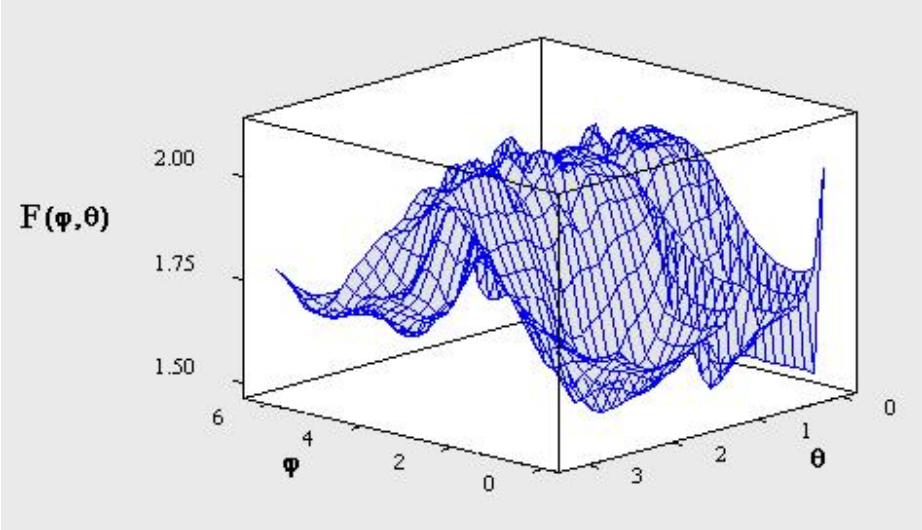


Figure 3.10: *F(φ,θ) for the two object scene, θ is the elevation angle of the light and φ is azimuth angle of the light.*

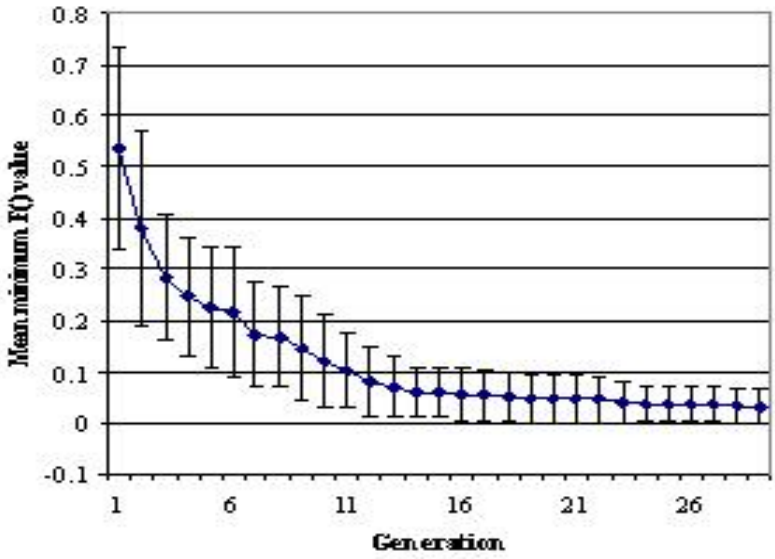
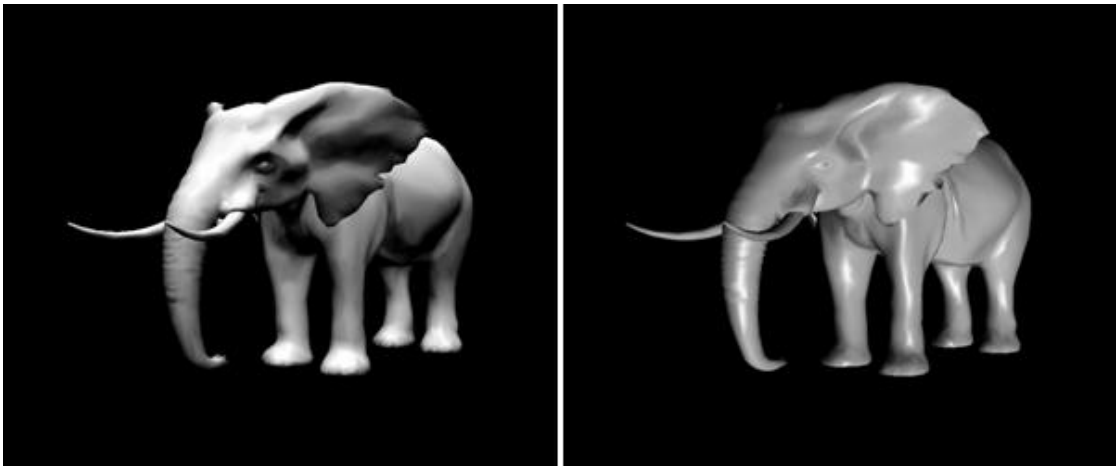


Figure 3.11: *Averaged results for 10 runs of a 30 member population, the value of F() for the best member and standard deviation in this value across the runs.*



(a) Greedy optimization of the Shacked & Lischinski [29] base function.

(b) GA optimization on the Shacked & Lischinski [29] base function.

Figure 3.12: *Results of the optimization of the same scene using different optimization techniques.*

The nature of the optimization problem suggests a requirement for a more general optimization strategy such as a Genetic Algorithm (GA) or Simulated Annealing (SA). GA belongs to a class of global search heuristics inspired by evolutionary biology that include inheritance, mutation, selection and crossover. GAs are generally significantly better that greedy approaches at avoiding local optima as they allow the evaluation stage to move to a random point rather than simply to a local point where objective function takes the lowest value. Elitism or inheritance mechanisms allows a GA to carry forward some of the best solutions into the next generation. Hence, in the worse case the solution found would be the equivalent to that found by steepest decent technique. The encoding of the lighting problem in a GA is straightforward. The free variables θ_k, φ_k, I^d_k, and I^s_k are encoded directly in the chromosome as real numbers as alleles. We have evaluated the optimization problem with varying population sizes and configurations for the GA. From our experiments, we established that a population size of 30 as illustrated in Figure 3.11, 10% elitism and crossover and mutation rates of 80% and 20% respectively were sufficient.

Note that the process of optimizing a single (key) light takes place under the constraint that the light position is limited in the values of elevation and azimuth angles (e.g. the position of a light is limited to a quarter sphere in front and above the centre of the scene). This constraint must be respected in all stages of the optimization, but

particularly at initialization and during the generation of random mutations (members of the population whose values do not reside in constrained ranges are rejected). Note that as the constrained region is convex, crossover cannot yield offspring that violate the constraint.

Although the selection of population size, mutation and crossover are problem specific, our particular configuration was established through experimentation on a variety of scenes. Evidence of satisfactory convergence of the GA is illustrated in Figure 3.11, which shows a plot of *F()* for the best members of the population (averaged over ten runs) at each generation where the population size was 30. The GA exhibited consistently better results than the greedy search, both in terms of the optimisation result and the visual quality of the solution. Figure 3.12 shows a direct comparison of the greedy and GA optimisation results, although care must be taken in generalising from specific examples such as these. The performance of a genetic algorithm strongly depends on selection of the parameters for mutation, crossover and elitism. With a common configuration for optimization problems of this size (a population size of 30, 10% elitism, and crossover and mutation rates of 80% and 20% respectively), GAs often give better (in this case, lower) solutions in comparison with steepest descent strategies, but with the drawback of relatively long run times.

An alternative to the optimization problem is simulated annealing (SA) which is a stochastic search technique inspired by the annealing process in metallurgy [53, 54]. Simulated annealing is a generic probabilistic meta-algorithm for the global optimization problem that is good at locating an approximation to the global optimum of a given objective function in a large search space, especially when the search space is discrete. The underlying idea of SA is to bring the system, from an arbitrary initial state at an initial high temperature, to a state with the minimum possible energy by employing a cooling scheme. The decision on making the transition from current state to the next candidate state depends on an acceptance probability function, which is determined by the energies of the current state and candidate states, and global time-varying parameter called the temperature. One essential requirement of SA is that the system may move to the new state even when it is worse than the current one. This feature allows SA to avoid becoming stuck in a local optimum.

The performance of simulated annealing depends on the selection of the initial temperature parameter *T* and the cooling schedule. In our approach, *T* was empirically determined such that the uphill probability was made equal to downhill probability at a random initial state. Temperature *T* was repeatedly reduced by 10% for each iteration. Simulated annealing often converges to a solution better than that of steepest descent, and has a significantly better performance than a GA (yielding similar, and in many cases a more optimal, solutions).

As anticipated, the steepest descent scheme converges fastest, but usually to a local minimum. Figure 3.13 illustrates the performance characteristics for the three schemes on 5 examples. Each scheme was run 10 times and the average values of the objective function are plotted for each iteration.

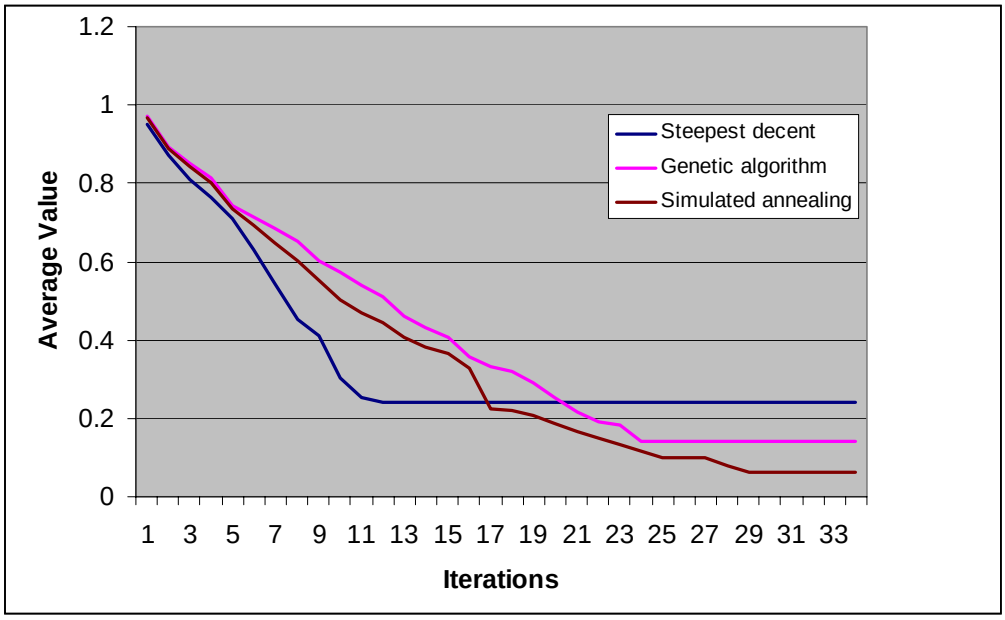


Figure 3.13: *Plot of the objective function (y-axis) for each iteration of the optimization schemes. Steepest descent converges fastest, but to a local minimum. Simulated annealing converges to the most optimal value.*

3.2.5 Shadow processing using shadow-map

Shadows have not been considered in most previous automatic lighting design approaches, probably because of the complexity of the implementation and the rendering times required. As described earlier in section 2.1, Poulin and Fournier [9] investigated

information from shadows and proposed an inverse method for designing light shape and light positions through the specification of shadows in a 3D scene.

In practice, shadows are one of the key ecological features of real-world vision that needs to be incorporated in the lighting design process. They serve a practical purpose in most scenes by showing spatial relationships between objects. Shadows on an object can be self shadows, cast by the object itself, or shadows cast by other objects in the scene. In computer graphics, shadows computed by basic ray tracings, which compute the shadows at a point only by evaluating the visibility of the point to the light sources without taking into consideration the vicinity effects of surrounding objects, yield the hard shadows, as compared to the soft shadows computed by advanced methods such as ray tracing with cones [55] and distributed ray tracing [56]. As already discussed, soft shadows aid spatial perception [57] and look considerably more natural than hard shadows. However, in automatic lighting design, shadows are problematic as they actually contribute regions of low luminance in the rendered scenes. For 2D image-based optimization techniques in lighting design, shadows tend to result in what would traditionally be considered as 'non-ideal' brightness levels in rendered images. Specifically, shadows significantly impact on the edge (F_{edge}), mean luminance (F_{mean}) and luminance variance (F_{var}) components of the objective function used in our approach.

With respect to the mean luminance component, shadows give rise to low brightness regions in the image that result in lowered values of the mean luminance component. For the same reason, shadows also tend to drive the luminance variance into a lower range of intensity since the target value of intensity variance constrains the intensity of the rendered image to a predefined width of intensity range. During the optimization process, lighting parameters are thus influenced such that the mean brightness of the rendered image increases to reach the target value component during optimization process. This leads to a defect whereby object surfaces are over-lit, and the final image of the optimization process is too bright. Edges are also significantly weakened in shadow regions and need to be enhanced using particular techniques [58].

We propose a solution to these problems whereby, for the brightness and intensity variance components (F_{mean} and F_{var}) in the objective function, the pixels in shadow

regions are not used in the calculation of the mean brightness and intensity variance components. For the edge component (F_{edge}), edge pixels are weighted differently for two types of edge pixels: that is, edge pixels that are in and out of shadow regions. The edge pixels in shadow regions are given a weight whose value is higher than that of edge pixels not in shadow regions. As a consequence, edges pixels in shadow regions have significantly more influence on the objective function.

A data structure called *shadow map* is used in the implementation of this approach. The *shadow map* is actually an array whose elements show which pixels are in shadow and which are not in shadow. The shadow map is derived by casting shadows into the scene with a depth buffer-based technique. Figure 3.14 shows an example of a *shadow map*.

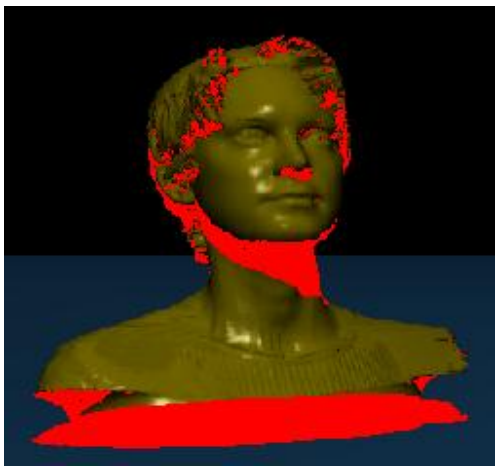


Figure 3.14: *Shadow map example. Red pixels are pixels in shadow regions.*

3.3 Implementation Details

In this section, a full discussion of the technical implementation is presented, which involves a colour-code based algorithm and related data structures for detecting visible surface and constructing pixel type maps. Normal maps and their application to setting up initial light positions are also discussed.

3.3.1 Colour-code for fast visible surface detection and pixel type map

In the implementation, a colour-code-based method is used to detect visible surfaces of objects in a scene and for creating the pixel type map. Objects are rendered without lighting, and each polygon of an object is rendered with a unique and exclusive colour. To obtain an exclusive colour for each polygon, a colour is inferred from an incremental integer variable I_c. A pseudo code for rendering objects with a colour-code method and creating a colour-code array is presented in figure 3.15.

The data structure, which contains information about colour-codes associated with each polygon of an object in the scene, is as follows:

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

`ObjectID is the` identification of an object

`FaceID` is the identification of a polygon of an object.

`Red` is the red component of a colour of a polygon.

`Green` is the green component of a colour of a polygon.

`Blued` is the blue component of a colour of a polygon.

An array of colour-code structure, denoted *CCArray*, is used to save information about the colour-codes for all objects. Figure 3.16 shows an example of an object rendered with the colour-code method. In our system, a colour-code image is saved in a matrix for later processing. This colour-code image is obtained by accessing the colour buffer of OpenGL. Each point in the matrix contains three colour components of corresponding pixel of the rendered image. This data structure can be described as follows:

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

```
 Byte Blue;
}
```

Visible surface detection

Visible surface detection is used to determine what is visible within a scene from a given viewpoint. There are commonly two approaches: (1) object-space methods which decide which object, as a whole, is visible; (2) image-space methods in which the visibility is determined point-by-point. The visibility of a polygon of an object can be easily inferred from information saved in the colour-code array and a colour-code image. The visibility of a surface is determined by checking the colour-code of the surface against available colours in colour-code image. If the colour-code of the surface exists in the colour-code image then that surface is visible. In our system, the visibility of polygons is used to calculate a normal map of visible polygons which is used for initializing the light positions in the optimization process. The pseudo code of the algorithm for detecting the visibility of a polygon of an object is illustrated in figure 3.18.

Pixel type map

A pixel type map indicates the type of a pixel in a 2D rendered image. An edge map of objects in a 3D scene is extracted by applying an edge detection operator to the depth buffer [44]. Sobel and Laplace edge detection operators are combined to enhance the accuracy of edge detection. Figure 3.17 illustrates a result of edge detection using depth buffer. A pixel type map is derived by combining an edge map and a colour-code image matrix. With colour-code, we know which points belong to objects. Points belonging to objects consist of edge and surface points. With edge maps, we know which points are edge points. Finally, a pixel type map contains three types of pixel, as follows:

- EDGE type: a pixel belonging to an edge of an object.
- SURFACE type: a pixel belonging to a surface of an object.
- BACKGROUND: a pixel not belonging to any object in the scene; this type appears when objects are rendered in the background (background properties are not considered during the optimization process).

Normal map and light position initialization

In our approach, normals are used to determine the initial positions of lights. The normal map is a data structure containing information about normals of visible polygons of objects. Visible surface detection is described in the previous section. Therefore, normal calculation of visible polygons is easily achieved. Depending on the number of lights used in the optimization process, normals of visible polygons are assigned a number of clusters. Centroids of clusters are considered as initial positions of lights. In our system, a standard K-Mean clustering method is used for clustering normals. A normal in our system is represented as a normalized 3D vector (rather than using a spherical representation). The K-Mean method takes as input an array of normalized 3D vectors and a desired number of clusters, and returns the centroids of clusters.

An alternative is to use curvature information for surfaces to determine the initial positions for the lights. Surfaces of objects are segmented according to the curvature information, and light positions are determined such that the specular highlights are maximized, as the specular component changes fastest compared to ambient and diffuse components (the ambient colour does not vary across an object, as the diffuse colour varies according the Lambertian law). Using an empirical approach is another alternative to initializing the light positions, whereby light positions are constrained such that they light from above and in front of objects.

```
Procedure RenderWithColorCode
Begin
Integer k;
Integer Ic, red, green, blue;
Ic = 0;
Nj: Number of polygons of the object j
No: Number of objects
ColorCode CCArray[0.. No-1];
For j=0 to No do
For k=0 to Nj do
Begin
   red = Ic&0xff0000;
   red = red >>16;
   green = Ic &0x00ff00;
   green = green >> 8;
   blue = Ic &0x0000ff;
   CCArray[Ic].ObjectID = j;
   CCArray[Ic].PolygonID = k;
   CCArray[Ic].Red = red;
   CCArray[Ic].Green = green;
   CCArray[Ic].Blue = blue;
       Render-Polygon k of object j with a
          colour comprised of three colour
             components(red, green, blue);
 Ic = Ic +1;
End;
End
```

Figure 3.15: ***Render scene with colour-code based method.***

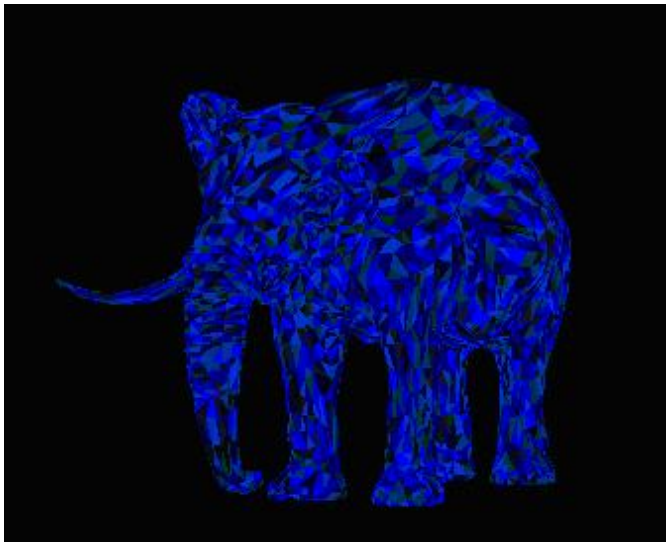


Figure 3.16: *An example of an object rendered with colour-code-based method.*

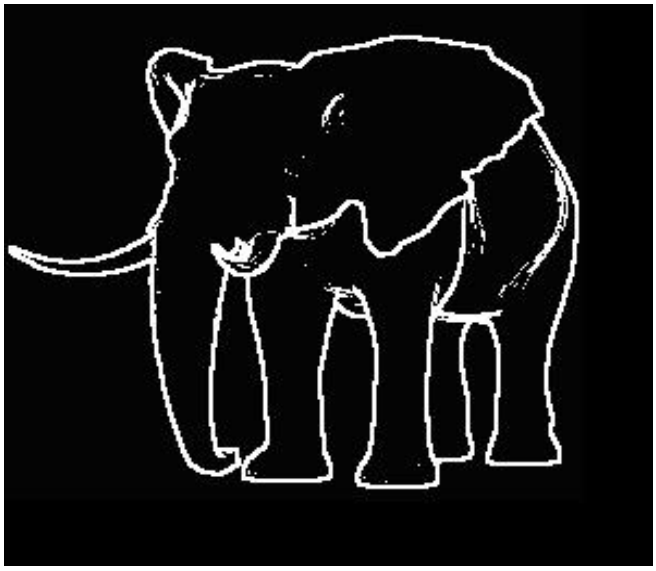


Figure 3.17: *Edge detection using depth buffer.*

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

Figure 3.18: *Colour-code based visible surface detection algorithm.*

3.4 Chapter Conclusion

In this chapter, a number of enhancements to the perception-based lighting design approach proposed by Shacked and Lischinski [29] have been presented. However, we note a number of shortcomings, both in the framework and the philosophy of lighting design through objective optimization. With respect to the framework, many aspects of visual perception are not accounted for and the approach lacks ecological validity. Contrast between objects gives the information about relative positions between objects. The contrast component has been added to the objective function with the goal of enhancing depth-perception and edge visibility of the objects in the scene. An algorithm has been proposed to calculate the contrast component. A backlighting technique, which is a practical lighting technique used in studio lighting, has been integrated into the lighting design process to enhance the silhouette visibility of the objects. Perceptually non-uniform colour spaces such as RGB have been widely used in graphic applications. The main drawback of the perceptually non-uniform colour spaces is that two points at a certain distance apart in a colour space may be perceived to differ by more or less than two other points the same distance apart in another region of the colour space. We addressed this issue by using a perceptually uniform colour space in our extended system. The impacts of shadows on the objective function have been considered in our work, in which the edges in the shadows are differently weighted and the surface pixels in the shadows are not used in the computation of the mean brightness component. By analyzing the nature of the objective functions for some sample scenes, the shape of the objective functions was found to be complex with multiple local extrema even for a scene with geometrically simple objects. Several optimizations such as steepest decent, genetic algorithm and simulated annealing have been implemented and evaluated in an optimization framework. There was a trade-off between using different optimization techniques, in terms of performance and quality. Also, a special treatment for edges was developed in which edge pixels in shadows were given higher weights than those that were not in shadows. The appropriateness of the extended features, compared to the original formulation, is described in a user study presented in chapter 7.

Chapter 4: Perception-Based Lighting-By-Example

4.1 Introduction

Ideal lighting is one of the approaches to the lighting design problem, in which lighting parameters are optimized in order to reveal visual properties of objects in the final 2D images. Spatial properties of objects are conceptually defined as different kinds of information that 2D images convey, such as depth information and information about the shape of objects. Spatial information is apparent in visual properties of 2D images such as shading gradients, the visibility of edge, contrast, average luminance, and histogram of 2D images. Ideal lighting approaches try to maximize visual characteristics of objects in a scene by optimizing an objective function, which is composed of components that correspond to physical properties of 2D images [29,59]. In practice, the notion of *ideal lighting* is only meaningful with respect to a small range of real-world graphics applications. Such applications include the automatic lighting of 3D visualizations where the number, position and orientation of 3D glyphs or other objects cannot be predicted in advance (and are typically not textured). Although in such visualization applications the colour, spatial properties and other physical characteristics of the elements capture all the information to be represented (and the role of lighting is to reveal these), in many domains subtle changes in lighting are used to convey mood, emotion, and factors other than the raw geometric and visual properties of the scene elements. The underlying idea of *lighting-by-example* stems from the fact that "ideality" is not always the best choice.

4.1.1 Lighting and expression from a cinematography perspective

Live-action lighting techniques have been the main focus of research on cinematic lighting; many live-action concepts can be applied to lighting for synthetic cinema. However, there are significant differences between light design for live-action and synthetic cinema in many aspects, such as how tasks are completed and by which roles of the design team. In live-action cinema, the director and cinematographer work collaboratively and simultaneously on lighting design and other tasks, such as the staging and framing of a shot. Coordination between different activities is critical as they are heavily dependent on each other. In order to achieve the best results of the lighting

design process, other activities in the production process can be altered. Synthetic cinema often employs a pipeline approach, whereby involving activities are carried out in sequence, and the lighting design is normally the last stage of the pipeline. The lighting design becomes more involved in the storytelling process as it begins sooner in the production process. An art director in synthetic lighting takes more responsibilities than a live-action art director. Apart from what an art director does in live-action, in synthetic cinema the art director often plays an important role in lighting decisions on lighting themes for individual scenes, as well as the movie as a whole. This is due to the fact that a stylized, illustrative quality of lighting in a synthetic movie depends more on hand-drawn animation than live-action cinema.

In cinematography, the director, who acts as a storyteller, decides on what the lighting designer is attempting to reveal. In order to achieve good cinematic lighting, it is vital for directors and lighting designers to have a thorough understanding of the story-point behind each scene, and its relation to the whole story. Lighting design for a synthetic film does not simply mean that the scene is illuminated such that the viewers can see what is happening, or that the lighting effects make objects in the scene look attractive. A much more important role of cinematic lighting is to drive the attention and emotion of audience by intentional lighting effects in order to emphasize the action and to set the mood. The following six lighting goals have been found vital to good lighting design [60]: “Directing the viewer’s eye, enhancing mood, atmosphere and drama, creating depth, conveying time of day and season, revealing character personality and situation, complementing composition”.

In almost all narrative films, or static images with a narrative theme, lighting positions and the subsequent spatial distribution of luminance in an image plays an important role in conveying both mood and emotion. In film, this is particularly important as the mood which a shot evokes plays a crucial role in supporting the narrative goals of the director, and is used to great effect to influence the subjective judgments of the audience as to the emotional tone of both particular scenes and the work in its entirety. A pleasing composition gives a sense of prosperity, a feeling of happiness. An unbalanced-lit composition could evoke a feeling of tension, which can be useful to build story tension or to convey the tensional state of a character. A progressive

building of visual tension could give a feeling that something bad is going to happen. Feelings about a dramatic change of the situation in a film can be driven by a sudden change in lighting that causes the changes in visual tension. Contradictions in lighting drive the audience to uncomfortable emotional states, which is useful if the intention is to take viewers out of context and to cause them to be shocked.

Enhancing mood, atmosphere and drama is vital to cinematography. There are many aspects of an image that have an impact on its mood and dramatic qualities. Lighting is one of the components which can convey mood and emotion of the story being told. Lighting design can create a pleasing mood or it can illustrate a contradictory mood to drive the context. The emotion and mood of a story-point can be implicitly driven by lighting design. A gloomy scene may not be discernibly perceived if it is not emphasized by a preceding inspirational scene. Likewise, it would not be appropriate to light a happy and glorious scene dimly.

An infinite number of combinations of different properties of lights, such as position, direction, colour, intensities, and throw pattern, can create a wide range of visual and emotional effects. Hence, a set of generalized rules have been used to describe lighting styles such as tonal range, which is the grey intensity in between darkest and brightest areas. Other elements such as colour shadows can be used to define lighting style. The distribution of intensity values of tonal range within a frame has a strong influence on the character and mood of a scene.

The decision on which lighting style should be used is normally motivated by the dramatic content and intent of the story. Soft fill light and light or soft shadows characterize the high-key lighting style, which is suitable for a light-hearted or comedic story. This lighting style results in a scene with minimal suspense. On the contrary, a scene with a dominant dark area emphasized with some bright area is a typical low-key lighting style that is usually deployed to stimulate the imagination of the audience and direct the attention of the viewers. Other than overall brightness or darkness used to describe the lighting style, contrast range can also be used to characterise lighting styles. High-contrast scenes, which are composed of a wide range of light and dark areas with a narrow middle range of greys, have a dramatic graphic quality and could elicit a sense of

conflict. Low-contrast scenes, which contain middle tones of shades, could evoke a feeling of calmness or deserted domination. Lighting patterns can be used to create the atmospheric effects that in turn enhance both depth and mood. The moods these create are strongly dependent on context and light colour. Bright, warm shafts of light give a cosy feeling while cool shafts yield a cloudy and hazy feeling.

Figure 4.1: *The pleasing balance in the lighting of this composition evokes a sense of prosperity.*

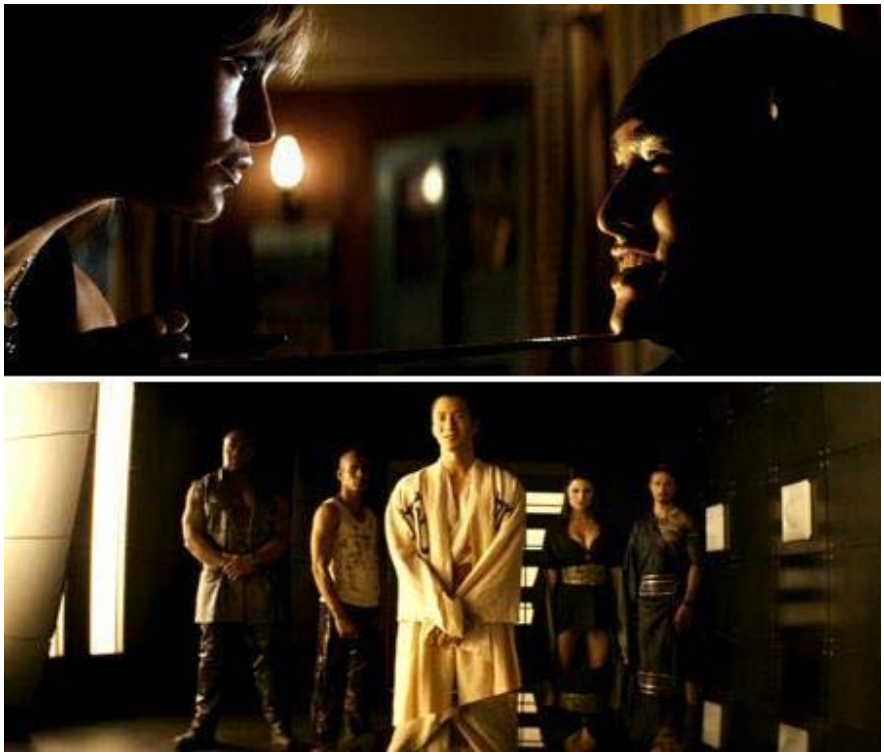


Figure 4.2: *A composition with little unbalance in lighting, can create a feeling of tension and apprehension (© 2004 20th Century Fox).*

4.1.2 Lighting-by-example

The goal of a general research program into lighting design is not an accurate (and empirically verified) objective function for ideally illuminated scenes, but a framework for the specification of lighting using example 3D scenes and even photographs – tools to allow artists and graphic designers to interactively modify scene lighting through inverse design. Such approaches presume the ability to model target scenes in the form of a perceptually meaningful objective function, and to optimize source scenes using these objectives. The *lighting-by-example* approach uses a perception-based lighting framework that was initially developed within the *ideal lighting* framework, with which lighting parameters can be optimal with respect to a set of target values for different components in an objective function. As such, the *lighting-by-example* approach recognizes that perceptual optimality is rarely an appropriate or meaningful notion when 3D artists are engaged in lighting design.

Automatic lighting design aims to provide users with semi-automated approaches, and easy-to-use tools, to configure lighting for 3D scenes. Following a perception-based lighting design framework, which models image quality using cognitively-inspired objective functions, we present a new approach to lighting design that can be devised which both: (1) allows the declarative specification of lighting and (2) facilitates an intuitive and natural interactive control of scene lighting. The approach should enable users to select the desired lighting for a scene using exemplar 3D scenes or 2D images and use the perceptual properties of these scenes as the target values of an initial optimization step.

Even lay viewers are highly sensitive to the emotional tone of an image arising from its lighting, although non-expert viewers will have little or no insight into the configuration of lights with which such effects are created. Indeed, in photographic and film production, the subtleties of scene lighting are often the result of highly artificial configurations of lights on a studio set (and post production editing). In short, we know what we want when we see it, but have little idea of how to reproduce it. This is the observation on which we base the *lighting-by-example* approach – that lighting is best configured for 3D scenes on the basis of existing exemplar images and not through direct manipulation of lighting types, positions and luminance.

4.2 Previous Example-Based Approaches

There have been a number of approaches which can be considered either examples of, or strongly related to, example-based lighting design. Schoeneman et al [2] addressed lighting design as an inverse problem. Users were able to configure a set of desired properties that are expected to appear in the final image and the system tried to find a solution whose properties are closest the set of desired properties. The user paints on the surfaces of the rendered scene causing a change in the surface radiance functions. These after-painted surface radiance functions are used as target values in the optimization process that followed. Painted surfaces in the rendered image are given more weight, which biases the optimization towards solutions to those with properties that best match the painted surfaces. Painted surfaces can be considered as *examples* affecting the target radiance surface functions, though in this approach Schoeneman et al only address the problem for finding matching light intensities and colours for fixed light positions. A

different approach to optimizing lighting for individual objects, and to inverse lighting design through the manipulation of object properties (such as shadows), is Design Galleries [16]. Marks et al's goal was the design of an interactive system to allow a user to interactively reduce the design space for light configurations; this would be done through the use of a mapping function between an input vector containing light position, light type, and light direction, and an output vector containing a set of values that summarizes the perceptual qualities of the final image.

During the optimization step, lights are moved from one predefined position to another. At each position a light type is selected from a set of light types, and a corresponding image is generated. Final images are then arranged in clusters on the basis of the perceptual distance between images. Design Galleries can be considered to be in the spirit of an example-based approach despite the fact that there is no specific target used as the basis for an objective function (to be optimized). Through its generation of a wide range of clustered images as examples, Design Galleries presents sets of exemplars for users to perform selection on as part of a render-selection loop. Thus, there is no explicit specification of what the appearance of the final image should be, but the user has the opportunity to select good candidates.

Image-based lighting can also be considered as another form of example-based approach. Supan and Stuppacher [61] present an approach to lighting augmented environments in which virtual objects fit seamlessly into a real environment. A key challenge in such applications is how to consistently coordinate the lighting between virtual objects and a real environment. Image-based lighting approaches attempt to capture lighting information from a real environment and use it somehow to light virtual objects, such that consistency in the lighting of the virtual object and the real-world objects can be obtained. At the heart of this approach is an environment map which represents lighting information of the real environment. To obtain the environment map, a mirrored sphere is set up in a real environment such that the surrounding scene can be seen as a reflection on the sphere, and the image of the mirrored sphere is captured with a camera. Specular and diffuse sphere maps can be created using a radial blur technique in which image captured from the mirrored sphere is mapped to a virtual sphere of unit radius. Shadows are also addressed in this approach, and to calculate the shadows cast

by virtual objects light positions in the real environment are identified. Supan and Stuppacher use an intensity distribution-based technique for estimating the light positions from the environment map. A high dynamic range image derived by combining several images captured under different lighting exposures was used to enhance the accuracy of the light position estimation. The final rendering process requires only small modifications in the calculation of the lighting (obtained by looking up values in the environment map and using Lambert's Law). This approach can be considered as a class of lighting-by example, that is, a lighting optimization problem in which lighting parameters for a virtual environment are optimized on the basis of lighting information captured from a real environment.

4.3 *Lighting-by-example* with Perception-Based Methods

As mentioned earlier, the aim of the *lighting-by-example* approach to lighting design is to provide users with a means to express their desired characteristics through the use of pre-lit exemplars. A user selects a target image from a set of examples provided by the lighting system in the form of 3D scenes or 2D images. Properties of the selected example are used in an initial optimization step. The optimization process seeks to discover a configuration of the lighting parameters such that the components of the objective function, extracted from a rendered scene, have values that are close to that of the target example.

4.3.1 Target property extraction

Our first proposal for *lighting-by-example* is based on our core perception-based lighting design framework. This in turn is an extension to the approach proposed by Shacked and Lischinski [29]. In their perception-based lighting design scheme, the position and intensity of light sources (specular and diffuse components of a local illumination model) are optimized using an evaluation function that characterizes separate aspects of low-level processing in the segmentation and recognition of objects. This approach uses an objective function that is the linear combination of five distinct image properties. The first step of the *lighting-by-example* pipeline is the computation of target values for the six objective function properties: edge distinctness, mean brightness, mean shading gradient, intensity range, image intensity distribution, and contrast. The key data

structure for this process is the *pixel type map* which records the type of pixels in a rendering. A pixel type map for renderings of 3D scenes is extracted by applying an edge detection operator to the depth buffer [44]. Sobel and Laplace [62] edge detection operators are combined to enhance the accuracy of edge detection. Points on the surface of an object in the scene are detected by an algorithm using colour-coded polygon identifiers [59, 63]. The resulting pixel type map contains three types of pixels: EDGE (pixels on the edge of an object), SURFACE (pixels on the surface of an object), and BACKGROUND (pixels corresponding to objects and scene elements that are considered to be in the background). The pixel type map is used in the calculation of all components of the objective function.

Edge extraction

Boundaries in images are perceived via edges, and edges detection is the fundamental problem in image processing. Edges in images are regions with dramatic changes in intensity from one pixel group to the adjacent ones. Many ways of performing edge detection have been studied. However, the majority of different methods may be classified into two categories, gradient and Laplacian. The gradient method identifies the edges by searching for the maximum and minimum in the first derivative of the image function $I(x,y)$. The Laplacian method searches for zero crossings in the second derivative of the image function $I(x,y)$ to find edges. Figure 4.4 illustrates the idea of gradient-based and zero crossings-based methods of edge detection. Obviously, the first derivative shows a maximum positioned at the centre of the edge in the original signal. This edge detection method is the characteristic of the "gradient filter" family of edge detection filters. A pixel location is considered as an edge location if the value of the gradient is greater than some threshold. Edges should have higher pixel intensity values than those surrounding it; hence, once a threshold is set, the gradient value can be compared to the threshold value in order to detect an edge whenever the threshold is exceeded. Mathematically, when the first derivative of a function is at an optimum, the second derivative of the function is zero. As a result, locating the zeros in the second derivative of an image function is another way of finding the location of an edge.

In this approach, given a 2D rendered image, edges are detected by combining first and second order derivatives of the image function $I(x,y)$. The first derivative of $I(x,y)$,

the intensity gradient *G(i,j)*, detects the local change in luminance at a pixel *p(i,j)*. This component is calculated on the basis of the Sobel operator described in equations (4.1-4.4)

$$G_x(i,j) = I(i+1,j+1) + 2I(i,j+1) + I(i-1,j+1) - I(i-1,j-1) - 2I(i,j-1) + I(i+1,j-1) \quad (4.1)$$

$$G_y(i,j) = I(i+1,j-1) + 2I(i+1,j) + I(i+1,j+1) - I(i-1,j-1) - 2I(i-1,j) + I(i-1,j+1) \quad (4.2)$$

$$|G(i,j)| = \sqrt{G_x{}^2(i,j) + G_y{}^2(i,j)} \quad (4.3)$$

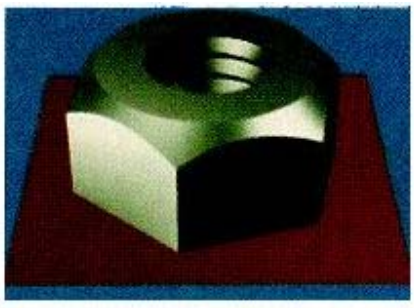
$$|G(i,j)| = |G_x(i,j)| + |G_y(i,j)| \quad (4.4)$$

Where:

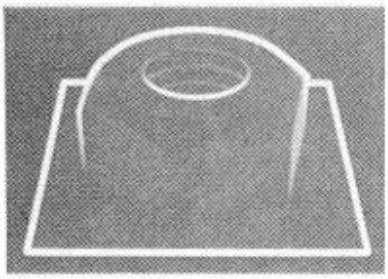
$|G(i,j)|$ is the magnitude of the gradient.

I(i,j) is the value of the image function at row *i* and column *j*.

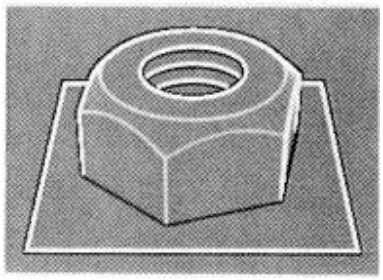
The magnitude of the gradient is calculated in equation (4.3) and approximated in equation (4.4).



(a) source image



(b) 1st order differential



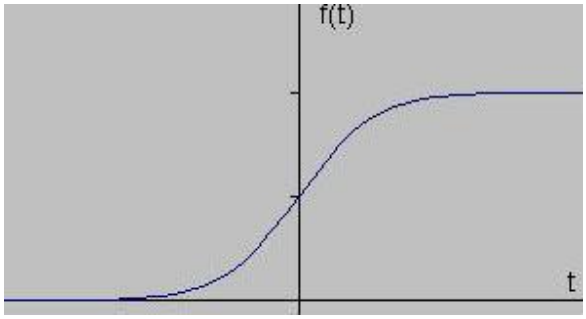
(c) 2nd order differential

Figure 4.3: *Effects of different edge detection methods.*

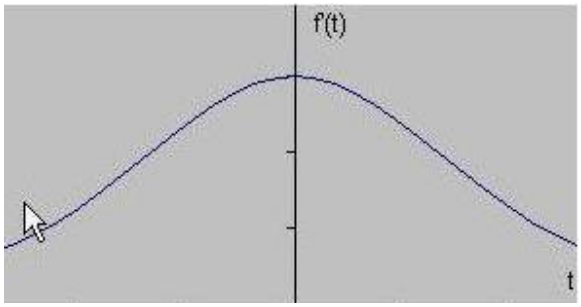
The second order derivative addresses discontinuities in the first derivative [44] and is calculated as in equation (4.5).

$$L(i,j) = \frac{8I(i,j) - I(i+1,j-1) + I(i+1,j) - I(i+1,j+1) - I(i,j-1) + I(i,j+1) - I(i-1,j-1) - I(i-1,j) - I(i-1,j+1)}{8} \quad (4.5)$$

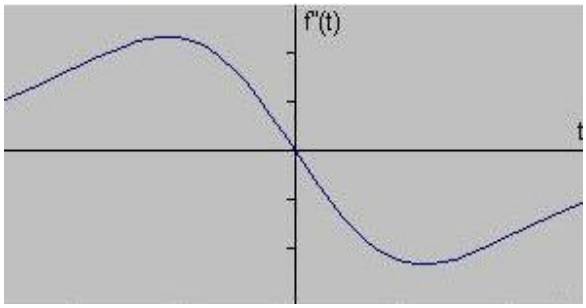
L(i,j) is the second derivative calculated at row *i* column *j* of the image.



(a) Signal with an edge shown by the jump in intensity.



(b)First derivative of the signal.



(c) Second order derivative of the signal.

Figure 4.4: *First and second order derivative of a function f(t).*

In lighting-by-example, the aim is not to maximize the edge contrast but to minimize the edge difference in the normalized edge ratio.

Shading gradient extraction

Perceptual psychologists have extensively studied issues relating to shading used to extract depth and shape information by the human visual system [31, 41]. In our lighting design framework, the shading gradient component of the objective function serves to enhance perception of depth. The shading gradient at each pixel *p(x,y)* is calculated and the final shading gradient is derived by averaging shading gradients over the whole image.

$$T_{grad} = \sqrt{\frac{1}{N_s} \sum_{p(i,j) \in S} |\nabla_{i,j}|^2} \qquad (4.6)$$

T_{grad}: is the target shading gradient component extracted from an example.

p(i,j) is the pixel at the i^{th} row and j^{th} column in an image.

$\nabla_{i,j}$ is the shading gradient of image function *I(x,y)* at a pixel *p(i,j)*.

S is the set of surface pixels derived from the pixel type map.

N_s is the number of surface pixels.

Mean luminance extraction

Luminance is not a perceptual quantity but a physical measure used to define an amount of light in the visible region of the electromagnetic spectrum. Hence, it can be directly measured by a scientific measuring device. The human visual system is not equally sensitive to all wavelengths. The human visual system is about one hundred times less sensitive to light at the wavelength of 450 nanometres than it is at the wavelength of 510 nanometres. Human spectral sensitivity is measured in terms of the equivalent of visual effects and wavelengths, which means two equal fields of the same wavelength and radiance should be equally perceived by viewers in all respects. The mapping characteristics between light spectrum and visual effects can be characterised by mathematically interpolating the raw curve established according to equivalences of many pairs of light spectrums. The sensitivity of the visual system at a randomly selected wavelength can be inferred from the mapping characteristics, which is the

relative *spectral sensitivity function* for the human visual system. Luminance takes into account two major factors: the energy of the chromatic light and the spectral sensitivity of the observer. Luminance is a measurement of light energy weighted by the spectral sensitivity function of the human system. The spectral sensitivity function is very useful as it provides a close match to the combined sensitivities of the individual cone receptor sensitivity functions. Reasonably, the spectral sensitivity function can be considered as measuring the luminance efficiency of the first stage of an extended process that ultimately allows the human visual system to perceive useful information, such as surface lightness and the shapes of the surfaces. Technically, it defines how the sensitivity of the so-called luminance channel varies with wavelengths. The luminance channel is an important theoretical concept in vision research; it is held to be the foundation for most pattern perception, depth perception, and the motion perception [64].

In this approach, the mean luminance component captures the global luminance of the rendered image. The mean luminance of an example is calculated as follows:

$$T_{mean} = \frac{1}{N_s} \sum_{p(i,j) \in S} I(i,j) \tag{4.7}$$

T_{mean} is the target mean component extracted from an example.

$p(i,j)$ is the pixel at the i^{th} row and j^{th} column in an image.

$I(i,j)$ is the value of the image function at a pixel $p(i,j)$.

S is the a set of surface pixels derived from the pixel type map.

N_s is the number of surface pixels.

Luminance variance extraction

The human visual system is particularly sensitive to a narrow range of luminance around a certain average luminance value. The range of light intensity levels to which a human visual system (HVS) can adapt is on the order of 10^{10}. Subjective brightness, intensity as perceived by the HVS, is a logarithmic function of the light intensity incident on the eyes. However, the HVS cannot operate over such a range of intensity simultaneously. For any given set of conditions, the current sensitivity level to which the HVS adapts is called *brightness adaptation level*, and the HVS is most sensitive to intensities around this level. The HVS can distinguish between variations in brightness at any specific

adaptation level. At low levels of illumination brightness, discrimination is poor because rods enable HVS to operate at low levels of illumination and they are not involved in colour vision; brightness discrimination improves significantly as background illumination increases as cones, which best operate at high levels of illumination, are sensitive to colour. The typical observer can perceive one to two dozen different intensity changes which are the number of different intensities a person can discern at any one point in a monochrome image. Works on adaptation levels by Tumblin & Rushmeier [36] and Stevens [37] theorized that the adaptation level can be estimated by the expected value of log_{10} of the luminance visible on the retina. The HVS is not sensitive to intensities at values far below this level. The number of illumination levels that appear in a rendered image can be broad. However, the HVS cannot discern all illumination levels simultaneously. This means that we should not create an image with a large range of intensities. Consequently, variance of the luminance of pixels should be minimized for an image.

The variance component is calculated as in equation (4.8):

$$T_{\text{var}} = \sqrt{\frac{1}{N_s} \sum_{p(i,j) \in S} \left(I(i,j) - T_{mean} \right)^2} \tag{4.8}$$

T_{var} is the target luminance variance component extracted from an example.

$p(i,j)$ is the pixel at i^{th} row and j^{th} column in an image.

$I(i,j)$ is the value of image function at a pixel $p(i,j)$.

S is the set of surface pixels derived from pixel type map.

N_s is the number of surface pixels.

T_{mean} is the target mean component extracted from an example.

Brightness histogram extraction

A histogram can demonstrate whether or not the image has been properly exposed, or whether the lighting is harsh or flat. Each pixel in an image has a colour which resulted from some combination of the primary colours such as red, green and blue (RGB). Each of these colours can take a brightness value ranging from 0 to 255 for a digital image with a bit depth of 8-bits. A greyscale value can be calculated from values of three

colours. A greyscale histogram can be obtained when the computer scans through each of these greyscale values and counts how many are at each level from 0 through 255. The basic layout of a histogram of an image is illustrated in Figure 4.5.

The region where most of the brightness values appear is called the *tonal range*. Tonal ranges are drastically different from image to image. There is no one *ideal histogram* which all images should try to mimic; histograms should only be representative of the tonal range in the scene and what the image conveys. A histogram contains information about contrast of an image as the spread of the histogram relates to contrast. Contrast is a measure of the sudden changes in brightness between light and dark areas in a scene. A broad histogram reflects an image with significant contrast, whereas a narrow histogram illustrates less contrast of a flat image. From the peak of the histogram, we can draw information about whether the image is a *low key* one or a *high key* one. Images where most of the colour intensities occur in the shadows are called low key, whereas high key images have most of the colour intensities in the highlights.

Histograms are used to characterise other aspects of an image such as contrast and tone key. In cinematography, those characteristics are related to viewers' perceptions of the emotions conveyed by the image. High key lighting is normally used for pleasing scenes, whereas low key lighting is the choice for slow or tensional scenes. A brightness histogram is used to represent the distribution of brightness value over pixels. The histogram comprises of 256 bins, and the value of each bin represents the number of pixels at a certain brightness range. The histogram is normalized by dividing the value of every bin by total number of pixels used in the histogram.

$$T^{k}_{hist} \underset{k\in[0,255]}{} = \frac{N_k}{N_t} \tag{4.9}$$

T^{k}_{hist} is the target histogram value of bin k extracted from an example.

N_k is the number of pixels at brightness level k.

N_t is the total number of pixels used for calculating histogram. If the example is a 3D-model, this is the total number of edge and surface pixels. If the example is a 2D image, this is the total number of pixels in the 2D image (see section 4.3.2).

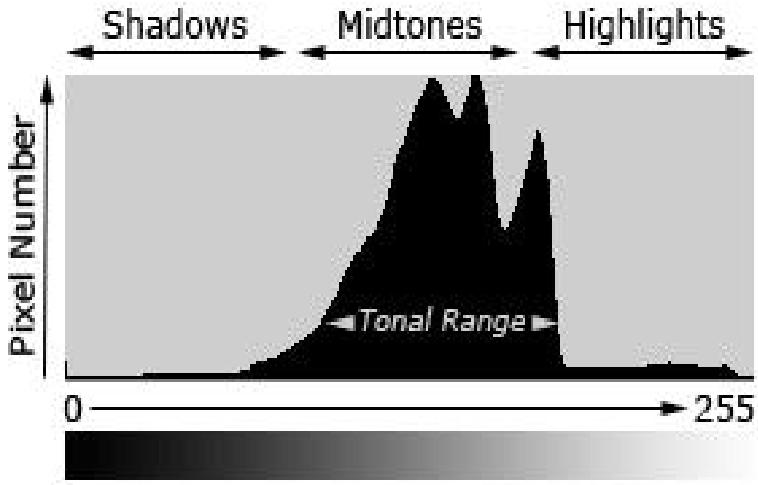


Figure 4.5: *Histogram of an image.*

Contrast extraction

Empirical studies of visual cognition have demonstrated that object perception depends on both absolute amount of luminance and the difference of the object's luminance from its background [31]. The relative visibility of an object in an image or a scene is partially influenced by local variations in luminance, or brightness contrast. Generally, objects with a stronger contrast have been found to draw more visual attention than other objects with a weaker contrast [65]. Correlation between relative visibility and perceived depth is confirmed by observations showing that the apparent depth of a given region within the visual field is decided by variations in local brightness or hue [66]. Schwartz and Sperling [67] have demonstrated by experiments on the kinetic depth effect that the visual system uses brightness contrast to render this depth phenomenon perceptually non-ambiguous; that is, to resolve the problem of relative distance from objects in an image or a scene to the HVS in the stimulus. This observation, which was termed *proximity-luminance-covariance* in binocular viewing, was a motivation for other psychophysical studies of contrast as a depth cue. O'Shea et al [68] have proven that the higher-contrast stimulus of a pair of stimuli appears nearer than the lower-contrast stimulus in monocular viewing. They claimed that relative visibility, or contrast, should be sufficient as a pictorial depth cue because it simulates the optical consequences of aerial perspective. Possible interactions of contrast with other cues such as interposition

or partial occlusion, which are considered as major determinants of pictorial depth [69, 70], were not taken into account in these studies. The FACADE (Form-And-Colour-And-Depth) theory introduced by Grossberg [71, 72] has clarified the mechanism of the visual cortex giving rise to 3D percepts of objects separated from their backgrounds.

The FACADE theory provides a model for the formation of 3D percepts from 2D images. After monocular pre-processing, the visual input is fed in parallel into two subsystems of the cortical network: the BCS (Boundary Contour System) and the FCS (Feature Contour System). Figure 4.6 illustrates the model proposed in the FACADE theory. The BCS is orientation-selective and pools opposite contrast polarities. It generates early representations of contour groupings in the image. The FCS generates visible surface representations that are sensitive to contrast polarity. At an early stage of processing, monocular outputs of both subsystems cooperate and compete to determine which boundaries and surfaces will be selected. A relatively strong grouping signal that coincides with a relatively strong contrast signal stands a better chance of winning the competition than weaker coinciding grouping signals. The binocular form representations that emerge after this competition are stored and activate different representations of surface depth within the visual cortex. The stronger groupings that have survived the competition before binocular integration are predicted to be perceived as ‘nearer' by an observer. The weaker groupings that have lost the competition are predicted to be perceived as 'further away'. A useful result drawn from this analysis was the demonstration that the same cortical mechanisms clarify how 2D pictures give rise to percepts of objects separated from, and in front of, their backgrounds. A major theme of the theory is that several different types of cooperative and competitive processes, whose interactions give rise to 3D scenic percepts and 2D pictorial percepts, can be activated by pictorial cues, such as contrastive and geometrical relationships among contours.

As studied by Grossberg in [72], the model predicts that contrastive and geometrical properties of a 2D picture can affect the boundary contour system and feature contour system in different ways, and thereby alter the ensuing figure-ground percept. Dresp et al [73] explored this possibility on the basis of psychophysical data. In three experiments, interactions of contrast and contour factors with other pictorial depth cues such as interposition and partial occlusion were tested. The expected outcome was that the

emergence of near-far percepts in briefly presented, two-dimensional, images may be strongly influenced by the contrast of a given visual object. However, the contrast cueing was proven to cooperate or compete with other pictorial cues, including the geometrical relationships among image contours, in the manner predicted by the FACADE theory. The satisfying result drawn from this study was that figural contrast was proven to be a principal factor for the emergence of percepts of near versus far in pictorial stimuli, especially when stimulus duration is brief.

Contrast Extraction Implementation

In our approach, the notion of contrast was extended through the provision of a means of evaluating difference in luminance between adjacent parts of an object and incorporating this in our objective function. As described in section 3.2.1, the contrast between two parts of an object is given by:

$$C_{ij} = \frac{(Y_i - Y_j)}{Y_j} \tag{4.10}$$

C_{ij} is the contrast between part *i* and part *j*.

Y_i is the mean luminance of part *i*.

The mean luminance of a part is calculated as follows:

$$Y_i = \frac{1}{N_i} \sum_{I_{x,y} \in P_i} I(x, y) \tag{4.11}$$

P_i is the part *i* of an object.

N_i is the number of pixels of part *i* that are visible.

p(i,j) is a pixel at row *x* column *y* in the image.

I(x,y) is the value of image function at pixel *p(x,y)*.

Based on the assumption that the edges in the pixel type map correspond to boundaries between parts of an object, an algorithm (see section 3.2.1) was developed to calculate the contrast between adjacent parts of a 3D object using the pixel type map.

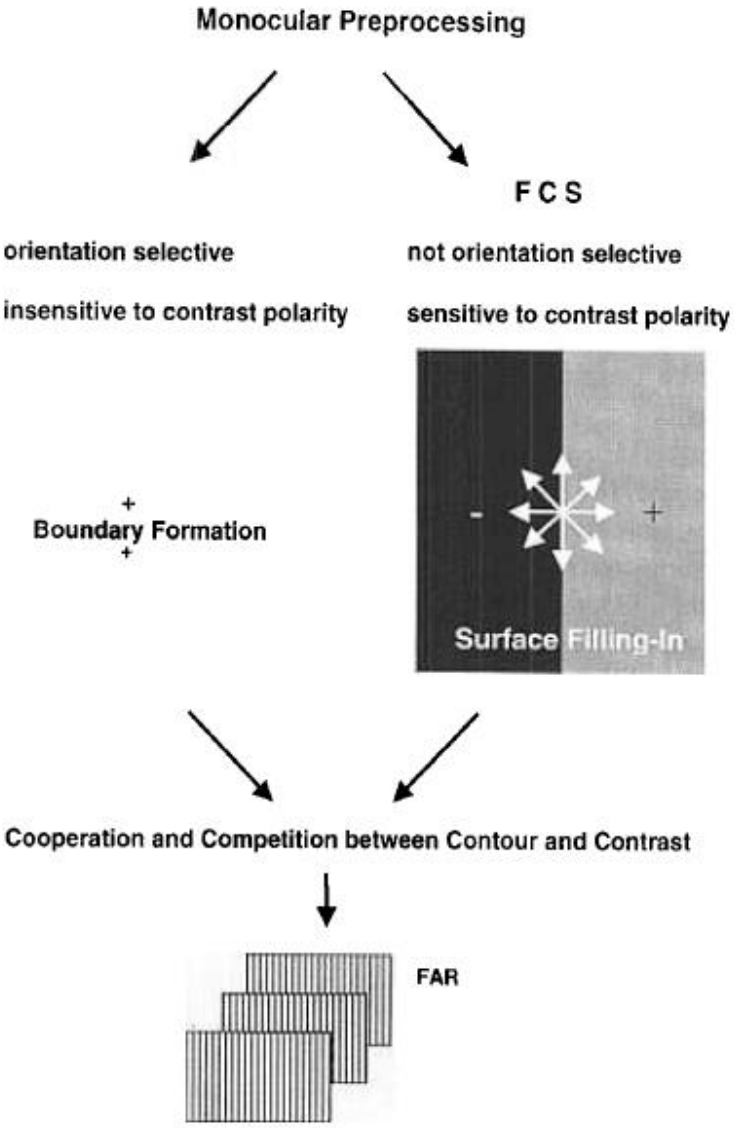


Figure 4.6*: A model f or the formation of 3D percepts from 2D images proposed in FAÇADE theory [72].*

4.3.2 3D versus 2D examples

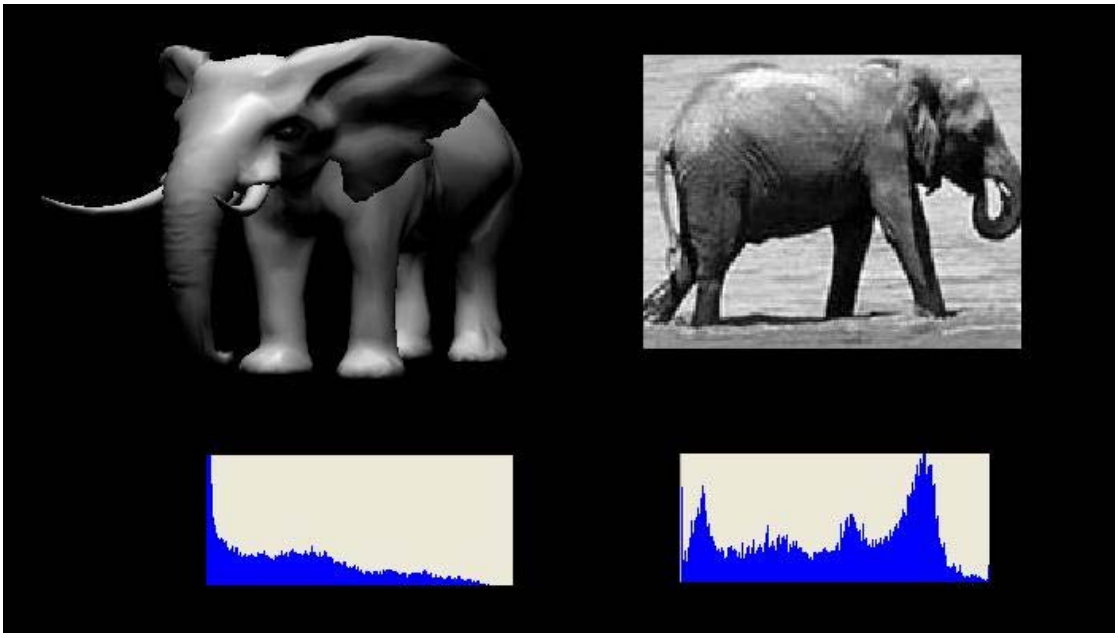
A 3D example is created by rendering a sample model with user-defined lighting parameters where the model has been created, either for a separate purpose or for the specific purpose of being used as a *lighting-by-example* exemplar. The components of the objective function are extracted and saved in a configuration file. The rendered image is also saved as a bitmap. In practice, the 3D model can be rendered multiple times with different configurations of lighting parameters in order to create different examples of the same model (as in *Design Galleries* [16]).

2D exemplars are 2D images created either by rendering 3D models or using photographic images of real scenes. Other than the edge and contrast components, which are not used in 2D exemplars, the remaining target components are extracted in the same way for both 3D and 2D examples. Target values for components of the objective function are calculated over all pixels of the 2D image example. For 3D examples, the pixel type map is used to decide which pixels should be taken into consideration in the calculation of the target values for components of the objective function. Note that for 2D image exemplars, S (a set of surface pixels) is computed using a set of pixels of the whole 2D image excluding only the set of edge pixels, which can be derived by applying a standard edge detection technique to the example image. For implementation, a 3D exemplar is composed of two separate files: one is a normal bitmap which is displayed in the interface for users to know the lighting effects of the exemplar, while the other text file contains target values for the components extracted from the target bitmap, and the target values are used for optimization process. Due to the difference between 2D and 3D exemplars in the number of target values for the components of the objective function, the effects given by using 2D exemplars are different from those given by using 3D exemplars. Figure 4.7 shows a 3D exemplar and a 2D one with their corresponding target values.

A 2D exemplar does not include edge and contrast target values for components of the objective function. Hence, those edge and contrast components are not active during the optimization process, and the optimization process will not draw the objective function to the point where the prominence of the edges of the objects in the scene is enhanced. This probably results in a lighting configuration for the scene whereby the visibility of edges of the objects in the scene is limited. In particular, there is a possibility that the silhouettes are partially unlit. Due to the lack of target value for the contrast component, using 2D exemplars in *lighting-by-example* might result in a lighting configuration in which contrast between parts of an objects and between objects is not enhanced.

Figure 4.8 and

Figure 4.9 show examples of results of the approach for different classes of exemplar (2D, 3D and photograph). The recognizable properties of exemplars, such as overall luminance and shading gradient, are captured to optimized images.



F_{edge} = 0.2714 ; F_{mean} = 85.0; F_{grad} = 6.5084 ; F_{var}= 59.7420; F_{const}= 2.8576;

F_{mean} = 124.0000; F_{grad} = 15.4108 ; F_{var}= 70.8087;

Figure 4.7: *3D & 2D exemplars and their corresponding target values and histograms (bottom images). Although the edge component takes values in the range [0,1], the other components take values in the range [0,255].*

Figure 4.8 shows the results in which target 2D exemplars are used, Figure 4.9 shows the results in which target 3D exemplars are used, and Figure 4.10 shows the results in which target 2D photos are used. Images in the first row are targets, and those in the other rows are corresponding results. Results in columns 1 and 2 of Figure 4.8 and Figure 4.9 have luminance that is quite similar to that of targets. The ranges of brightness of results in columns 1 and 2 of Figure 4.8 and Figure 4.9 are recognizably equivalent to those of targets, and that means the luminance variance and histogram components have significant impacts on optimization process. Looking at results in columns 1 and 2 of Figure 4.8, the average luminance of those results are pretty much the same, but the shading effects are significantly different due to the impacts of the shading gradient component on optimization process.

4.4 Chapter Conclusion

In this chapter, we have presented a *lighting-by-example* approach to lighting design. In this approach, users are presented with a set of intuitive images as examples of lighting effects. The user can select the example of lighting effects that he wants to have in the scene to be lit. There are several kinds of examples used in the system: a 3D example derived by rendering a 3D scene and saving the target values to a file, or a 2D example created by either rendering a 3D scene and saving the colour buffer to a bitmap or using a photograph. The only difference between using 3D examples and 2D examples is that edge and contrast components are not used in the objective function when using 2D examples. A perception-based objective function was used at the core of the approach to capture the properties over an image function. Extended features such as shadow processing have been incorporated into the processing pipeline.

In the *lighting-by-example approach*, users are empowered with an intuitive way of expressing their desired lighting effects. This is due to the fact that we normally realize what we want when we see something similar but we do not know how to manipulate the lighting parameters to produce desired lighting effects. With *lighting-by-example*, a naïve user is able to optimize lighting parameters for a scene, as all he has to do is to select an exemplar that has lighting effects similar to what he want in the source scene. The potential application of a program of research into the design of lighting includes the specification of lighting in which photographs are used as targets, though future tools should incorporate controls that help graphic designers interactively modifying scene lighting through inverse design.

This approach to *lighting-by-example* takes the view that target scenes can be modelled in the form of a perceptually meaningful objective function, and the lighting of source scenes can be optimized using these objectives. Spatial frequency, that is, the spatial distribution of pixels at certain luminance levels, is not captured by this approach. An alternative approach to *lighting-by-example* that utilizes spatial frequency information is the topic explored in chapter 5.

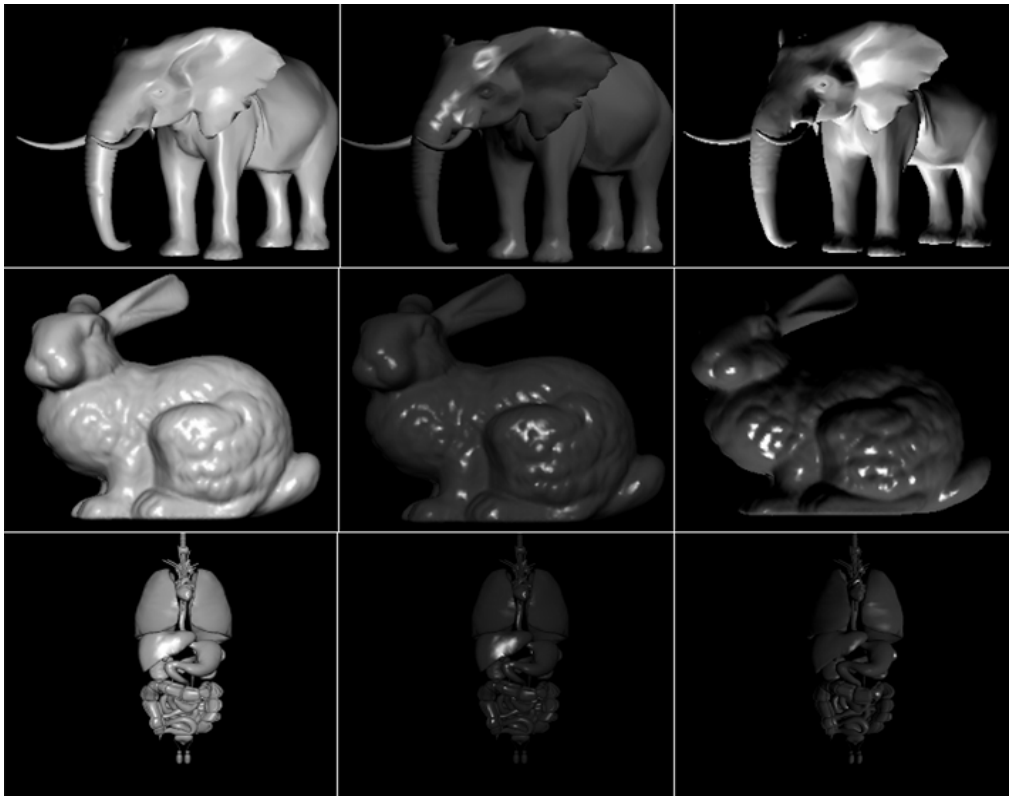


Figure 4.8: *Lighting-by-example with 2D targets. Images in the first row are target 2D images. Images in the other rows are results derived by optimizing different 3D scenes with corresponding targets in the first rows.*

F_{mean} = 134.00; F_{grad} = 8.005 ; F_{var}= 45.026	F_{mean} = 56.00; F_{grad} = 4.681 ; F_{var}= 27.122	F_{mean} = 112.00; F_{grad} = 11.752 ; F_{var}= 65.629
F_{mean} = 144.00; F_{grad} = 9.894 ; F_{var}= 49.973	F_{mean} = 42.00; F_{grad} = 6.510 ; F_{var}= 22.780	F_{mean} = 66.00; F_{grad} = 10.354 ; F_{var}= 44.168
F_{mean} = 113.00; F_{grad} = 12.572 ; F_{var}= 49.697	F_{mean} = 28.00; F_{grad} = 7.448 ; F_{var}= 23.018	F_{mean} = 28.00; F_{grad} = 7.227 ; F_{var}= 17.751

Table 4.1: *The quantitative values of the components of the corresponding image shown in figure 4.8*

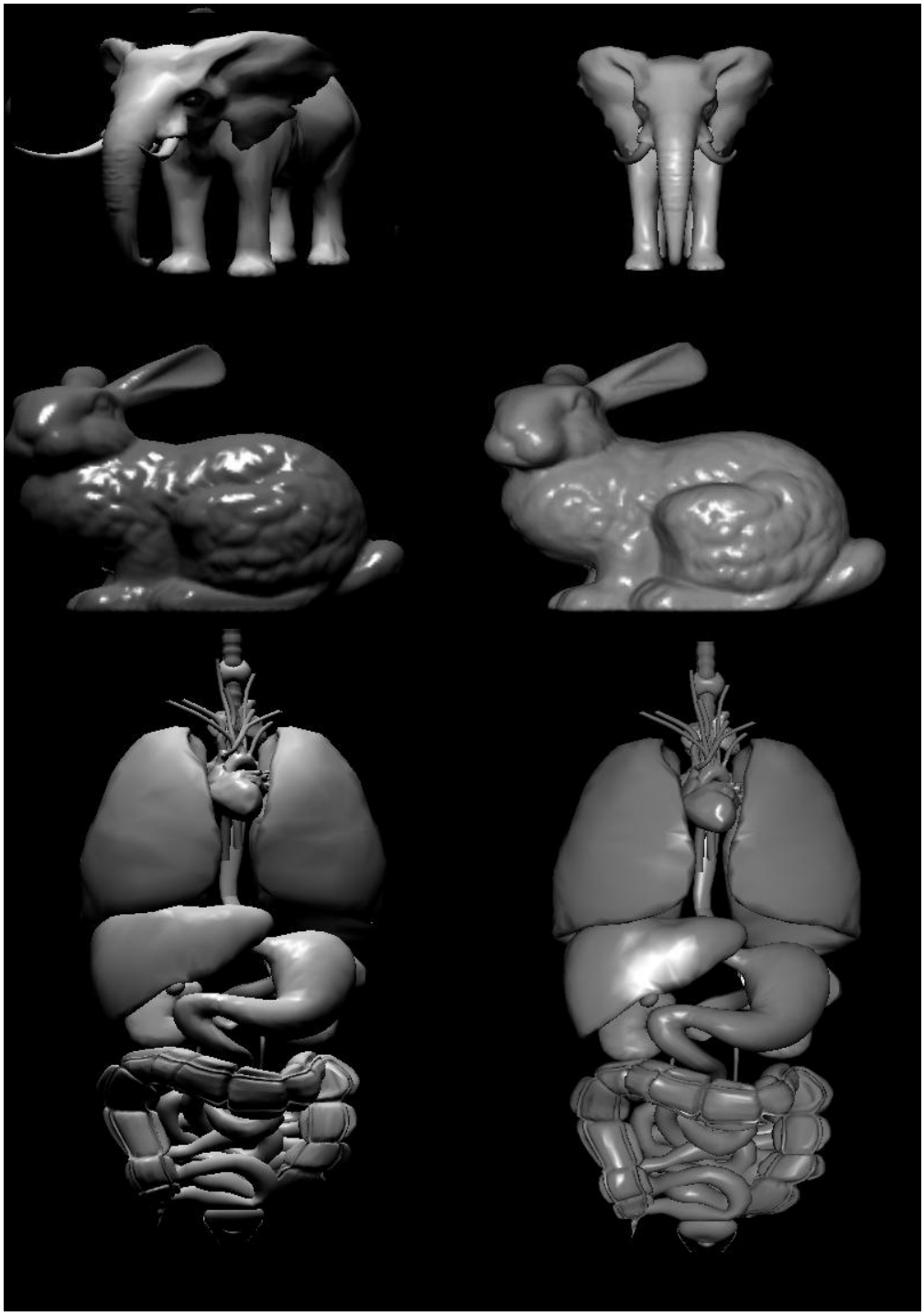


Figure 4.9: *Lighting-by-example with 3D targets.. Images in the first row are target 3D image. Images in the other rows are results derived by optimizing different 3D scenes with corresponding targets in the first rows.*

F_{edge} = 0.141 ; F_{mean} = 66.0; F_{grad} = 4.372 ; F_{var}= 23.603; F_{const}= 0.655;	F_{edge} = 0.271 ; F_{mean} = 85.00; F_{grad} = 46.508 ; F_{var}= 59.742; F_{const}= 2.8576;
F_{edge} = 0.163 ; F_{mean} = 61.0; F_{grad} = 5.131 ; F_{var}= 25.312; F_{const}= 0.578;	F_{edge} = 0.263 ; F_{mean} = 92.00; F_{grad} = 48.716 ; F_{var}= 57.672; F_{const}= 3.043;
F_{edge} = 0.152 ; F_{mean} = 72.0; F_{grad} = 3.931 ; F_{var}= 24.153; F_{const}= 0.510;	F_{edge} = 0.238 ; F_{mean} = 78.00; F_{grad} = 42.193 ; F_{var}= 62.100; F_{const}= 2.062;

Table 4.2: *The quantitative values of the components of the corresponding image shown in figure 4.9*

F_{mean} = 124.00; F_{grad} = 15.410 ; F_{var}= 70.808	F_{mean} = 167.00; F_{grad} = 4.181 ; F_{var}= 55.052
F_{mean} = 167.00; F_{grad} = 14.815 ; F_{var}= 75.955	F_{mean} = 173.00; F_{grad} = 14.565 ; F_{var}= 72.672
F_{mean} = 133.00; F_{grad} = 12.406 ; F_{var}= 83.117	F_{mean} = 155.00; F_{grad} = 12.937 ; F_{var}= 80.921

Table 4.3: *The quantitative values of the components of the corresponding image shown in figure 4.10.*

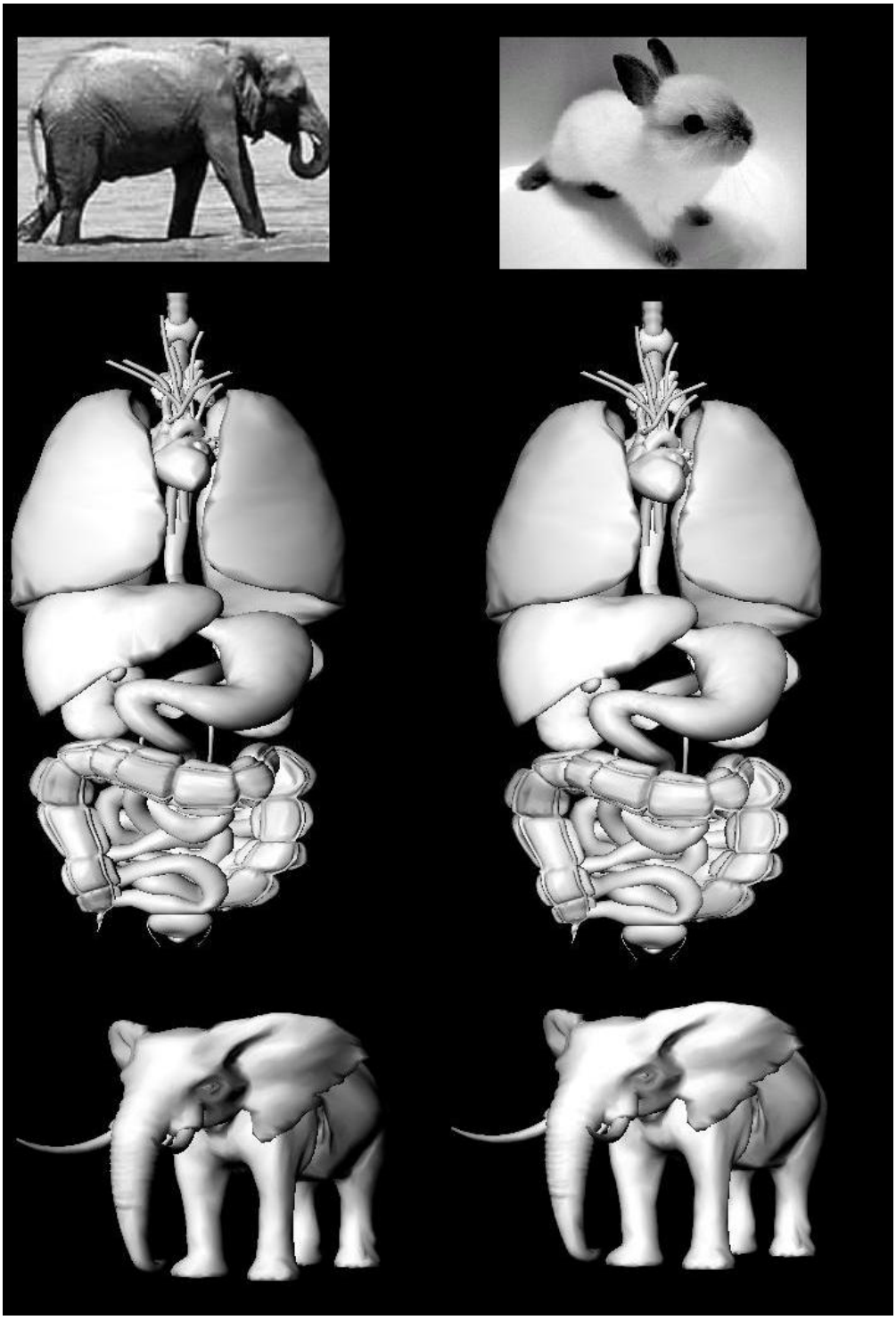


Figure 4.10: *Lighting-by-example with 2D photo targets. Images in the first row are target 2D photos. Images in the other rows are results derived by optimizing different 3D scenes with corresponding targets in the first rows.*

Chapter 5: Wavelet-Based Lighting-By-Example

5.1 Introduction

The requirements of an example-based lighting framework have been carefully analyzed in chapter 4 and existing approaches have also been discussed. The *lighting-by-example* system presented in chapter 4 was based on a perception-based framework in which the objective function comprised components motivated by research on human perception. The clear drawback of this approach is that, other than information about edge prominence, average luminance, shading gradient, histogram and contrast, spatial frequency information in the target image is not captured in the example nor transferred to the target. In other words, the spatial distribution of pixels at a certain luminance level is not captured. In reality, images that have the same edge prominence, average luminance, shading gradient, histogram and contrast properties but different spatial information are perceived very differently by viewers.

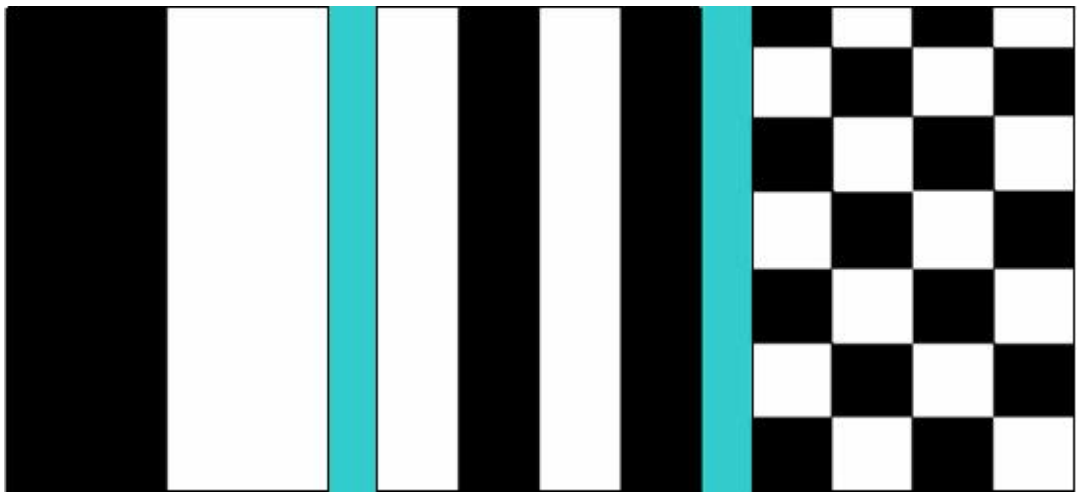


Figure 5.1: *Images with equal histogram, contrast, average luminance but very different spatial frequency distributions.*

In the three patterns shown in figure 5.1, properties such as average luminance, shading gradient and histogram are identical; however, luminance patterns are distinctly different. The difference lies in a contrast in spatial frequencies, the spatial distribution of pixels, at certain luminance levels in the images. The perception-based *lighting-by-*

example cannot capture the spatial frequencies from the target to the source except the target values for the components in the perception-based objective function. Therefore, the optimization process for perception-based *lighting-by-example* may converge to a solution in which the values for visual properties of the target are close to those of the rendered image, but their spatial distribution of luminance patterns are much different from each other, and this leads to the target and result images giving different perceptions of lighting effects to viewers. It is necessary to develop a new implementation for *lighting-by-example* which can capture the spatial distribution of luminance patterns from the target to the source.

The spatial distribution of luminance patterns is captured by the spatial frequency distribution of the image function. One mathematical tool that has been widely applied to different areas of science, especially signal processing, is the wavelet. Wavelets have been widely used to capture the energy distribution of a multi-dimensional function over both frequency domain (which can be obtained by Fourier transform), and dimensions of function space (where positional information can not obtained by a Fourier transform).

In this chapter, a wavelet-based *lighting-by-example* framework is presented. The objective function for the optimization process has been formulated on the basis of wavelet transform which can capture the spatial frequencies of the image functions from the target to the source.

5.2 Wavelet-Based Lighting Design Framework

Other than “Painting with light” [2], which actually utilizes significant user interaction, no fully automatic example-based lighting design method has to date been proposed. Furthermore, existing approaches to lighting design have concentrated on the development of metrics for optimizing the distribution of luminance on single objects so as to enhance the salient geometric features, such as edges and curvature. In practice, the illumination characteristics of a whole scene are of equal, if not greater, importance.

5.2.1 Motivations

In wavelet-based *lighting-by-example* (WLBE), 2D images, either rendered scenes or photographs, are selected as exemplars (targets), for lighting 3D scenes. During the

optimization process, the system tries to minimize an objective function that captures information in the target that is important to human visual system. A fundamental component of the workings of the HVS, in reconstructing the structure of objects, is the use of rapid relative changes in information from one region of the optic array to another. The HVS filters out low frequency temporal and spatial frequencies; indeed, in broad terms, the HVS can be considered as a spatial filter [74].

Thus, given a static image, an image function contains spatially distributed patterns of light intensity, which remain when the image is transformed to the frequency domain. Our basic observation, as to the use of target images in WLBE, is that this spatial frequency information is of primary importance. That is, the final image of a rendered 3D model derived by optimizing lighting parameters relative to a target image must have a spatial frequency distribution close to that of the target. In using an image as an exemplar, we need not only consider the relative distribution of an image function energy among spatial frequencies for the whole image function, but also the relative distribution of an image function energy over specific frequencies at specific locations in the image. To achieve this, we must convert the images (source and target) into an appropriate representation and for this we use a wavelet transform.

5.2.2 Wavelet formulation

Wavelet transforms have been used widely in signal processing [75, 76, 77]. The image function can be considered as a signal with two spatial variables. A wavelet transform of an image function provides us with information about the distribution of energy in the image function over different spatial frequencies and different subspaces in an image. Wavelet transforms aim to represent an image function by a set of basis functions. Each basis function is weighted by a coefficient; therefore, applying the same set of basis functions to different image functions results in different sets of coefficients.

The larger the coefficient of a certain basis function, the more harmonics there are that have a spatial frequency similar to the basis function at the location pertaining to that function. Therefore, by analyzing a set of the coefficients, we know which basis functions dominate an image function at a specific location in the image. In other words, we know the energy spectrum of an image function over basis functions. To achieve this,

a set of basis functions used for a wavelet transform must be designed to capture information about both the spatial and frequency properties of a signal.

The basis functions used are very application dependent. For example, in image retrieval, wavelets are widely used to characterize and model a number of image features; these include texture features [78], in which a wavelet-based texture retrieval method that is based on the accurate modelling of the marginal distribution of wavelet coefficients using generalized Gaussian density provides greater accuracy and flexibility in capturing texture information, and salient points and corners [79] in which a salient point detector was presented which is based on wavelet transform to detect global and local variations.

In this approach, the basis functions are derived by modifying the Haar basis [80, 81]. Haar basis is a set of orthogonal functions with the assumption that any function can be the linear combination of those orthogonal functions. We extend the Haar basis to create a mother function for the wavelet transform as follows:

$$B_w = \left[\phi_h(x,y) + \phi_v(x,y)\right] \tag{1}$$

Where $\phi_h(x,y)$ is the horizontal mother function for the wavelet transform defined in equation (2).

$$\phi_h(x,y) = \begin{cases} 1 & 0 \le x < 1/2 \quad and\ 0 \le y < 1 \\ -1 & 1/2 \le x < 1 \quad and\ 0 \le y < 1 \\ 0 & otherwise \end{cases} \tag{2}$$

$\phi_v(x,y)$ is the vertical mother function for the wavelet transform defined in equation (3).

$$\phi_v(x,y) = \begin{cases} 1 & 0 \le y < 1/2 \quad and\ 0 \le x < 1 \\ -1 & 1/2 \le y < 1 \quad and\ 0 \le x < 1 \\ 0 & otherwise \end{cases} \tag{3}$$

Scaling and shifting (2) and (3) yields the set of basis functions represented in equations (4) and (5). Scaling changes the frequency of the mother function to obtain basis functions at different frequencies. Shifting changes the position of the mother function. A combination of scaling and shifting operations will create basis functions at different frequencies and different locations.

$$\phi_h^{\,a,b}(x,y) = \phi_h\left(2^{b_h}x - a_h, 2^{b_v}y - a_v\right) \tag{4}$$

$$\phi_v^{a,b}(x,y) = \phi_v\left(2^{b_h}x - a_h, 2^{b_v}y - a_v\right) \quad (5)$$

b_h and b_v are scaling factors for the horizontal and vertical basis functions, respectively. a_h, and a_v are shifting factors for the horizontal and vertical basis functions, respectively. We can rewrite (1) as follows:

$$B_w^{a,b} = \phi_h^{a,b}(x,y) + \phi_v^{a,b}(x,y) \quad (6)$$

We can define ${}_iE^{a,b}$ to be the normalized energy spectrum coefficient of a wavelet transform of the image function *I(x,y)* corresponding to i^{th} basis function $B_w^{a,b}$ where:

$${}_iE^{a,b} = {}_iE_h^{a,b} + {}_iE_v^{a,b} \quad (7)$$

${}_iE_h^{a,b}$ *and* ${}_iE_v^{a,b}$ are the normalized energy spectrum coefficients of a wavelet transform of the image function *I(x,y)* respectively, corresponding to the horizontal and vertical components of the basis function $B_w^{a,b}$ where:

$${}_iE_h^{a,b} = \frac{2\times 2^{(b_h+b_v)}}{W\times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} I(x,y)\phi_h^{a,b}\left(\frac{x}{W},\frac{y}{H}\right) \quad (8)$$

$${}_iE_v^{a,b} = \frac{2\times 2^{(b_h+b_v)}}{W\times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} I(x,y)\phi_v^{a,b}\left(\frac{x}{W},\frac{y}{H}\right) \quad (9)$$

Where *I(x,y)* is the intensity image function, and *W* and *H* are the width and height of the image respectively.

Finally, we have a set *E* of normalized energy spectrum coefficients for a wavelet transform of the image function *I(x,y)* as follows:

$$E = \{ {}_iE^{a,b} \}, \; i = 0,1,\ldots,N-1 \quad (10)$$

Where *N* is the number of basis functions.

```
Integer current_depth;
Boolean BuildQuadTree(WLNode *pNode,
 Window wnd)
Begin
  Window wnd1,wnd2,wnd3,wnd4;
  current_depth = current_depth+1;
  If(current_depth > maxdepth)
   Return TRUE;.
  Calculate wavelettransform for wnd;
  Save energy to pNode->fenergyFreq ;
  Divide wnd into 4 equal quandrants;
  wnd1 = lower-left quadrant of wnd;
  wnd2 = lower-right quadrant of wnd;
  wnd3 = upper-left quadrant of wnd;
  wnd4 = upper-right quadrant of wnd;
  BuildQuadTree(pNode->child[0],wnd1);
  BuildQuadTree(pNode->child[1],wnd2);
  BuildQuadTree(pNode->child[2],wnd3);
  BuildQuadTree(pNode->child[3],wnd4);
  current_depth = current_depth-1;
  Return TRUE;
End
```

Figure 5.2: *Recursive wavelet transform implementation.*

5.3 Wavelet Transforms & Lighting Design

5.3.1 Wavelet-based objective function

Lighting-by-example aims to allow us to set up lighting parameters, comprising positions and specular and defuse intensities of lights, in order to recreate the illumination apparent in a selected exemplar (the target). Lighting effects can be considered as a distribution of different light intensity levels over different locations in images and ratios of average luminance between regions of different granularity.

As described in section 5.2, a wavelet transform captures information about the distribution of energy of image function over different frequencies at different locations

in an image function. Each wavelet transform of an image function is characterized by a set of normalized energy spectrum coefficients. The optimization process tries to make the set of energy spectrum coefficients of the rendered image close to that of the selected exemplar.

The Euclidean distance is used to measure the distance between two sets of energy spectrum coefficients. In other words, the optimization process tries to find a set of lighting parameters such that the distance between two sets of energy spectrum coefficients, those of the rendered image and those of the selected exemplar, is as small as possible.

Suppose we have a set of target values derived from the wavelet transform of the target image $T = \{t_i\}$, $i = 0, 1, .., N-1$. The objective function is given by:

$$F\left(\theta_k, \varphi_k, I^s{}_k, I^d{}_k, R_k\right) = \frac{1}{N}\sqrt{\sum_{i=0}^{N-1}\left({}_i E^{a,b} - t_i\right)^2} \qquad (11)$$

$F(\theta_k, \varphi_k, I^s{}_k, I^d{}_k, R_k)$ is the objective function.

θ_k is the elevation angle of k^{th} light.

φ_k is the azimuth angle of k^{th} light.

$I^d{}_k$ is the diffuse intensity of k^{th} light.

$I^s{}_k$ is the specular intensity of k^{th} light.

R_k is the distance k^{th} light (fixed for directional lights).

N is the number of basis functions.

5.3.2 Parameter selection

The shifting and scaling parameters (a_h, a_v), (b_h , b_v) of the wavelet transform are selected, such that at each spatial frequency the whole image is scanned without using overlapping sections of windows for each basis function. A full quadtree is used to represent normalized energies derived by a wavelet transform of an intensity image function $I(x,y)$. Nodes at each level of the quadtree represent normalized energy spectrum coefficients at a spatial frequency, at different locations in the image, as specified by the shifting parameters a_h, a_v.

The number of basis functions can be determined by the maximum spatial frequency used for wavelet transform. The highest spatial frequency is calculated using a *2×2* pixel window. The higher the maximum spatial frequency used for wavelet, the greater the computational complexity. Each node of a quadtree is defined as a structure:

```
struct WLNode{
  Real fenergyFreq;
  Window wind;
  WLNode *child[10];
};
```

Where `fenergyFreq` is the normalized energy spectrum coefficient of a wavelet transform of the image function corresponding to a basis function, and `wind` is the location of the basis function. The wavelet transform is implemented recursively (see the pseudo-code in Figure 5.2). Given an input image, the algorithm will construct a full quadtree, and populate each node of the quadtree with information as to the energy of a specific spatial frequency at a specific location of the image function.

5.4 Results and Discussion

The system has been tested on a range of 3D scenes, and Figure 5.3, Figure 5.4, and Figure 5.5 present three examples lit using this approach, under local illumination, utilizing two lights and different exemplars (target images). In broad terms, the results demonstrate that spatial information such as the ratio and positions of light and dark regions in the exemplar images are replicated in the rendered images. Note that, as in these example, the highest spatial frequency used in the wavelet transform was a *2×2* pixel window, thus the wavelet transform captures information about shading gradients.

We have tested different 3D models with target images that are shots from well-known movies[1] and sample results of using a simulated annealing optimization scheme are shown in Figure 5.3, Figure 5.4, and Figure 5.5. In Figure 5.3 the scene to be lit includes smooth un-textured objects with planar background polygons, Figure 5.4

[1] *Entrapment* (© 20th Century Fox); *Cat Woman* (© Warner Brothers); *Secret Window* (© Columbia Tristar Films).

contains mostly planar background objects with no foreground features, and Figure 5.5 contains a mixture of foreground and background objects with mid-range geometric complexity typical of a 3D computer game. The spatial distribution of image function energy over different spatial frequencies in the exemplars is clearly replicated in the optimized results. Figure 5.6 to Figure 5.14 show the convergence of the distance function between the sources and the targets of the example shown in Figure 5.3, Figure 5.4, and Figure 5.5.

In row 3 of Figure 5.3, and row 2 of Figure 5.4, we notice that the upper regions of both the exemplar and the optimized result are significantly darker than the lower half. This demonstrates the ability of this approach to capture the distribution of image function energy over low spatial frequencies in the vertical direction. Likewise, a close viewing right-hand-side region of the images in rows 1 and 3 of Figure 5.4, and row 2 of Figure 5.5 reveals a replication of this effect in the horizontal direction, mirroring the low frequency distribution of luminance in the target scenes.

A different effect can be seen in rows 1 and 3 of Figure 5.4 in which small dark and bright regions alternate in both the exemplars and the optimized results. This can be explained by the fact that the distribution of image function energy over higher spatial frequencies in the exemplars is replicated in the optimized result. The examples show that the dark and bright regions of the exemplar are not exactly mapped to the optimized result, in terms of their precise spatial locations, but the relative distribution of dark and bright areas as well as luminance of each area are captured.

The highest spatial frequency used in wavelet transform utilizes 2×2 pixel window. Here, the wavelet transform captures information about shading gradients and, as illustrated by Figure 5.5 (the only scene with a significant number of geometric features), and also Figure 5.3, the shading gradient information of the exemplar images is replicated (particularly in row 1 of Figure 5.3, and row 1 of Figure 5.5).

5.5 Chapter Conclusion

In the example-based relevant approaches such as “Painting with light” [2], users directly paint on the surface of the objects, which causes changes in the reflectance function. The drawback of this approach is that it requires a 3D model and users must

knows exactly which part of the scene should be painted with certain lighting effects. In the perception-based *lighting-by-example* presented in chapter 4, the system cannot capture the spatial distribution of the lighting effects from the target to the source. In this chapter, we have presented the wavelet-based *lighting-by-example* approach to lighting design. A wavelet-based objective function was used at the core of the approach to capture the distribution of spatial frequencies over an image function. A set of basis functions for the wavelet transform has been developed, specifically targeted to this application.

As the examples demonstrate, both high and low spatial frequency effects (not just overall luminance levels) are important factors in establishing the mood of a scene – and such high level notions have to date been beyond the scope of automated lighting design systems. The use of wavelet in this example-based approach was intended to capture the spatial frequencies or the spatial distribution of lighting patterns from the targets to the sources. The spatial frequencies of the image function also carry information about shading and the texture. The initial positive results have demonstrated the efficiency of the wavelet in this application.

Figure 5.3: *Test scene comprising geometric objects and planar walls, ceiling and floor.*

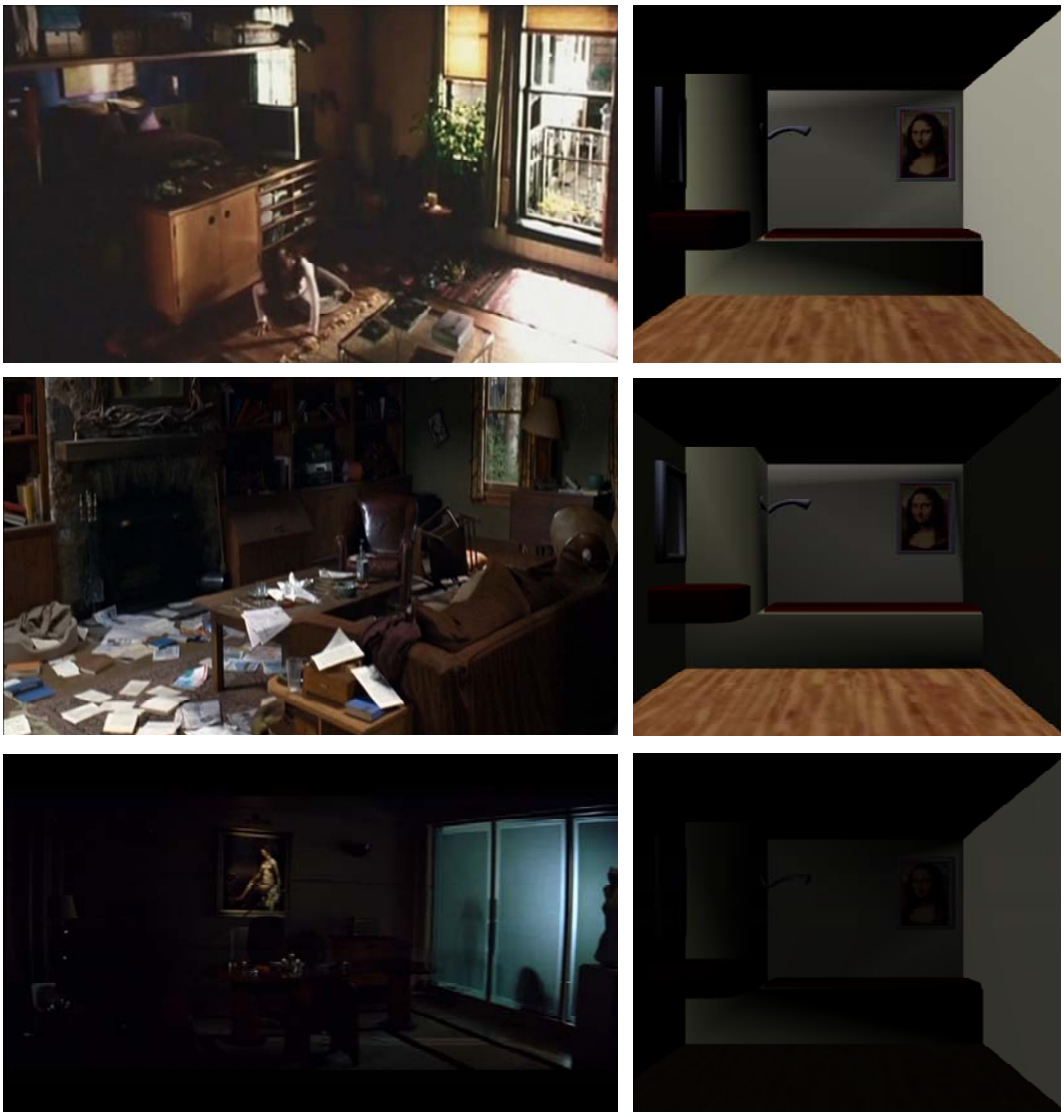
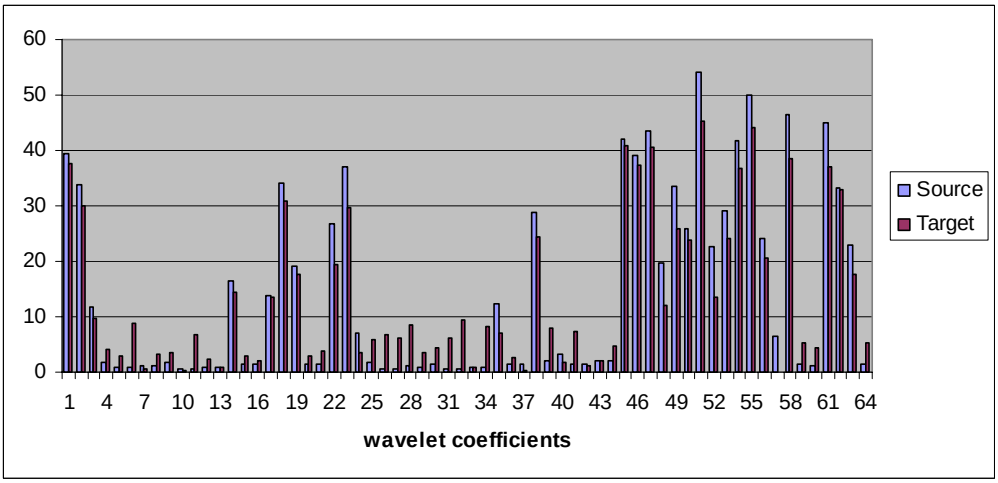


Figure 5.4: *Test scene comprising simple geometric objects and planar walls.*

Figure 5.5: *Test objects of moderate geometric complexity and non-planar background objects (27706 polygons).*

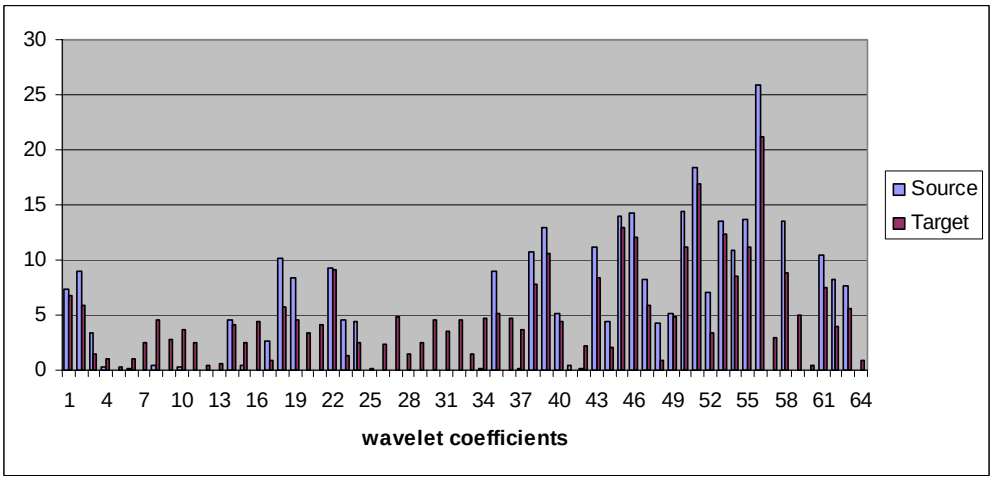
(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of the target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.6: *Characteristics of the example shown in row 1 figure 5.3.*

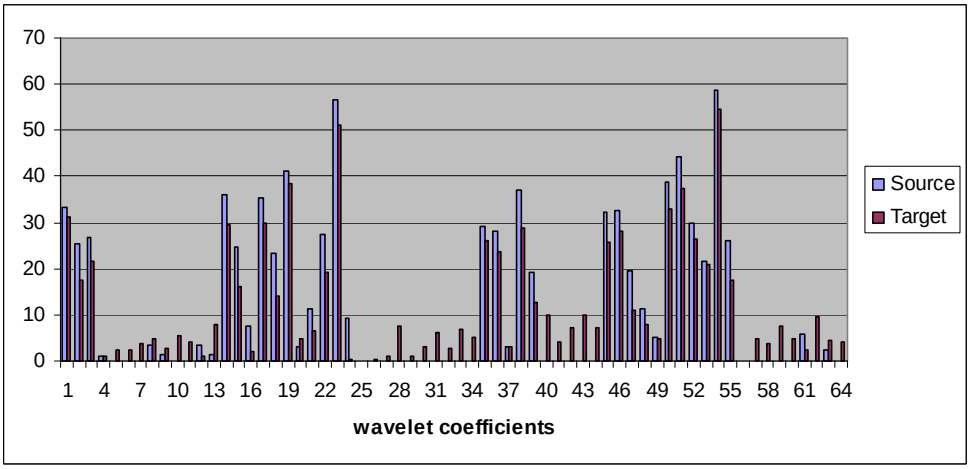
(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

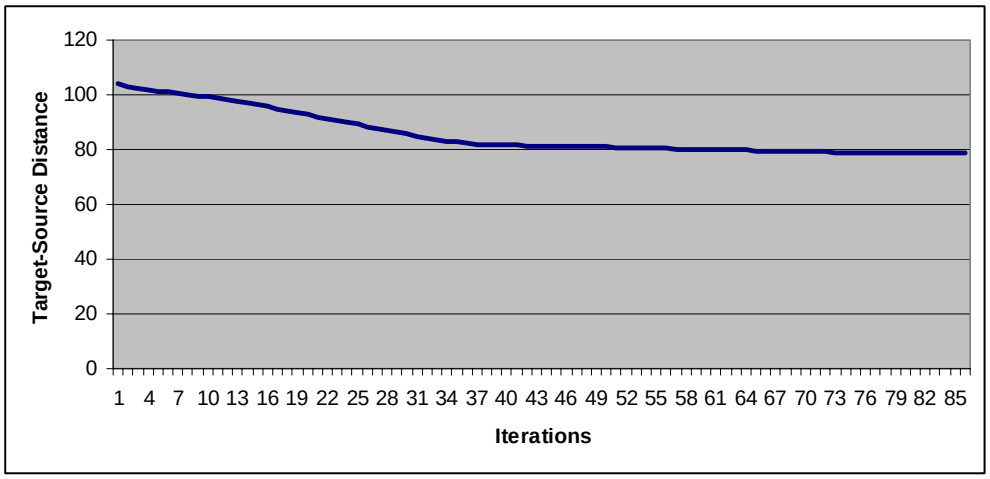
Figure 5.7: *Characteristics of the example shown in row 2 figure 5.3.*

(a) The convergence of the distance between target and source

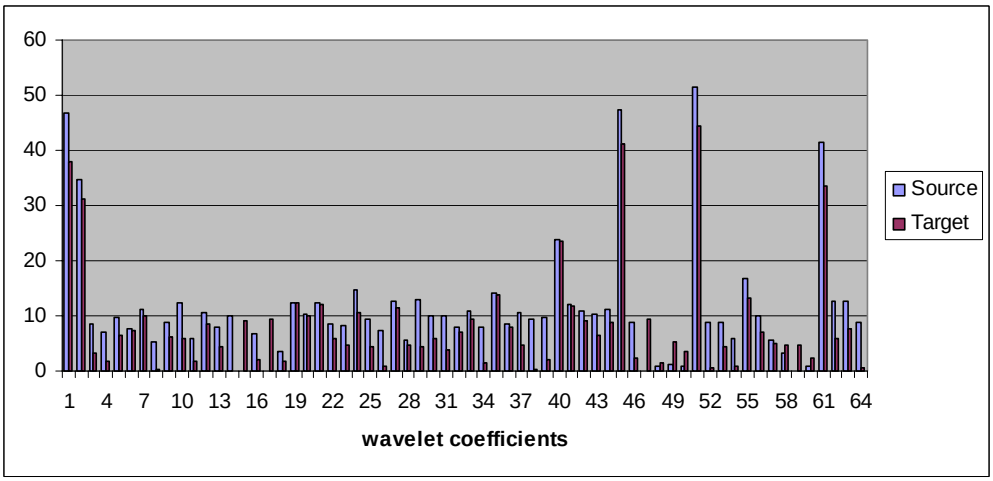


(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.8: *Characteristics of the example shown in row 3 figure 5.3.*

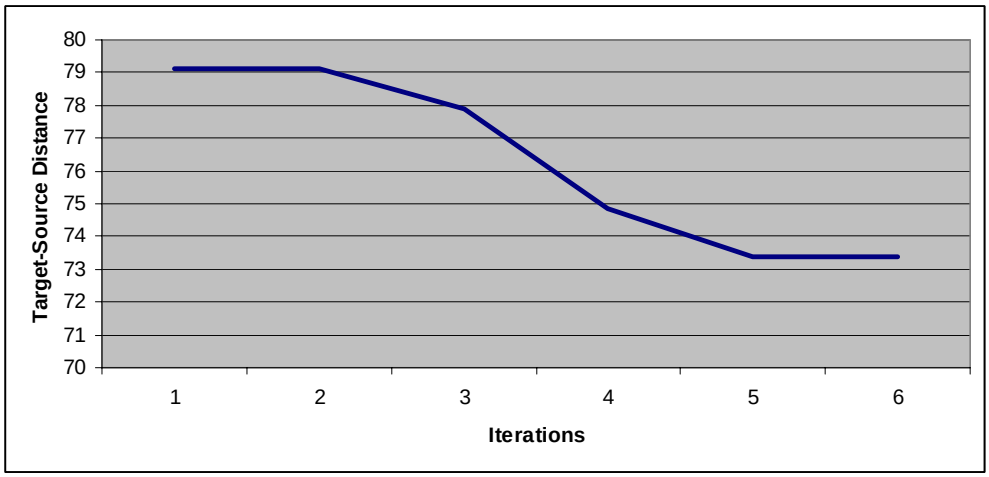


(a) The convergence of the distance between target and source

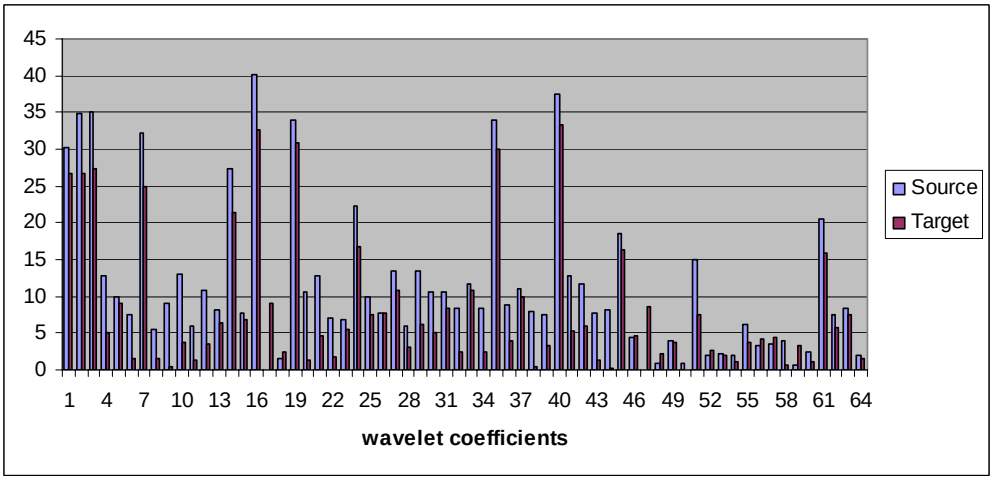


(b) The first 64 wavelet coefficients of target and optimized source image, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.9: *Characteristics of the example shown in row 1 figure 5.4.*



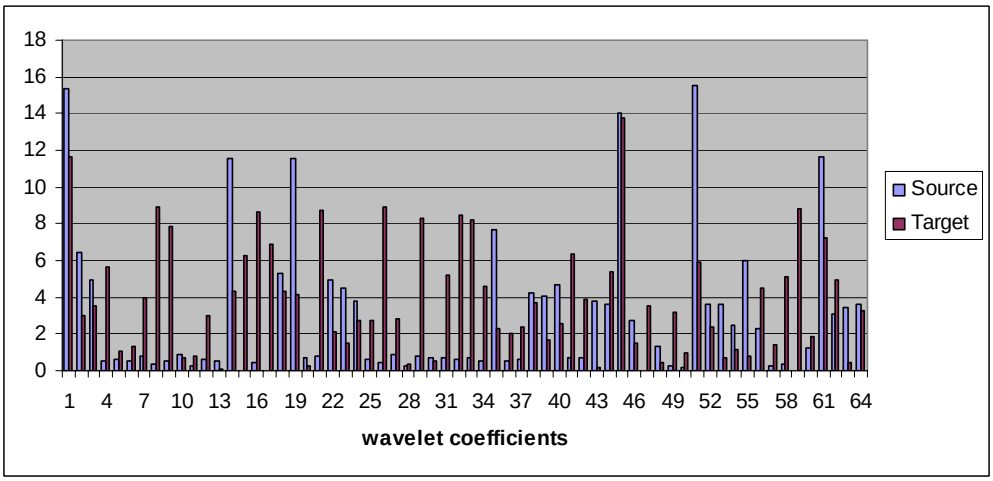
(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.10: *Characteristics of the example shown in row 2 figure 5.4.*

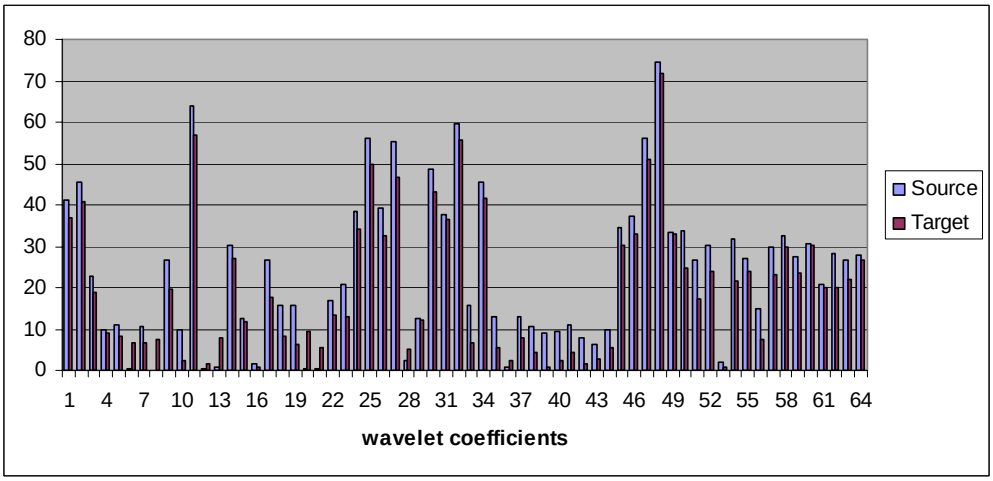
(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

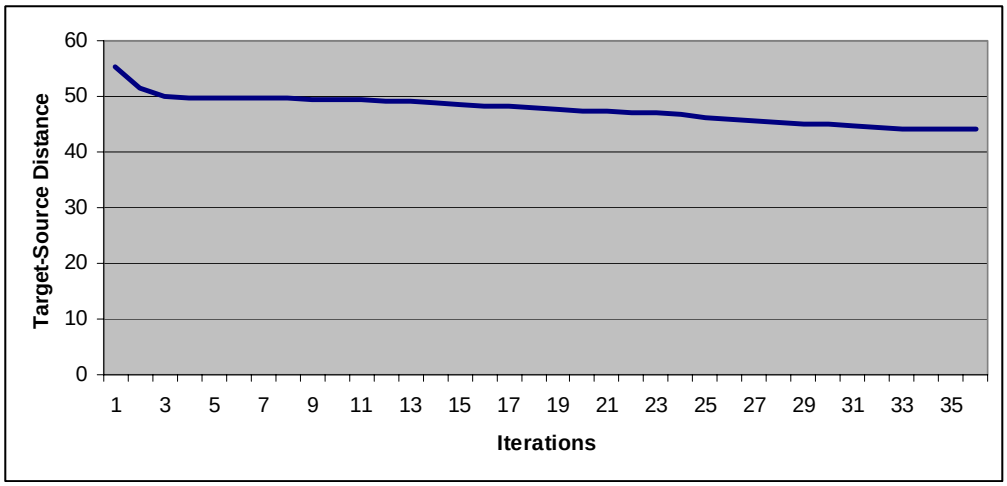
Figure 5.11: *Characteristics of the example shown in row 3 figure 5.4.*

(a) The convergence of the distance between target and source

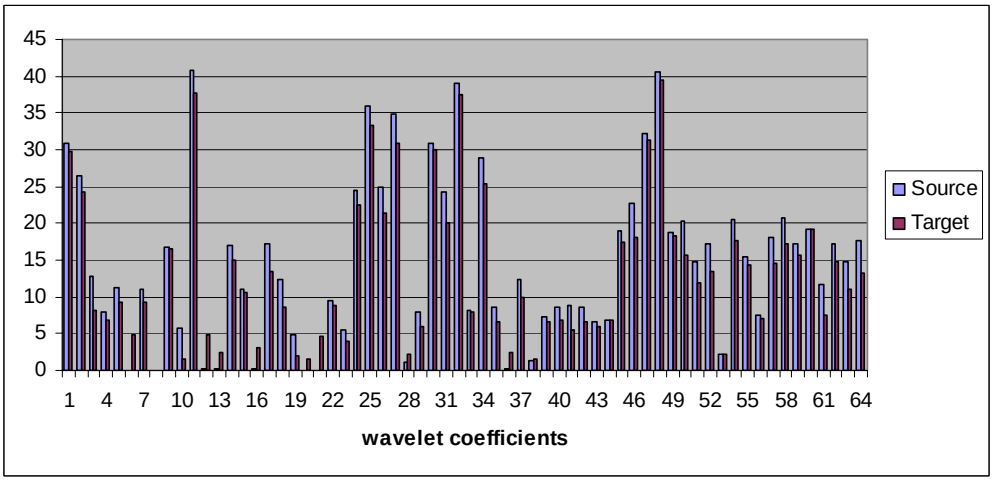


(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.12: *Characteristics of the example shown in row 1 figure 5.5.*



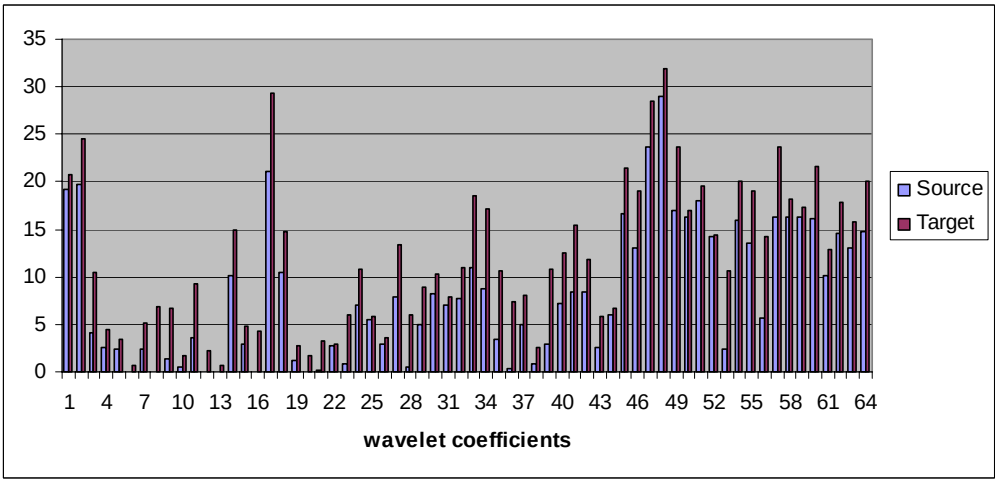
(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.13: *Characteristics of the example shown in row 2 figure 5.5.*

(a) The convergence of the distance between target and source



(b) The first 64 wavelet coefficients of target and optimized source images, the vertical axis is the normalized value of the wavelet coefficients.

Figure 5.14: *Characteristics of the example shown in row 3 figure 5.5.*

Chapter 6: Interactive Lighting Design

Since the main focus of this research has been on the automation of lighting design for scenes as a whole, the question how we might control and specify of lighting as an interactive process has not yet arisen. This chapter reviews the requirements of an interactive lighting design system, and presents an extension of our scene-based optimization framework which allows users to declaratively specify the lighting of individual objects.

6.1 Interactive Lighting Design

Conventional lighting design is a repeated process of manipulating lighting parameters and testing the changes in lighting effects; the lighting process stops when desired lighting goals have been achieved. Apparently, conventional lighting design is a knowledge-based process rather than a random process, as experienced users would know how to adjust light parameters better than inexperienced users. Manipulating scene parameters in general, and lighting parameters in the lighting design process in particular, requires heavy interactions between users and graphic tools. Hence, interactions in graphic tools have been continually improving in order to equip users with convenient ways of interactions [86].

In graphic tools such as 3D Studio Max, lighting work starts with identifying the types of light sources and their characteristics that will be used in a scene. By investigating the purpose and intent of lights, a user finds a real-world counterpart for each of every light used in the scene. Lights are then positioned at intended locations. The next step will be editing the properties of the lights. In commercial graphic tools, most of the light properties such as intensities, colour, attenuation and shadow parameters are manipulated through a window using a keyboard. For some properties of lights, such as light position and direction, there is an alternative for specifying them through mouse-based interactions, such as dragging and dropping. Indeed, mouse-based interactions for light positioning would be more efficient and intuitive than specifying the values through a window using a keyboard, as users normally know relatively where the light should be and which direction the light should point at rather than the exact

values for light position and direction. In reality, commercial graphic tools such as 3D Studio Max, Maya and Light Wave support mouse-based interactions for manipulating light position and direction. Where mouse-based interactions can not be easily applied, such as specifying values for falloff and hotspot of lights, context menus are normally used in graphic tools to enhance the performance of accessibility to functionalities.

Conventionally, shadows and highlights in the lighting design process are normally computed on the basis of the positions of lights used in the scene. In other words, light positions should be specified prior to the step of designing highlights and shadows. There was another approach to designing positions of lights that was based on shadows and highlights. In this case, shadows and highlights must be designed prior to the process of specifying light positions. An interactive sketch-based interface has been developed by Poulin et al [10] that enabled users to automatically design light positions according to the desired shadow areas sketched with a mouse pointer. The sketching process starts with selections of an object to cast a shadow and a light. A user sketches strokes on a 2D image plane projected directly from a 3D scene; the light position is recalculated at every stroke. An objective function was defined in such a way that the shadow region for a computed point light (and also some extended light geometries) bounds the sketched regions as tightly as possible. The objective function is based on distance between sketch points and the perspective position of a light, which is maximized in optimization process. Sketch-based interaction is very user friendly, as it allows users to express their desired goals in a natural way rather than using rigid forms. For instance, drawing a curve using sketch-based interactions would be more natural than using limited types of curve described by mathematical equations provided by the system. Of course, the arbitrary sketch-based expressions are eventually approximated by mathematical models in the system. However, users have the chance to express the desired goals in a natural way.

The ultimate goal of research on human-machine interactions is to empower users with convenient ways of achieving the desired goals with less effort using limited resources. In particular, interactions in lighting design aims to provide users with convenient ways of achieving desired lighting effects by taking advantage of the combination of existing lighting design approaches.

6.2 Interactive Design Through Independent Object Lighting

Interactive lighting design is problematic as, although we may have a clear notion of how we want each individual object in a scene to be lit, any change in lighting parameters affects the illumination of all objects simultaneously. One solution, which is already a common practice in graphic design, is to produce 2D images from 3D scenes and use 2D image processing techniques to merge different source images (i.e. source images that have been differently lit or have been manipulated in some way). There are a number of drawbacks to working with 2D images; in particular, 3D information is not taken into consideration. Indeed, when 2D images are merged together without taking advantages of 3D information, unanticipated effects may occur (for example, if no account is taken of object occlusion).

In fact, creating lighting effects, either using 2D compositing or by modifying the parameters of scene lights directly, is not an intuitive process, even for experienced graphic artists. To address this problem, we propose the extension of the work presented so far. By integrating the optimization schemes developed in chapters 3, 4 and 5 (for both *perception-based lighting* and *lighting-by-example*) we have developed an interface that allows users to design the scene lighting on an object-by-object basis. Developing an interactive approach requires us to extend our optimization framework and develop mechanisms by which we can more or less independently specify and modify the lighting for different objects in a scene.

Two methods are proposed which can be contrasted in two respects: the degree to which they treat each object in the scene independently, and the nature of the result of the lighting process. Both approaches take as a starting point (if desired) either the results of a perception-based lighting optimization process, or a lighting-by-example process. In this way, we envisage *interactive lighting design* as a matter of fine tuning the illumination of each object in a scene. In the first approach, 3D Interactive Lighting Design (3D-ILD), the outcome of the lighting design process is a 3D scene in which objects are lit differently using spotlights, as is commonly used in interactive graphics applications. In the second approach (2D-ILD) a set of lights that is specific to each object is optimized, and the visual results of these separate processes are combined (utilizing 3D depth information) in the seamless generation of a 2D image.

6.3 3D Interactive Lighting Design (3D-ILD)

For most graphics applications, we anticipate that the result of a lighting design system is a configuration of lights with respect to the scene elements. Though this design process takes place by reference to a single viewpoint (that is, the optimization process), the viewpoint selected is typically one that is characteristic or exemplifies the contexts in which an object is likely to be viewed. Thus the resulting lighting configuration can be integrated into a scene which includes other objects and light. We have realized this 3D interactive lighting design process through the use of spotlights which are used to light each object individually. A spotlight provides localized and directional illumination. Illumination is restricted to within a specified cone (i.e. the beam of light). This beam of light can be controlled through the specification of both width of cone and the nature of the drop-off function.

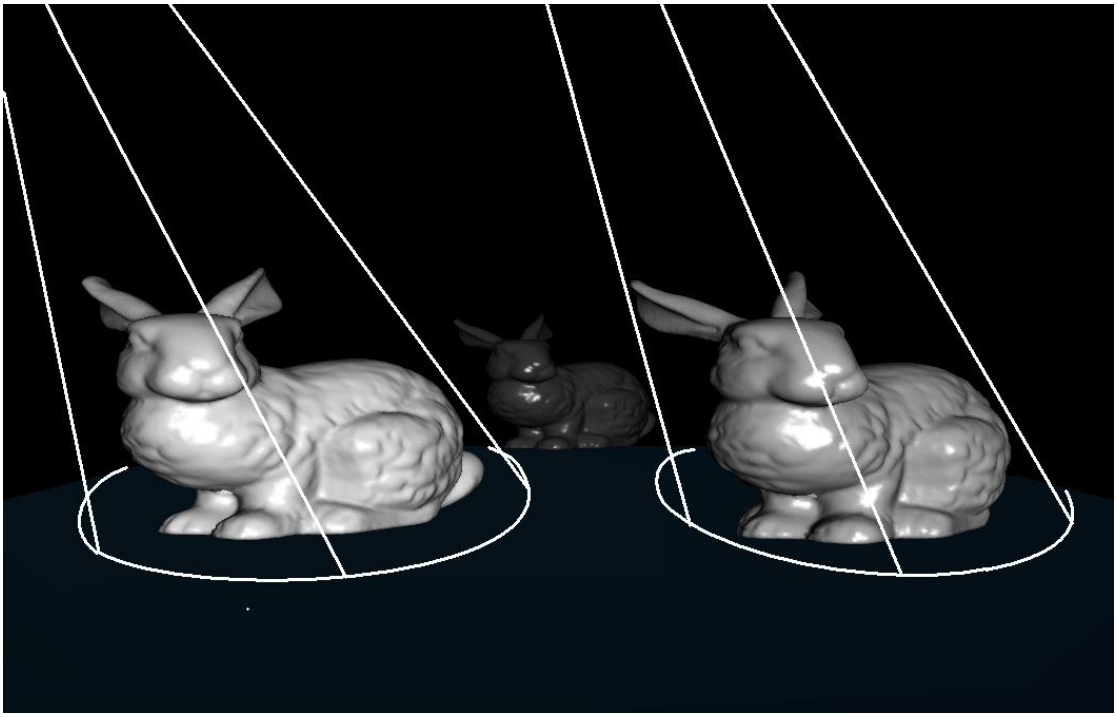


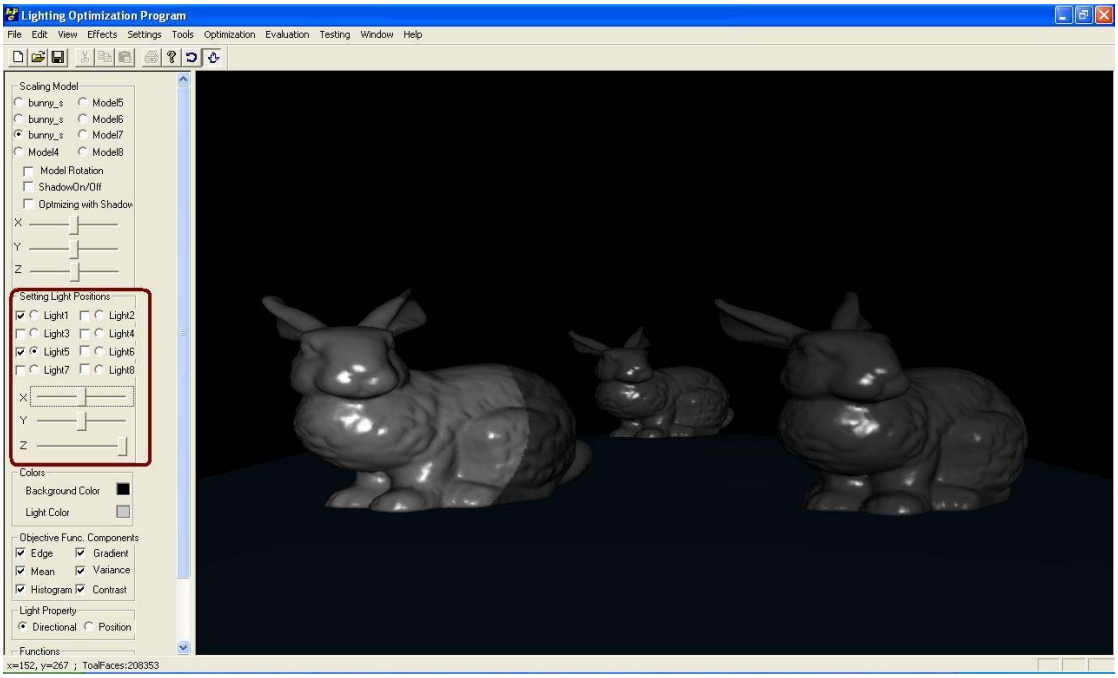
Figure 6.1: *Each object is lit by a spotlight with an appropriate focus cone.*

Figure 6.1 illustrates the idea of using spotlights to optimize lighting for objects separately. In this scenario, there are three objects in the scene and, for simplicity, two spotlights are used to light the leftmost and rightmost objects, and an ambient light was

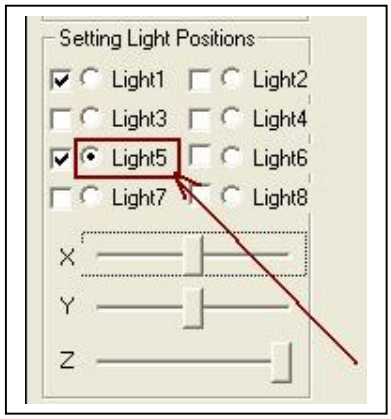
used to create the background illumination. The cone around each object represents the focus of a spotlight. Apparently, the cone parameters must be adjusted such that the whole object is in the cone of the spotlight.

The first step of the interactive design process involves setting up appropriate parameters for the spotlight, the position, orientation, and cut-off angle, such that the cone of the spotlight focuses on the object of interest without lighting adjacent objects (see figure 6.2(a)). This is sometimes impossible due to the close proximity of other objects, and occlusions, in the 3D scene. In such cases, the user simply has to adjust the light so as to keep lighting on unintended objects to a minimum. Once the parameters of spotlights for different objects have been set up, the next step is to select the spotlight to be optimized (see figure 6.2(b)).

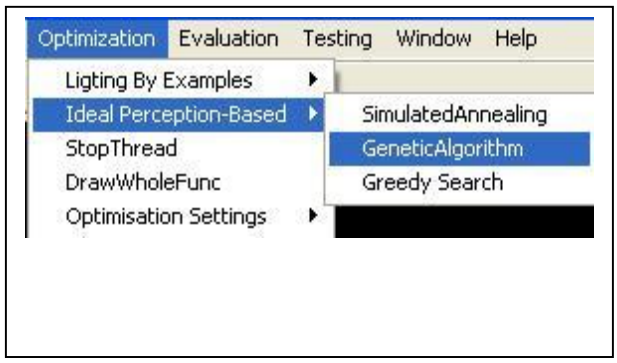
The final step is to choose the optimization technique to be employed. This can be either ideal *perception-based lighting optimization* (see chapter 3), which will optimize the lighting parameters of the spotlight such that the visual properties of the objects in the scene are maximized, or *perception-based lighting-by-example* (see chapter 4) which aims to capture the lighting effect of a 3D example and recreate it for the current object (see figure 6.2(c)). Our current experimental implementation supports up to six spotlights. Figure 6.3 shows an example in which lighting for objects are separately optimized using spotlights. Figure 6.3(a) shows the original scene, for which there is a default direction lighting originating from behind and above the camera. Figure 6.3(b) shows the result of applying the ideal perception-based lighting optimization approach to the leftmost bunny rabbit. Figure 6.3(c) shows the example object (an elephant) that is the target for lighting the rightmost bunny rabbit using perception-based lighting by example. Finally, figure 6.3(d) shows the final result for both processes. The figures demonstrate how the objects can be independently lit in a 3D scene without significantly affecting the lighting of the other objects in the scene.



(a) The leftmost object is not completely in the cone of the spotlight; users use light control interface (controls in the red square) to change spotlight parameters such that the cone of the spotlight covers the whole object



(*b) Select the light to be optimized*

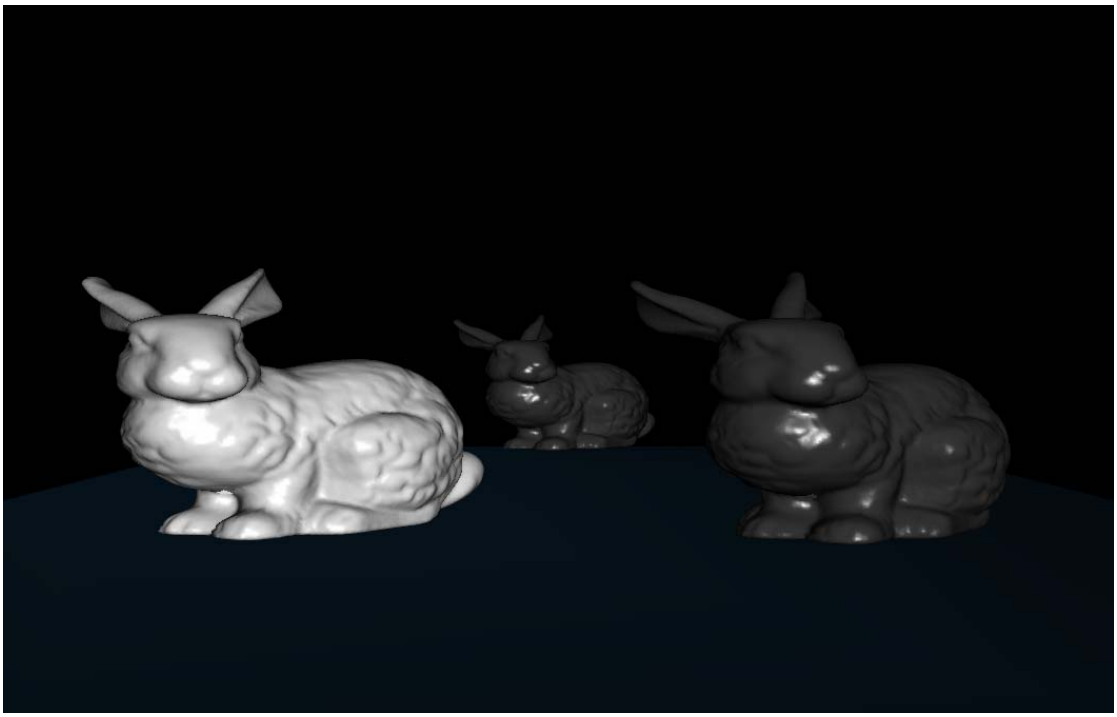


(*c) Select the optimization technique*

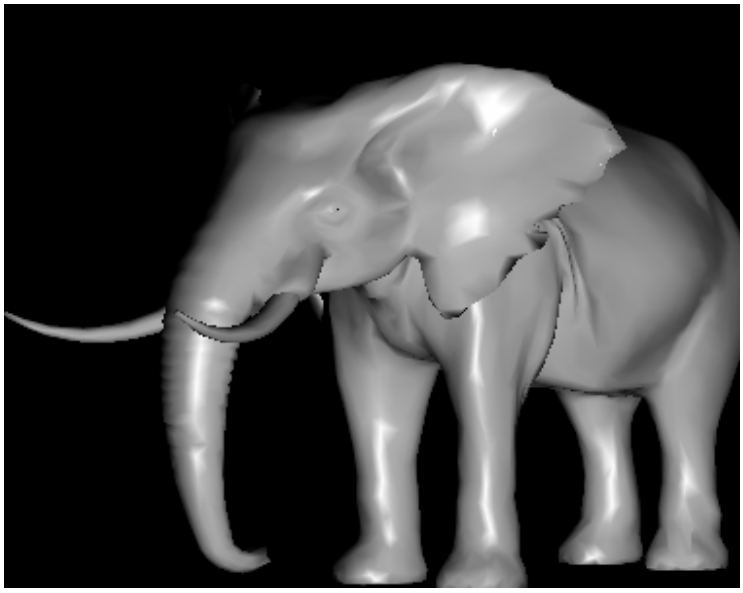
Figure 6.2: *The interface for optimizing lighting parameters for a scene using spotlights.*

The optimization steps of the interaction took 15 steps (93 seconds on a Windows PC, with P4 3.00GHz processor, 1G RAM and GeForce 7600 GT graphics card) for the ideal perception-based lighting optimization and 28 steps (154 seconds on a Windows PC, with P4 3.00GHz processor, 1G RAM and GeForce 7600 GT graphics card) for the perception-based *lighting-by-example* optimization.

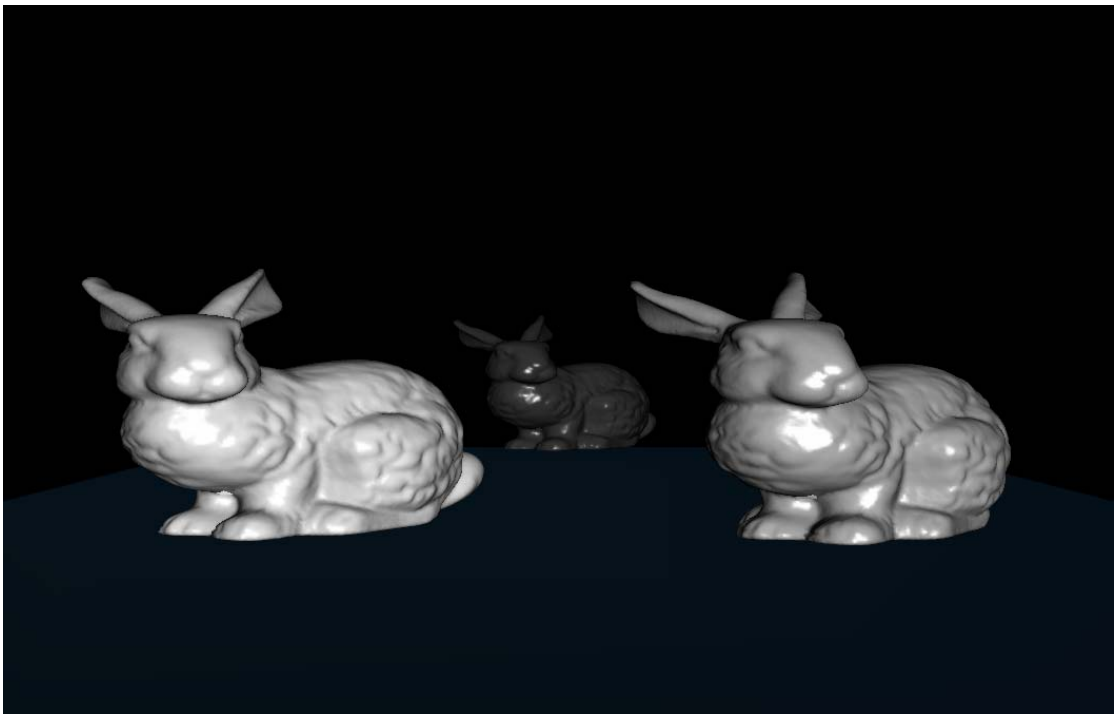
6.3(a) The original lighting setup



6.3(b) The result of applying the ideal perception-based lighting optimization approach to the leftmost bunny rabbit



6.3(c) The example object that is the target for lighting the rightmost bunny rabbit using perception-based lighting-by-example



6.3(d) The final result after both processes

Figure 6.3: *An example of lighting parameters for objects are separately optimized using spotlights.*

6.4 2D Interactive Lighting Design (2D-ILD)

Where the result of the lighting design process is a 2D image (for example, in graphic design for static digital media and print) we can design the lighting for each object in the scene by optimizing each object individually. Figure 6.4 illustrates this process, by which the other objects in a scene are hidden and the techniques that we developed in chapters 3 and 4 are deployed to optimize the lighting for a single object. Having optimized each object separately, the resulting 3D scenes are merged into a final (2D) composition using depth information, which is retained from the 3D renderings. The final image buffer is displayed in a separate window.

Figure 6.5 is a diagram that illustrates the use of an interactive interface to individually optimize lighting parameters for objects. Firstly, the objects of the 3D scene are loaded and the graphic designer then uses a context menu to hide the objects that are not to be optimized in the current loop. The objects for which lighting parameters are to be optimized in this loop remain visible. At this point the user can also manipulate the visible objects with operations, such as rotating, scaling and moving the objects. When the user is satisfied with the current set-up of the objects, he can proceed to the next step of choosing an appropriate lighting optimization, either ideal perception-based lighting, perception-based *lighting-by-example* or wavelet-based *lighting-by-example*. The optimization for the current setup of objects proceeds and the colour buffer is then merged to the final image buffer. The user reveals all the invisible objects, and if lighting parameters have been optimized for all objects then the optimization process stops. If not, the user will start to optimize lighting parameters for the next object and new objects may also be loaded to the scene. Figure 6.6 shows some results of the lighting optimization using interactive interface for the case in which the outcomes are 2D images.

Figure 6.7 shows a worked example in which the lighting for each of the faces on a two-headed bust is separately designed and then recombined into a final image. In this process a context menu for an object provides a number of options:

1. *Hide This Object*: hide the currently selected object, to allow the optimization of the remaining (visible) objects in the scene.

2. *Hide All Objects*: hide all objects of the 3D scene, used when a user is about to load a new object to the scene and lighting parameters are only to be optimized for the newly loaded object.
3. *Show All Objects*: show all objects of the 3D scene, used when the user wants to view the set-up of the whole 3D scene.
4. *Merge With Final Image:* merges the colour buffer with the final image buffer using available depth information to ensure that objects of the two buffers are merged in the right depth order.

Figure 6.7(a) shows the selection of the right head as the object to hide. In figure 6.7(b) the ideal perception-based framework is deployed, and in figure 6.7(c) the colour and depth information is merged in the final 2D image (and a separate depth buffer). Figure 6.7(d) shows the selection of the left head as the object to hide. As the perception-based *lighting-by-example* framework is deployed, figure 6.7(e) shows the exemplar, and figure 6.7(f) shows the result for the right head. Figure 6.7(g) shows the final merged image in which the spatial information for the scene is preserved, but the two heads have been lit very differently.

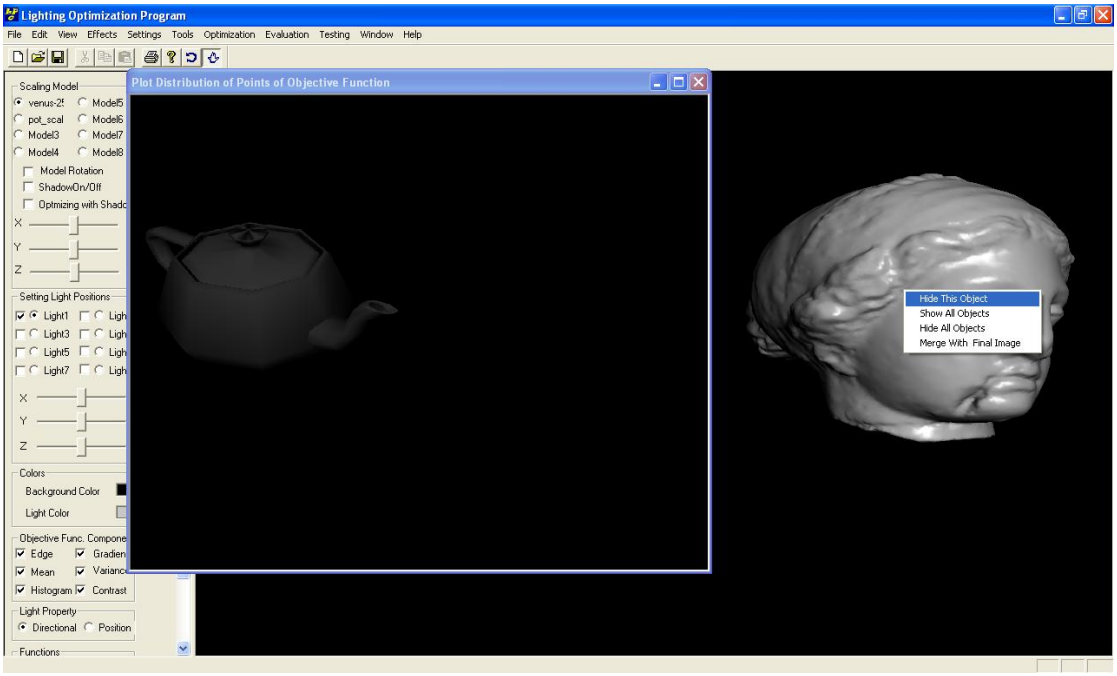


Figure 6.4 *Interactive Interface of the lighting design application.*

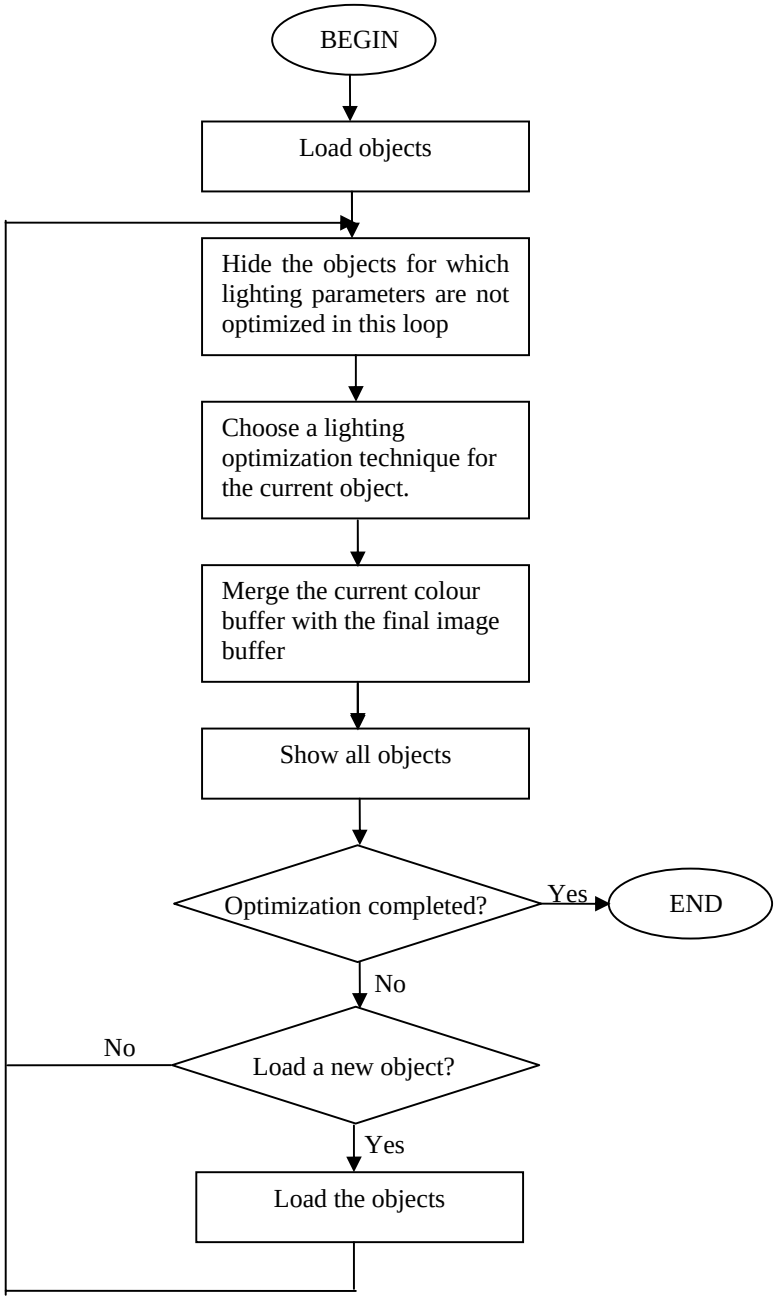


Figure 6.5: *The workflow for optimizing lighting parameters for an individual object in 2D Interactive Lighting Design.*

6.5 Chapter Conclusion

Interactive methods developed in this chapter aim to demonstrate some ways of using lighting design approaches developed in this research. There should be more different ways of using lighting design approaches developed in this research. In chapter 8 (Conclusion) we outline a number of proposals for exploiting optimization-based lighting design in interactive and real-time adaptive lighting design systems.

In this chapter we have explored different ways of using lighting design methods developed in this research. The use of spotlights in 3D context where the outcome is a 3D scene with optimized lighting parameters has been explored, in which each object in the scene was lit by a spotlight that can be individually optimized. The challenge of this approach is that the parameters of the spotlights must be carefully adjusted such that each spotlight can light only one object.

For 2D context in which the outcome is a 2D image, a set of interactive functions have been implemented that enable graphic designers to optimize lighting parameters for individual set of objects. In this scheme, a lighting optimization process for a 3D scene is broken down to a series of repeated steps; in each step, lighting parameters were optimized for only one object, or one set of objects, and the colour buffer was merged with the final image buffer with respect to depth order of objects. The advantage of this method is that different lighting methods can be employed for different objects in each repeated step. The use of depth information allows graphic designers to create different visual effects that could not be achieved when using only 2D image merging techniques.

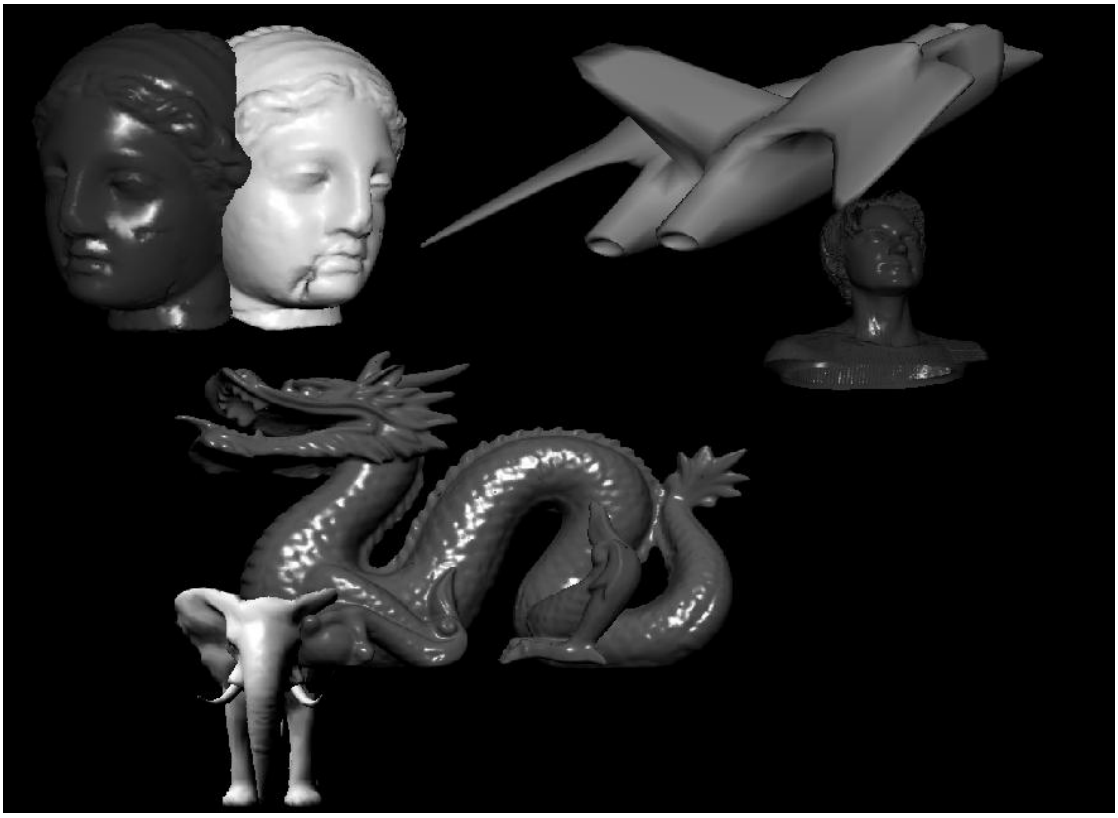
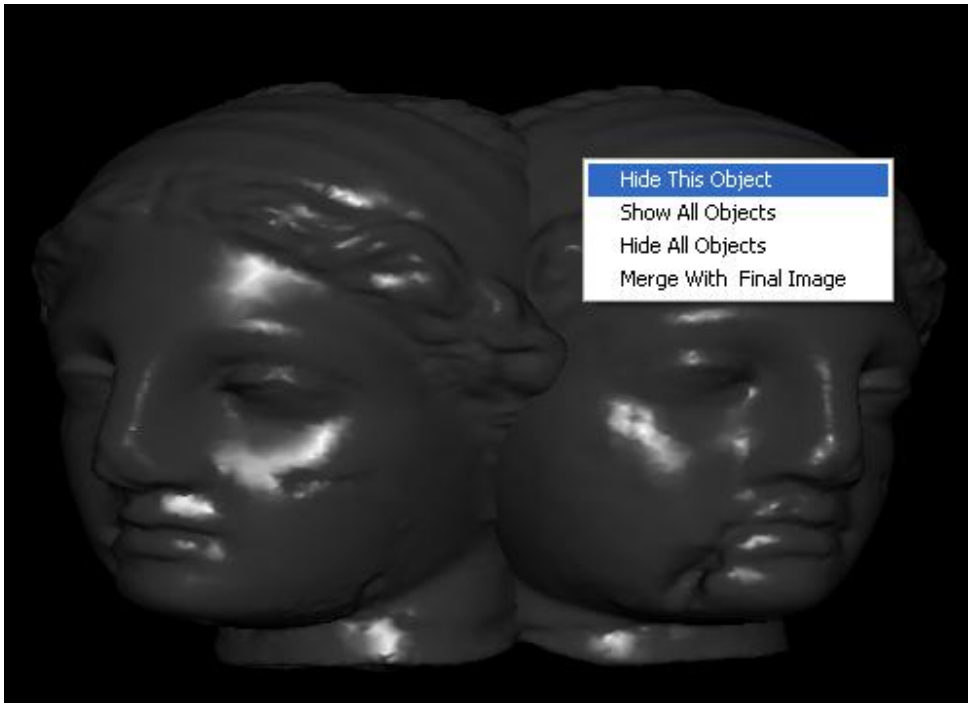
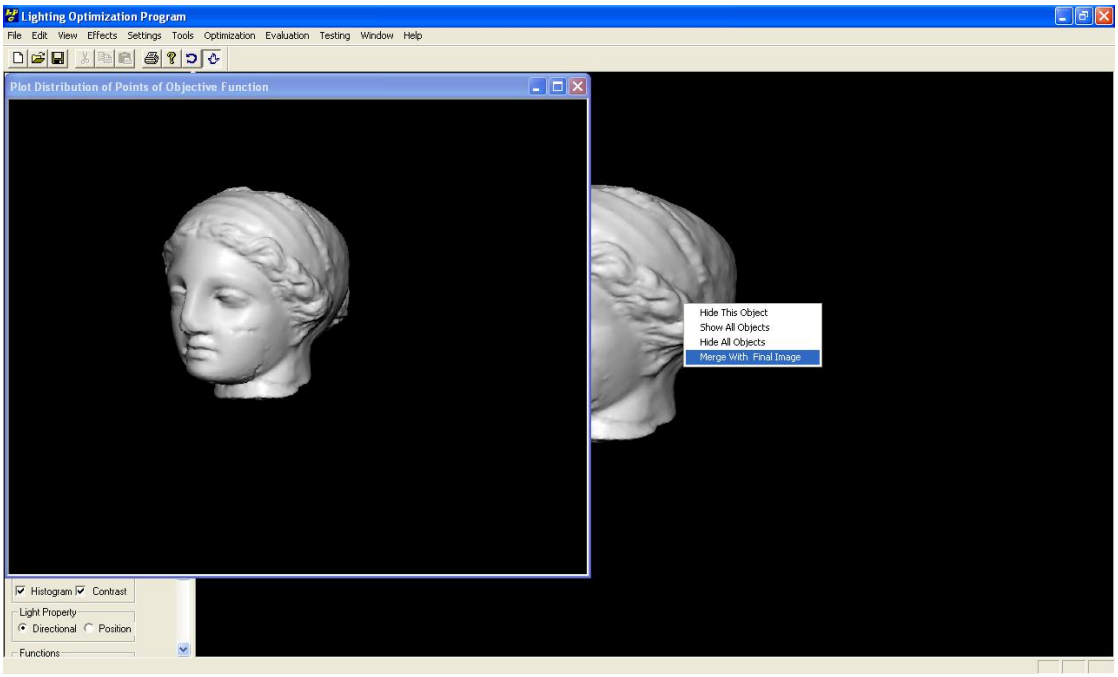


Figure 6.6: *Optimizing lighting parameters for objects individually with different lighting design technique using interactive interface when outcome is a 2D image.*

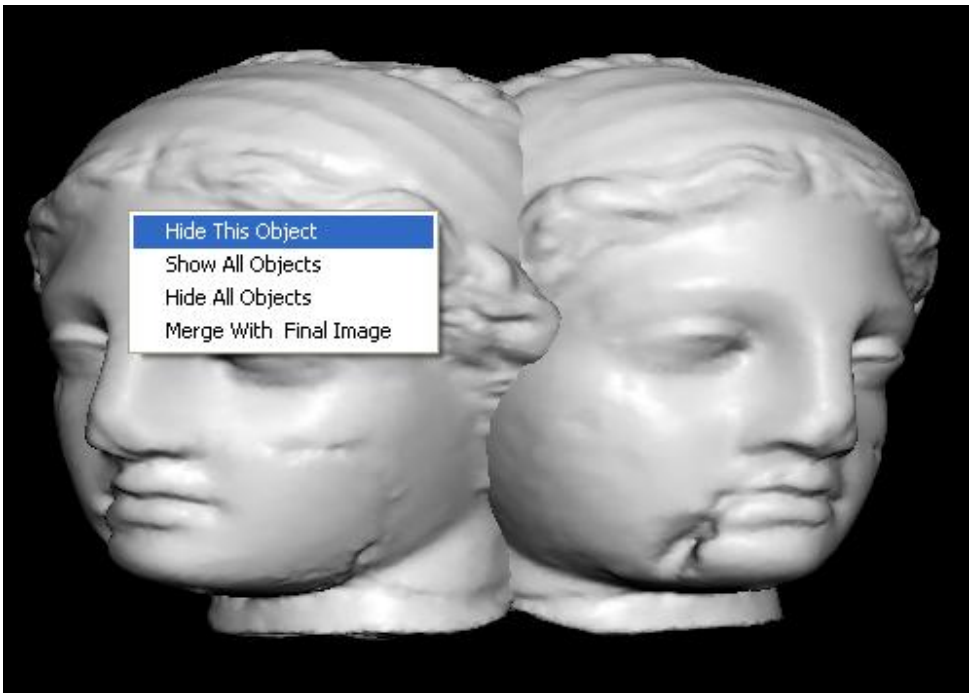


6.7(a) Hide the right head with the context menu

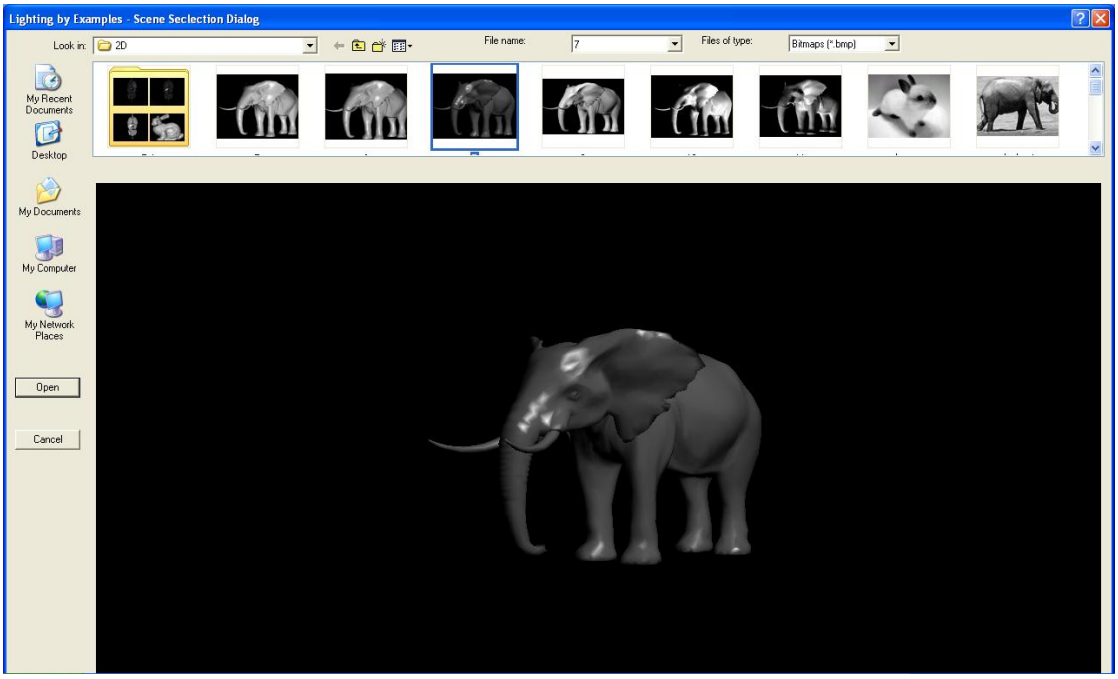
6.7(b) Optimized lighting for the left head using ideal perception-based framework



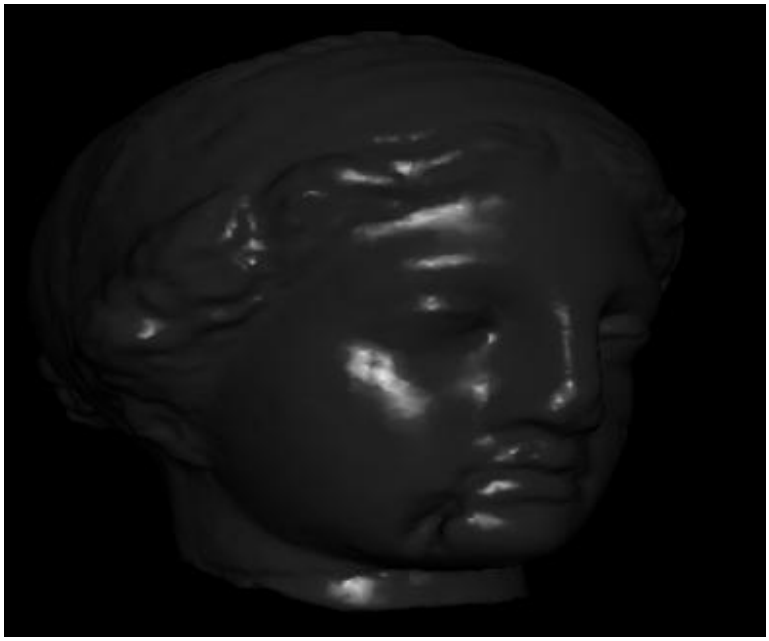
6.7(c) The left head image is merged in the final 2D-image using depth information



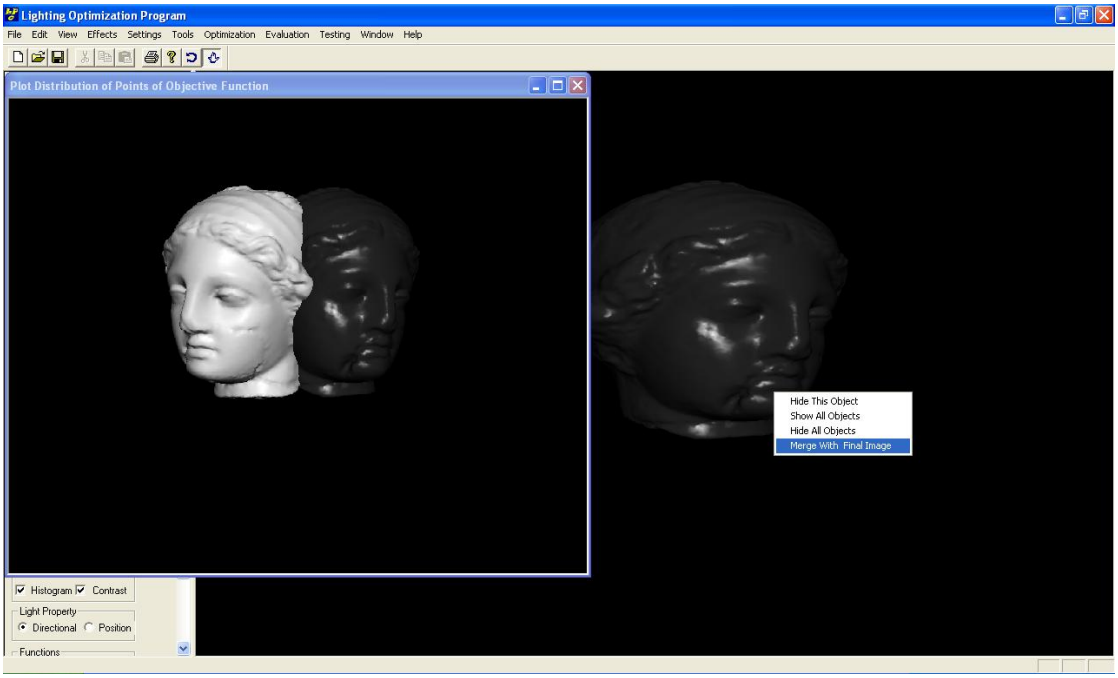
6.7(d) Hide the left head with the context menu



6.7(e) The exemplar is used for the perception-based lighting-by-example framework



6.7(f) Optimized lighting for the right head using perception-based lighting-by-example framework



6.7(g) The final merged image in which the two heads are lit very differently

Figure 6.7: *Process of lighting two heads differently using different lighting design approaches in which the output is a 2D-image.*

Chapter 7: Evaluation

We have developed a lighting design system that uses a number of different approaches, ranging from an ideal lighting design model to *lighting-by-example*, with both a perception framework and a wavelet framework. Due to the subjective nature of people's judgements about lighting quality, previous proponents of automatic lighting design schemes have failed to conduct any significant evaluations of their algorithms, simply preferring to present characteristic examples or use small number of subjects (< 15 subjects). In this, we present a number of experiments that explore the judgements of a large subject pool as to the configuration of the optimization scheme (i.e. the weights of the linear combination of factors in the ideal lighting framework) and their ratings of the two lighting-by-example schemes.

7.1 Experiment 1: Configuring the parameters for the ideal lighting scheme

As described in chapter 3 ("Extending the Perception-Based Approach"), at the heart of the perception-based framework is an objective function motivated by research on human visual perception. In our final version of the objective function we increased the number of components to six: *edge visibility*, *mean brightness*, *shading gradient*, *luminance variance*, *histogram*, and *contrast*. These six components are linearly combined using six weights: w_e, w_m, w_g, w_v, w_h, w_c. Each weight has the effect of controlling the impact of the associated component on the objective function. For instance, if the weight of the mean brightness component is increased, then it will have a more significant impact on the objective function. Therefore, the ratio of weights is vital to how the lighting configuration gives rise to the optimization framework. In simple terms, an inappropriate ratio of the weights will result in perceptually sub-optimal lighting.

For example, if the weight of the mean brightness component is too big, compared to that of the other components, this is likely to result in a lighting configuration where the lit scene has few depth cues. That is, the shading gradient and the contrast components have a reduced significance. Similarly, if the weight of the edge visibility component is overly dominant, this may result in a lighting configuration where the lit scene is in general too bright (though the edges stand out prominently from the background).

Although previous optimization-based approaches have relied on the intuition of the researchers themselves to select the values of these different weights in the objective function, our goal is to base these setting on the findings of perceptual experiments on significant numbers of actual human subjects. This is a challenging problem of experimental design. Theoretically, each weight can take any real value in the normalized range from 0.0 to 1.0, so with six weights even a relatively coarse discretisation of the parameter space gives rise to a large number of combinations where each combination is itself a potential "condition" of the experiment.

However, as already discussed, in practice it is the relative differences between weights that determine the behaviour of the objective function, rather than absolute values of the weights of the components in the objective function. This allows us to reduce the complexity of the problem, from that of finding absolute values for the weights of the components, to the search for a relative weighting of the components in the objective function. To reduce the number of combinations of weights that we need to evaluate, we use a binary system for the generation of the ratios of the weights in which each weight can take one of only two values; on the reasonable assumption that every component has impact on the objective function, we have chosen to evaluate participants' judgements for weights set to 0.2 and 0.8.

The different combinations of two values for each weight yields 64 distinct settings of weights in the objective function. Our experimental set-up lights a number of 3D scenes according to these 64 settings for the objective function. Samples of these 64 images, for different 3D objects, are used as stimuli in a study in which participants are asked to rate an image on a scale ranging from 1-5 (worst-best). The interface presented to the participants is shown in Figure 7.1. The experiments were implemented as a web-based application in PHP hosted on an Apache web server using MySQL. At the beginning of the experiment, users were given a brief guide to the evaluation of lighting quality. Figure 7.7 shows the homepage of the web-based application. The personal information of subjects was also collected.

Figure 7.2 shows the average scores of each 64 settings of the weights over the 40 subjects that took part in the experiment. Although a number of configurations of the weights yield high values, setting 45 has the highest rating score which means that the

ratio of weights used for optimizing the lighting parameters for the image in the question should be the best one. The values of the weights are: $w_e=0.2$, $w_m=0.8$, $w_g=0.2$, $w_v=0.2$, $w_h=0.8$, $w_c=0.2$. An example rendering, produced using this weight configuration, is shown in Figure 7.3. In order to re-evaluate the ratio of weights found in the experiment, we used this ratio to optimize lighting parameters for different scenes, and some results are shown in Figure 7.4. The results have shown that the ratio of weights found in the experiment works well with different scenes.

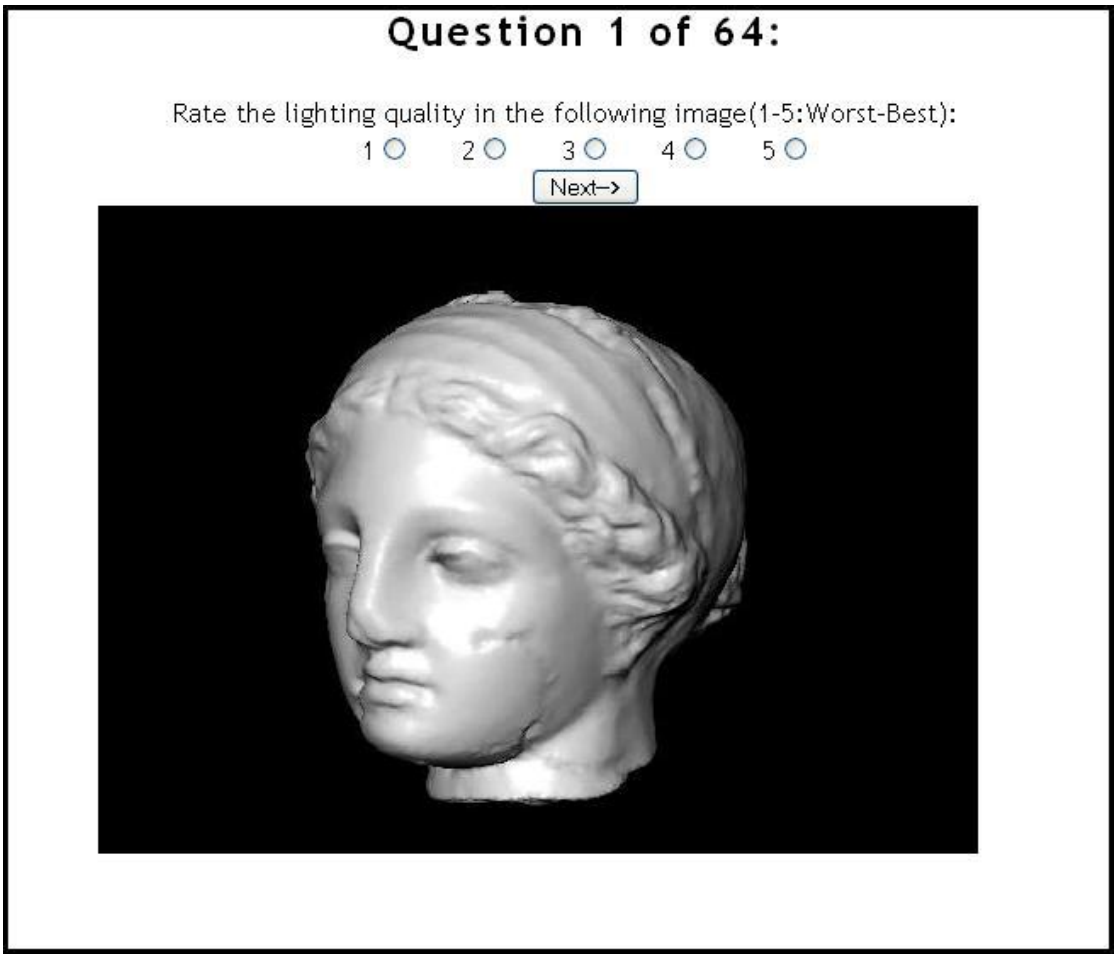


Figure 7.1: *Interface for experiment 1, users were asked to rate the lighting quality of the image on a scale from 1 to 5 (worst-best).*

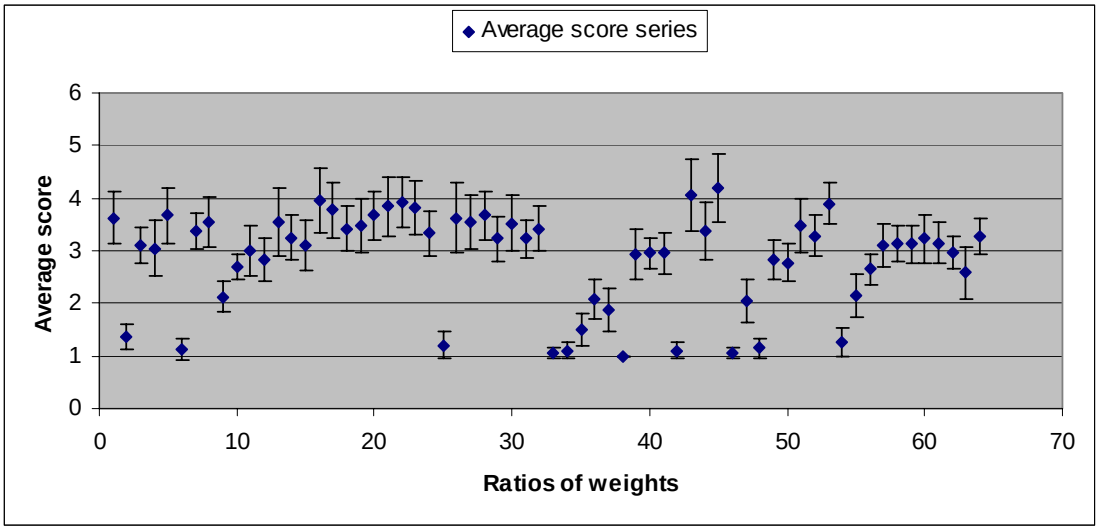


Figure 7.2: *Average scores of the 64 settings of weights evaluated in experiment 1.*

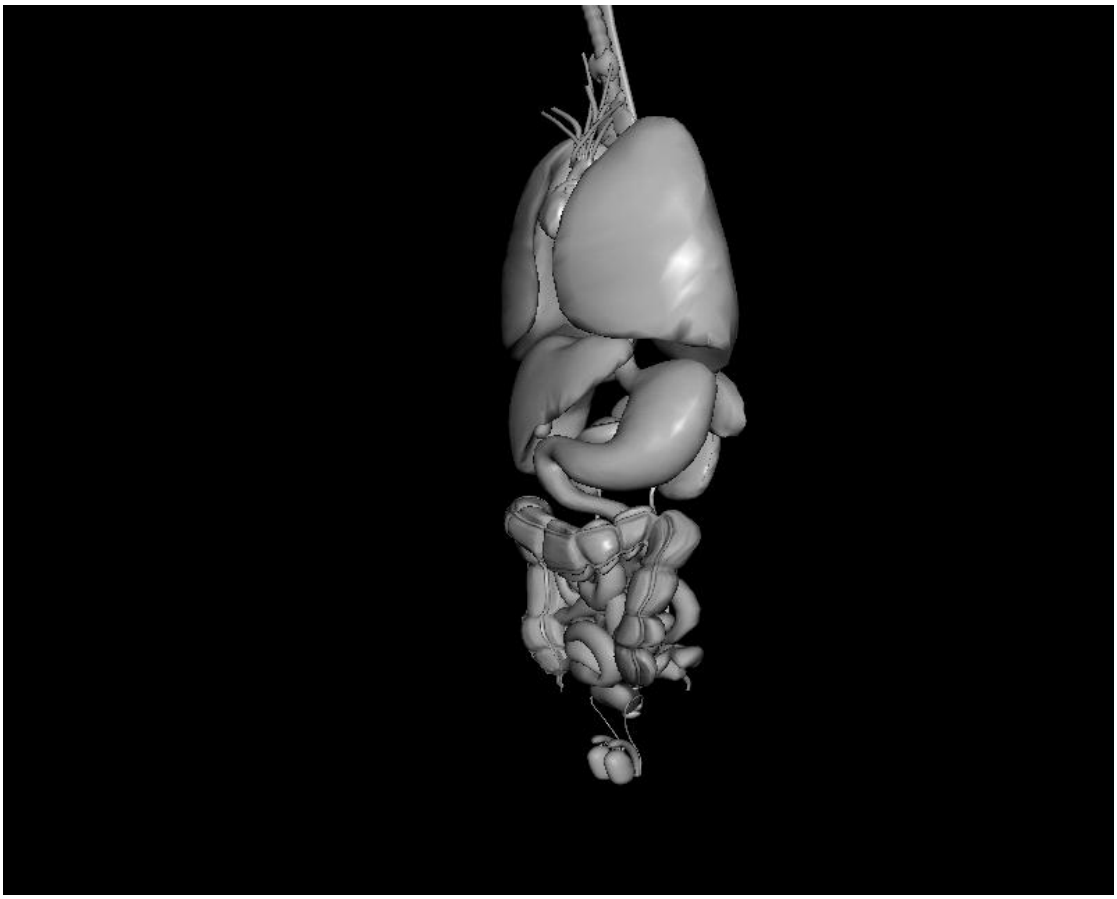


Figure 7.3: *Image used the weight settings with the highest average score across the 40 participants.*

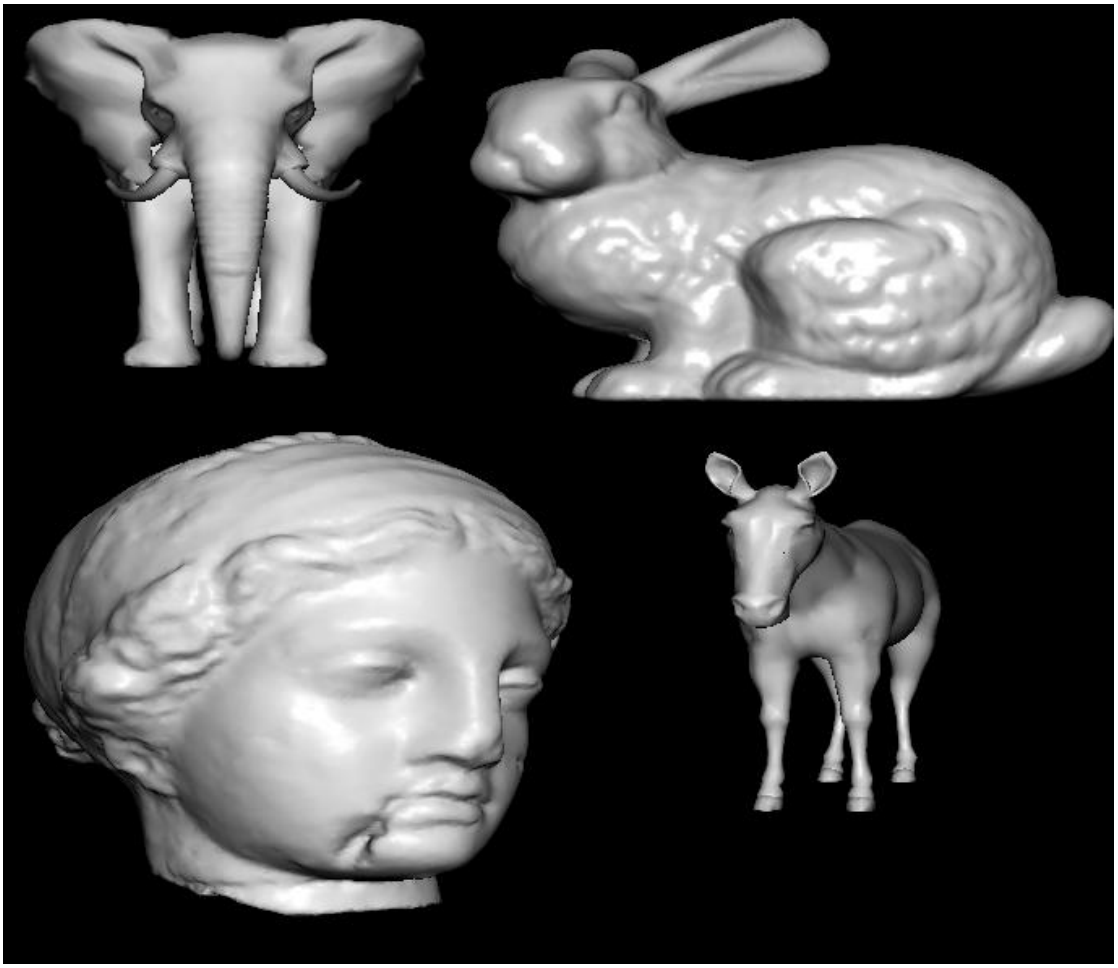


Figure 7.4: *Results of optimizing different scenes using the ratio of weights found in experiment 1.*

7.2 Experiment 2: Evaluating the extensions to perception-based lighting

Our extension to Shacked's original characterization of perception-based lighting design included a number of factors, most notably the addition of a contrast property to the objective function (contrast between different surfaces of an object). Other factors included back-lighting to enhance visibility of the silhouette of the objects, the use of a perceptually uniform colour space (instead of a simple RGB colour space), a more powerful optimization framework, and the inclusion of shadow processing.

In our second evaluation study, we sought to explore the cumulative impact of these extensions (when compared to Shacked's original proposal). Thirty pairs of images were shown, one-by-one, to the study participants. Both the images in each pair were the same rendered 3D model, from the same viewpoint. However, one was lighting optimized

using our implementation of Shacked's original framework, while the second used the fully extended perception-based framework as described in Chapter 3. Participants were asked to rate which of the pair of images they considered to be better lit. Figure 7.5 shows an example of the web-based interface with which the stimuli were presented to participants in this experiment.

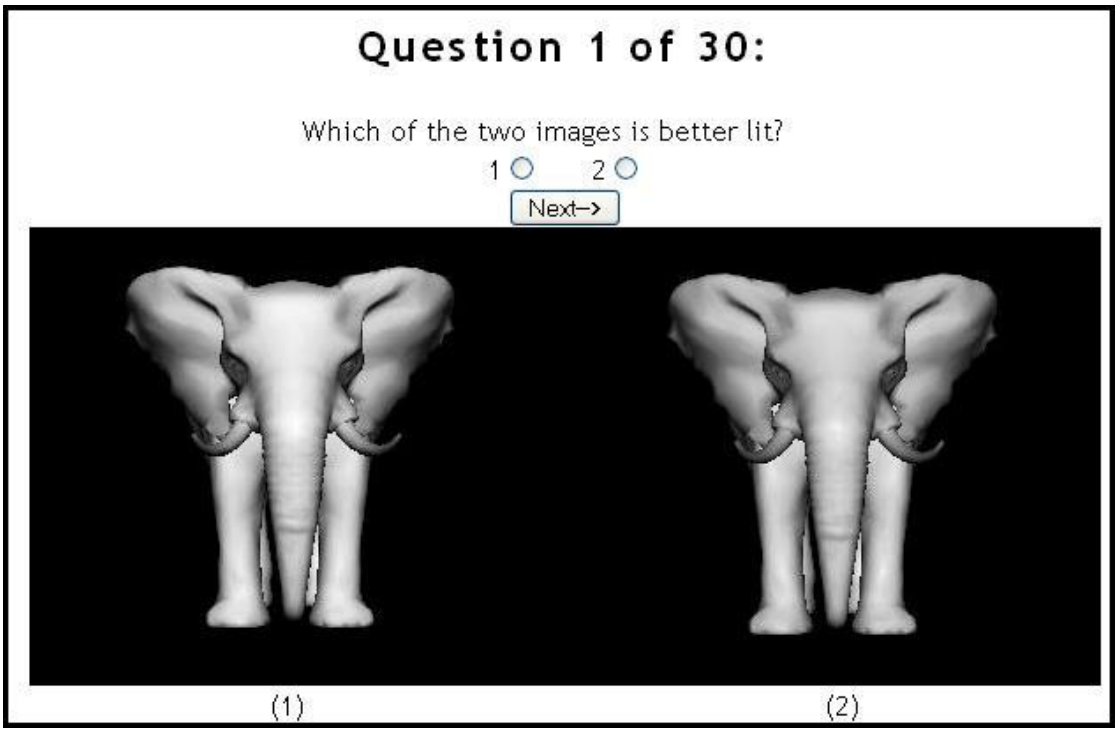


Figure 7.5: *Interface for experiment 2: participants were asked to compare the lighting quality of the two images.*

Table 7.1 shows the results of this experiment. The proportion of ratings in favour of the extended perception-based framework for each user is computed on the basis of the number of the answers, showing that the image generated by extended perception-based framework has better lighting quality than that generated by the original one, as follows:

$$P_e^k = \frac{NA_e^k \times 100}{NQ}\% \qquad (7.1)$$

P^k_e is the percentage of ratings in favour of the extended perception-based framework of the subject *k*.

NA^k_e is the number of answers in favour of the extended perception-based framework of the subject *k*.

NQ is the total number of questions in the experiment.

The average proportion of ratings in favour of extended perception-based framework across *N* subjects can be computed as in equation (7.2):

$$M = \frac{1}{N}\sum_{k=1}^{N} P_e^k \tag{7.2}$$

M is the average percentage of rating in favour of extended perception-based framework across *N* subjects.

N is the number of the subjects.

P^k_e is the percentage of rating in favour of the extended perception-based framework of the subject *k.*

The standard deviation of ratings in favour of the extended perception-based framework across *N* subjects can be calculated as in equation (7.3).

$$SD = \sqrt{\frac{1}{\mathrm{N}}\sum_{k=1}^{N}\left(P_e^k - M\right)^2} \tag{7.3}$$

SD is the standard deviation of rating in favour of extended perception-based framework across *N* subjects.

M is the average percentage of rating in favour of extended perception-based framework across *N* users.

N is the number of the subjects.

P^k_e is the percentage of rating in favour of the extended perception-based framework of the subject *k.*

	Mean (%)	Standard Deviation	p-value
Percentage of ratings that favour the extended framework	72.57	23.09	0.09

Table 7.1: *The result of the experiment for evaluating the effect of the extended perception-based framework versus original one.*

Table 7.1 shows that participants judge the lighting quality of the images optimized using extended perception-based framework as significantly better than those images optimized using Shacked's original perception-based framework. This is incontrovertible evidence that the cumulative impact of the additional features and the modification to the objective function has led to significant subjectively judged improvements in the resulting lighting.

7.3 Experiment 3: Evaluating approaches to lighting-by-example

The *lighting-by-example* approaches presented in chapters 4 and 5 were motivated by the fact that "ideality" is not necessarily (and, in fact, not usually) the lighting quality we are looking for in a rendered image. In the perception-based lighting design approach, lighting parameters are optimized in order to maximize the visual perception of the objects in the image, such as edge visibility, mean brightness, shading gradient, intensity range, ideal histogram and contrast. However, lighting designers do not simply seek to produce scenes that are in some way perceptually optimal, but instead seek to convey emotions, moods and expression (especially in cinematographic contexts).

Of course, in terms of visual imagery, we often know what we want (and like) when we see it as visual stimulus. *Lighting-by-example* allows a user to express their desired lighting qualities by selecting an exemplar that is lit in a manner desired. Our realization of *lighting-by-example* has been based on two quite different frameworks: the perception-based framework and the wavelet-based framework. While we expected that the wavelet-based framework should better capture the visual properties of a target, we designed a user-study in order to evaluate participants' judgments of the two *lighting-by-example* approaches in terms of the extent to which they can replicate the lighting of the target scene.

Forty participants took part in a study in which thirty sets of images, each set containing three source images and one target image, were presented to participants one-by-one. Participants were asked to match the target image with one of the three source images according their judgement of the similarity of the lighting. The three source images were generated by optimizing lighting parameters for a 3D scene using either a perception-based framework or a wavelet-based framework. Among three source images for comparison, only one was using the target image presented to them, while the others

were hand-crafted examples that we judged to be approximately similar to the target. Figure 7.6 shows the interface presented to participants.

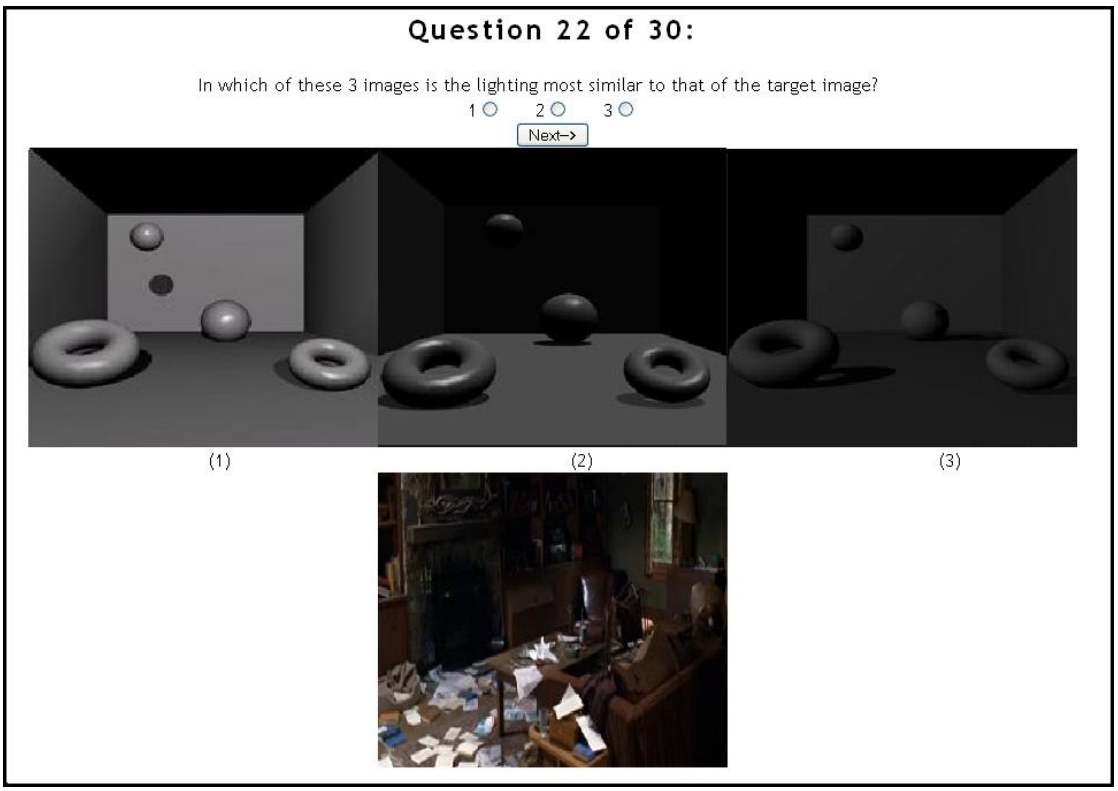


Figure 7.6: *Interface for experiment 3: users were asked to match the target at the bottom with one of three source images in the top row.*

Table 7.2 shows the results for experiment 3. The proportion of source images selected that matched the correct source image for the perception-based *lighting-by-example* or wavelet-based *lighting-by-example* scheme was calculated on the basis of the number of right answers for each case and the total number of questions for each case, as indicated in equation (7.4).

$$P^{k}{}_{pw} = \frac{NA^{k}{}_{pw} \times 100}{NQ_{pw}}\% \qquad (7.4)$$

$P^{k}{}_{pw}$ is the percentage of matching answers for the perception-based/wavelet-based *lighting-by-example* for subject *k*.

$NA^{k}{}_{pw}$ is the number of right answers for the perception-based/wavelet-based *lighting-by-example* for subject *k*.

NQ_{pw} is the total number of questions for perception-based/wavelet-based *lighting-by-example* in the experiment.

The average proportion of correct matches for the perception-based/wavelet-based *lighting-by-example* scheme across *N* subjects can be computed as in equation (7.5).

$$M_{pw} = \frac{1}{N}\sum_{k=1}^{N} P_{pw}^{k} \tag{7.5}$$

M_{pw} is the average percentage of rating for perception-based/wavelet-based *lighting-by-example* across *N* users.

N is the number of subjects.

P^{k}_{pw} is the percentage of rating for perception-based/wavelet-based *lighting-by-example* of the subject *k.*

The standard deviation in the proportion of correct matches for the perception-based/wavelet-based *lighting-by-example* schemes across *N* subjects can be calculated as in equation (7.6).

$$SD_{pw} = \sqrt{\frac{1}{\mathrm{N}}\sum_{k=1}^{N}\left(P_{pw}^{k} - M_{pw}\right)^{2}} \tag{7.6}$$

SD_{pw} is the standard deviation of rating for perception-based/wavelet-based *lighting-by-example* across *N* subjects.

M_{pw} is the average percentage of rating for perception-based/wavelet-based *lighting-by-example* across *N* users.

N is the number of subjects.

P^{k}_{pw} is the percentage of rating for perception-based/wavelet-based *lighting-by-example* of the subject *k.*

	Average (M_p) (%)	Standard Deviation	p-value
Wavelet-based LBE approach	95.88	9.79	0.16
Perception-based LBE approach	80.29	12.42	0.13

Table 7.2: *Results for experiment 3: comparing light-by-example approaches.*

The data in the Table 7.2 summarizes the main results of the experiment, that the perception-based LBE and wavelet-based LBE were judged to have been the closest match in 80.29% and 95.88%, respectively. The average scores show that the wavelet-based LBE is judged to perform better than perception-based LBE.

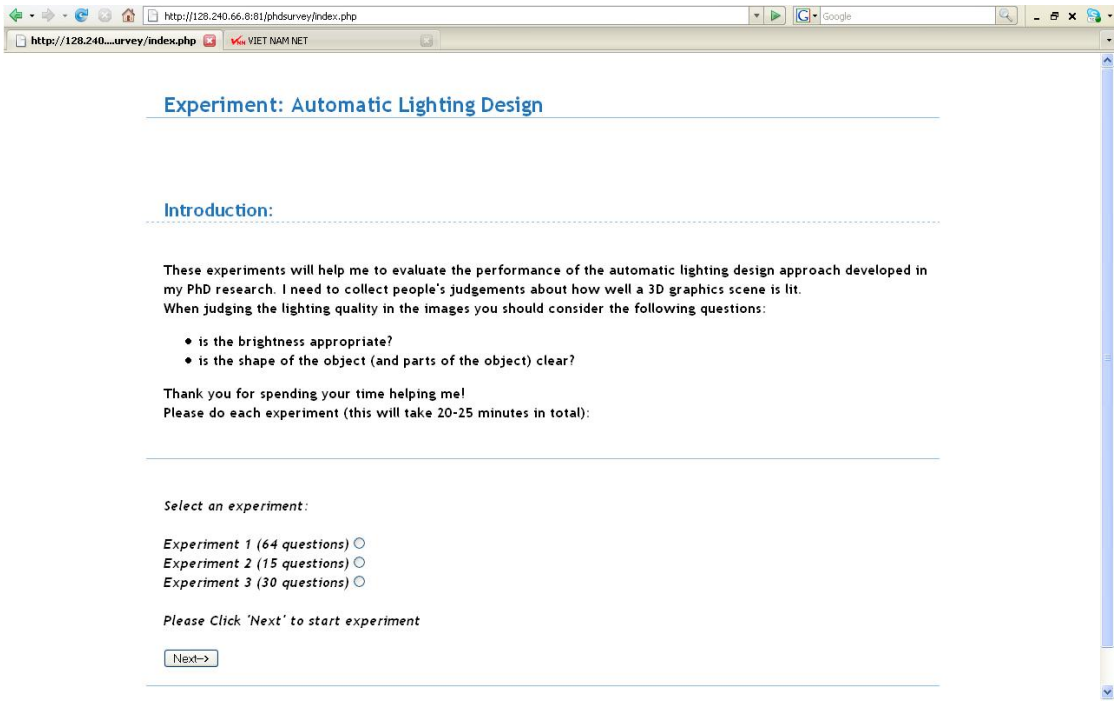


Figure 7.7: *The interface for the web-based application designed for the experiments.*

7.4 Chapter Conclusion

In this chapter, three experiments have been described and analyzed. The first experiment aimed to evaluate the most appropriate setting for the weights of the objective function for the perception-based lighting design framework. On the assumption that the relative ratio of the weights is more important than absolute values, the value space was two distinct values. The two values for each of the six parameters were used to generate a set of 64 distinct configurations for the weights in the experiment. Each question contained an image which was derived by optimizing lighting parameters for a 3D scene using an objective function and one of the 64 configurations of the weights. Subjects in the experiment were asked to rate the lighting quality of the image in questions from 1-5. The configuration of weights with the highest average score was considered to be the best candidate configuration for the objective function.

This configuration of weights was used to optimize the lighting for a number of other scenes and was found to yield qualitatively similar (and effective) results.

A number of extensions were added to the original perception-based lighting design framework, including the contrast between different surfaces of an object, back-lighting technique, use of perceptually uniform colour space, more powerful optimization framework and shadow processing. In our second experiment, the performances of this extended perception-based lighting design framework were contrasted with the original configuration. Users were asked to compare a pair of images which had been generated by optimizing the lighting parameters for the same 3D scene with the original perception-based framework and the extended one. The results clearly demonstrated the additional utility of the extended perception-based framework.

Finally, the *lighting-by-example* approach aims to provide users with a new way of designing lighting for 3D scenes. It was inspired by the fact that lighting can convey information about moods and emotions and that users normally know what they want when they see it rather than directly manipulating technical lighting parameters. The *lighting-by-example* approach has been realized using both a perception framework and wavelet framework and, in a third experiment, we sought to evaluate how well the lighting effects of target images were judged to have been captured. Participants were asked to match a target image to one of three source images (for which perception-based LBE or wavelet-based LBE had been used for one of the images). The average score of wavelet-based LBE was significantly higher than that of perception-based LBE, lending evidence to our intuition that capturing the spatial frequency distribution of lighting in a scene is a key element of a person's judgement regarding lighting similarity.

Chapter 8: Conclusions and Future Work

8.1 Main Contributions

We have successfully explored improvements on previous perceptual approaches to lighting design, by incorporating measure of contrast and using a perceptually uniform colour space that are significant element of a user's judgement of the lighting in an image. Explicit perceptual factors were also augmented by the use of a practical studio lighting heuristic, such as backlighting. Also, by having the objective function take special account of object edges that are in shadow (actually self-shadows) using a custom shadow processing framework, we have been able to take account of subtle lighting interactions not previously taken account of in earlier approaches. Furthermore, since the lighting design process is cast as an optimization problem, a systematic exploration of the objective function for a number of 3D scenes helped us justify our choice of a stochastic optimization scheme (for example, as compared to local gradient descent methods). Finally, for the very first time in the lighting design literature, a significant user study was designed in order to find an actual set of weight ratios for the components of the objective function and these were found to be applicable for a number of different 3D models that we evaluated.

The most significant contribution of this research is our intuitive *lighting-by-example* approach, which we have explored both in relation to a perceptual model of lighting (our extension of Shacked's approach) and a new wavelet-based framework. In the perception-based lighting design for ideal lighting, the optimization goal was a combination of the visual perceptual qualities of the image and we sought to minimise the difference between these for a source and target image. As we discussed, the approach of lighting design through the optimization of an objective that characterizes an "ideally lit scene" can be criticized for the naïve view it takes of the role that lighting plays in visual media. However, *lighting-by-example* was motivated by the fact that lighting can convey moods and emotions that are better expressed in terms of a 'variety of lighting'; more importantly, this *light-by-example* was also motivated by the fact that users normally know what they want when they see it. In the LBE-based system, users were able to express their desired lighting effects by selecting an image, from a set of

exemplars, which had lighting similar to the result they desired and to avoid the complications of setting lighting parameter values explicitly (which requires a subtle understanding on the impact each parameter might have on a scene). Our first attempt to realise LBE was based on the extended perception-based lighting design framework presented in chapter 3 ("Extending the Perception-Based Lighting Design"). The main drawback of using extended perception-based lighting design as the foundation for the LBE approach was its failure to sufficiently capture the spatial frequency distribution of the illumination on a target image. That is, it characterised a scene in terms of individual objects, and a set of relative values for perceptual metrics in relation to individual objects. Our wavelet-based mathematical framework extended the one-dimensional Haar basis functions to two dimensional wavelet functions and was applied in the wavelet transform of the source and target images. By seeking to match the two spatial frequency distributions, this wavelet-based optimisation process yields a *lighting-by-example* approach that best matches users' expectations.

We now observe that the outcome of a program of research into the design of lighting is not a cognitively accurate (and empirically verified) objective function for ideally illuminated scenes, but a framework for the specification of lighting using example 3D scenes and photographs – and tools to allow artists to interactively modify scene lighting through inverse design. Such approaches presume the ability to model target scenes in the form of a meaningful objective function, and to optimize source scenes using these objectives. In contrast to previous work, we have sought to refer back to users (the viewers of images) to both configure the parameters of our objective function and to verify the results of our approaches to lighting design. The first experiment enables us to find the best configuration for the parameters of the objective function (rather than using arbitrary, hand-crafted settings as is standard practice). The second experiment demonstrated that the additional features added to the perception-based lighting design framework brought significant benefits to the lighting design process in terms of users' judgements of quality of lighting in the generated images. The third and final user study evaluated the performance of the *lighting-by-example* approach. In this experiment, we saw that our *lighting-by-example* approach allowed us to transfer the lighting effects from targets to the source scenes, as judged by users.

We have explored different ways of using the lighting design techniques devised in this research program by developing an interactive interface for the lighting design process. Two approaches to interactive lighting design were proposed. In the 3D context where the anticipated output of the lighting design process is a 3D scene, we have proposed a method that uses a spotlight for each object in the scene. The lighting parameters of each spotlight are optimized separately using different lighting design techniques. Alternatively, in the 2D context, where the expected output of the lighting design process is a 2D image, a method was been proposed that is based on a merging technique that takes advantage of depth information. For this merging approach, interaction takes the form of a set of operations such as "showing" objects, "hiding" objects, and "merging buffers". Lighting design in a 2D context is a process in which the lighting parameters are optimized for certain objects in a scene by hiding unwanted objects. The rendered image of the optimized objects is then merged into final image buffer using depth information to put objects in the right depth order. This technique is a combination of 2D image processing and depth buffer processing techniques. All the approaches to lighting design presented were implemented within a single application framework that realised all the basic lighting design operations (i.e. analysis, extraction, rendering etc) and supports loading and processing of a number of file formats that are widely used in 3D graphics, such **.ply* and **.3ds*.

8.2 Implications for Future Research

This preliminary work has the potential to enable a number of exciting developments in the design and application of lighting systems in computer graphics. Several avenues for future work can be readily identified:

Dynamic real-time lighting: In applications that include dynamic scenes with moving objects, a dynamic real-time lighting framework is highly desirable. In such applications, changes of scene structure from frame-to-frame results mean that lighting parameters could be re-optimized in order to express the intent and purpose of current scene structure. The current lighting optimisation process is resource intensive and so we need

to develop alternative real-time approaches, for example using caching and light mapping techniques, constraining the locations of lights in an appropriate manner.

Implementation of Multi-Light Optimisation: In theory, the approaches developed, such as ideal perception-based lighting design and *lighting-by-example*, can support any number of lights. Due to our particular research focus being on the discovery of different lighting design algorithms (rather than performance and applications in their own right), our current lighting optimisation framework currently supports a maximum of eight lights for each scene, while each optimisation loop only supports two lights. In terms of practical application, it would be desirable to realise these techniques in the form of a plug-in for graphics tool such as 3D-Max and Light Wave.

Interaction for lighting design: In the current interactive interface, we only implemented a number of basic functions for the purpose of exploring their potential, and as a result our prototype interface was rather simple in the range of interactions it supports for the spotlights and the depth information-supported merging technique. We believe the further development of interaction techniques, to combine the lighting approaches, is likely to greatly enhance the appeal of such a system to novice and expert users. For example, an interface that allows users to change the target values for *lighting-by-example* approach would allow significantly more flexibility. Likewise, the interface for manipulation of light parameters is rather limited in the current interactive interface. In many cases, the light positions have been fixed and defined by users. A sketch-based interface for manipulating scene parameters would also contribute to making interactions far more natural for users. Also, in some applications, designing lighting parameters for multiple viewpoints is highly desirable, as this requires interaction and display techniques such as allowing multiple viewports for displaying results of the lighting design process from different viewpoints. Finally, in the current development framework, only standard lights are used in the system. However, for scenes that require complex lighting effects, extended light sources (i.e. physically extended light sources that are not point-like) are needed.

Developing the perceptual metrics: Many ideas explored were motivated by existing psychological findings in visual perception. Of course, the lighting effects themselves are only meaningful in relation to the realities of visual perception. There is significant

scope for the development of further perceptual metrics, both in terms of the low-level components of a perceptual model of lighting (i.e. for our *ideal lighting model*) and also in more holistic metrics such the distance between source and target components in the objective function, which would enhance LBE process directly.

Developing emotion/mood descriptors (a language for lighting): The fact that lighting effects can expressively convey mood and emotion has inspired our *lighting-by-example* approach. Within different visual cultures, there is some consistency in the representation of moods/emotions using different lighting effects and lighting properties. The integration of emotion/mood descriptors into the lighting optimisation process has the potential to increase the flexibility of both our *lighting-by-example* and direct lighting approaches by enabling users to communicate explicit goals with the system. Emotion/mood descriptors might be used either as a criteria to search for an appropriate exemplar or directly as a target for the lighting optimisation.

Lighting for non-photorealistic scenes: Non-photorealistic lighting is not constrained to physically correct lighting and shading. Lighting quality of a non-photorealistic scene is application dependent. For instance, supporting the communication of form and character means that many of the conventions of classical perception (incorporated in our objective function) would have to take second place to the stylised effects of lighting realised in cartoons and other non-photorealistic media. It is likely that we would need to develop an alternative ontology of lighting effects for such domains, but the underlying notions of lighting-by-example would still apply.

8.3 A Final "Word"

The main motivation for this research was the development of a new approach to lighting design that facilitates the work of computer graphics lighting designers. We also sought to develop algorithms and proof-of-concept software that would be the basis of an easy-to-use tool for optimizing lighting parameters for novice users, who might only have examples of the sort of lighting they would like. As described in Chapter 1 ("Introduction"), the design of the lighting for 3D computer graphics is a significant problem in adaptive and intelligent multimedia. The position and intensity of lights have a significant and complex bearing on a viewer's perception of a scene. The traditional approaches to lighting design fundamentally rely on trial-and-error, where lighting

parameters are incrementally manipulated and rendered until the desired visual properties of the scene are obtained. Due to the fact that the efficient manipulation of lighting parameters requires both a technical and aesthetic grasp of the impact of changes to lighting parameters, non-expert designers encounter significant difficulties in adjusting lighting parameters to achieve a desired lighting.

We realized that a deeper account of lighting, that took stock of properties such as contrast and the nonlinearity of our perceptual response to colour (which we modelled using a perceptually uniform colour space), could lead to improvements in the automated design of lighting. Similarly, less well-founded but practical techniques, as employed in studio lighting, have much to offer. Our novel approach to lighting design, which we term *lighting-by-example*, was motivated by our awareness that ideal lighting is not always what users need or want, and that lighting can be used to convey mood and emotion. We are very aware that credibility of this research is dependent on whether a user's perceptions of the results generated demonstrate the improvements predicted, and we have conducted user studies both to refine our approach (i.e. set the values of parameters of our ideal lighting objective function) and evaluate its effectiveness. Algorithmic developments do not, of course, preclude the need for creative tools and we have prototyped such a tool that combines our different lighting design approaches. Though the result is not an elegant integration of our methods, and is not particularly usable, it stands as a novel exemplar of the declarative control that future lighting design tools should provide. Although the system itself needs to be extended and refined before it is integrated in real world graphics authoring tools and applications, the research presented here on the automation of lighting design has laid the conceptual and practical foundations for such work.

References

1. Birn, J. (2000). Lighting & Rendering. New Riders, USA.
2. Schoeneman, C., Dorsey, J., Smits, B., Arvo, J., and Greenburg, D. (1993). Painting with light. In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 143-146.
3. Whitted, T. (1980). An improved illumination model for shaded display. Communications of the ACM, vol. 23(6), pp. 343-349.
4. Cindy, M.G., Kenneth, E. T., Donald, P. G., and Battaile, B. (1984). Modelling the interaction of light between diffuse surfaces. In Proceedings of the 11th Annual International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 213-222.
5. Hanrahan, P., Haeberli, P. (1990). Direct WYSIWYG painting and texturing on 3D shapes. In Proceedings of the 17th Annual International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp.215-223.
6. Kawai, J.K., Painter, J. S., and Cohen, M. F. (1993). Radioptimization-goal based rendering. In Proceedings of the 20th Annual International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 147-154.
7. Flynn, J.E., Hendrick, C., Spencer, T.J., and Martyniuk, O. (1979). A guide to methodology procedures for measuring subjective impressions in lighting. Journal of the Illuminating Engineering Society, vol. 8, pp. 95-110.
8. Flynn, J.E. (1977). A study of subjective responses to low energy and non-uniform lighting systems. Lighting Design and Application, vol. 7(1), pp. 167-179.
9. Poulin, P., Fournier, A. (1992). Lights from highlights and shadows. In Proceedings of Symposium on Interactive 3D Graphics, pp. 31-38.
10. Poulin, P., Ratib, K., and Jacques, M. (1997). Sketching shadows and highlights to position lights. In Proceedings of the Conference on Computer Graphics International (CGI), pp. 56-63.
11. Jolivet, V., Plemenos, D., and Poulingeas, P.I. (2002). Inverse direct lighting with a Monte Carlo method and declarative modelling. In Proceedings of International Conference on Computational Science (ICCS), pp. 3-12.

12. Shirley, P. (1991). Radiosity via ray tracing. Graphics Gems II, Arvo, J. Ed., Academic Press, San Diego, p. 306-310.
13. Graham, R.L. (1972). An efficient algorithm for determining the convex hull of a finite set of points in the plane. Information Processing Letters, vol. 1, pp. 132-133.
14. Plemenos, D. (1995). Declarative modelling by hierarchical decomposition. In Proceedings of International Conference (GraphiCon).
15. Desmontils, E.. (1995). Formulation des Propriétés en modélisation déclarative à l'aide des ensembles flous décembre. Rapport de Recherche IRIN.
16. Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., and Shieber, S. (1997). Design galleries: a general approach to setting parameters for computer graphics and animation. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 389-400.
17. Karypis, G., Kumar, V. (1995). Multilevel k-way partitioning scheme for irregular graphs. In Technical Report, Dept. of Computer Science, Uni. of Mennesota.
18. Garay, M.R., Johnson, D.S. and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. Theoretical Computer Science, vol.1(3), pp. 237-267.
19. Gooch, A., Gooch, B., Shirley, P., and Cohen, E. (1998). A non-photorealistic lighting model for automatic technical illustration. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 447-452.
20. Anderson, S., Levoy, M. (2002). Unwrapping and visualizing cuneiform tablets. IEEE Computer Graphics and Applications, vol. 22(6), pp. 82-88.
21. Stewart, A.J. (2003). Vicinity shading for enhanced perception of volumetric data. In Proceedings of IEEE Visualization (VIS), pp. 355-362.
22. Phong, B.T. (1975). Illumination for computer generated images. Communications of the ACM, vol. 18(6), pp. 311-317.
23. Reitsma, P.S.A., Pollard N. S. (2003). Perceptual metrics for character animation: sensitivity to errors in ballistic motion. ACM Transactions on Graph, vol. 22(3), pp. 537-542.

24. Farugia, J.P., Peroche, B. (2004). A progressive rendering algorithm using an adaptive perceptually based image metric. Computer Graphics Forum, vol. 23(3), pp. 605-616.
25. Gumhold, S. (2002). Maximum entropy light source placement. In Proceedings of IEEE Visualization (VIS). pp. 275-282.
26. Lee, C.H., Hao, X., Varshney, A. (2004). Light Collages: Lighting design for effective visualization. In Proceedings of IEEE Visualization (VIS), pp. 281-288.
27. Taubin, G. (1995). Estimating the tensor of curvature of a surface from a polyhedral approximation. In Proceedings of the 5th International Conference on Computer Vision (ICCV). pp. 902-907.
28. Mangan, A.P. and Whitaker, R. T. (1999). Partitioning 3d surface meshes using watershed segmentation. IEEE Transactions on Visualization and Computer Graphics, vol. 5(4), pp. 308-321.
29. Shacked, R., Lischinski, D. (2001). Automatic Lighting Design using a perceptual quality metric. Computer Graphics Forum, vol. 20(3), pp. 215-226.
30. Cavanagh, P. (1999). Pictorial art and vision sciences. In Proceedings of MIT Encyclopaedia of the Cognitive. pp. 644-646.
31. Gross, M. (1994). Visual Computing, Springer-Verlag, New York, USA.
32. Christou, C., Koenderink, J., and Doom, A. J. (1996). Surface gradients, contours and the perception of surface attitude in images of complex scenes. Perception, vol. 25(6), pp. 701-713.
33. Biederman, I. and Ginny J. (1988). Surface versus edge-based determinants of visual recognition. Cognitive Psychology, vol. 20, pp. 38-64.
34. Braje, W.L., Tjan, B.S. and Legge, G.E. (1995). Human efficiency for recognizing and detecting low-pass filtered objects. Vision Research, vol. 35(21), pp. 2955–2966.
35. Tjan, B.S, Braje, W.L., Legge, G.E, and Daniel, K. (1995). Human efficiency for recognizing 3-D objects in luminance noise. Vision Research, vol. 35(21), pp. 3053–3069.
36. Tumbling, J., Rushmeier, H. (1991). Tone reproductions for realistic computer generated images. In Technical Report GIT-GVU-91-13, Graphics, Visualization and Usability Centre, Georgia Institute of Technology.

37. Stevens, S.S., Stevens, J.C. (1963). Brightness function: effects of adaptation. Journal of Optical Society of America, vol. 53(3), pp. 375-385.
38. Millerson, G. (2005). Lighting for television and film. Focal Press, 3rd ed., GB.
39. Parker, J.R. (1996). Algorithms for image processing and computer vision. Wiley, New York, USA.
40. Kenneth, R.C. (1996). Digital Image Processing. Prentice Hall, USA.
41. Spillmann, L., Werner, J. S. (1990). Visual perception the neurophysiological foundations. Academic Press, San Diego, USA.
42. Mather, G. (2006). Foundations of perception. Psychology Press, New York, US.
43. Rohaly, A.M., Wilson, H.R. (1999). The effects of contrast on perceived depth and depth discrimination. Vision Research, vol. 39(1): pp. 9-18.
44. Saito, T., Takahashi, T. (1990). Comprehensible rendering of 3-D shapes. ACM SIGGRAPH Computer Graphics, vol. 24(4), pp. 197-206.
45. Kahrs, J., Calahan, S., Carson, D. and Poster, S. (1996). Pixel cinematography: A lighting approach for computer graphics. In Course Notes of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH).
46. Winkler, S. (2001). Visual fidelity and perceived quality: Towards comprehensive metrics. In Proceedings of SPIE Human Vision and Electronic Imaging, pp. 114-125.
47. Sarifuddin, M., Missaoui, R.A. (2005). New perceptually uniform colour space with associated colour similarity measure for content-based image and video retrieval. In Proceedings of Multimedia and Information Retrieval (MIR).
48. Ying, C., Hao, P., and Dang, A. (2004). Optimal transform in perceptually uniform colour space and its application in image coding. In Proceedings of Image Analysis and Recognition (ICIAR), pp. 269-276.
49. Kolpatzik, B.W., Bouman, C. A. (1995). Optimized universal colour palette design for error diffusion. Journal of Electronic Imaging, vol. 4(2), pp. 131-143.
50. Zhang, X., Wandell, B.A. (1997). A spatial extension of CIELAB for digital colour image reproduction. Journal of the Society for Information Display, vol. 5(1), pp. 61-63.
51. Hall, R. (1993). Illumination and colour in computer generated imagery. Springer-Verlag, New York, USA.

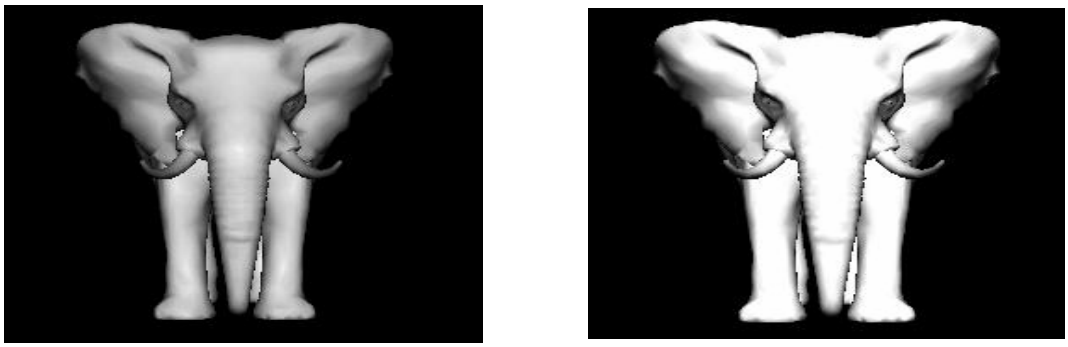
52. William, H.P., Saul, A.T., William, T.V., and Brian, P.F. (1992). Numerical recipes in C: The art of scientific computing, 2nd ed., Cambridge University Press, GB.
53. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. Journal of Science, vol. 220, pp. 671-680.
54. Kirkpatrick, S. (1984). Optimization by simulated annealing: quantitative studies. Journal of Statistical Physics, vol. 34, pp. 975-986.
55. Amanatides, J. (1984). Ray tracing with cones. ACM SIGGRAPH Computer Graphics, vol. 18(5), pp. 129-135.
56. Robert, L.C., Porter, T., and Carpenter, L. (1984). Distributed ray tracing. ACM SIGGRAPH Computer Graphics, vol. 18(3), pp. 137-145.
57. Kersten, D., David, C.K., Mamassian, B., and Bülthoff, I. (1996). Illusory motion from shadows. Nature, vol. 379, pp. 31-38.
58. Brajovic, V. (2004). Brightness perception, dynamic range and noise: A unifying model for adaptive image sensors. Computer Vision and Pattern Recognition, vol. 2(27), pp. 189-196.
59. Ha, H.N., Olivier, P. (2006). Explorations in Declarative Lighting Design. In Proceedings of the 6th International Symposium on Smart Graphics (SG), pp. 160-171.
60. Lowell, R. (1999). Matters of Light & Depth. Lower Light Management, New York, USA.
61. Supan, P., Stuppacher, I. (2006). Interactive image based lighting in augmented reality. In Proceedings of the Central European Seminar on Computer Graphics for students.
62. Petrou, M., Bosdogianni, P. (1999). Image Processing: The Fundamentals, John Wiley & Sons, GB.
63. Ha, H.N., Olivier, P. (2006). Perception-based lighting design. In Proceedings of Theory and Practice of Computer Graphics (TPCG), pp. 63-69.
64. Mullen, K. (1985). The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. Journal of Physiology, vol. 359, pp. 381-400.

65. Yantis, S., Jones, E. (1991). Mechanisms of attentional selection: Temporally modulated priority tags. Journal of Perception and Psychophysics, vol. 50. pp. 166-178.
66. Egusa, H. (1983). Effects of brightness, hue, and saturation on perceived depth between adjacent regions in the visual field. Journal of Perception, vol. 12, pp. 167-175.
67. Schwartz, B.J., Sperling, G. (1983). Luminance control of the perceived 3D structure of dynamic 2D displays. Journal of Psychonomic Society, vol. 21, pp. 456-458.
68. O'Shea, R., Blackburn, S.G., and Ono, H. (1994). Contrast as a depth cue. Vision Research, vol. 34, pp. 1595-1604.
69. Kanzisa, G. (1979). Organization in vision: Essays in Gestalt perception. Praeger Press, New York, USA.
70. Kanzisa, G. (1985). Seeing and thinking. Acta Psychologica, vol. 59, pp. 23-33.
71. Grossberg, S. (1994). 3D vision and figure-ground separation by visual cortex. Perception and Psychophysics, vol. 55, pp. 48-120.
72. Grossberg, S. (1997). Cortical dynamics of 3D figure-ground perception of 2D pictures. Psychology Review, vol. 104, pp. 618-658.
73. Dresp, B., Durand, V., and Grossberg, S. (2002). Depth perception from pairs of overlapping cues in pictorial displays. Journal of Spatial Vision, vol. 15(3), pp. 255-276.
74. Bruce, V., Patrick, R.G., and Mark, A.G. (1997). Visual Perception. Psychology Press, GB.
75. Ronald A. D. and Bradley, J.L. (1992). Image compression through wavelet transform coding. IEEE Transactions on Information Theory, vol. 38, pp. 719-746.
76. Chambolle, A., Bradley, J. L. (2001). Interpreting translation-invariant wavelet shrinkage as a new image smoothing scale space. IEEE Transactions on Image Processing, vol. 10, pp. 993-1000.
77. Namyong, L., Bradley, J. L. (2001). Wavelet methods for inverting the radon transform with noisy data. IEEE Transactions on Image Processing, vol. 10, pp. 79-94.

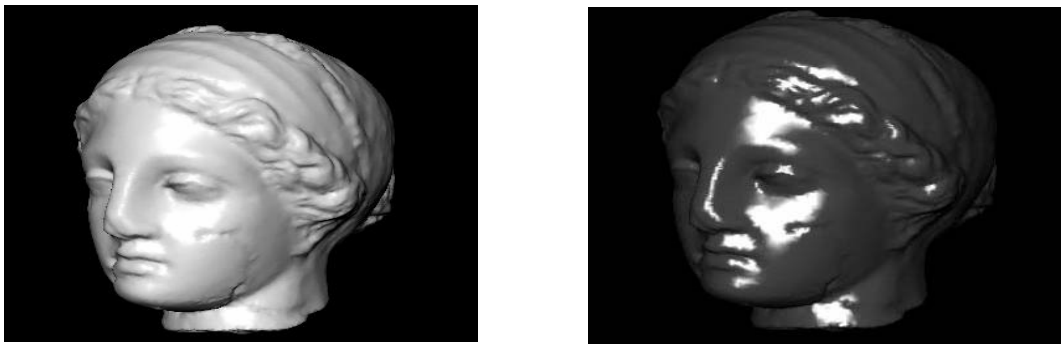
78. Do, M.N., Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. IEEE Transaction on Image Processing, vol. 11(2), pp. 146-158.
79. Tian, Q., Sebe, N., Lew, M.S., Loupias, E. and Huang, T.S. (2001). Image retrieval using wavelet-based salient points. Journal of Electronic Imaging, vol. 10(4), pp. 835-849.
80. Haar, A.Z. (1910). Theorie der orthogonalen Funktionensysteme. Journal of Math, vol. 69, pp. 331-371.
81. Crandall, R.E. (1996). Topics in advanced scientific computation. Springer-Verlag, USA.
82. http://www.dennisflood.com/photos/gallery/sausage_lake/large/sausage_lake_3-Shadows.jpg
83. http://www.shuttersstock.com/pic-1879689.html
84. Millerson, G. (2005). Lighting for television and film. Focal Press, 3rd ed., GB.
85. Boughen, N., 3DMax Lighting, Worldware Publishing, USA.
86. A Lighting Approach for Computer Graphics (SIGGRAPH 96 - Course 30)
87. Gershbein, R. and Hanrahan, P. (2000). A Fast Relighting Engine for Interactive Cinematic Lighting Design. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH).

Appendix A

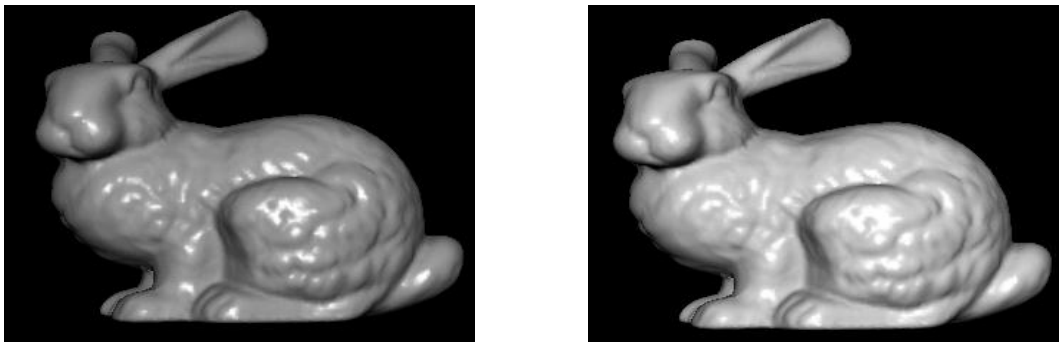
Models & target images for experiments 1, 2, and 3.



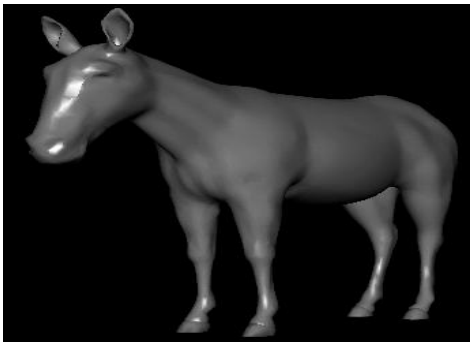
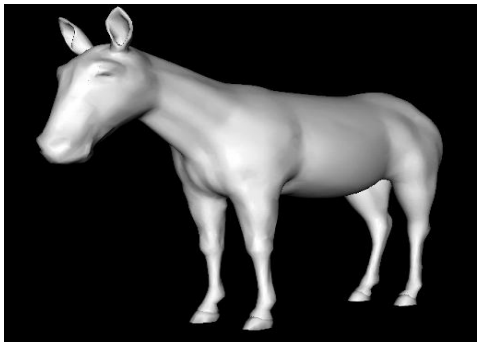
Elephant model with 10000 triangles was rendered with different lighting parameters.



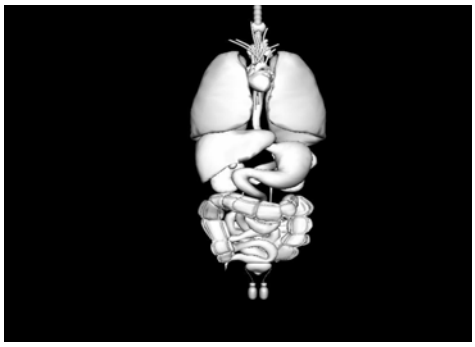
Head model with 67170 triangles was rendered with different lighting parameters.



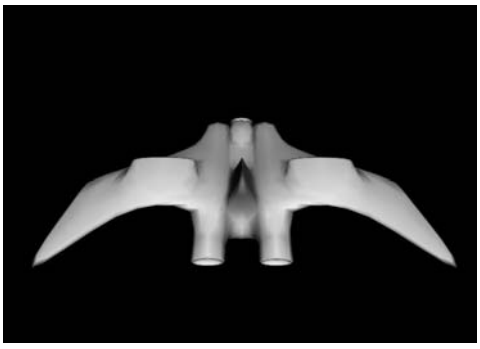
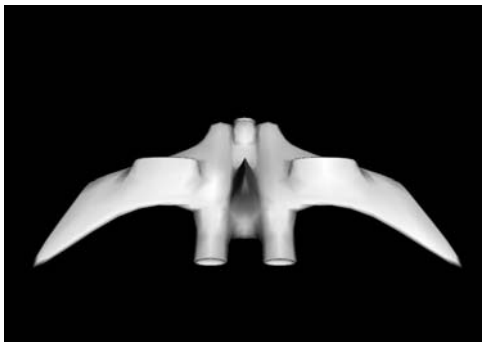
Bunny model with 69451 triangles was rendered with different lighting parameters.

Horse model with 12684 triangles was rendered with different lighting parameters.

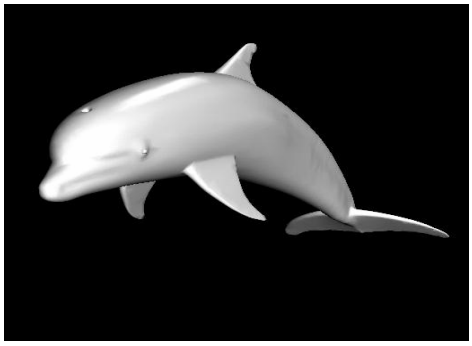
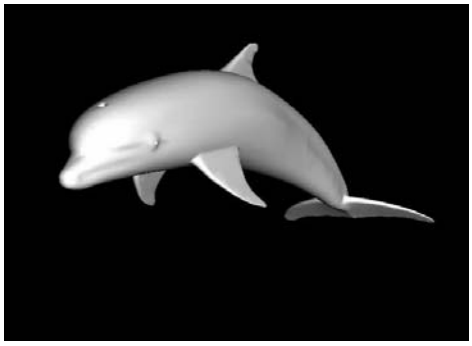



Body model with 129530 triangles was rendered with different lighting parameters.

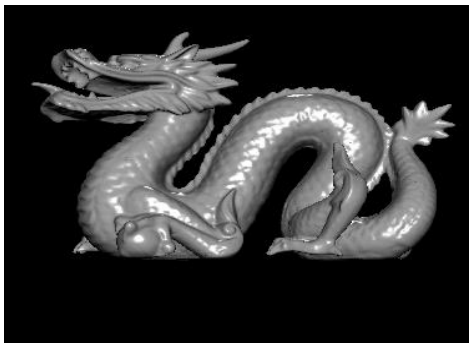
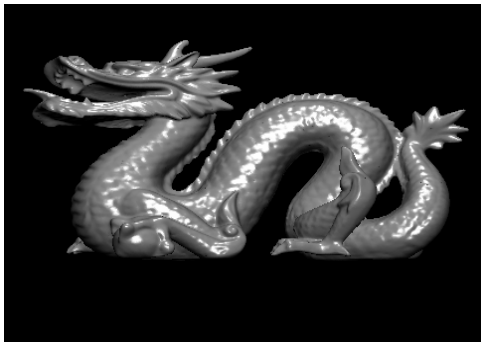



Air craft model with 2884 triangles was rendered with different lighting parameters.

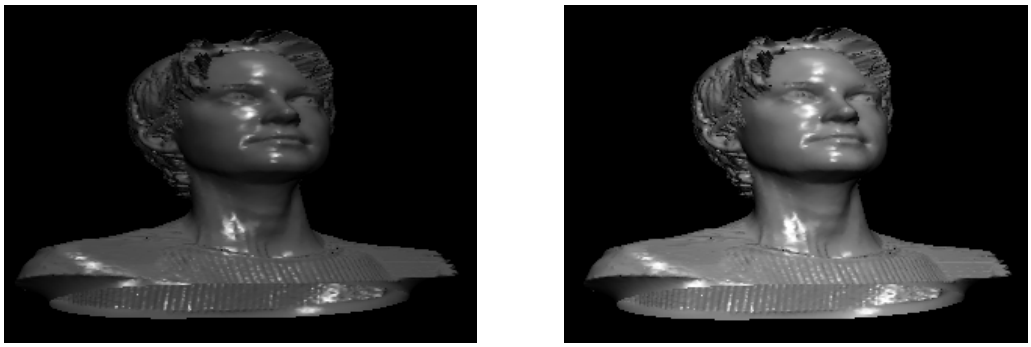

Buddha model with 1087716 triangles was rendered with different lighting parameters.

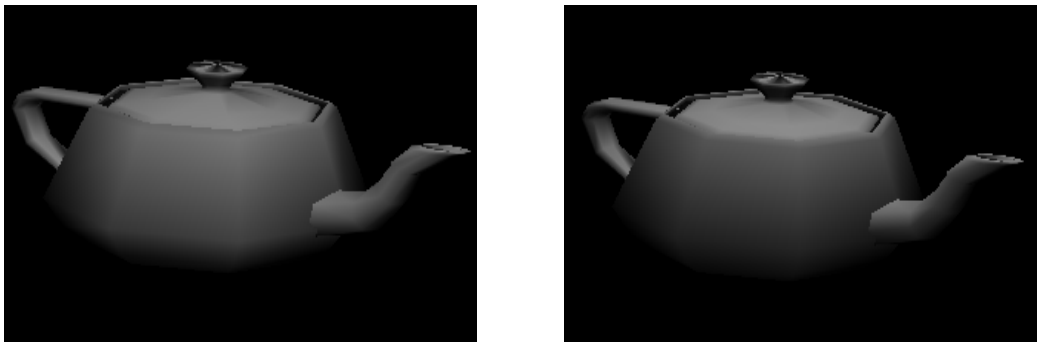
Dolphin model with 10000 triangles was rendered with different lighting parameters.

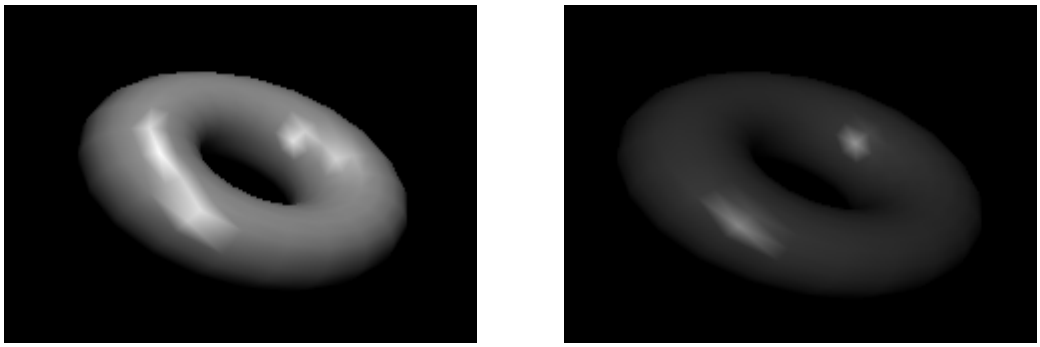
Dragon model with 871414 triangles was rendered with different lighting parameters.



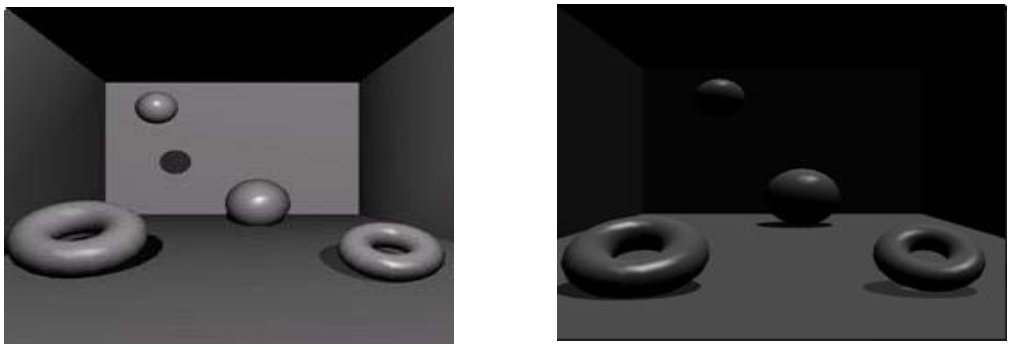
Haft female body model with 151270 triangles was rendered with different lighting parameters.



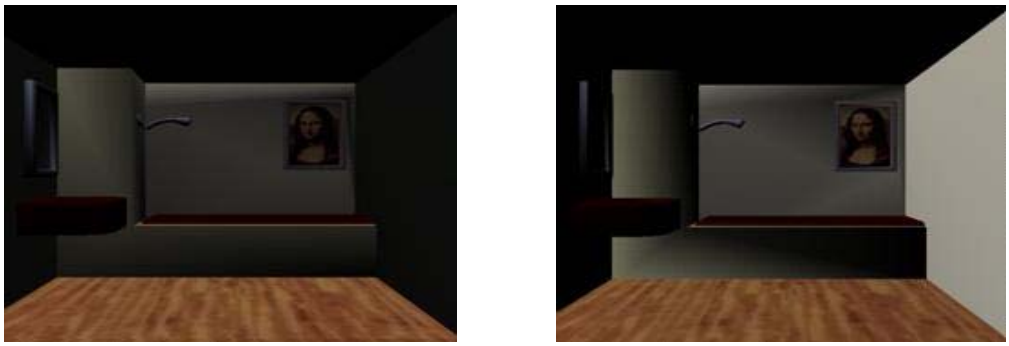
Pot model with 604 triangles was rendered with different lighting parameters.



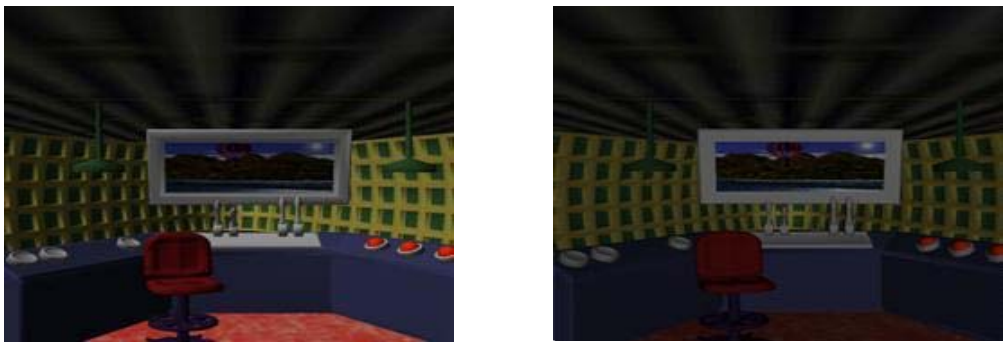
Torus model with 800 triangles was rendered with different lighting parameters.



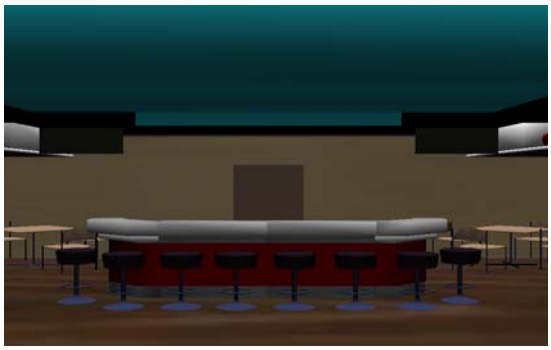
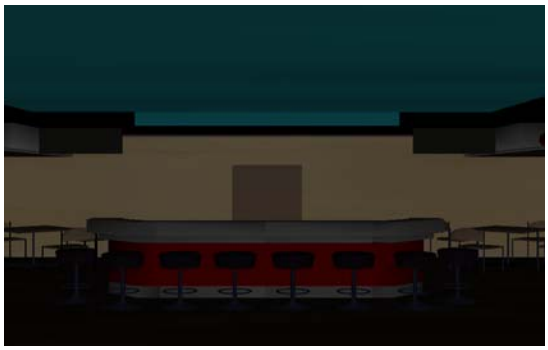
Synthesised scene with 61214 triangles was rendered with different lighting parameters.



Bathroom scene with 1092 triangles was rendered with different lighting parameters.



Cabin scene with 27706 triangles was rendered with different lighting parameters.

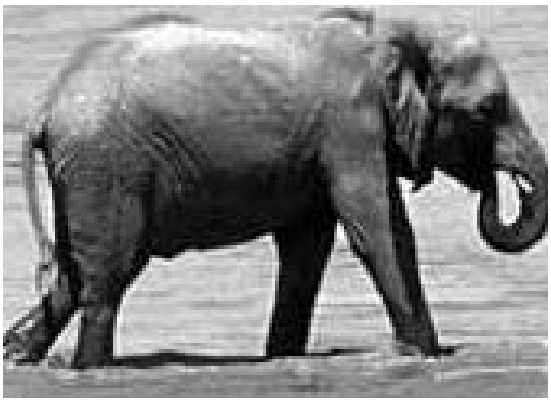



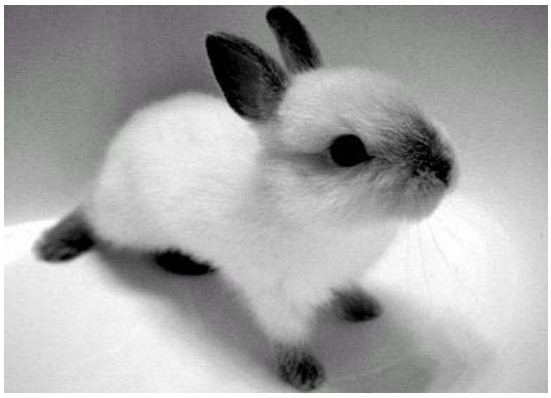
Bar scene with 24380 triangles was rendered with different lighting parameters.



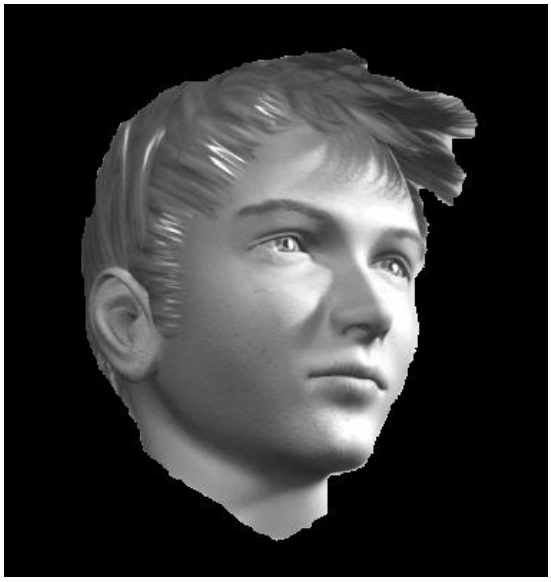



Images captured from the movies "Entrapment", "Secret Window", and "Cat Women" were used as the target exemplars in lighting-by-example.





Elephant and mouse photos were used as the target exemplars in lighting-by-example.



Face photos were used as the target exemplars in lighting-by-example.

Appendix B

Collected Data of Experiments

Collected Data of Experiment 1 (Question 1 to 21):

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
1	5	1	3	2	5	1	4	5	3	3	5	4	5	4	4	5	5	4	4	5
2	5	1	2	3	4	1	3	3	2	3	4	4	5	4	4	5	5	4	4	5
3	4	1	4	5	3	1	3	2	1	3	1	1	1	1	1	1	2	3	2	3
4	4	2	4	3	2	1	3	2	2	3	3	2	3	2	2	4	4	3	2	4
5	4	2	3	3	2	1	3	3	2	3	3	2	2	2	2	2	3	3	2	3
6	4	2	3	3	2	1	2	2	3	3	2	2	2	3	2	2	2	3	4	2
7	3	2	3	4	2	1	3	2	2	3	2	2	2	2	2	2	3	4	4	3
8	2	1	3	4	3	1	2	2	1	3	1	2	1	3	2	3	4	3	2	3
9	4	1	3	2	4	1	4	4	2	2	4	4	5	4	4	5	5	4	4	4
10	4	1	3	3	5	1	4	4	2	3	4	4	5	4	4	5	3	2	4	5
11	2	2	2	1	4	1	4	4	2	2	4	3	4	4	4	4	3	3	3	3
12	3	1	3	2	5	1	4	4	2	2	3	3	4	4	4	5	4	3	3	3
13	2	2	2	2	5	1	4	5	2	3	3	3	5	4	4	5	3	2	3	3
14	2	1	3	2	5	1	4	4	3	2	3	3	4	4	4	5	4	3	4	4
15	5	1	3	2	4	1	3	4	2	3	3	3	4	3	3	4	5	5	5	5
16	5	1	4	3	4	1	3	4	2	2	3	3	3	4	4	5	5	5	5	5
17	4	1	4	5	4	1	4	4	3	3	3	3	3	3	3	4	5	4	5	4
18	3	1	3	2	4	1	4	4	2	2	4	3	4	4	4	4	3	3	3	3
19	3	1	3	2	4	1	3	4	2	2	3	3	4	4	4	4	3	2	3	3
20	4	1	3	3	4	1	4	4	2	3	3	3	4	3	3	5	4	4	5	4
21	4	1	4	5	3	1	4	4	2	3	3	3	4	3	3	4	5	4	4	3
22	5	2	3	3	5	3	4	5	3	3	5	4	5	4	3	5	5	4	3	5
23	4	2	2	3	4	2	3	3	2	3	4	4	5	3	4	5	5	4	3	5
24	5	1	4	5	3	1	3	2	1	3	1	1	1	1	1	1	2	3	2	3
25	4	2	4	4	2	2	3	2	2	3	3	2	3	3	2	4	4	3	1	4
26	4	1	3	3	2	1	3	3	2	3	2	2	2	3	2	2	2	2	3	3
27	4	2	2	3	2	1	2	3	3	3	2	2	2	2	2	2	2	3	4	2
28	3	2	3	4	2	1	3	2	2	3	2	2	3	2	2	3	3	4	3	3
29	2	1	3	4	3	1	2	3	1	3	1	2	1	3	2	3	4	3	2	3
30	4	1	3	2	4	1	4	4	2	2	4	4	5	4	4	5	5	4	4	4
31	4	2	4	3	5	2	4	4	3	3	4	4	5	3	4	5	3	2	4	5
32	3	2	2	2	4	1	3	4	2	2	3	3	4	4	3	4	3	4	3	3
33	2	1	3	3	5	1	3	4	2	3	3	3	5	4	4	5	4	4	3	3
34	2	2	2	2	5	1	4	5	2	3	3	3	5	4	4	5	3	2	3	3
35	3	1	3	2	4	1	4	4	3	2	3	2	4	4	3	5	4	4	4	4
36	4	1	3	3	4	1	3	4	2	3	3	3	4	3	3	4	5	5	5	5
37	5	1	4	3	4	1	3	4	2	2	3	3	3	4	4	5	4	5	5	5
38	4	1	4	5	4	1	4	4	3	3	3	3	3	3	3	4	5	3	5	4
39	4	1	4	5	3	1	4	4	2	3	3	3	4	3	3	4	5	4	4	3
40	3	1	3	2	4	1	4	4	2	2	4	3	4	4	4	4	3	3	3	3

Collected Data of Experiment 1 (Question 22 to 42):

User	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Q35	Q36	Q37	Q38	Q39	Q40	Q41
1	4	4	3	1	5	4	4	4	5	3	4	1	1	2	2	2	1	3	3	3
2	5	5	4	1	5	5	4	4	4	3	3	1	1	3	2	2	1	3	3	3
3	3	3	2	1	1	3	3	3	1	3	1	1	1	1	1	1	1	5	3	1
4	4	4	5	2	3	3	5	3	4	3	4	1	1	1	4	2	1	5	4	3
5	2	2	2	1	1	2	2	2	2	2	2	1	2	2	3	3	1	4	3	2
6	2	2	3	3	2	3	3	4	2	3	3	2	2	2	2	2	1	2	3	4
7	4	4	4	2	2	2	2	2	2	2	3	1	1	1	3	2	1	3	3	3
8	4	2	3	1	2	1	2	1	3	2	3	1	1	2	3	4	1	2	3	2
9	5	5	5	1	5	4	4	3	4	4	5	1	1	1	1	1	1	3	3	2
10	5	5	4	1	5	5	4	4	5	4	4	1	1	1	2	2	1	3	3	4
11	4	4	3	1	4	4	5	4	4	4	4	1	1	1	2	1	1	2	3	4
12	4	4	3	1	5	4	4	3	4	4	4	1	1	2	2	1	1	2	2	3
13	3	3	3	1	4	4	4	3	4	3	3	1	1	2	2	1	1	2	2	3
14	4	4	3	1	4	3	4	4	4	3	3	1	1	2	2	2	1	4	4	3
15	5	5	4	1	4	4	5	4	4	4	4	1	1	1	2	2	1	2	3	3
16	5	5	4	1	5	4	4	3	4	3	3	1	1	1	2	3	1	3	4	4
17	5	5	3	1	4	3	4	3	4	4	4	1	1	1	1	2	1	3	3	3
18	3	3	3	1	4	5	4	4	4	4	4	1	1	1	2	1	1	3	3	4
19	3	3	2	1	4	4	3	3	3	4	4	1	1	1	1	1	1	3	2	3
20	4	4	3	1	3	4	4	4	4	3	3	1	1	1	2	2	1	2	2	3
21	4	4	3	1	4	4	3	3	3	3	4	1	1	2	2	2	1	2	2	2
22	4	4	3	1	5	4	4	4	5	3	4	1	1	2	2	2	1	3	3	3
23	5	5	4	1	5	5	4	4	4	3	3	1	1	3	2	2	1	3	3	3
24	3	3	2	1	1	3	3	3	1	3	1	1	1	1	1	1	1	5	3	1
25	4	4	5	2	3	3	5	3	4	3	4	1	1	1	4	2	1	5	4	3
26	2	2	2	1	1	2	2	2	2	2	2	1	2	2	3	3	1	4	3	2
27	2	2	3	3	2	3	3	4	2	3	3	2	2	2	2	2	1	2	3	4
28	4	4	4	2	2	2	2	2	2	2	3	1	1	1	3	2	1	3	3	3
29	4	2	3	1	2	1	2	1	3	2	3	1	1	2	3	4	1	2	3	2
30	5	5	5	1	5	4	4	3	4	4	5	1	1	1	1	1	1	3	3	2
31	5	5	4	1	5	5	4	4	5	4	4	1	1	1	2	2	1	3	3	4
32	4	4	3	1	4	4	5	4	4	4	4	1	1	1	2	1	1	2	3	4
33	4	4	3	1	5	4	4	3	4	4	4	1	1	2	2	1	1	2	2	3
34	3	3	3	1	4	4	4	3	4	3	3	1	1	2	2	1	1	2	2	3
35	4	4	3	1	4	3	4	4	4	3	3	1	1	2	2	2	1	4	4	3
36	5	5	4	1	4	4	5	4	4	4	4	1	1	1	2	2	1	2	3	3
37	5	5	4	1	5	4	4	3	4	3	3	1	1	1	2	3	1	3	4	4
38	5	5	3	1	4	3	4	3	4	4	4	1	1	1	1	2	1	3	3	3
39	4	4	3	1	4	4	3	3	3	3	4	1	1	2	2	2	1	2	2	2
40	3	3	3	1	4	5	4	4	4	4	4	1	1	1	2	1	1	3	3	4

Collected Data of Experiment 1 (Question 43 to 64):

User	Q43	Q44	Q45	Q46	Q47	Q48	Q49	Q50	Q51	Q52	Q53	Q54	Q55	Q56	Q57	Qu58	Q59	Q60	Q61	Q62	Q63
1	5	3	4	1	2	1	3	3	3	4	4	1	2	3	3	3	3	3	4	3	2
2	5	4	5	1	2	1	3	3	4	4	4	1	2	3	3	3	3	3	3	3	2
3	1	1	2	1	4	1	2	1	3	3	3	3	5	3	5	3	3	5	2	5	5
4	2	2	4	1	4	2	5	4	4	4	4	1	2	4	3	2	2	5	3	3	2
5	2	1	1	1	2	1	2	2	1	2	3	2	3	3	2	2	2	3	4	3	2
6	4	4	4	2	1	2	3	3	4	1	3	2	2	2	3	4	4	2	4	2	2
7	2	2	2	1	3	2	3	3	3	3	3	1	3	3	2	2	2	4	2	3	4
8	2	3	2	1	2	1	3	3	5	3	4	2	2	2	1	2	2	3	1	2	1
9	4	3	5	1	1	1	4	4	4	4	4	1	1	2	3	3	3	3	3	3	2
10	5	4	5	1	1	1	3	3	3	3	3	1	1	2	3	4	4	3	3	3	3
11	5	5	5	1	2	1	2	3	3	3	3	1	2	2	4	4	4	3	4	3	2
12	4	4	5	1	2	1	2	2	3	3	3	1	2	2	3	3	3	2	4	3	2
13	5	4	5	1	1	1	2	2	3	3	4	1	2	2	3	3	4	2	4	3	2
14	5	4	5	1	2	1	3	3	3	4	5	1	2	3	4	4	4	3	4	3	2
15	5	4	5	1	2	1	3	3	5	4	5	1	2	3	4	4	4	3	3	3	2
16	5	4	5	1	2	1	3	3	5	4	5	1	2	3	3	3	3	3	3	3	4
17	5	4	5	1	2	1	3	3	5	4	5	1	2	2	3	3	3	4	3	3	4
18	5	4	5	1	2	1	3	3	3	3	4	1	2	3	4	4	4	3	3	3	3
19	5	4	5	1	2	1	2	2	3	3	3	1	2	2	3	3	3	2	3	3	2
20	5	3	5	1	2	1	3	3	4	4	5	1	1	2	3	3	3	3	3	3	3
21	5	4	5	1	2	1	2	2	2	3	5	1	2	3	3	3	3	5	3	3	3
22	5	3	4	1	2	1	3	3	3	4	4	1	2	3	3	3	3	3	4	3	2
23	5	4	5	1	2	1	3	3	4	4	4	1	2	3	3	3	3	3	3	3	2
24	1	1	2	1	4	1	2	1	3	3	3	3	5	3	5	3	3	5	2	5	5
25	2	2	4	1	4	2	5	4	4	4	4	1	2	4	3	2	2	5	3	3	2
26	2	1	1	1	2	1	2	2	1	2	3	2	3	3	2	3	2	3	4	3	2
27	4	4	4	2	1	2	3	3	4	1	3	2	2	2	3	4	4	2	4	2	2
28	2	2	2	1	3	2	3	3	3	3	3	1	3	3	2	2	2	4	2	3	4
29	2	3	2	1	2	1	3	3	5	3	4	2	2	3	1	2	2	3	1	2	1
30	4	3	5	1	1	1	4	4	4	4	4	1	1	2	3	3	3	2	3	3	2
31	5	4	5	1	1	1	3	3	3	3	3	1	1	2	3	4	4	3	3	2	3
32	5	5	5	1	2	1	2	3	3	3	3	1	2	2	4	4	4	3	4	3	2
33	4	4	5	1	2	1	2	2	3	3	3	1	2	3	3	3	3	3	4	2	2
34	5	4	5	1	1	1	2	2	3	3	4	1	3	2	3	3	4	2	4	3	2
35	5	4	5	1	2	1	3	3	3	4	5	1	2	3	4	4	3	3	4	4	2
36	5	4	5	1	2	1	3	3	5	4	5	1	2	3	4	4	4	3	3	3	2
37	5	4	5	1	2	1	3	3	5	4	5	1	2	3	3	3	3	3	3	3	4
38	5	4	5	1	2	1	3	3	5	4	5	1	2	2	3	3	3	4	3	3	4
39	5	4	5	1	2	1	2	2	2	3	5	1	2	3	3	3	3	5	3	3	3
40	5	4	5	1	2	1	3	3	3	3	4	1	2	3	4	4	4	3	3	3	3

Collected Data of Experiment 2 (Question 1 to 15):

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
1	2	1	2	1	2	2	1	1	1	1	2	1	2	1
2	2	2	1	2	1	2	2	1	1	2	2	2	2	2
3	1	2	2	1	2	2	2	2	2	2	1	2	2	1
4	1	1	1	1	1	1	1	1	2	2	1	1	2	1
5	2	2	1	1	1	2	1	1	1	1	2	2	2	2
6	2	1	2	1	2	2	1	1	1	1	2	1	2	1
7	1	1	1	1	2	1	1	1	2	1	1	1	2	1
8	1	1	1	2	1	2	2	1	1	1	1	1	2	1
9	1	1	2	2	1	1	1	1	1	1	2	1	2	2
10	1	1	1	1	1	1	1	2	2	2	1	2	2	1
11	2	1	2	1	1	2	1	1	1	1	2	1	2	1
12	2	1	2	1	2	2	1	1	1	1	2	1	2	1
13	2	1	1	1	1	1	2	1	1	1	2	1	2	1
14	2	2	1	1	2	1	2	1	2	1	2	1	2	1
15	2	2	1	1	2	1	1	1	2	1	2	1	2	1
16	2	1	2	1	2	1	2	1	2	1	2	1	2	1
17	2	1	2	1	2	1	2	1	2	1	2	1	2	1
18	2	1	2	1	2	1	2	1	2	1	2	1	2	1
19	2	2	1	1	2	1	2	1	1	1	2	1	2	1
20	2	2	1	1	1	2	1	1	2	1	2	1	2	1
21	2	2	2	1	2	1	2	1	2	1	2	1	2	1
22	2	2	1	1	2	1	2	1	2	1	2	2	1	1
23	1	1	2	2	1	1	1	1	1	1	2	1	2	2
24	1	1	1	1	1	1	1	2	2	2	1	2	2	1
25	2	1	2	1	1	2	1	1	1	1	2	1	2	1
26	2	2	1	2	1	2	2	1	1	2	2	2	2	2
27	1	2	2	1	2	2	2	2	2	2	1	2	2	1
28	1	1	1	1	1	1	1	1	2	2	1	1	2	1
29	2	2	1	1	2	1	2	1	1	1	2	1	2	1
30	2	2	1	1	1	2	1	1	2	1	2	1	2	1
31	2	2	2	1	2	1	2	1	2	1	2	1	2	1
32	2	1	2	1	2	2	1	1	1	1	2	1	2	1
33	2	2	1	2	1	2	2	1	1	2	2	2	2	2
34	1	1	1	2	1	2	2	1	1	1	1	1	2	1
35	2	2	1	1	2	1	2	1	2	1	2	2	1	1
36	1	2	2	1	2	2	2	2	2	2	1	2	2	1
37	1	1	2	2	1	1	1	1	1	1	2	1	2	2
38	2	1	2	1	1	2	1	1	1	1	2	1	2	1
39	2	2	1	1	2	1	2	1	1	1	2	1	2	1
40	1	1	1	1	2	1	1	1	2	1	1	1	2	1

Collected Data of Experiment 2 (Question 16 to 30):

User	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29
1	1	1	1	2	1	2	1	2	2	1	2	2	1	2
2	1	1	1	2	2	2	2	1	2	2	2	2	2	1
3	2	2	2	1	2	2	1	2	1	2	1	2	1	2
4	1	2	1	1	1	2	1	1	2	1	2	2	2	2
5	1	2	1	2	1	1	1	1	1	1	2	2	1	2
6	1	1	1	2	1	1	2	1	1	1	2	2	2	1
7	1	2	2	2	1	1	1	1	1	2	2	1	2	1
8	1	1	1	1	1	2	2	1	2	1	2	1	1	1
9	1	1	1	2	1	2	2	2	1	1	2	1	2	1
10	2	2	2	1	2	2	1	1	1	1	1	2	1	2
11	1	2	1	2	1	2	1	2	1	2	1	2	1	2
12	1	1	1	2	1	2	1	2	1	2	1	2	1	2
13	1	2	1	2	1	2	1	2	1	2	1	2	1	2
14	1	2	1	2	1	2	1	2	1	2	1	2	2	1
15	1	1	1	2	1	2	1	2	1	2	1	2	1	2
16	1	1	1	2	1	2	1	1	2	1	1	2	1	2
17	1	2	2	2	1	2	1	2	1	2	1	2	1	2
18	1	2	2	2	1	2	1	2	1	2	2	1	1	2
19	1	1	1	2	1	2	1	1	2	1	2	2	1	2
20	1	2	1	2	1	2	1	2	1	2	1	2	1	2
21	2	2	1	2	1	2	1	1	1	2	1	2	1	2
22	1	2	2	1	1	2	1	2	1	2	1	2	1	2
23	1	1	1	2	1	2	2	2	1	1	2	1	2	1
24	2	2	2	1	2	2	1	1	1	1	1	2	1	2
25	1	2	1	2	1	2	1	2	1	2	1	2	1	2
26	1	1	1	2	2	2	2	1	2	2	2	2	2	1
27	2	2	2	1	2	2	1	2	1	2	1	2	1	2
28	1	2	1	1	1	2	1	1	2	1	2	2	2	2
29	1	1	1	2	1	2	1	1	2	1	2	2	1	2
30	1	2	1	2	1	2	1	2	1	2	1	2	1	2
31	2	2	1	2	1	2	1	1	1	2	1	2	1	2
32	1	1	1	2	1	1	2	1	1	1	2	2	2	1
33	1	1	1	2	2	2	2	1	2	2	2	2	2	1
34	1	1	1	1	1	2	2	1	2	1	2	1	1	1
35	1	2	2	1	1	2	1	2	1	2	1	2	1	2
36	2	2	2	1	2	2	1	2	1	2	1	2	1	2
37	1	1	1	2	1	2	2	2	1	1	2	1	2	1
38	1	2	1	2	1	2	1	2	1	2	1	2	1	2
39	1	1	1	2	1	2	1	1	2	1	2	2	1	2
40	1	2	2	2	1	1	1	1	1	2	2	1	2	1

Collected Data of Experiment 3 (Question 1 to 15):

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
1	2	1	1	2	3	2	2	3	1	2	2	1	2	1
2	2	1	2	2	3	2	2	3	1	2	2	1	2	1
3	3	2	1	2	1	2	2	1	1	2	3	1	2	1
4	2	1	1	2	3	2	2	3	1	2	3	1	2	3
5	2	3	1	1	1	2	3	3	1	2	3	1	1	3
6	2	3	1	2	3	2	3	3	1	2	3	1	2	3
7	2	3	2	2	3	2	2	3	1	2	3	1	2	3
8	2	3	1	2	3	1	2	3	1	2	3	1	3	3
9	2	3	1	2	3	2	2	3	1	2	3	1	3	3
10	2	3	1	2	3	2	2	3	1	2	3	1	1	3
11	2	3	1	2	3	1	2	3	1	2	3	1	1	1
12	2	3	2	2	3	2	2	3	1	2	2	1	2	3
13	2	3	1	2	1	2	2	3	1	2	3	1	1	3
14	2	3	1	2	3	2	2	3	1	2	3	1	2	3
15	2	3	1	2	3	2	2	1	1	2	3	1	2	3
16	2	3	1	2	3	2	2	3	3	2	3	1	3	3
17	2	3	1	2	3	2	2	3	1	2	2	1	1	3
18	2	1	2	2	3	2	2	3	1	2	2	1	2	1
19	1	1	1	2	3	2	2	3	1	2	2	1	2	1
20	2	2	1	2	1	2	2	1	1	2	3	1	2	1
21	2	1	1	2	3	2	2	3	1	2	3	1	2	3
22	2	3	1	1	1	2	3	3	1	2	3	1	1	3
23	2	3	2	3	3	2	3	3	1	2	3	1	2	3
24	2	3	1	2	3	2	2	3	1	2	4	1	2	3
25	3	3	1	2	3	1	2	3	1	2	3	1	3	3
26	2	3	1	2	3	2	2	3	1	2	3	1	3	3
27	2	3	1	2	3	2	2	3	1	2	3	1	1	3
28	2	3	1	2	3	1	2	3	1	2	3	1	1	1
29	2	3	21	2	3	2	2	3	1	2	2	1	2	3
30	2	3	1	2	1	2	2	3	1	2	3	1	1	3
31	2	3	1	2	3	2	2	3	1	2	3	1	2	3
32	2	3	1	2	3	2	2	1	1	2	3	1	2	3
33	2	3	1	2	3	2	2	3	3	2	3	1	3	3
34	2	3	1	2	3	2	2	3	1	2	2	1	1	3
35	1	3	1	2	3	2	3	3	1	2	3	1	2	3
36	2	3	1	2	3	2	2	3	1	2	3	1	2	3
37	2	3	2	2	3	1	2	3	1	2	3	1	3	3
38	2	3	1	2	3	2	2	3	1	2	3	1	3	3
39	2	3	1	2	3	2	2	3	1	2	3	1	1	3
40	2	3	1	2	3	2	3	3	1	2	3	1	2	3

Collected Data of Experiment 3 (Question 16 to 30):

User	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29
1	2	3	1	3	3	1	2	3	1	2	3	1	2	3
2	2	3	1	3	3	1	2	3	1	2	3	1	2	3
3	3	1	3	3	3	2	3	3	3	2	3	1	3	3
4	2	3	1	2	3	1	2	3	1	2	3	1	2	3
5	2	3	1	3	3	2	2	1	1	2	3	1	2	3
6	2	3	1	2	3	1	2	3	1	2	3	1	2	3
7	2	3	1	2	3	1	2	3	1	2	3	1	2	3
8	2	3	1	1	3	1	2	3	1	2	3	1	2	3
9	2	3	1	2	3	1	2	3	1	2	3	1	2	3
10	2	3	1	2	3	1	2	3	1	2	3	1	2	3
11	2	3	1	2	3	1	2	3	1	2	3	1	2	3
12	2	3	1	2	3	1	2	3	1	2	3	1	2	3
13	2	3	1	3	3	1	2	3	1	2	3	1	2	3
14	2	3	3	1	3	1	2	3	1	2	3	1	2	3
15	2	3	1	2	3	1	2	3	1	2	3	1	2	3
16	2	3	1	1	3	1	2	3	1	2	3	1	2	3
17	2	3	1	1	3	1	2	3	1	2	3	1	2	3
18	2	3	1	3	3	1	2	3	1	2	3	1	2	3
19	2	3	1	3	3	1	2	3	1	2	3	1	2	3
20	3	1	3	3	3	2	3	3	3	2	3	1	3	3
21	2	3	1	2	3	1	2	3	1	2	3	1	2	3
22	2	3	1	3	3	2	2	1	1	2	3	1	2	3
23	2	3	1	2	3	1	2	3	1	2	3	1	2	3
24	2	3	1	2	3	1	2	3	1	2	3	1	2	3
25	2	3	1	1	3	1	2	3	1	2	3	1	2	3
26	2	3	1	2	3	1	2	3	1	2	3	1	2	3
27	2	3	1	2	3	1	2	3	1	2	3	1	2	3
28	2	3	1	2	3	1	2	3	1	2	3	1	2	3
29	2	3	1	2	3	1	2	3	1	2	3	1	2	3
30	2	3	1	3	3	1	2	3	1	2	3	1	2	3
31	2	3	3	1	3	1	2	3	1	2	3	1	2	3
32	2	3	1	2	3	1	2	3	1	2	3	1	2	3
33	2	3	1	1	3	1	2	3	1	2	3	1	2	3
34	2	3	1	1	3	1	2	3	1	2	3	1	2	3
35	2	3	1	2	3	1	2	3	1	2	3	1	2	3
36	2	3	1	2	3	1	2	3	1	2	3	1	2	3
37	2	3	1	1	3	1	2	3	1	2	3	1	2	3
38	2	3	1	2	3	1	2	3	1	2	3	1	2	3
39	2	3	1	2	3	1	2	3	1	2	3	1	2	3
40	2	3	1	2	3	1	2	3	1	2	3	1	2	3