A PAINTING INTERFACE FOR ROOM ILLUMINATION BY ROBOTIC LIGHT ARRAY

コンピュータ制御によるグリッド状の照明の操作のためのペインティングインタフェース

by

Seung-Tak Noh
盧 承鐸

A Master Thesis
修士論文

Submitted to
the Graduate School of the University of Tokyo
on August 29th, 2012
in Partial Fulfillment of the Requirements
for the Degree of Master of Information Science and Technology
in Computer Science

Thesis Supervisor: Takeo Igarashi 五十嵐 健夫
Professor of Computer Science

ABSTRACT

We propose a painting interface that enables users to design an illumination distribution for a real-world room using an array of computer-controlled lights. Users specify which area of the room is to be well-lit and which is to be dark by painting a target illumination distribution on a tablet device displaying the image obtained by a camera mounted on the ceiling. The painting result is overlaid on the camera image as contour lines of the target illumination intensity. The system then runs an optimization to calculate light parameters to deliver the requested illumination condition. We implemented a GPU-based parallel search to achieve real-time optimization. In our system, we used aimable lights that can change the lighting direction to generate the requested illumination condition more faithfully than static lights. We built a miniature-scale experimental environment and ran a user study to compare our method with a standard direct manipulation method using GUI widgets. The experimental results showed that the users preferred our method for informal light control.

論文要旨

本研究では、コンピュータ制御されるグリッド状の照明を利用し、実世界における室内照明の制御を行うためのペインティングインタフェースを提案する。本システムでは、天井に設置されたカメラから得られた照明状態が画像としてタブレット上に表示され、ユーザはこの画像の上でどの領域を明るくしてどの領域を暗くするかをペインティングで指定する。ユーザの行ったペインティングの結果はカメラ画像上に等高線として表示される。システム側はユーザの求める結果が得られるような照明パラメータを最適化によって求める。この最適化の実装では、GPUによる並列探索を用いてリアルタイム処理を可能とした。 本システムは、通常の固定された照明でなく、向きを変えることのできる照明機構を用いることにより、より忠実にペイントされた結果を再現することができる。 提案手法の評価に当たって、ミニチュア規模の実験環境を作り、通常のGUIウィジェットによる照明の直接操作インタフェースと比較するユーザスタディを行った。その結果、ユーザは正確さの要求されない照明の操作においては提案手法を好むことが分かった。

Acknowledgements

It would have been impossible to complete this thesis without the help of many people to whom I am indebted.

First and foremost, I would like to express my deepest gratitude to Prof. Igarashi, who is my academic adviser. Without his insightful and rigorous advices, I could have not been achieved this thesis. Not only as an academic adviser, he is the most creative and hard-working person who I have ever met. He will forever be an inspiration to me during my life.

Second, I would like to appreciate to coauthors of this work. I would like to thank Sunao Hashimoto, who showed a good starting point to this research theme. I also appreciated using his miniature scale prototype implementation with his courtesy. I would like to thank Daiki Yamanaka, who supported the first implementation of preliminary interface and making the video for submission. I would also like to Prof. Masahiko Inami, who gave a hint of a single screen idea. His criticism was on our first system was painful at first time, but it revealed that it was a necessary step in making more advanced system. Without discussions with them and their coorporation and advices, this research could not have been completed.

Third, I would like to thank the participants of my research. I would like to thank Daniel Saakes, Mi Haipeng, Kouki Nomura, Jordan Tewell, and Ryohei Suzuki, who are the participants of informal user study and made useful comments. Especially Daniel, I owe him a very important debt on correcting the manuscript during the rebuttal period. I would also like to appreciate Kentaro Ishii, Keita Watanabe, Makoto Nakajima, Akira Ohgawara, Yutaro Hiraoka, Yuki Koyama, Genki Furumi, Naoki Sasaki, Fangzhou Wang, and Naoki Ishikawa, who are the participants of my final user test and give priceless comments on the research.

Fourth, I would like to thank the former and present members of User Interface Laboratory and JST ERATO Igarashi Project, who encourage to keep this research. I would like to thank Assistant Prof. Daisuke Sakamoto, Kenshi Takayama, Nobuyuki Umetani, Jun Kato, who are or had been members of User Interface Laboratory. I would also like to thank Myung-geol Choi and Youichi Kamiyama who are or had been members of JST Erato Igarashi Project. Especially, I would like to appreciate Daisuke Sakamoto, who gave me a priceless comments on user test evaluation and impress me with positive attitude.

Finally, I would like to thank my family for their financial support and encouragements. For two and a half year during my master course, I almost depended on my family for making my studying and living in Japan.

This work was funded by grants from JST (Japan Science and Technology Agency) Erato Igarashi Project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Light is indispensable element for the visual perception. Unlike outdoor environment where the main light sources are external lights such as sunlight, the light source for indoor environment is light appliances equipped in environment in advance. This thesis discusses the user interface for this important lighting task.

The currently dominant user interfaces for lighting control are uniform switches and dimmers. These interfaces are not problematic when the room has few lights within a relatively small area. However, such interfaces become inconvenient when the number of lights increases. In addition, when the position and direction of the lights are controllable, the degree-of-freedom (DOF) of the lighting design increases, making control even more difficult. A typical lighting console used in specialized facilities such as theaters is shown in (Figure 1.1). There are numerous widgets on this console for controlling detailed lighting.

Figure 1.1: Conventional lighting console (Behringer EUROLIGHT LC2412)

There are two fundamental problems in such interfaces, according to [18]; grouping and mapping problem. Grouping problem is caused by difficulties in establishing correspondence between switch and function. Mapping problem is caused by difficulties in establishing correspondence between switch and light. We do not know these correspondences until undergoing several trials. For reducing this complexity, it is possible to assign a switch to a predefined group of lights, but then it becomes difficult to achieve customized control. Adaption to this interface requires long training so it is not practical for casual usage by novice users. Even for professional users, they must need time for learning to control other environments where they are not familiar with. How can we address this complexity in user control of lighting?

In this thesis, we propose a painting interface that enables users to design an illumination distribution for a real-world room using an array of computer-controlled lights. Users specify which area of the room is to be well-lit and which is to be dark by painting a target illumination distribution on a tablet device displaying the image obtained by a camera mounted on the ceiling. The painting result is visualized as contour lines of the target illumination intensity overlaid on the camera image. The system then runs an optimization to calculate light parameters to deliver the requested illumination condition.

Our method is inspired by the goal-based lighting optimization approach in computer graphics [10, 19, 23, 25]. We have adapted this approach for lighting control in a living space and address the problems observed in the previous methods. In conventional painting interfaces, the user paints a target image that is separate from the image showing the rendered result. In contrast, in the proposed method, we continuously show the real illumination result captured by a camera, and the painting result is shown as contour lines overlaid on top of the camera image. This approach makes it easier for the user to understand the relation between user input and the system result. Another difference is that while conventional painting interfaces have the user paint absolute pixel values, we have the user paint illumination intensity values relative to the maximum brightness possible with the given lights. This prevents the user from painting an unrealistic target illumination distribution. To achieve real-time interaction with user painting, we implemented a GPU-based parallelized search.

We built a miniature-scale environment featuring twelve lighting units with intensity and orientation that are controlled independently. We ran a user study comparing our method with a standard direct manipulation method using sliders. The target illumination condition was given through natural language instructions such as “Illuminate this table”. The test users preferred our painting interface to direct manipulation for such informal light control tasks.

Our technical contributions are summarized as follows:

- We present a room lighting system which is controlled by user painting. We also tested for its feasibility with a miniature-scale prototype (Chapter 3).
- We present a painting interface that enables users to design the room lighting configuration. Unlike previous painting interface for lighting, our painting interface uses a single screen for painting and visualizing the result to mitigate the user confusion. In our interface, painting result is overlaid on top of camera images as contour lines (Chapter 4).
- We introduce GPU-based parallel search for achieving real-time optimization with user painting (Chapter 6).
- We compared our painting-based lighting control interface with conventional interface which consist of GUI widgets, and confirmed that our interface is better than conventional interface. (Chapter 7).

Chapter 2

Related Work

In this chapter, we review the previous researches on lighting control. We classify them into four classes; goal-based lighting design, real-world object illumination, lighting simulation, and lighting control system. In the goal-based lighting design section we describe the lighting control except of control lighting parameter directly. In real-world object illumination section, we describe several systems for illumination system in real world. Lighting simulation is used for predicting the lighting result in our system. We also provide the information of sensor network approach in the lighting control system section.

2.1 Goal-based Lighting Design

Goal-based lighting is the approach which gets the goal as input instead of raw lighting parameter and generates a lighting result that satisfies the input features by computer. By its industrial needs, this approach mainly researched in a computer graphics field.

The system proposed in [10] enables users to indicate the desired illumination on a known 3D model by sketching, after which a knowledge-based lighting design system recommends which lights to use and where to place them. Pellacini et al. [24] presented a user interface for controlling lighting effects by interacting with the rendered image: for example, by dragging a shadow. The approach by Shacked and Lischinski [26] automatically determines the values of various lighting parameters by optimizing a perception-based image quality metric. Painting interfaces for lighting design have been presented for many systems [19, 23, 25] and tend to feature users painting the desired illumination result and the system computing a lighting configuration by optimization. Kerr and Pellacini [12] observed that such painting interfaces are not effective for reproducing a given result, although they might be suitable for quick, exploratory design. Recently, Pellacini [22] presented a method to quickly edit captured environment maps.

We also take a goal feature as input in our painting interface, but our painting result is achieved in a real-world environment, not in a virtual environment.

2.2 Real-world Object Illumination

Researchers have also developed various systems to control real-world illumination. Ghosh et al. [7] proposed a method for actively controlling the illumination in a real room so that it is consistent with a virtual world in a 3D game. Amano and Kato [1] used a projector camera system to achieve irradiance correction and appearance enhancement for real scenes. The light stage system [4] controls the spherical array of lights to reproduce environmental light for live acting. These researches have quite different standpoint on lighting, so it is hard to compare with our approach directly.

There exist few researches which have similar concept that use painting as an input. Anrys and Dutré [2] presented a system for controlling an array of colored lights around an object. Mohan et al. [16] presented a system for synthesizing a photograph of an object by blending images captured with varying illumination conditions obtained using an actuated light.

In both systems, the purpose is achieving the artistic representation of an object with lighting. In this context, input painting is a means of conveying the result, so modification will be restricted in small area and user already know the exact same solution is not achieved in real-world lighting result. Taking sufficient time to optimize is also justified in this posing of a problem.

By contrast, we use the painting interface as substitution of lighting controller instead of switches or dimmers in a living environment, so the area of modification becomes the entire scene. If we use the absolute illumination intensities in our system, then user can easily paints the infeasible illumination result. Our method uses relative illumination intensity to avoid this problem. In our context, the real time performance also becomes important factor because nobody expects much response time in controlling switches or dimmers.

2.3 Lighting Simulation

Lighting simulation is mainly researched for generating photorealistic images, and it is also a large research topic itself. We are not aim to generate of realistic image but aim to predict the lighting result.

An aim of lighting simulation in the real world is often to measure a detailed appearance model of the target object (reflectance field), as in [3, 21, 30]. These methods are useful for synthesizing images from arbitrary viewing angles and lighting conditions, but the capture process is rather expensive. We, on the other hand, use simple image-based synthesis and search within the space composed of the synthesized images to bypass expensive measurement procedures, as in [2, 16].

Sheng et al. [27] presented an augmented reality interface for architectural daylighting design. With this interface, users build a physical prototype of a building by placing walls in the desired locations, and the system then captures the geometry and simulates the lighting in the environment. The result of the simulation is shown to the user by projecting illumination distribution to the walls of the physical prototype.

2.4 Lighting Control System

Lighting control has also been investigated in the context of sensor network systems. Park et al. [20] presented a lighting system for stage lighting in which the user specifies the desired lighting result by using a formal language and the system runs optimization to find a lighting configuration that can deliver the desired result. Singhvi et al. [28] presented an office lighting system that optimizes user comfort and energy usage. Their algorithm relies on the locality of each light. Wen and Agogino [29] presented a similar system that considers global lighting effects. In their system, the input is the desired light intensity at specific office worker locations; it does not provide painting and optimization on a radiance map obtained from a ceiling camera. Finally, we should note that all the systems mentioned here use fixed lights that cannot change their angles.

Chapter 3

System Overview

Our proposed system consists of multiple aimable lights, a ceiling mounted camera, a computer, and a painting interface (Figure 3.1). Each light is able to change its direction and has a micro-computer that controls the brightness and angle. Lights can be controlled individually by an input signal given by the computer. The image obtained from the camera is used to measure the radiance of the scene, after which the user can design the illumination distribution of the room with the user interface provided by the computer.

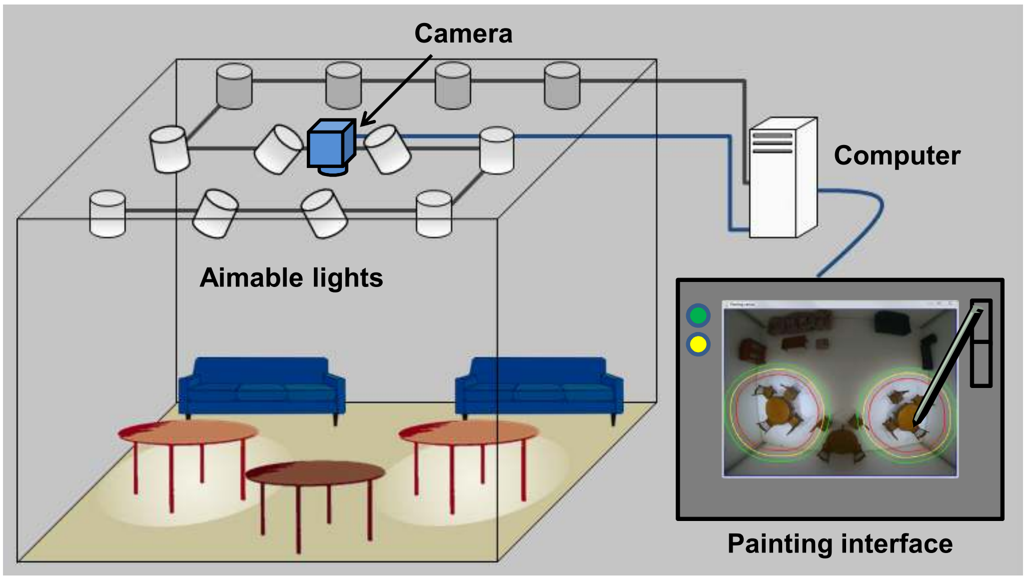


Figure 3.1: System overview

The control system consists of the painting interface, an optimization module, and a light control module. The system runs automatic calibration once when it is installed or when the furniture arrangement is changed. Calibration process consumes much time (about 15 minutes), but we think this is acceptable because it can run at anytime when no one is using the room. The painting interface allows the user to design the illumination distribution of the room by painting the desired illumination result directly onto the image. The optimization module calculates the parameters of the lights that will satisfy the desired illumination distribution request. The light control module then sends the calculated parameters to the lights.

3.1 Algorithm Overview

In our system, the user can design the illumination distribution by painting interface (Chapter 4). The overall algorithm is shown in Figure 3.2. When it is installed in an indoor environment or the furniture arrangement is changed, the system runs the calibration. We use an image-based simulation. The system illuminates the room with selected lighting configurations (brightness, angle) and captures the resulting radiance maps. This database is used to predict the illumination result for a given lighting configuration (Chapter 5). In a runtime, the user specifies an illumination distribution map by painting interface. The system then creates a user request illumination distribution based on the painted distribution map and base illumination which is captured during the calibration. The system searches for a configuration that minimizes the sum of the differences between the user request and the predicted illumination result (Chapter 6).

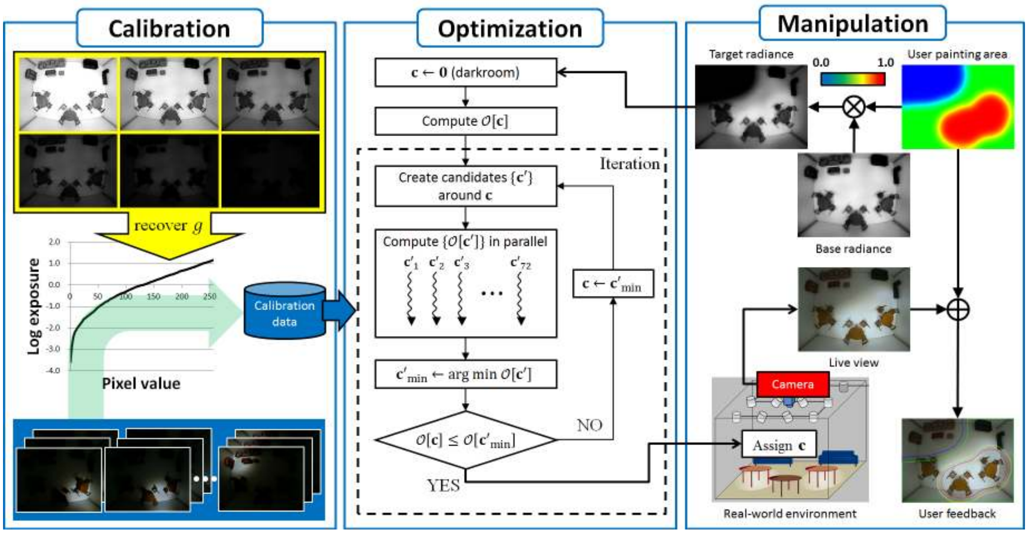
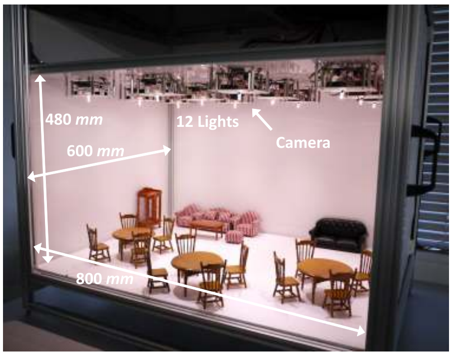


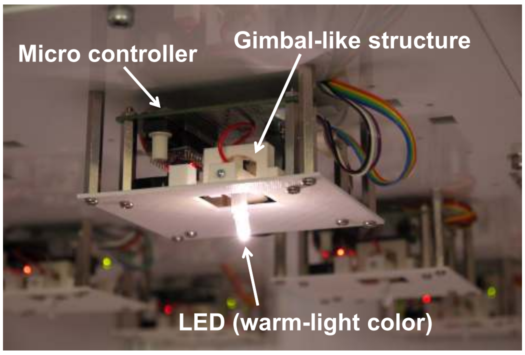
Figure 3.2: Algorithm overview

3.2 Miniature Scale Environment

We built a miniature room at 1/12 scale ($800 \times 600 \times 460$ mm^3; $9.6 \times 7.2 \times 5.5$ mm^3 in real scale), shown in Figure 3.3. Twelve lights are mounted on the ceiling at even intervals of 200mm. The resolution of each light direction is 5×5 (-25, -15, 0, 15, and 25 degrees for the x and y directions) and the resolution of light brightness is 9 (0, 32, 64, 96, 128, 160, 192, 224, and 255 for 8-bit input value of light). We mounted a wide range camera (Watec WAT231S2 camera with TAMRON 13FM22IR lens) to the center of the ceiling to capture the top-down view image. The miniature room is a box enclosed by six white panels to shield it from environment lights. Each lighting unit consists of a light-emitting diode (LED), two servo motors, a micro controller (Arduino), and a mechanical structure. We opted for a warm light color LED (NICHIA NSPL510DS). The viewing angle of the LED is 50-degree. The gimbal like structure and two servo motors provide two rotational DOFs for controlling the direction of the light. These units communicate with the host computer via serial bus.



(a)



(b)

Figure 3.3: (a) Miniature scale environment. (b) Each light can change its direction.

All data in this thesis are generated from this environment; however, we believe that our approach has generality in every indoor environment.

Chapter 4

User Interface

We initially implemented a painting interface with two screens, but we obseved several usability problems. Hence, we merged separated screen into a single screen. We first explain the two screen system and its problems. Next, we describe our single screen system.

4.1 Two Screen User Interface

Overview

In our preliminary study, we implemented a painting interface for lighting referring previous researches [25, 2, 12, 16, 23]. We call it as *two screen system* in this thesis. Figure 4.1 shows our two screen system. In this implementation, we have two similar but different screens. The left window shows the live view from the ceiling mounted camera (a), the right window shows the painting canvas where the user specifies the illumination request (b), and the bottom window shows the tool panel with basic painting tools (brush, squirt, and clear) (c). We also attached registration function for using the same setting after that time.

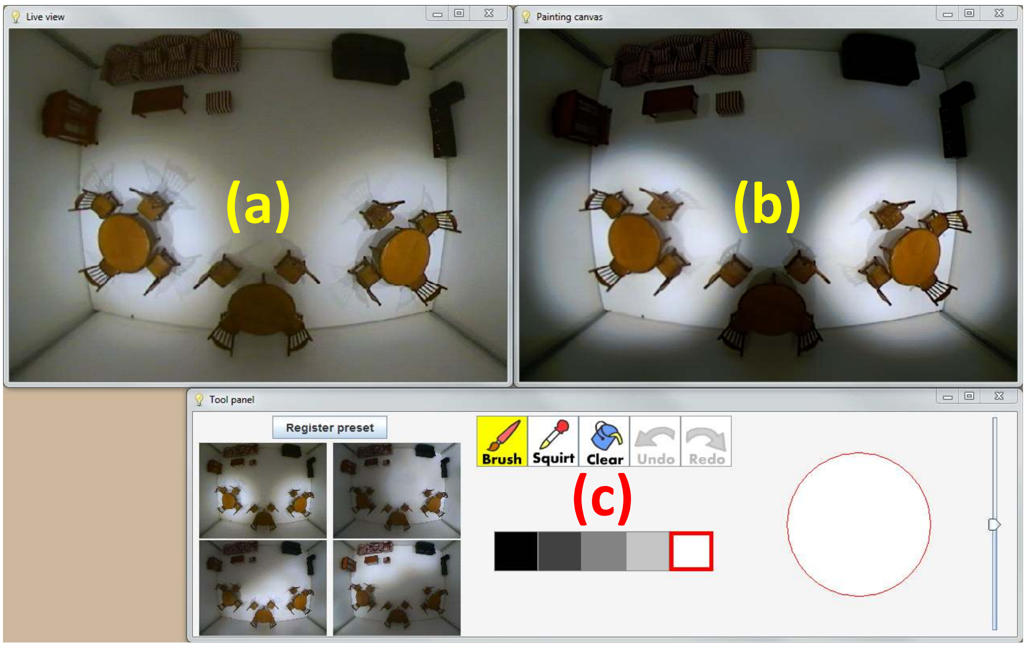


Figure 4.1: Our preliminary user interface

In our two screen system, we were not only following previous researches, but also devised three ideas in painting canvas to mitigate *sketch rather than paint* problem identified by [12].

First, we restricted user painting domain. In our two screen system, the user can paint radiance intensity relative to the radiance captured during calibration. Roughly speaking, zero corresponds to complete darkness and one corresponds to a fully lit image. Previous painting interfaces for lighting design usually have the user directly specify absolute pixel values, as in standard paint systems such as PaintBrush, PhotoShop, and GIMP. The problem with these methods is that it is quite easy for users to paint unrealistic results with these direct paint methods. For example, occlusion keeps some part of the room from becoming very brightly, the user can paint it very bright. In contrast, in our interface the user is not allowed to paint such a location brighter than can be obtained. According to this modification, the meaning of color in the tool panel is changed to the brightness level in our painting interface.

Second, we devised few ideas in brush manipulation. The brush size is set intentionally large (a circle of 160×160 to 240×240 pixels) and its boundary is blurred since it is impossible to reproduce a detailed illumination result with limited DOF lights, and therefore a detailed painting with a small, sharp brush would result in unrealistic expectations. The large brush also encourages the user to paint over a large area. After each user manipulation such as stroke, user can acquire feedback on both of painting canvas and live view. The result of painting canvas is generated by virtual photograph algorithm (will be described in Chapter 5) based on user painting under the same shutter speed with the camera. The system takes this painting result as target and adjusts the room light configuration (brightness and angle) as output so that the illumination result shown in the camera view matches the one on the painted canvas as closely as possible via optimization.

Third, we set optimization run interactively; that is, the system continuously updates the room light configuration during as the user paints so that the user can use the feedback to steer the optimization toward a more satisfactory result [12].

Informal User Feedback

We asked five test users (three students and two researchers in computer science), who are different participants with our final user test, to try our two screen system and asked them to paint various illumination configurations. They generally accepted the concept of system and found the ability to control room lighting with painting interface intriguing. Although users accepted our concept finally, they also complained several points of our preliminary system. Here we summarize comments.

- Two screens: The major complain was about the visual difference between painting canvas and live view. Many testers complained infeasible painting is achievable on the canvas but system still works and achieve weird result by this painting. They said that the feedback on canvas by virtual photograph makes an impression of simulation, but they could not confidence it because infeasible painting can be achieved. Furthermore, few of them had a doubt on the necessity of painting canvas feedback. They said that canvas has no meaningful information when the painting is obviously different. They also informed that they hesitated to find input screen when the painting is near to live view.

- Color as brightness level: Testers accepted the concept our color as brightness level without reluctance, on the contrary to the existence of two screens.

- About brush: Testers generally found the current maximum brush size is appropriate, but they complained that the minimum brush size was too large to paint details. They were aware that small painting cannot be achieved in real world lighting, but they need small brush for detailed editing.

- Setting registration: They all identified the necessity to be able to register painted results and select them later instead of painting from scratch every time.

Problems of Two Screen System

We estimated that our two screen system had a problem in fundamental level. At least, all of concepts were not problematic because some part of the system (i.e. color as brightness level) was accepted smoothly. We focused on difference between live view and painting canvas.

Figure 4.2 illustrates an example of visual difference situation. Note that, although our interface is designed to restrict the user's painting to within the realm of possibility for physical lights, some of the painting is still not feasible. The main issue is difference of the scope of influence. In the painting manner, the scope is the size of brush; however, in lighting, enclosed environment makes the entire scene be affected even by very small illumination. If he or she has no previous knowledge on this effect, one would have a sense of incongruity on canvas feedback. It is also unreasonable assumption that user can paint the lighting scene with consideration of those effects. We may adapt these effects approximately by sophisticated algorithm, however, it does harm in painting manner itself; It is no longer painting at all.

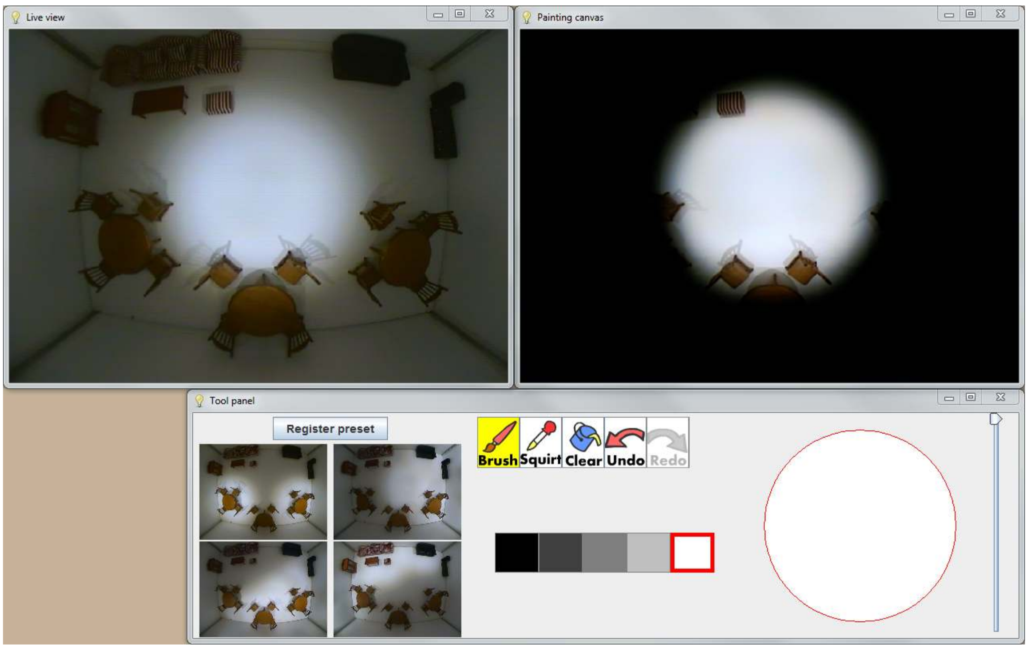


Figure 4.2: Problemetaic situation in two screen system

It is also questionable that the optimizer can or cannot search the same painting in the searching space. In other words, it is prerequisite knowledge that painting is just target in optimization and it cannot be achieved in lighting situation in two screen system. In this circumstance, placing two different screens side by side causes to make user focus on the difference, not on the resemblance.

4.2 Single Screen User Interface

We observed usability problems with the two screen system has a problem in way of user interaction. The feedback on canvas does not have a merit but causes of confusion. We therefore merge the live view and painting canvas, without changing the main framework.

In merging the two screens, the problem is how to show the user painting with live view. Because of scope difference as we mentioned in previous section, it is hard to understand the behavior of system if we only show the live view. To address this problem, we use contour line to visualize the user painting.

Our current user interface is shown in Figure 4.3. We call it a *single screen system* in this thesis. The left widow shows the live view from the ceiling mounted camera, and the right window shows the tool panel with basic painting tools including brush, squirt, clear, undo, redo and pallet. The user specifies the desired intensity on the live view by painting.

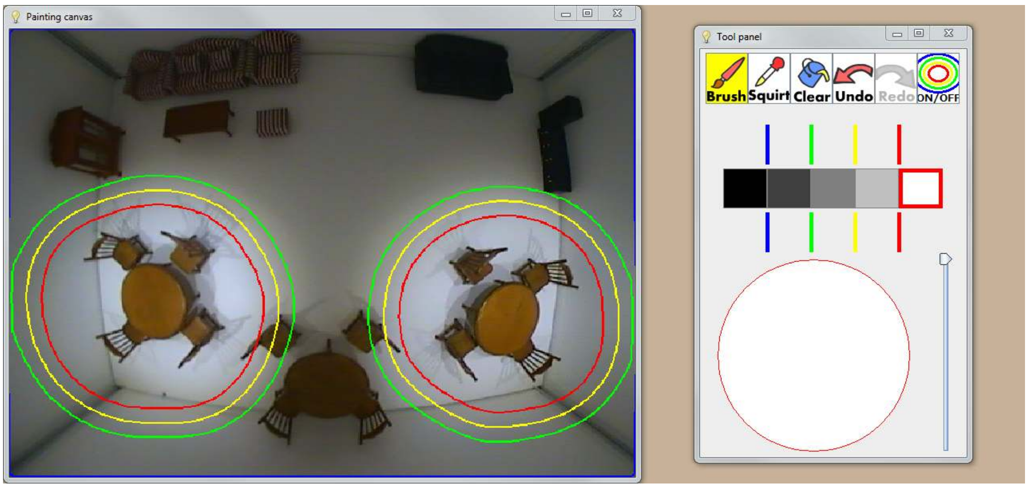


Figure 4.3: Our current user interface

In our single screen system, both live view and painting canvas are merged into one screen and the painting result is visualized as contour lines of the target illumination intensity overlaid on the live view. The relative intensity is indicated by color: a warm color indicates a brighter area, and a cold color indicates a darker area. Each line is generated in the middle of each color. To avoid bothering the user examination, we also added toggle function in the tool panel. As merging two screens into one screen, the different view problem is mitigated in our single screen system because the system shows the essence of user request as contour lines and the illumination result in live view simultaneously, making it easy for the user to understand what is happening. This merging will be also helpful using on small screen such as a smartphone; on the contrary previous approach consumes a lot of screen space.

In addition to merging two screens, we added running optimization and controlling the lights on the fly while the user is hovering the stylus tip on the screen for efficient exploration. It shows what happens if the user touches the screen with the stylus at the location before actually making a commitment (i.e., actually touching the screen). Some previous systems (e.g. [12]) continuously updated the result while the user was painting, but our system provides further assistance by showing feedback even before the user starts painting.

We changed minimum brush size from 160×160 to 120×120 to reflect informal user feedback besides. Smaller setting is also possible, but we choose this size without incurring a problem on contour visualization. On the painting area, the boundary of stroke is smoothed and the contour may not appear when the brush size is too small. There is no problem on size 120×120 as described above.

Chapter 5

Image-based Lighting Simulation

In this chapter, we give a brief of the image-based simulation method [17, 5, 2, 16]. Although this method is fully studied in theory, we also provide a practical knowledge for reproducibility with specific camera having its own constraints.

5.1 Background

We need a prediction of room lighting status during optimization process in preparatory to control lighting units at runtime. We have few options to do this: (1) sampling several spots with illuminator and interpolation, (2) estimating light effect using basis function [6], and (3) image-based simulation using captured images by camera. The important point is that our system will be used under unknown lighting units with changeable furniture layout defined by user. From this point of view, sampling approach is impractical by its hardness of automation. Estimating strategy is partly acceptable; however, it needs burdensome calibration on light source position and furniture geometry in each environment. Compared these two approaches, image-based simulation approach will be achieved easily by using the ceiling mounted camera. Because of this, we estimate the influence of each lighting parameter on the result by image-based simulation in this research. Figure 5.1 shows the schematic diagram of the simulation. We can achieve lighting simulation with multiple lights through synthesizing the individual lighting results using linearity [17].

The problem is that we cannot use pixel values from an image directly at synthesizing. Although they come from the actual radiance, the values lose linearity by several non-linear mapping through the acquisition. This is why we need to recover camera response curve for recovering radiance map from image at first. We apply the method described in [5] to recover the camera response curve. The relationship between pixel value Z and radiance E can write down as:

$$Z = f(E\Delta t) \tag{5.1}$$

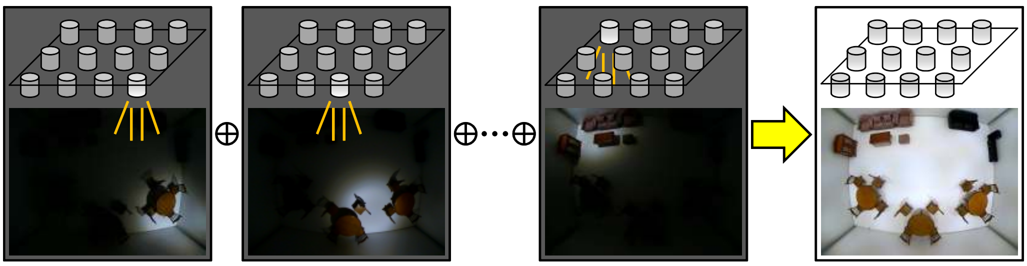


Figure 5.1: Schematic diagram of image-based lighting simulation

where f is a characteristic curve which maps the radiance to pixel value and Δt is the shutter speed at photograph taken. Equation (5.1) can be rewritten as follows using an inverse function $g = \ln f^{-1}$ for separating the right hand side:

$$g(Z) = \ln E + \ln \Delta t \tag{5.2}$$

Debevec and Malik [5] use least squares method with equation (5.2) as objective function under the same scene in different shutter speeds for recovering camera response curve g. Using this curve, we can convert the image $\mathbf{Z}$ to radiance map $\mathbf{E}$:

$$\mathbf{E} = \exp\left(g(\mathbf{Z}) - \ln \Delta t\mathbf{I}\right) \tag{5.3}$$

where $\mathbf{I}$ is unit matrix. Once we obtain radiance maps by equation (5.3), we can synthesize the effect of each light by linearity:

$$\mathbf{E}_{\text{synth}} = \sum_i \mathbf{E}_i \tag{5.4}$$

For checking the validity of simulation, we may export the virtual photograph, as follows:

$$\mathbf{Z}_{\text{synth}} = g^{-1}(\ln \mathbf{E}_{\text{synth}} + \ln \Delta t_{\text{out}}\mathbf{I}) \tag{5.5}$$

where Δt_{out} is the arbitrary shutter speed for output photograph.

5.2 Calibration Detail

Our calibration procedure is divided into three steps: Aperture control, recovering response curve g, and taking the effect of each lighting parameter.

First step is aperture control. The essential point in this step is the adjustment of exposure. Exposure is total amount of light sensitized on the image sensor. This value is determined by the aperture size (f-number) and exposure time (shutter speed). Excessive exposure setting influences the precision of simulation by acquisition problem (e.g. saturation by overexposure). Although modern digital camera has automatic exposure adjustment, it is not recommendable in the purpose of recovering the radiance [5]. Because of this, we performed the follow approach: (i) turn on a single light and set the camera at the longest shutter speed, (ii) determine the aperture size manually paying attention to overexposure and underexposure, and (iii) turn on the entire lights and find the adequate shutter speed for live view. Since recovered response curve g has inevitable error at both ends, step (ii) executed carefully.

Second step is recovering the camera response curve g using the method described in [5]. We take photographs under different shutter speeds, and pick out the part of them by means of intensity histogram. In our case, 6 photographs were used for recovering g (Figure 5.2). Figure 5.3 shows the resulting g. It is also important to check the monotonicity of obtained g.

Last step is calibration of each lighting parameter under the longest shutter speed (in our case, 1/60). In this research, we represent the configuration of i-th lighting unit as c_i. This includes the 8-bit brightness level b_i and directions x_i and y_i:

$$c_i = [b_i, x_i, y_i] \tag{5.6}$$

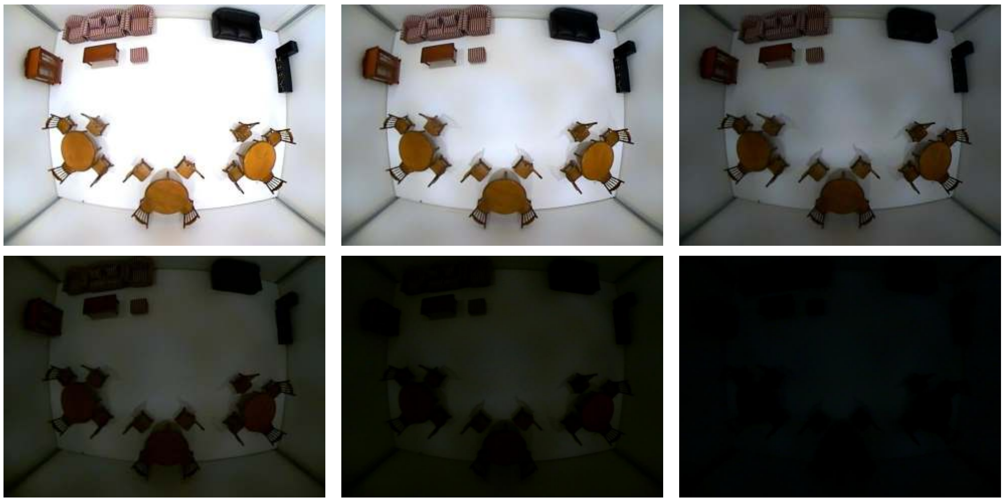


Figure 5.2: Photographs for recovering camera response curve. All images are captured under the same condition except of shutter speed (from upper left to lower right; 1/60, 1/100, 1/250, 1/500, 1/1000, 1/2000).

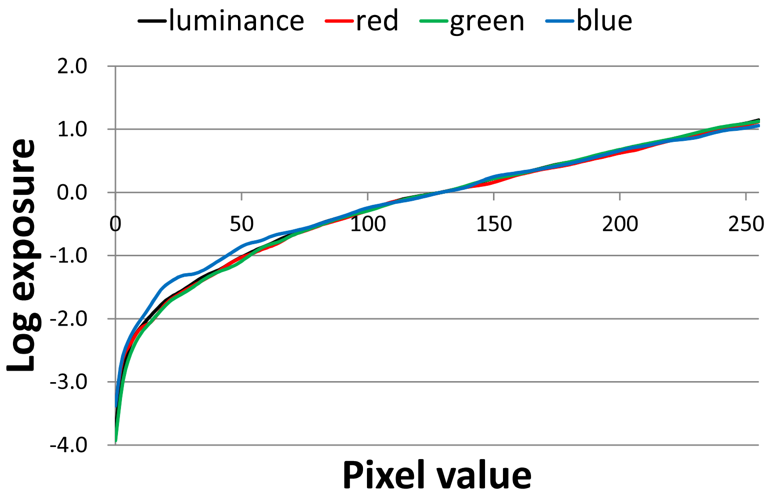


Figure 5.3: Our camera response curve

We set the resolution of the light direction to 5×5 and that of the brightness to 9, meaning that the possible selection of each light is $5 \times 5 \times 9 = 255$. We therefore need to take 2700 images in total ($12 \times 255 = 2700$) in straightforward fashion. However, this way incurs not only time consuming but also the source of error when capturing faint brightness by camera threshold. Here, we only capture the full-on status image and use the linearity of light. We actually need to capture 300 images ($12 \times 5 \times 5 \times 1 = 300$). To do this, equation (5.3) is denoted as follows:

$$\mathbf{E}_i[c_i] = \exp\left(g(\mathbf{Z}_i[255, x_i, y_i]) \cdot w(b_i) - \ln \mathit{\Delta} t\mathbf{I}\right) \tag{5.7}$$

where w is weight function calibrated by illuminometer (Figure 5.4). Since our lighting unit has a linear relationship between brightness level and radiance level, equation (5.7) is simplified as follows:

$$\mathbf{E}_i[c_i] = \exp\left(g(\mathbf{Z}_i[255, x_i, y_i]) \cdot \frac{b_i}{255} - \ln \mathit{\Delta} t\mathbf{I}\right) \tag{5.8}$$

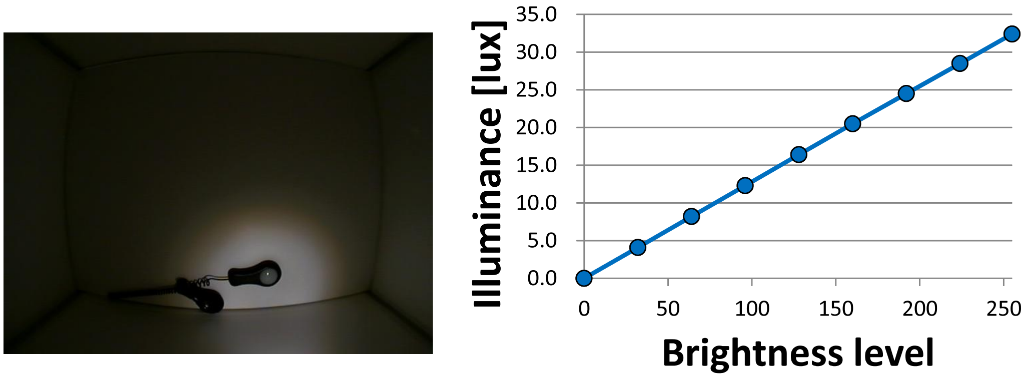


Figure 5.4: Calibration of lighting unit and its result

Once the entire images are captured, we convert them into single channel (luminance) radiance maps and downscaled for optimization (will be described in Section 6.3). In our case, it takes about 10 minutes for capturing and 5 minutes for converting.

5.3 Results

Generally, it is hard to evaluate the validity of simulation. We take on strategy that picking up several examples and comparing with the simulated result. Although we have no need to simulate color because our environment has homogeneous light sources, we tested our simulation in full-colored domain for further study.

We show five lighting examples and simulation results of them in Table 5.1. These synthesized images are generated under the same shutter speed with live view. It is sufficient to predict the lighting result, even though there is color tone mismatching and the entire scene becomes slightly darkened.

5.4 Discussion

In our simulation result, there exists much error in blue channel. It was likely to be generated by the distorted section between 0 and 50 in g curve, although it is also monotonic. In order to obtain better result, we may need more careful pixel sampling in recovering the response curve g.

Our environment is enclosed by walls, so we can only consider the effect of single light in calibration process. However, ordinary room in real world has window and room lighting is affected by external lights. If we want to cancel this effect, we can use following equation:

$$\mathbf{E}_i[c_i] = \mathbf{E}_i^{\delta}[c_i] - \mathbf{E}_i^{\delta}[0,0,0] \tag{5.9}$$

where δ is unknown external lighting effect. By this subtraction, we can obtain the pure effect of each parameter. This idea is also useful for faint illumination. If we cannot extract the correct effect of light caused by underexposure though using the longest shutter speed, we can exploit other lights for calibrating. In this case, we may capture two photographs while all the other lights are turn on with small brightness level. The equation is as follows:

$$\mathbf{E}_i[c_i] = \mathbf{E}_i[c_i = c, c_{j\neq i} = [\delta,0,0]] - \mathbf{E}_i[c_i = [0,0,0], c_{j\neq i} = [\delta,0,0]] \tag{5.10}$$

Table 5.1: Comparison of real and simulated results

	Lighting result (real)	Simulation result (synth)
Case 1		
Case 2		
Case 3		
Case 4		
Case 5		

(a) (b) (c)

Figure 5.5: The effect of compensation

For testing this idea, we captured images with 25% intensity under the same camera setting. Determining this delta value is a quite issue. If this value is too small the images will not be bright enough and if it is too big it will saturate them. We create multiple base radiance maps with different δ values and examined the accuracy of synthesized radiance maps using these radiance maps. We then selected the best value ($\delta = 12$ in our case) for our current system. Figure 5.5 shows the comparison of compensated result. The image at the center (b) is ground-truth image which is captured by camera. The leftmost image (a) is synthesized result without compensation. The rightmost image (c) is synthesizing result with compensation. Although both synthesizing results suffer color tone mismatching, compensated result is more sufficient to reproduce the real lighting result. Note that this compensation is only alternative option; taking longer shutter speed or widening aperture are more effective measures in most cases.

The validity of simulation is definitely depended upon the accuracy of the response curve g. Although we acquired moderately satisfactory result, we may need more accurate prediction in real-scale environment. We expect that this problem will be addressed by using the HDR camera, even though there are other criterions such as ease for re-organizing or installation, merit in price, etc.

Chapter 6

Optimization

In our system, one can control lighting units by painting method. We provide the algorithm detail and computation performance in this chapter. We first describe the way of generating the target radiance map and contour lines. Then, we formulate the problem as cost minimization problem and provide the principles in our optimization. We downscale the resolution and introduce GPU-based local searching algorithm for achieving the real-time performance. The followings are analysis of computation performance and influence of initial seed.

6.1 Generation of Target Radiance Map

The user specifies the target illumination via the painting interface. There are two related – pixel and radiance – domains for manipulation. At a glance, there is no difference between these domains. However, it cannot be handled the exceeded value which has under 0 or over 255 in pixel domain. Manipulation on radiance domain, by contrast, has a merit because value is not dependent on the shutter speed. Our ultimate goal is controlling the real world lighting, not the image on the screen, so we use radiance domain as our painting area.

In our painting interface, the user is not allowed to specify the absolute radiance values because the achievable maximum brightness is different for each pixel by physical limitation. Instead, the user specifies relative radiance for each pixel with respect to the value obtained when the lights are fully lit. This idea is for mitigating the "sketch rather than paint" phenomenon discovered by Kerr and Pellacini [12]. Specifically, the target radiance map is given by $\mathbf{E}_{\text{target}} = \{E^j_{\text{target}}\}$, as

$$E^j_{\text{target}} = \beta^j E^j_{\text{base}} \tag{6.1}$$

where j is the index of a pixel and $\beta^j (\geq 0)$ is the weight applied to pixel determined by user painting. In this research, we set base radiance map ($\mathbf{E}_{\text{base}}$) as the radiance map provided by a configuration in which all lights have maximum brightness and are angled downward:

$$\mathbf{E}_{\text{base}} = \sum_i \mathbf{E}_i[255, 0, 0] \tag{6.2}$$

Note that β can theoretically take a value higher than 1.0, which is possible when multiple lights are aimed at the specific area. However, we did not allow the user to paint with β higher than 1.0 in our user study because it inevitably reduces the brightness in other places, which can confuse the user. It is also difficult to capture and show the effect of large β at the single shutter speed because pixel values are almost saturated with $\beta = 1.0$ in our live view. To visualize the contour lines, we use marching square algorithm which is the simplified version of marching cubes [14] on β area.

6.2 Formulation

The system computes a configuration that minimizes the difference between the user request and the predicted illumination result. We formulate the problem as one of finding the best configuration $\mathbf{c}$ that minimizes the following objective function:

$$\mathcal{O}[\mathbf{c}] = \left\| \mathbf{E}_{\text{target}} - \sum_i \mathbf{E}_i[c_i] \right\|^2 \tag{6.3}$$

There exist tremendous methods for optimization. The important point in selecting algorithm is the understanding the characteristics of problem. We first listed the factors of our system for determining the optimization method.

- Combinatorial optimization: State of each light is defined as discrete status $\mathbf{c}$, and each light is permitted having its own status. Besides, one light does not affect other lights' status. Nevertheless, the room lighting result is determined by all of lights.
- Necessity for fast response: We suppose that our interface substitutes for switches and dimmers in real world lighting. The response time of those controllers is an instant; our interface should have fast response time at interactive rate to avoid user's incongruity.
- Ambiguity in request information: Although we contrived a novel idea on painting canvas for generating realistic target radiance map, the user painting does not correspond to lighting result in essential, as we described in Chapter 4.

For combinatorial optimization, we can use meta-heuristic searching algorithm. We chose local search by hill climbing for our optimization method. Although this is quite an ad-hoc approach, it reproduced reasonable results in our experiments. There are more advanced algorithms for finding global optimum, such as simulated annealing [13], a genetic algorithm [8], and particle swarm optimization [11], but these methods generally require enormous computation and iteration until convergence, meaning they are not practical for interactive performance. Moreover, we have no merit to find global optimum, because we cannot expect that there is perfect matching combination with user painting.

6.3 Method Detail

Computation Cost Reduction

Since our environment has homogeneous light sources, we have no need to compute three different color channels. We simply convert all color images into grayscale and also recover a response curve in these grayscale images. Each pixel of a radiance map is stored as single precision floating point value.

In the matter of resolution, full resolution (640×480) causes too much performance and memory space overhead. In addition, full resolution is not really necessary because of light diffusion. Nonetheless, too much downscaling causes a discrepancy between the simulation and real-world results. We tested few variations and selected 80×60 as our optimization resolution.

As a result, our calibration data size is reduced from $640 \times 480 \times 4 \times 3 \times 300 = 1.03\text{GB}$ to $80 \times 60 \times 4 \times 300 = 5.49\text{MB}$.

GPU-based Parallel Search

Even though we opted for a local searching approach, the computation time of evaluating neighbors is an obstacle to real-time interaction. We therefore introduced a GPU-based parallel searching technique [15]. We examine multiple candidates $\mathbf{c}'$ around the current solution $\mathbf{c}$ in each iteration. Each candidate decreases or increases one of three parameters b_i, x_i, or y_i of one of twelve lights. We examined a total of 72 candidates ($2 \times 3 \times 12 = 72$).

There are two main factors for speed improvement by GPU implementation; parallelism and memory transfer. Modern GPUs have a lot of cores to support executing numerous threads. Local search algorithm has high parallelism, so it is quite reasonable to implement on GPU. A difficulty is on memory transfer, since host (CPU) and device (GPU) has independent memory spaces. Reducing memory traffic between host and device is the key to improve the running speed.

Figure 6.1 shows the overview of our GPU-based searching method. We transfer all radiance maps data obtained in the calibration to the device (GPU) memory. We also transfer the target radiance map to the device at the beginning of the optimization. In each iteration, only the list of candidate configurations $\{\mathbf{c}'\}$ will be sent to the device, and the objective function values $\{\mathcal{O}[\mathbf{c}']\}$ of those candidates will be sent to the host (CPU) memory. There are two steps for computing $\{\mathcal{O}[\mathbf{c}']\}$. The first step is summation and subtraction, which is described as $\mathbf{E}^j_{\text{target}} - \sum_i \mathbf{E}^j_i[c'_i]$. In this step, all threads are parallelized at the pixel level. The second step is summation $\left\|\mathbf{E}^j_{\text{target}} - \sum_i \mathbf{E}^j_i[c'_i]\right\|^2$ using parallel reduction [9]. This step is parallelized at the candidate level.

6.4 Implementation and Performance Evaluation

We implemented our optimization module using JAVA™ on a desktop PC with a 2.80-GHz Core i7-930 CPU and 4.0GB RAM. We also used jCUDA library for integrating GPU implementation into the system with NVIDIA GeForce GTX 480.

In point of performance evaluation, the ambiguity in user painting keeps us from evaluating in a qualitative manner. In addition, there are many error sources except of optimization method (see Figure 6.2). We input achievable target radiance maps by synthesizing described in Chapter 5. In this way, we can compare the target with its solution and estimate the effect of downscaling, etc.

We first evaluated the performance by downscaling factor. In this evaluation, we set the darkroom status as initial seed and full-on with downwards status as target. Even though the result is trivial, we can estimate the general search time by this evaluation. The result is shown in Table 6.1. Thanks to GPU implementation, we can achieve interactive rate at all resolutions. The issue is a balance of solution quality and searching time.

Although every downscale setting achieves interactive rate, there are still uncertainties in our local searching approach. One is downscaling does or does not have a side-effect. Another issue is starting configuration in optimization. The quality of optimum is depended upon the starting point in the local search. We

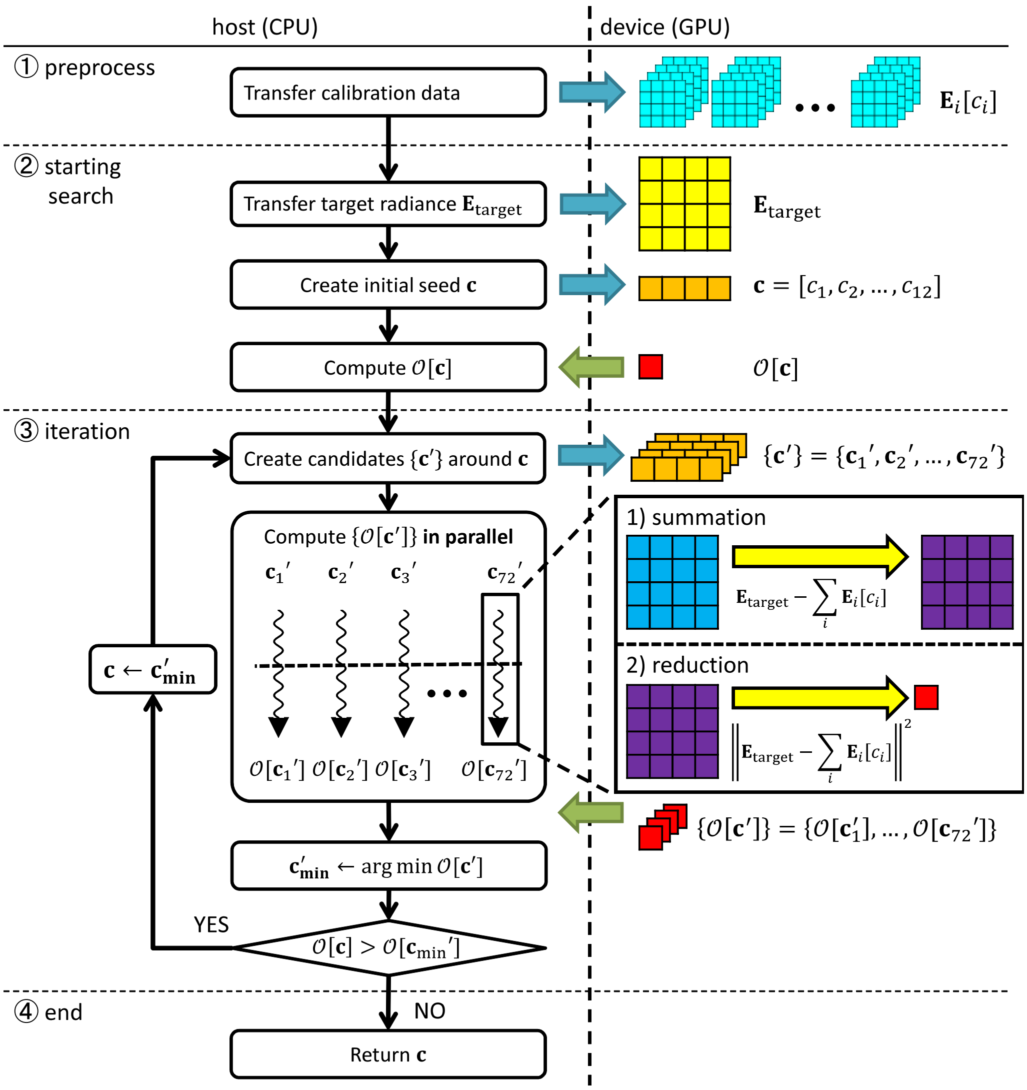


Figure 6.1: GPU-based parallel search

then tested the effect of downscaling with different starting configurations. For evaluation the quality of optimum, we defined visual error rate as the error-to-signal ratio, as follows:

$$\frac{\|\mathbf{E}_{\text{local}} - \mathbf{E}_{\text{target}}\|^2}{\|\mathbf{E}_{\text{target}}\|^2} \tag{6.4}$$

where $\mathbf{E}_{\text{local}}$ is obtained radiance map with local search. The visual error rate is 0 when the illumination result exactly matched the target and 1 when the illumination result is completely darkness.

Table 6.2 to Table 6.4 shows the result images. In many cases, downscaling scarcely affects in the result of both of configuration and illumination. Although few results had affected in configuration, the illumination result and visual error rate of them are very close.

6.5 Discussion

In this research, we only focused on hill-climbing for optimization method. There is a possibility that another type of algorithm can achieve the real-time perfor-

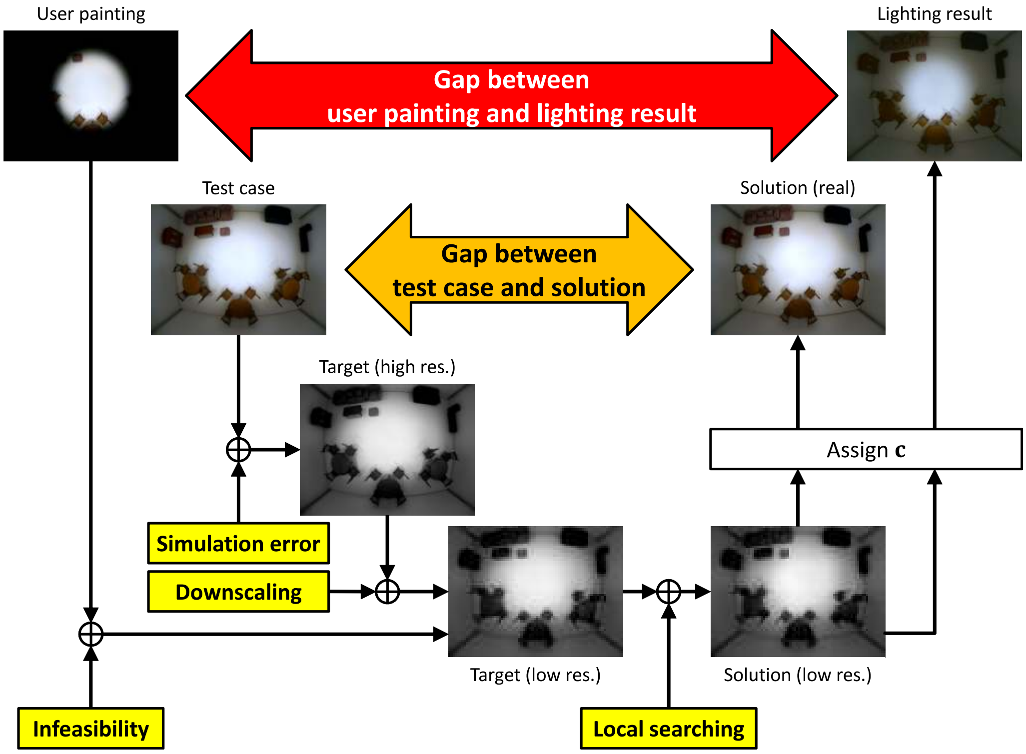


Figure 6.2: Schematic diagram of our calculation domain. Note that local searching is not a solitary error source in our system.

Table 6.1: The performance onto downscaling block

Resolution	640×480	320×240	160×120	80×60
Downscale	1×1 (none)	2×2	4×4	8×8
Image quality				
Memory	351.86MB	87.89MB	21.97MB	5.49MB
GPU time	1584 [ms]	439 [ms]	150 [ms]	63 [ms]
CPU time	179494 [ms]	45373 [ms]	11492 [ms]	2910 [ms]

Table 6.2: Results starting from darkroom (0% intensity) status

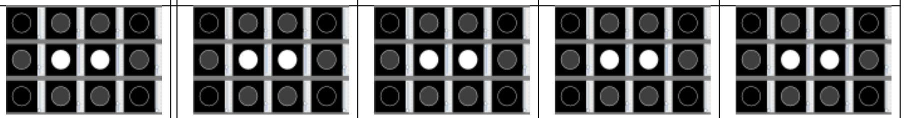
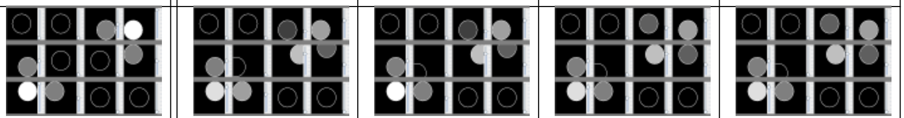
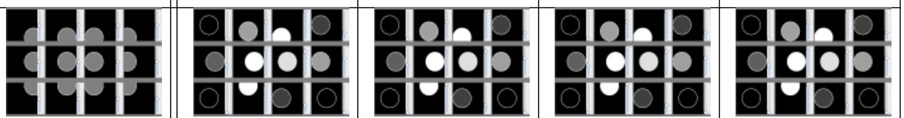
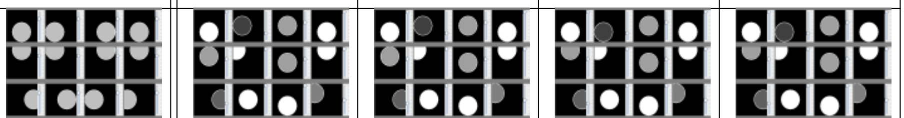
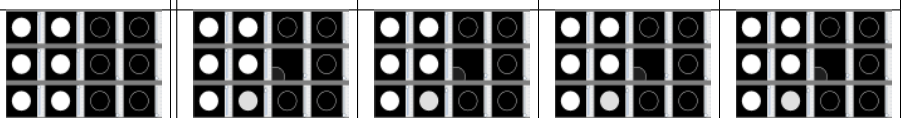
		target	640 × 480	320 × 240	160 × 120	80 × 60
Case 1	Image quality					
	Config.					
	Visual error	–	0.0000	0.0000	0.0000	0.0000
Case 2	Image quality					
	Config.					
	Visual error	–	0.0033	0.0030	0.0079	0.0079
Case 3	Image quality					
	Config.					
	Visual error	–	0.0036	0.0036	0.0036	0.0036
Case 4	Image quality					
	Config.					
	Visual error	–	0.0051	0.0051	0.0048	0.0048
Case 5	Image quality					
	Config.					
	Visual error	–	0.0007	0.0007	0.0007	0.0007

Table 6.3: Results starting from mid-on (50% intensity) status

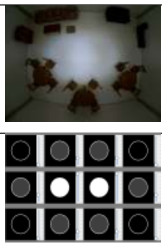
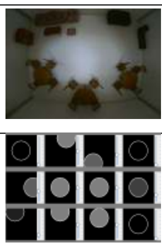
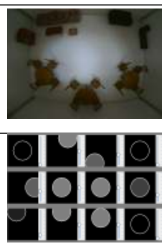
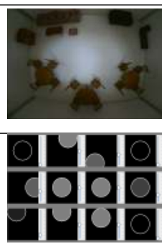
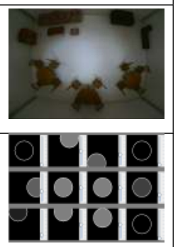
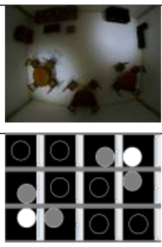
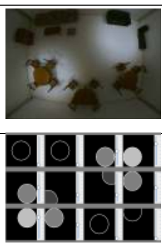
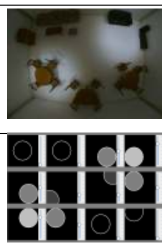
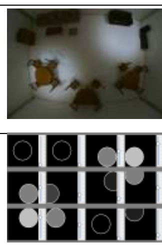
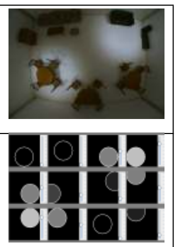
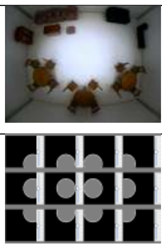
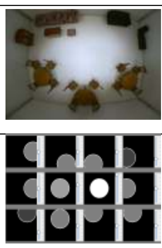
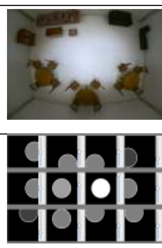
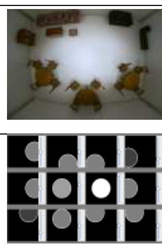
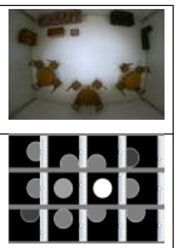
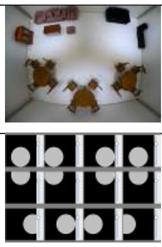
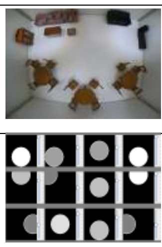
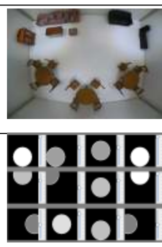
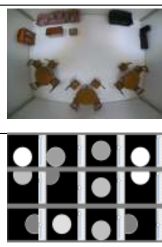
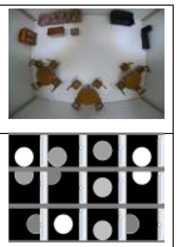
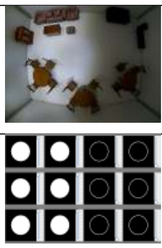
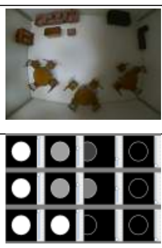
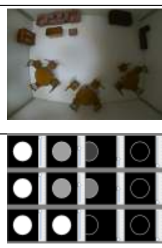
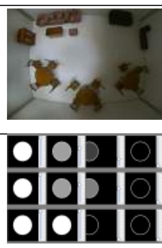
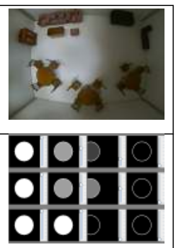
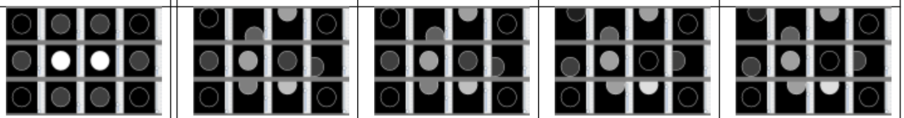
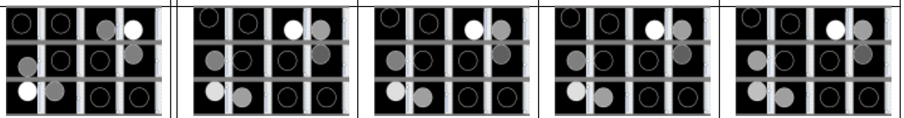
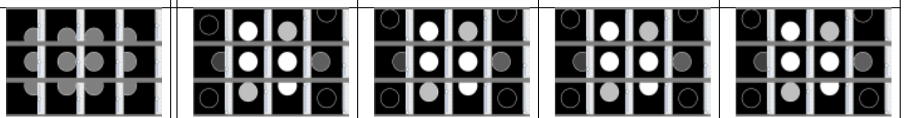
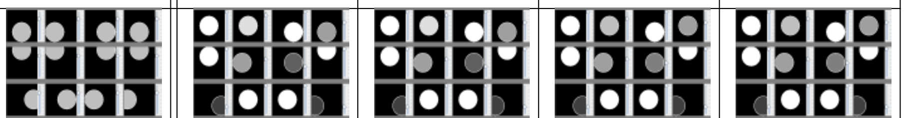
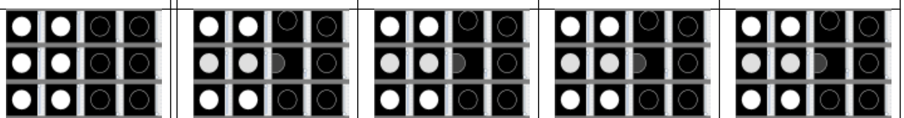
		target	640 × 480	320 × 240	160 × 120	80 × 60
Case 1	Image quality					
	Config.					
	Visual error	–	0.0034	0.0034	0.0034	0.0034
Case 2	Image quality					
	Config.					
	Visual error	–	0.0027	0.0027	0.0017	0.0017
Case 3	Image quality					
	Config.					
	Visual error	–	0.0025	0.0025	0.0025	0.0025
Case 4	Image quality					
	Config.					
	Visual error	–	0.0049	0.0049	0.0049	0.0049
Case 5	Image quality					
	Config.					
	Visual error	–	0.0019	0.0019	0.0019	0.0019

Table 6.4: Results starting from full-on (100% intensity) status

		target	640×480	320×240	160×120	80×60
Case 1	Image quality					
	Config.					
	Visual error	–	0.0044	0.0044	0.0061	0.0061
Case 2	Image quality					
	Config.					
	Visual error	–	0.0045	0.0045	0.0045	0.0048
Case 3	Image quality					
	Config.					
	Visual error	–	0.0057	0.0057	0.0057	0.0057
Case 4	Image quality					
	Config.					
	Visual error	–	0.0063	0.0063	0.0060	0.0060
Case 5	Image quality					
	Config.					
	Visual error	–	0.0010	0.0010	0.0010	0.0010

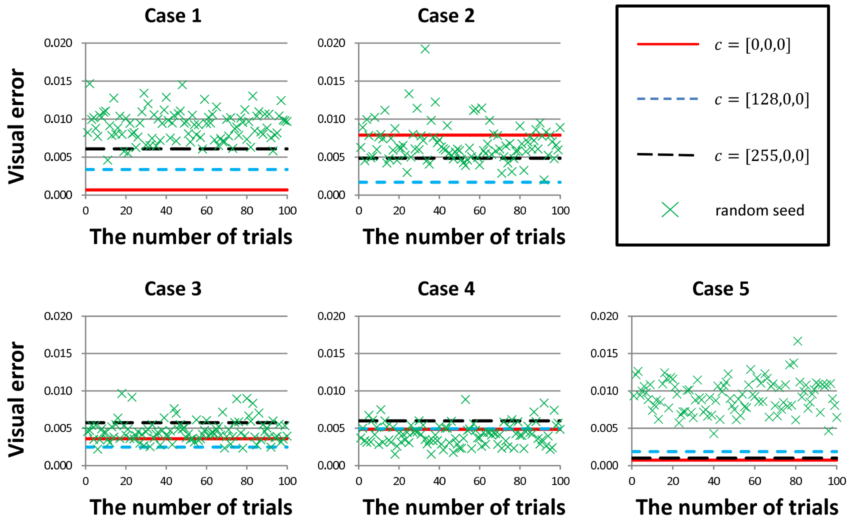


Figure 6.3: Comparison with searching from random seeds

mance, of course. The most important factor on choosing method is not a finding global optimum but plausibility with a real-time performance, in our perspective.

First of all, it is unreasonable assumption that user can paint considering all of the lighting effect such as diffusive light. Those effects are completely depending on the environment. Even if one can obtain the sense by numerous trials, one also has to do trial and error when he or she comes across another environment. Consequently, the painting by user is inevitably becomes uncertain (i.e. Figure 4.2) in painting manner. Because of this, it is no merit on finding global optimum.

We temporarily conclude that it is sufficient to local search from darkroom status on 80×60 downscaled resolution. Here we also provide other clues for justifying our local searching with a single seed. We tested local searching from random seeds for searching global optimum. Figure 6.3 shows the result. Many random seeds converge to local minimum with small error-to-signal. Except of case 2, local minimum obtained by darkroom seed have a good result in most cases.

Searching from random seeds occasionally achieve a better result than darkroom seed. We may exploit better optimization result by adopting this searching in our idle process time. However, we slightly oppose on it because he or she would have no confidence on the entire system by this behavior.

In a similar perspective, we take a skeptical view on adoption of MSHC (multi-start hill climbing) searching although we had already implemented it. We estimate that there also exist numerous local minimums with infeasible painting (i.e. Figure 4.2), but it is uncertain that a global minimum in this situation does or does not give similarity to user.

Our user interface reliefs this problem by not showing the target illumination directly. For these reasons, we tentatively concluded that it is sufficient to achieve our purpose by searching with SSHC (single-start hill climbing). We also did our user study with SSHC.

Chapter 7

User Study

We ran a user study using our miniature scale environment to compare our method with a baseline method and obtain feedback from the participants for further improvement. We implemented a conventional direct manipulation method as baseline which consists of multiple GUI widgets as a baseline method. We asked the participants to control the lights using these two methods following instructions given in natural language and then measured task completion time and user satisfaction.

7.1 Baseline Method

As a baseline method, we implemented a standard direct manipulation user interface to control the each light unit. Hardship of lighting control is classified into two types. One problem is hard to grasp correspondence between lighting parameter and controller. Another problem is hard to know the influence on the entire scene of a single lighting parameter. The latter one is a difficult problem even for professional lighting designers. We concentrated on the former one for fair comparison. In this point of view, we thought that replicating a the conventional lighting console used for professional lighting designer is not good for novice users. Most consoles are equipped with uniform dimmers, and it is likely that novice users will be confused on correspondence between the device and lighting parameter. We excluded this confusing from our comparison since it is trivial. We are rather interested in the fundamental hardship in lighting control.

A screenshot of our direct manipulation interface is shown in Figure 7.1. In order to prevent user confusing, we arranged the control widgets for single light unit in position on the screen. The left window shows the live view obtained by the ceiling mounted camera, and the right window is the light control widgets. There are 1-DOF sliders to control brightness and 2-DOF direction controllers for the twelve lights. Each slider can be adjusted to one of 9 brightness levels. Users change the direction a light is facing by dragging the corresponding circular handle in 2 dimensions. When the circle is dragged to the left, the light is pointing left in the live view. The arrangement of the controllers corresponds to the arrangement of the lights in the top down view. The user manipulates this interface on a tablet device with a stylus.

It might be possible to overlay these widgets on the live view screen that enables user to focus on lighting result. However, overlaying will also obscure the view of lighting result. In meaning of real-world reflection, our current direct interface is reasonable baseline method because conventional lighting console is separated with view screen. Even if we have no problem on concealing, indirect illumination by other lights still keeps the user from identifying of individual

Figure 7.1: Interface of direct manipulation method

lighting in manipulation. In separated screen, by contrast, user can certificate the status of each light at a glance. From these points of view, this interface is sufficient to compare with our painting interface.

7.2 Experiment Design

Participants

We recruited 10 participants aged 20 – 35 years old to participate in our study. All of them are familiar with using a computer in their daily lives. Participants were paid $21 for their time and the sessions lasted from 1 to 1.5 hours.

User Task

Each task was to specify an illumination distribution for the target scene in the miniature room. The target illumination condition was given via the natural language instructions shown in Table 7.1. All lights were initialized to 50% intensity and zero degrees in xy-tilting (looking downwards) at the beginning of each trial (Figure 7.2).

Previous study on painting interfaces for illumination control [12] showed images to the participants to specify the target illumination results. We instead gave instruction in natural language because our target application is illumination control in a living space, and so quick, informal control is more appropriate than precise control with a specific target. If we showed a target image, the participant would try to produce a faithful replication, but this is probably not a typical use case in a real living environment.

Procedure

The comparison was performed under a within-subjects condition, where each participant tested both methods. For each method, we instructed participants how to use the interface and had them practice it using two training tasks. We then asked them to complete five tasks. We used the same seven tasks in a fixed order for all participants and both methods.

We allowed a maximum of three minutes for each trial. Participants completed a trial by pushing the finish button when they were satisfied with their design. For each trial, we recorded the task completion time and asked participants to rate their satisfaction on a seven-point Likert scale with high scores positive. We also recorded the user manipulation history and screenshots for post-test observation.

Table 7.1: Task instruction

Training 1	Make the entire room brighter than the initial status. However, the sofas on the upper left side should be darker.
Training 2	Make the right side of the room darker while making the circular table on the left the brightest spot.
Trial 1	Make the left and right circular tables brighter while keeping the remaining area the same brightness.
Trial 2	Make the center and right circular tables illuminated well. The upper left area where the sofas and cupboard exist should be darker.
Trial 3	Make the entire room brighter. However, the areas around the left circular table and upper right sofa should be dark.
Trial 4	Make the left side of the room bright and the right side dark, gradually changing the brightness in the middle.
Trial 5	Make the upper right sofa bright and radially decrease the brightness around the sofa.

Figure 7.2: Initial status of each trial

After completing the five main tasks, we asked participants to answer the questionnaire shown in Table 7.2. This questionnaire also featured a seven-point Likert scale (1 = strongly disagree, 7 = strongly agree). Additionally, we asked them of usefulness of contour lines with seven-point Likert scale. We also asked when they used toggling the contour.

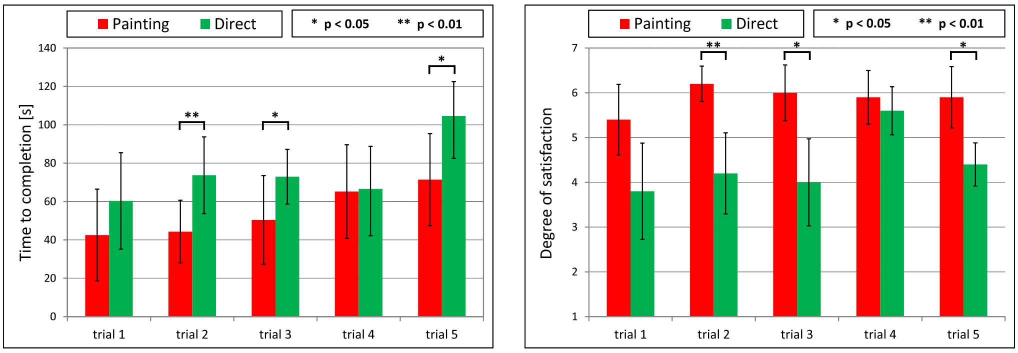
The two methods were used in a balanced order: five participants tested our proposed method first and the other five tested the direct manipulation method first. We interviewed them after all the trials were completed.

Table 7.2: Questionnaires for the conditions

Question 1	This method is a natural way to control indoor lighting.
Question 2	This method reflects my intent well.
Question 3	A lot of training is necessary to use this method.
Question 4	This method makes me tired.
Question 5	I want to control the indoor lighting by this method.

7.3 Result and Analysis

The task completion time (a) and satisfaction result by each participant (b) are shown in Figure 7.3. For a raw data, see Appendix A. The 80 percent of tasks were finished within 90 seconds. Most participants completed their tasks within the 180-second time limit, but one participant reached the limit in some tasks. We analyzed the data with Student's t-test. It showed that our method was significantly faster than the direct manipulation method in trial 2 ($t_9 = 7.17$, $p < 0.000$), 3 ($t_9 = 2.97$, $p < 0.05$) and 5 ($t_9 = 2.87$, $p < 0.05$). Trial 1 and 4 did not show significant difference ($p = 0.095$ and $p = 0.869$). We also tested the significance between the user interfaces by Student's t-test. Interestingly, the same trials (2, 3, and 5) showed significant difference in user satisfaction. Trial 2 has significant difference at 99% level (t_9=3.35, $p < 0.01$). Trial 3 ($t_9 = 2.27$, $p < 0.05$) and 5 ($t_9 = 2.87$, $p < 0.05$) has significant difference at 95% level. Trial 1 and 4 did not show significant difference ($p = 0.065$ and $p = 0.496$).



(a) (b)

Figure 7.3: (a) time to completion and (b) satisfaction result

The questionnaire results are shown in Table 7.3. Question 3 and Question 4 are negative questions. We tested the significance between the user interfaces by Wilcoxon signed-rank test. Statistically significant differences were observed in Question 1 and Question 4. The answers to Question 1 show that the participants felt our method is more natural than direct method ($p < 0.01$). The answers to Question 4 show that direct manipulation makes the participants more tired than painting ($p < 0.05$).

Participants commented that they performed the task more confidently when using our method because the target illumination distribution was visualized on the screen, thus reassuring them that they were giving correct instructions to the system. They also commented that our method is good for quickly changing the lighting condition of a wide area. One participant commented that he can focus on lighting status with our method while it was difficult with direct manipulation because it required him to see screen and the widgets alternatively. Full list of the comments are provided in Appendix B.

Participants used the contour lines visualization often and appreciated it. Contour line visualization was enabled in approximately 86 % of the total operation time in the painting method. The score of the question about the utility of contour line visualization was 6.9 on average. Most participants commented that they enabled contour lines during painting because it was hard to understand the behavior of our system without the contour lines. Many participants also commented that they disabled contour lines after painting to examine the result. Few participants, who did not disabled the contours in trial, also mentioned the

Table 7.3: Questionnaire result analysis using Wilcoxon signed rank test

Participants		Question 1		Question 2		Question 3		Question 4		Question 5	
		P	D	P	D	P	D	P	D	P	D
Paint ↓ Direct	#1	6	4	6	2	1	7	1	7	5	1
	#3	6	3	4	5	3	2	4	7	3	3
	#5	5	4	7	4	2	4	2	6	4	2
	#7	7	4	7	1	1	5	1	6	7	4
	#9	6	3	5	4	2	5	3	7	5	4
Direct ↓ Paint	#2	6	4	5	3	3	6	2	4	6	3
	#4	6	7	1	6	6	1	7	5	2	7
	#6	6	4	5	3	2	5	2	4	4	3
	#8	6	3	7	3	2	6	2	7	5	2
	#10	7	4	7	2	1	1	1	7	7	2
W		52		35		26		51		28	
N_r		10		10		9		10		9	
Z-score		2.625		1.758		-		2.574		1.629	
significance		$p < 0.01$						$p < 0.05$			

necessity of the ability to disable the lines.

Nine participants answered in the post-test interview that they would use the painting method in the future for this task rather than the direct manipulation method. Their main reason was that the painting method required less control, which made it possible to complete a task faster without much fatigue compared to the direct manipulation method. In contrast, one participant preferred the direct manipulation method because it allowed them more precise control.

7.4 Discussion

Differences among Task Performing

It is interesting to see that some tasks showed statistically significant differences while others did not. Here we examine the details of the user behavior observed in the study to identify the strength and weakness of the two methods.

First, let us examine the difference between the trial 4 and 5. They were similar in that both requested gradual brightness changes but the results were quite different. The trial 5 showed significant difference but the trial 4 did not. In our observation, we found that the most participants did not change the light directions in trial 4 in the direct manipulation method. This was because the spatial arrangement of lights was aligned well with the target illumination distribution. The user simply made the lights on the left bright and the lights on the right dark without changing their directions. However, the target illumination in the task 5 was radial, which was quite different from the layout of the lights. The users had to change the direction of the lights to obtain the radial illumination distribution, which was difficult and time consuming.

Second, let us examine why the painting method showed better results in the trial 2 and 3. These tasks involved both brightening and darkening operations. One can achieve brightening and darkening in a similar way in the painting method. The user can simply paint the scene with bright brush or dark brush. In contrast, brightening and darkening required quite different operation in the direct manipulation method. Brightening an area was relatively easy. The user can simply increase the brightness level of light and direct the light to the target area. However, in order to darken an area, the user had to carefully change the orientation of the nearby lights so that they looked away from the target area. Some participants actually pointed a light with brightness level 0 to the area to be darkened. This was a completely wasteful operation, which represents the difficulty experienced by the users in the direct manipulation method.

Last, let us examine what had happened in the trial 1. We observed two types of behaviors in this task. Most participants changed the illumination and completed their task quickly, while two participants repeatedly changed the configurations very carefully spending a very long time. It seemed that these few participants took the word “keeping” very seriously and tried to achieve it very precisely. It seems that the large difference caused by these two behaviors masked the difference between the methods difficult. This suggests that we have to be more careful in designing natural language instruction to obtain a consistent result.

Misconception on Darkening Task

In our experiment, there are three persons who manipulate the off-status light with direct interface. Two of them manipulate it one or two times and spend relatively small amount of time, while one manipulates frequently and spend long time comparing these two participants. In case of two participants, we are unsure that it occurred from intention or just mistake. By contrast, in case of one particular participant, we can slightly draw a conclusion that it was intended manipulation. This participant tended to aim the off-status light to area where can be darkened in trial. This participant experienced painting interface before direct interface, so it is likely that he got a misconception on darkening operation in lighting when he used painting interface. However, our experimental population is small, so we need further study to conclude whether it was related to the trial order or not.

Limitation of Our Evaluation Method

Since our tool is not designed for precisely control the light setting, we show the instruction as natural language. It is closer to design process than the experiment in [12]. As a consequence, each participant set his own goal and modified lighting scene. For this reason, we had no chance to analyze user lighting result or workflow in quantitative manner such as L2-error. We may use the result of each trial as an ideal solution, but we conclude that it is deliberate interpretation so we did not such an evaluation.

Most participants have no problem on lighting control, but one particular participant who reached time limitation in trial had a difficulty in lighting control with both of interfaces. However, it is a quite questionable that the problem exists in the interface or lighting itself. In post-test interview, he commented that the entire lighting scene would not be able to control in contrary to his expectation. We guess that he imagined impossible lighting result and roamed to achieve it. To avoid this meaningless roaming, we may show the goal image. However, if we use image directly, we will repeat the same mistake in [12]. Currently, we conclude that it is a dilemma in evaluation of the system for creative process in graphical design, not only for a lighting design.

Chapter 8

Limitations and Future Work

In this chapter, we describe the limitation of our approach. We also describe the future work of our painting system.

8.1 Limitations

Our current system uses a fixed top-down view from the ceiling for calibration. If we change the camera position or furniture arrangement, we have to calibrate the entire system again. One solution is to use multiple cameras, but then co-ordination among these cameras becomes an issue. Our current system assumes that it controls all light sources itself, but real-world environments are affected by external light sources, most notably sunlight. In the source of light, we only concerns with lights from ceiling. We tested our method using only a miniature prototype, and it remains our future work to perform tests with a real-sized room.

8.2 Future Work

Manipulation Method

Our current system has fixed number of brightness level in brush and weight parameter in each brightness level. Although we achieved effective lighting result exploration by painting, there is room for improvement. It is not a good idea to increase the number of brightness level. It will be needed more contour lines on the canvas then them obscure the view of lighting result. Rather, we think that more effective exploration can be achieved by flexible coupling level and weight . Figure 8.1 shows our idea on improved interface. The value in the slider under color level is meaning weight in our algorithm. Once the value is changed, the system launch optimizer again and assign the parameter to lighting array. In this way, one can change entire illumination quantity without changing the painting.

In our current implementation, contour lines are automatically generated by painting result. We consider that another interesting research direction is to manipulate contour lines directly. It must provide a slightly different user experience. We expect that it may be useful on modifying task, instead of initial rough sketch. In advanced implement this, we consider that it is important to think several scenarios in this contour manipulation. These scenarios lead the appropriate data structure for implementation.

Color Treatment

Since our environment has homogeneous light sources, we actually need not each color values but luminance in our optimization. However, we expect that our

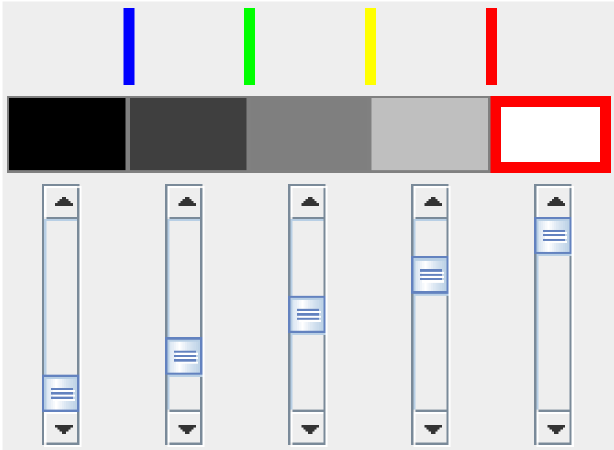


Figure 8.1: Flexible coupling of brightness level and weight value

system is also applicable in colorized lighting system. There are two problems remaining on software part to make it possible.

One is the problem on precision of the camera response curve. As we mentioned in Section 5.3, there are some color tone mismatching in our simulation. This problem is dependent upon the precision of camera response curve, as [5] already noted. [5] also suggested mending this problem by using luminaire of known color. However, we need more empirical study on recovering response curve unless we cannot obtain such a value.

Another is the problem on the way of painting. In our current implementation, we use different colors in contour lines to show relative brightness level. However, if we use colorized lighting units in our system, there is also issue of confusing on the meaning of color. How can we address this problem? One idea is using the line thickness as an indicator: a thick line indicates a brighter area, and a thin line indicates a darker area. However, we have some doubts about the clarity of this visualization technique. We may need further invention on the visualization or interaction technique for colorized system.

Miscellaneous

We restricted the user painting weight (β) from zero to one. Simple extension the range of weight makes inevitable reducing lights in some other area, so it cannot be answer until we cannot solve this inconsistency. We consider that another manipulation manner is needed for handling this in our system.

If aiming multiple light to specific area, which is derived by $\beta > 1.0$, makes overexposure pixel values in the live view image. In this case, we cannot acquire the detail of lighting result. Generally we can take shorter shutter speed for overexposure, darkening the entire image by changing shutter speed makes confusion on brightness perception. In order to mitigate the problem, we may use the HDR tone mapping approach for image consistency. However, it will be bother in real-time approach because we need time to capture several images for HDR synthesizing.

Showing Goal Images in the User Study

In our experiment, natural language instruction may make user roam to achieve the infeasible lighting. Showing the goal image will be mitigated this problem, but it also makes the user replicate the result precisely.

One way for evading this dilemma is using the goal image as a supplementary means only. The key idea is showing the goal image at the beginning of trial with natural language instruction in very short time. After that, the only instruction will be shown. This evaluation plan makes it possible to calculate quantitative image error. The verification of this approach, we need to test whether it is valid or not.

Chapter 9

Conclusions

We would like to conclude this thesis. First, we summarize our approach and results of this research. Next, we provide the design implications we are discovered in this research.

9.1 Summary

In this research, we presented a painting interface for controlling room illumination with an array of computer-controlled robotic lights. We designed this interface for reducing the complexity of user operation in lighting design by simple painting manner. Our system is unique compared with conventional painting interfaces for lighting control in that users paint relative radiance intensities and the result of painting is shown as contour lines overlaid on top of the input camera image. We built a physical miniature prototype system and achieved a real-time performance with a GPU implementation of the optimization process. We ran a user study comparing the proposed painting interface with a conventional direct manipulation interface, in which users controlled the lights following instructions given in natural language. The results showed that the proposed painting interface was faster and more preferable than direct manipulation in most cases. The results also showed that the contour line visualization was useful.

9.2 Design Implications

In this research, we focused on a lighting design system by painting interface. However, we believe that the knowledge obtained in this study has a generality for designing the user interface in many cases. Here we briefly summarize them.

Classification of Design Stage

Work by human being has high dimensionality, so it is hard to regulate the definition of term ”design” in a specific way. We think that design has, at least, two different stages. One is a rough sketch stage, and another is detail improving stage.

We here classify the two stages, but these two stages are not totally different. They also have a common point; the goal is not visible at first, but revealed gradually through the process. Compromise between ideal status with constraints will appear in anywhere on these processes. Because of this, we argue on the conclusion by [12]; the experiments in those researches just evaluate reproducing ability, not design ability.

The difference of two stages depends on the workflow. In rough sketch stage, the workflow starts from the abstract goal and has much diversity in materializing.

As a result, prototyping with several different prototypes and choosing the better one is common in this stage. Details are less important in this stage. Therefore, the speed and variety in prototyping will be important factor. It will be clear that our experimental result or comments of open-ended trials in [12].

In detail improving stage; by contrast, all workflows start from a previous prototype and improve in part of it with a specific way. As a result, the characteristic of workflow is based on one-by-one manipulation. Details become more important in this stage, so high fidelity and resolution in user manipulation will be also emphasized. Although we argue the experiment methodology and conclusion of [12], there was some truth in that. We cannot achieve high fidelity in control with optimization approach only. In this point of view, it may be needed direct manipulation for the artist-quality.

However, the most essential point in the user interface design is necessity of circumstantiation. In our experiment, most participants preferred painting interface than direct interface because they had no necessity on quality of lighting result except of illumination distribution. Namely, our painting interface will be sufficient to respond the lighting task for novice persons. We believe that it is also useful to professional users in their rough stages. In that case, we may need the merged interface as shown in Figure 9.1.

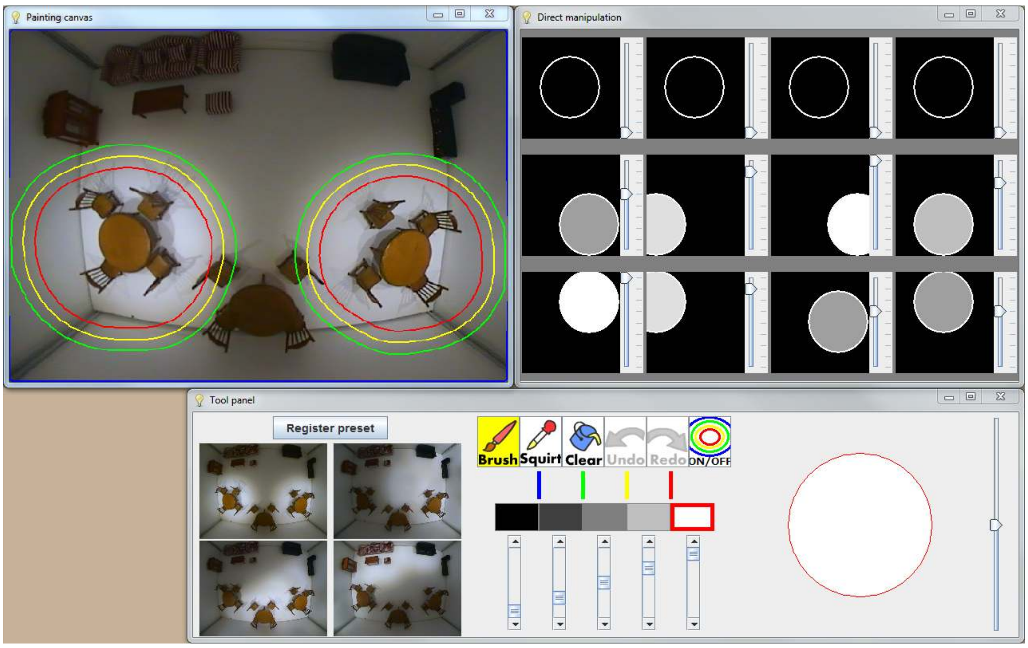


Figure 9.1: Merged interface for lighting with two different types of interfaces

Our classification in this section is in a hypothesis stage, but we believe that further studies and discussions on it are important in interface for design in the future.

Optimization and Interaction

Recently, there are many researches which use optimization to achieve complex feature improvement. In this trend, we would like to make a reasonable-but-forgettable remark: optimization is one of the means. Designing is fundamentally a creative process done by human. Input by human has also errors in physical or logical aspects, so finding global optimum becomes a wasteful method in this circumstance. In other words, the optimization is inevitable but supplementary things in a design system.

The way of interaction becomes more important factor, in this principle. Ordinary, discussion on the quantity of the information in the system tended to extend newer one. We would like to indicate that abbreviation of the unnecessary information in the system is important because excessive information leads to confusion of users. For instance in our research, the painting canvas feedback is not necessary in the two screen system. Our painting range is restricted with maximum lighting constraint but every painting is not achievable, so visualization of both of live view and painting canvas is not meaningful.

References

[1] T. Amano and H. Kato. Appearance control using projection with model predictive control. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2832 –2835, aug. 2010.

[2] F. Anrys and P. Dutré. Image-based lighting design, 2004.

[3] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 145–156, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[4] Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. A lighting reproduction approach to live-action compositing. *ACM Trans. Graph.*, 21(3):547–556, July 2002.

[5] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[6] Yoshinori Dobashi, Kazufumi Kaneda, Hideki Nakatani, Hideo Yamashita, and Tomoyuki Nishita. A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum*, 14(3):229–240, 1995.

[7] Abhijeet Ghosh, Matthew Trentacoste, Helge Seetzen, and Wolfgang Heidrich. Real illumination from virtual environments. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[8] David E. Goldberg and John H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3:95–99, 1988. 10.1023/A:1022602019183.

[9] Mark Harris. Optimizing parallel reduction in cuda.

[10] Thomas Jung, Mark D. Gross, and Ellen Yi luen Do. Light pen -– sketching light in 3d. In *In Proc. of CAAD Futures*, pages 327–338, 2003.

[11] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948 vol.4, nov/dec 1995.

[12] William B. Kerr and Fabio Pellacini. Toward evaluating lighting design interface paradigms for novice users. *ACM Trans. Graph.*, 28(3):26:1–26:9, July 2009.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[14] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987.

[15] Thé Van Luong, Nouredine Melab, and El-Ghazali Talbi. Parallel local search on gpu. Research Report RR-6915, INRIA, 2009. www.lifl.fr/luong.

[16] A. Mohan, R. Bailey, J. Waite, J. Tumblin, C. Grimm, and B. Bodenheimer. Tabletop computed lighting for practical digital photography. *Visualization and Computer Graphics, IEEE Transactions on*, 13(4):652 –662, july-aug. 2007.

[17] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey. Efficient re-rendering of naturally illuminated environments. In *IN FIFTH EUROGRAPHICS WORKSHOP ON RENDERING*, pages 359–373, 1994.

[18] Donald Norman. *Design of Everyday Things*. Doubleday Business, reissue edition, 2 1990.

[19] Makoto Okabe, Yasuyuki Matsushita, Li Shen, and Takeo Igarashi. Illumination brush: Interactive design of all-frequency lighting. In *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pages 171 –180, 29 2007-nov. 2 2007.

[20] Heemin Park, Jeff Burke, and Mani B. Srivastava. Design and implementation of a wireless sensor network for intelligent light control. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 370–379, New York, NY, USA, 2007. ACM.

[21] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[22] Fabio Pellacini. envylight: an interface for editing natural illumination. *ACM Trans. Graph.*, 29:34:1–34:8, July 2010.

[23] Fabio Pellacini, Frank Battaglia, R. Keith Morley, and Adam Finkelstein. Lighting with paint. *ACM Trans. Graph.*, 26(2), June 2007.

[24] Fabio Pellacini, Parag Tole, and Donald P. Greenberg. A user interface for interactive cinematic shadow design. *ACM Trans. Graph.*, 21(3):563–566, July 2002.

[25] Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo, and Donald Greenberg. Painting with light. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 143–146, New York, NY, USA, 1993. ACM.

[26] Ram Shacked and Dani Lischinski. Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum*, 20(3):215–227, 2001.

[27] Yu Sheng, T.C. Yapo, C. Young, and B. Cutler. A spatially augmented reality sketching interface for architectural daylighting design. *Visualization and Computer Graphics, IEEE Transactions on*, 17(1):38 –50, jan. 2011.

[28] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett, Jr., and H. Scott Matthews. Intelligent light control using sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 218–229, New York, NY, USA, 2005. ACM.

[29] Yao-Jung Wen and A.M. Agogino. Wireless networked lighting systems for optimizing energy savings and user satisfaction. In *Wireless Hive Networks Conference, 2008. WHNC 2008. IEEE*, pages 1 –7, aug. 2008.

[30] Tim Weyrich, Jason Lawrence, Hendrik Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 80:1–80:119, New York, NY, USA, 2008. ACM.

Appendix A

Raw Data of Each Trial

Table A.1: Time to completion [s]

Participants		Trial 1		Trial 2		Trial 3		Trial 4		Trial 5	
		P	D	P	D	P	D	P	D	P	D
Paint ↓ Direct	#1	13	30	26	46	28	42	33	35	26	74
	#3	176	175	135	180	180	143	175	180	180	180
	#5	35	34	57	71	47	63	129	80	109	114
	#7	43	56	41	69	31	80	67	64	54	96
	#9	21	33	37	61	32	71	44	51	39	120
Direct ↓ Paint	#2	26	126	37	71	46	76	31	59	48	132
	#4	35	38	28	48	32	67	61	28	112	87
	#6	22	19	32	70	38	60	52	74	42	104
	#8	35	58	31	84	45	85	34	70	63	92
	#10	19	34	26	37	25	42	26	25	41	47

Table A.2: Satisfaction score of each task

Participants		Trial 1		Trial 2		Trial 3		Trial 4		Trial 5	
		P	D	P	D	P	D	P	D	P	D
Paint ↓ Direct	#1	7	1	6	2	6	1	7	5	7	4
	#3	5	2	7	5	4	7	5	7	4	4
	#5	6	7	7	5	6	5	5	4	6	5
	#7	7	7	7	6	7	5	5	6	7	5
	#9	5	5	5	5	4	6	5	6	6	4
Direct ↓ Paint	#2	5	4	6	5	7	4	7	6	6	5
	#4	2	3	6	5	7	2	7	5	3	5
	#6	4	2	5	3	5	3	4	4	6	2
	#8	6	5	6	6	7	5	7	6	7	5
	#10	7	2	7	1	7	2	7	7	7	5

Table A.3: Trial 1

		Result by painting interface	Result by direct interface
painting → direct	participant #01	satisfaction: 7 	satisfaction: 1 
	participant #03	satisfaction: 5 	satisfaction: 2 
	participant #05	satisfaction: 6 	satisfaction: 7 
	participant #07	satisfaction: 7 	satisfaction: 7 
	participant #09	satisfaction: 5 	satisfaction: 5 

direct → painting	participant #02	satisfaction: 5 	satisfaction: 4 
	participant #04	satisfaction: 2 	satisfaction: 3 
	participant #06	satisfaction: 4 	satisfaction: 2 
	participant #08	satisfaction: 6 	satisfaction: 5 
	participant #10	satisfaction: 7 	satisfaction: 2 

Table A.4: Trial 2

		Result by painting interface	Result by direct interface
painting → direct	participant #01	satisfaction: 6 	satisfaction: 2 
	participant #03	satisfaction: 7 	satisfaction: 5 
	participant #05	satisfaction: 7 	satisfaction: 5 
	participant #07	satisfaction: 7 	satisfaction: 6 
	participant #09	satisfaction: 5 	satisfaction: 5 

direct → painting	participant #02	satisfaction: 6 	satisfaction: 5 
	participant #04	satisfaction: 6 	satisfaction: 5 
	participant #06	satisfaction: 5 	satisfaction: 2 
	participant #08	satisfaction: 6 	satisfaction: 6 
	participant #10	satisfaction: 7 	satisfaction: 1 

Table A.5: Trial 3

		Result by painting interface	Result by direct interface
painting → direct	participant #01	satisfaction: 6 	satisfaction: 1 
	participant #03	satisfaction: 4 	satisfaction: 7 
	participant #05	satisfaction: 6 	satisfaction: 5 
	participant #07	satisfaction: 7 	satisfaction: 5 
	participant #09	satisfaction: 4 	satisfaction: 6 

direct $\rightarrow$ painting	participant #02	satisfaction: 7 	satisfaction: 4 
	participant #04	satisfaction: 7 	satisfaction: 2 
	participant #06	satisfaction: 5 	satisfaction: 3 
	participant #08	satisfaction: 7 	satisfaction: 5 
	participant #10	satisfaction: 7 	satisfaction: 2 

Table A.6: Trial 4

		Result by painting interface	Result by direct interface
painting → direct	participant #01	satisfaction: 7 	satisfaction: 5 
	participant #03	satisfaction: 5 	satisfaction: 7 
	participant #05	satisfaction: 5 	satisfaction: 4 
	participant #07	satisfaction: 5 	satisfaction: 6 
	participant #09	satisfaction: 5 	satisfaction: 6 

direct → painting	participant #02	satisfaction: 7 	satisfaction: 6 
	participant #04	satisfaction: 7 	satisfaction: 5 
	participant #06	satisfaction: 4 	satisfaction: 4 
	participant #08	satisfaction: 7 	satisfaction: 6 
	participant #10	satisfaction: 7 	satisfaction: 7 

Table A.7: Trial 5

		Result by painting interface	Result by direct interface
painting → direct	participant #01	satisfaction: 7 	satisfaction: 4 
	participant #03	satisfaction: 4 	satisfaction: 4 
	participant #05	satisfaction: 6 	satisfaction: 5 
	participant #07	satisfaction: 7 	satisfaction: 5 
	participant #09	satisfaction: 6 	satisfaction: 4 

direct → painting	participant #02	satisfaction: 6 	satisfaction: 5 
	participant #04	satisfaction: 3 	satisfaction: 5 
	participant #06	satisfaction: 6 	satisfaction: 2 
	participant #08	satisfaction: 7 	satisfaction: 5 
	participant #10	satisfaction: 7 	satisfaction: 5 

Appendix B

Comments from Participants

	Painting interface	Direct interface
Pros	• It is easy to use, so I can achieve fast manipulation and did not become tired. (#2, #3, #4, #7, #10) • It is suitable to control the entire scene lighting. (#2, #4, #5, #6, #8) • I did not have to concern each lighting units by this interface. (#1, #2, #6) • It is easy to understand the feedback. (#3, #6) • Is is independant interface upon the environment. (#1, #2) • It is easy to generate the illuminated gradation which is not associated from light array, so it is easy to try several lighting patterns quickly. (#2,#9)	• I can achieve the detailed control by this interface. (#2, #3, #4, #6, #8, #9) • I feel the sense of control using this interface. (#4, #7, #9) • The lighting units are arranged in grid, so it is easy to make gradation according to this grid. (#5, #10) • I become happy if I achieve the lighting result as I expected. (#7)

Cons	• It is unsuitable to achieve detailed control. (#2, #4, #7, #8, #9) • I cannot imagine how to manipulate the canvas if I obtain the unexpected result. (#3, #6, #7)	• I need some time and experience for trial and error. (#1, #3, #6, #9, #10) • It is cumbersome to control all of lighting parameters, so I feel a fatigue. (#1, #3, #5, #6, #10) • I cannot imagine how to modify the lighting parameter. (#3, #4, #7) • It is hard to recognize the effect of single light illuminance. (#5, #10) • It is hard to achieve the rough control. (#8) • It is hard to reset the lighting status. (#3)