Lighting Estimation from a Single Image Containing Multiple Planes

Pin-Cheng Kuo
National Tsing Hua University
Hsinchu, Taiwan
vul3kuo@hotmail.com

Hsin-Yuan Huang
National Tsing Hua University
Hsinchu, Taiwan
misharkiller@gmail.com

Shang-Hong Lai
National Tsing Hua University
Hsinchu, Taiwan
lai@cs.nthu.edu.tw

ABSTRACT

In this paper, we present a novel lighting estimation algorithm for the scene containing two or more planes. This paper focuses on near point light source estimation. We first detect planar markers to estimate the poses of the 3D planes in the scene. Then we estimate the shading image from the captured image. A near point light source lighting model is used to define an objective function for light source estimation in this paper. The output of the proposed method is the lighting parameters estimated from minimizing the objective function. In the experiments, we test the proposed algorithm on synthetic data and real dataset. Our experimental results show the proposed algorithm outperforms the state-of-the-art lighting estimation method. Moreover, we develop an augmented reality system that includes lighting estimation by using the proposed algorithm.

CCS Concepts

• **Computing Methodologies** → Mixed /augmented reality;

Keywords

Augmented reality, lighting estimation, light source estimation.

1. INTRODUCTION

Recent years have witnessed a rise in the demand of mobile and wearable devices (e.g. smart phone, tablet). Augmented Reality (AR) has been widely used in conjunction with a mobile device in recent years. More and more AR applications have been developed to improve our living experiences from many perspectives.

In these AR applications, it is obvious that delivering a visually coherent rendering plays an important role in the field of Augmented Reality. Thus, accurate estimation of the illumination condition of the scene from the acquired images is very crucial to the rendering of 3D virtual models for display with the real image. However, relatively little work has been done for the problem of estimating near light source. Previous work often focused on shape and reflectance estimation (e.g. Oxholm and Nishino [1], Johnson and Adelson [2]). These works estimate the object shape and reflectance by natural illumination information. Lalonde and Matthews [14] is an example of lighting estimation from outdoor images. They aim at outdoor lighting and propose a low-dimensional parametric model that captures outdoor illumination conditions.

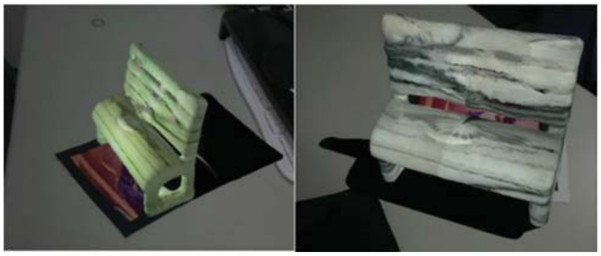

Figure 1. The augmented reality system with a plausible illumination using our proposed lighting estimation algorithm.

Nevertheless, there was very few previous works focused on lighting estimation for indoor and near light source. Since the popularization of Augmented Reality, it is important to estimate the near light source and render images for 3D virtual objects with plausible illumination. In this paper, we propose a novel lighting estimation algorithm which captures near light source information. In the experiment, we fuse it with AR system and show the estimation result by rendering virtual objects with the estimated illumination condition.

In the field of AR, two major problems must be addressed: 3D pose estimation and plausible illumination rendering. The main focus in this work is the lighting estimation of a near light source. For AR, it often requires estimating the illumination conditions (e.g. light radiations, positions) for delivering visually coherent rendering. Knowledge of the real world lighting plays a significant role in the second problem: plausible illumination through lighting and shading in AR. In this paper, we focus on estimating the near point light source at indoor scene. Figure 1 depicts an example of the plausible illumination using the lighting parameter estimated by the proposed lighting estimation algorithm.

The flow of the proposed Augmented Reality system is shown in Figure 2. Our AR system can be separated into two processes: lighting estimation and augmented reality. After capturing an input image from a camera, in the lighting estimation process, we execute the gradient filtering to recover the shading image of the environment. Next, we use the shading image as the input to estimate the initial guess of lighting parameters. Then, these parameters are optimized by solving a non-linear least squares minimization problem. For AR, it first searches the markers from the input, and uses these markers to estimate the 3D pose of the marker relative to the camera. Then we project the 3D model from

MMSys'16, May 10 - 13, 2016, Klagenfurt, Austria.

ACM 978-1-4503-4297-1/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2910017.2910613

the 3D world coordinate to 2D image coordinate. At same time, identifying markers help us extend more applications in AR. Finally, we fuse the results of two processes, and output the frame which includes virtual contents with plausible illumination rendering by the estimated lighting parameters.

The following is the main contribution of this paper. First, we propose an image-based approach that faithfully estimates the illumination condition of near point light source for indoor scene. Second, we generalize the original lighting estimation algorithm for a 3D plane to 3D scenes containing two or more planes. Finally, we present an Augmented Reality system which renders the virtual objects with plausible illumination after estimating the illumination conditions from real world. The rest of this paper is organized as follows. In section 2, some related works on lighting estimation are reviewed. In section 3, we present the proposed algorithm based on a shading model that closely approximates the illumination of environment, which is then used in the optimization approach. Section 4 shows some experimental results and provides quantitative evaluation to demonstrate the performance of the proposed method. In the end, some conclusions are drawn in section 5

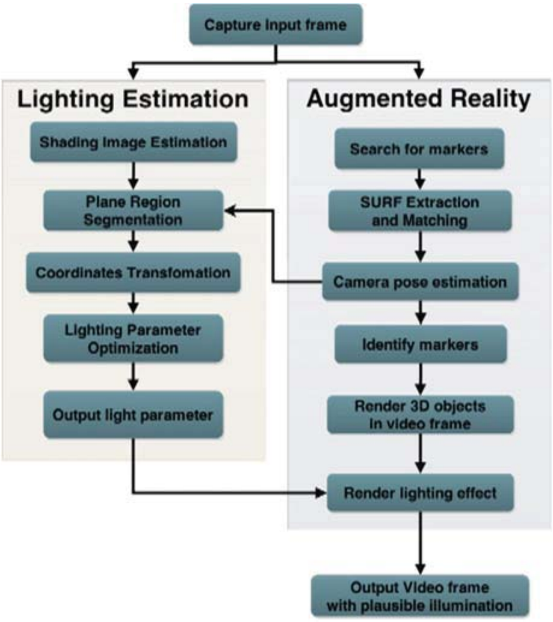
Capture input frame
Lighting Estimation
Augmented Reality
Shading Image Estimation
Plane Region Segmentation
Coordinates Transfomation
Lighting Parameter Optimization
Output light parameter
Search for markers
SURF Extraction and Matching
Camera pose estimation
Identify markers
Render 3D objects in video frame
Render lighting effect
Output Video frame with plausible illumination

Figure 2. The system flow chart of the proposed AR system.

2. RELATED WORKS

In the Augmented Reality system, visually coherent rendering plays a crucial role to render impressive results. One challenge in achieving this is to obtain correct lighting condition. In the following, there are five primary research directions related to lighting estimation problem.

2.1 Lighting Estimation from Light Probes

Typical lighting estimation methods often used a light probe to obverse the illumination of the environment, since spherical probes have a property of simplifying the estimation problem. Debevec [5] were among the first to estimate lighting by using a sphere. They capture the lighting environment map by photographing a mirror sphere, and relight where all incoming distant illumination was modeled. It is also common to assume that the light source is a point so that reducing the problem to that of estimating the 3D point position. Powell et al. [6] and Takai et al. [7] calibrated the near point light source by capturing images with two spheres. They analyze the incoming and outgoing angles of light on a sphere, respectively, and find the position of a point light source by mixing the incoming light vector. Ackermann et al. [4] presented a new method for accurately modeling reflections on a sphere. Others, such as Kato et al. [8], Wong et al. [9], and Wu and Tang [15], also used reflective spheres to recover light directions and camera poses. However, the input images used by this approach have to include the light probes. It may be less intuitive and convenient to be used for the Augmented Reality system.

2.2 Lighting Estimation from Shadows

Another approach to estimate light sources is to observe the shadows of the objects in an image. The principle of this idea is based on the geometry of the shadow caster and correct segmentation from the shadows and background. The work of Haller et al. [12] is an example of using the geometry with known objects to analyze shadows. Wang and Samaras [13] presented a method for estimating multiple directional lights, from known geometry and Lambertian reflectance. However, this approach can only be applied to static images and necessarily know the geometry information of objects.

2.3 Lighting Estimation from Outdoor Images

For outdoor scene, the representation of illumination environment can be modeled as the fusion with sun and sky. Lalonde and Matthews [14] introduced a practical low-dimensional parametric model that accurately captures outdoor lighting. They regard sun and sky as the directional light and ambient light, respectively, and propose a Hemispherical lighting model to model it. Thus, the results can be obtained from using this model to derive priors on the likely illumination conditions of a scene. The directional lighting model of this paper is inspired by the work in [14], which will be provided in section 3.

2.4 Lighting Estimation from HDR Images

Meilland et al. [21] used an RGB-D camera as a dynamic light-field sensor, based on a dense real-time 3D tracking and mapping approach, which avoids the need for a light probe or the observation of reflective surfaces. The radiance map of the scene is derived by fusing a stream of low dynamic range images (LDR) into an HDR image. Unfortunately, fish-eye HDR cameras are expensive and not commonly available.

2.5 Lighting Estimation from Arbitrary Geometry

Pilet et al. [16] presented a fully automated approach for geometric and photometric calibration by waving an arbitrary textured planar pattern in front of the cameras. In this work, the light probe is constrained to a planar surface. They applied the light map to model real illumination conditions. The surface albedo of the planar pattern is necessarily known before lighting calibration.

Estimating the illumination from a single image of an arbitrary object is a challenging task. Lombardi and Nishino [18] proposed a probabilistic formulation that seamlessly incorporates such constraints as priors to estimate the reflectance and illumination. Without any knowledge of the 3D geometry or reflectance, Chen et al. [17] decompose a single image to the intrinsic component including shading image and reflectance image. Then the radiance map of the scene is inferred based on the scene geometry and intrinsic components.

In recent years, Park et al. [20] presented an algorithm for calibrating a near point light source rigidly attached to a camera using a single plane. They first recover shading images by filtering high frequency gradients in the input image that correspond to albedo edges. The position of light source can be inferred by mixing the bilateral symmetry axis detected from the shading. Nevertheless, this work is concentrated on calibrating the light source rigidly attached to a camera. The angle between the dominant axis of the light source and camera direction may not be too great.

Moreover, Gruber et al. [19] tried to use RGB-D camera to estimate dynamically changing environment lighting from an unknown scene. By using the RGBD camera, the information of surface normal can be used to compute the distant light field easily. The final result of rendering a surface point can be obtained by solving a linear system. However, besides RGB data, the inputs of this work also include depth image. In contrast, our work only utilizes the RGB information and does not require any extra device

3. PROPOSED METHOD

As mentioned in section 1, our augmented reality system can be divided into two major components: lighting estimation and augmented reality process. In this section, we introduce our shading model for lighting estimation, and then describe our method for estimating lighting conditions of a near point light source using several markers. Finally, we also give the augmented reality process which provides image rendering with a plausible illumination by applying the lighting conditions estimated from the scene by using the proposed algorithm.

3.1 Shading Model

We propose a novel lighting estimation method by using planar markers. Our goal is to derive the lighting conditions in real scene from our proposed method. In this paper, we assume that the observed scenes are Lambertian surfaces and they are illuminated by a nearby, isotropic point light source. Inspired by the work of Lalonde and Matthews [14], we employ a simple directional lighting model to be described as follows. The intensity at the pixel (x, y) in the image I is given by

$$I(x, y) = \rho(x, y)(a * o(x, y) + d\langle n, l(x, y)\rangle^{+}) \qquad (1)$$

where $\rho(x, y)$ and $o(x, y)$ are the diffuse albedo and ambient occlusion at pixel (x, y) respectively. To simplify the problem, we assume the ambient occlusion o can be ignored in our method. In addition, the albedo ρ can also be eliminated by replacing the input image by the shading image recovered in section 3.2.1. As a result, the shading model in our method can be modified to as follows:

$$I(x, y) = a + d\langle n, l(x, y)\rangle^{+}, \qquad (2)$$

where

$$l(x, y) = X_l - X(x, y). \qquad (3)$$

In Eq. (2), a and d are parameters in the illumination model. For a single plane, each pixel (x, y) has the same surface normal n and light direction $l(x, y)$. The light direction $l(x, y)$ for the 3D point $X(x, y)$ corresponding to the image pixel (x,y) is along the 3D vector from the its 3D position t $X(x, y)$ to the 3D near light source position X_l , both in world coordinates. Here, $\langle\cdot,\cdot\rangle^{+}$ denotes the (strictly positive) dot product representing the foreshortening term. The unknowns in Eq. (2) and Eq. (3) are the lighting parameters $\theta = [\mathrm{a}, \mathrm{d}, n, X_l]$, which consists of the ambient and diffuse light intensity, surface normal and the light direction.

To summarize, the goal of our proposed method is to find a pair of reasonable parameters θ to express the real illumination environment. We will present the proposed lighting estimation method in the next section.

3.2 Lighting Estimation Algorithm

We now describe our method for lighting estimation by using planar markers. Consider that most of the real scenes are not always a single plane. Algorithm 1 summarizes the proposed method. Intuitively, by using planar markers, we can handle the 3D scene containing two or more planes in an image. More details will be described in this section. In addition, to simplify our problem, we use the gradient filtering technique to obtain the shading image from the input image. Therefore, by replacing the input image with shading image in our lighting estimation optimization process, we can eliminate the diffuse albedo ρ from our shading model Eq. (1).

Algorithm 1. Lighting estimation method

```
Input: An image $I$ to be estimated, the corners of each marker in image coordinates $C$.
Output: the intensity of ambient $a$ and diffuse $d$, the set of the surface normal $N$ and light source position $X_l$.
1: Estimate the shading image $B$ from the input image $I$ by minimizing the Eq. (4).
2: Segment plane region $S$ using the markers in the input image $I$.
for each plane $s$ in the set of plane region $S$
    3: $C_s$ ← find the correspondences from $C$
    4: $H_s$ ← compute the homography matrix by $C_s$
    for each pixel x in the plane $s$
        5: $\mathrm{X}^{(s)}$ ← Transform $x$ from image coordinates to corresponding marker coordinates by $H_s$
        if $\mathrm{s} \neq 1$
            6: Convert $\mathrm{X}^{(s)}$ to the major marker coordinates $\mathrm{X}^{(1)}$
        end if
    end for
end for
7: Estimate the lighting parameters $\theta = [a, d, N, X_l]$ by minimizing the error function Eq. (10)
```

3.2.1 Shading Image Estimation

Since we attempt to eliminate the effect of the diffuse albedo ρ in our shading model, given in Eq. (1), and to simplify the problem, we extract the shading image from the input image using gradient filtering. Inspired by the work in Park et al. [20], we remove high frequency gradients in the input image that correspond to albedo edges. The shading image $\hat{I}$ can be recovered by minimizing the following objective function.

$$\hat{\mathrm{I}} = \underset{I}{\operatorname{argmin}} \sum_{p \in P} \left\{ \left(\frac{\partial I_p}{\partial x} - f\left(\frac{\partial O_p}{\partial x}\right) \right)^2 + \lambda w_p \left(I_p - O_p\right)^2 \right\} \qquad (4)$$

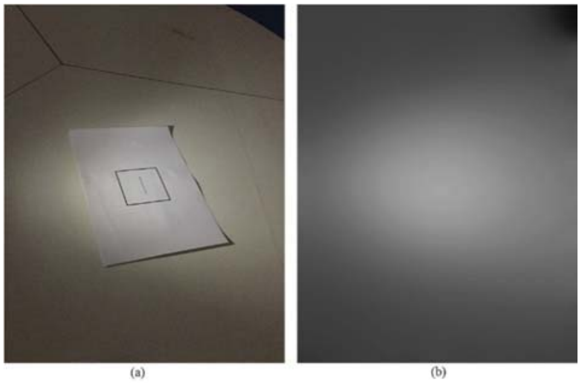

Figure 3. (a) An input image. (b) The corresponding shading image computed from the gradient of the input image.

where

$$f(x) = \begin{cases} x & \text{if } \|x\|_2 < \tau \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $f(\cdot)$ is gradient clipping function defined in Eq. (5) and O is the input image. In Eq. (4), the first term in the objective function encourages the gradients of I to match the clipped gradients of O in Eq. (4). On the other hand, the second term makes the intensity of both images as similar as possible. The weight $w \in [0,1]$ controls the second term as follows.

$$\mathrm{w_p} = 1 - |O_p - G * O_p|, \quad (6)$$

where G is a $\kappa \times \kappa$ Gaussian kernel, and $'*'$ is the convolution operator. $|O_p - G * O_p|$, the high pass filter response on O, implies that the flat pixels have lower values because they have less influences on Gaussian kernel. In contrast, the edge pixels would have higher values, so it would has less pixel weight on the edge pixels. It encourages the strong gradients in the albedo map to be eliminated in the recovered shading image. The constant λ and τ are the weight of the second term and the gradient clipping threshold for $I \in [0,255]$, respectively. An example of a recovered shading image is shown in Figure 3.

3.2.2 Plane Region Segmentation

For most of the situations in real scene, the input image does not always include only a single plane. Thus, we aim to handle the image of a 3D scene containing two or more planes. An intuitive idea is to give a different surface normal for each pixel on a different plane. Since we stick markers on each plane, the plane equation can be obtained by the augmented reality system (section 3.3.1). Here, we use the technique of augmented reality to segment the input image. In the flow of AR, the projection matrix of the transformation from 3D coordinate to 2D coordinate is computed in the procedure of camera pose estimation. We just render a large rectangle on the plane of each marker respectively. As shown in Figure 4, a reasonably good image segmentation for plane regions can be obtained by projecting these rectangles from world coordinates to image coordinates by the projection matrix estimated from camera pose estimation.

3.2.3 Coordinate Transformation

In this section, we focus on transforming the image pixels from image coordinates to world coordinates. Intuitively, by square markers, we can use homography transformation to solve this problem. Figure 5 shows that the pixels are transformed from image coordinates to marker coordinates. By searching four corners of the square marker, we can compute the homography matrix for each marker. Each homography matrix $\mathrm{H_s}$ corresponds to a plane region $s \in S$ which is segmented based on the method described in section 3.2.2. With the homography, we transform each pixel $p \in P_s$ from image coordinates to the corresponding marker coordinates.

$$\begin{bmatrix} \omega x'_p \\ \omega y'_p \\ \omega \end{bmatrix} = H_s \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}, \ z'_p = \frac{ux'_p + vy'_p}{-w} \quad (7)$$

$$\mathrm{X}_p^{(\mathrm{s})} = \begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix}, \quad (8)$$

where, $\mathrm{X}_p^{(\mathrm{s})}$ is the 3D position represented in the coordinates of the s-th plane (marker), and ω is the scale factor of homogenous coordinates. Note that the z value in marker coordinates $\mathrm{z'_p}$ is assigned by the plane equation with the s-th surface normal $\mathrm{N_s} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$. After transforming to the marker coordinates, we select a marker as the major marker, whose coordinates is regarded as the world coordinates. However, it should be converted to the coordinates of the major marker for the pixels which represent in the coordinates of other markers. Here, we respectively compute the rotation matrix $R^{(i)}$ between the surface normal of the ith marker N_i and the surface normal of the major marker N_1. The transformation can be formulated as follows.

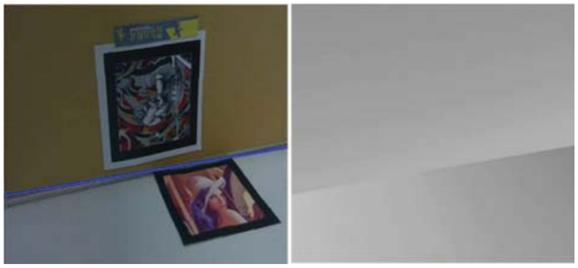

Figure 4. (a) An input image overlaid with the boundary of the two estimated planes. (b) Rendering of the two estimated planes in our AR system.

$$X_j^{(1)} = R^{(i)} X_j^{(i)} + t^{(i)}, \quad (9)$$

where, $X_j^{(i)}$ is the jth pixel represented in the coordinate of the i-th marker (Let $i = 1$ is the major marker) , and $t^{(i)}$ is the translation parameter adjusting the offset from the origin. Therefore, we can use Eq. (9) to transform the pixels from other marker coordinates to the coordinates of the major marker (world coordinates). There is an example shown in Figure 6. In this example, there are two markers in the input image, the image pixels can be separated into two plane regions, i.e. $\mathrm{s_1}$ and s_2. Intuitively, we can assign the initial guesses of the surface normal $\mathrm{N_1} = [0,0,1], N_2 = [0,-1,0]$. We choose the marker 1 as the major marker whose coordinate is regarded as the world coordinate. For the pixels $\mathrm{p} \in \mathrm{s_1}$, their positions in the world coordinate can just assign the z value to zero. In contrast, the pixels $\mathrm{p} \in \mathrm{s_2}$ should be transformed from marker 2 coordinates to marker 1 coordinates. By Eq. (9), we rotate the pixels $\mathrm{p} \in \mathrm{s_2}$ into the plane of marker 1, and translate them to a proper position. In summary, to make all the pixels represented in the same world coordinate, we first need to transform the pixels from image coordinate to the coordinate of its corresponding marker by Eq. (7)

and Eq. (8). Next, by Eq. (9), we convert these pixels from their original coordinates to world coordinates.

3.2.4 Lighting Parameters Optimization

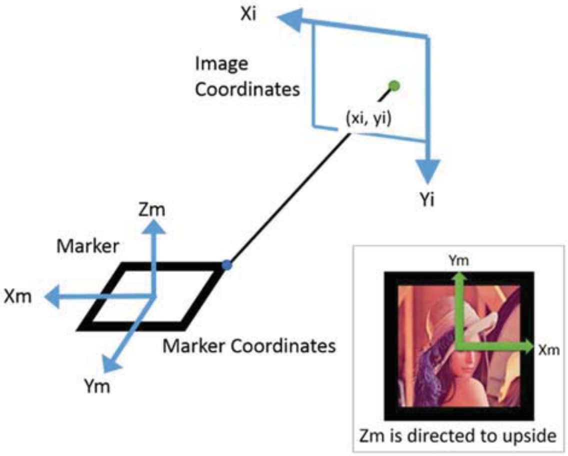

Figure 5. The relationship between marker coordinates and image coordinates observed from the screen. Here, (Xi,Yi) are the x axis and y axis respectively in image coordinates. Likewise, (Xm, Ym) are the axes in marker coordinates. We can transform the pixels by the homography computed from four points of the marker's corner found in the input image

We now describe the final step of our proposed lighting estimation method. This step is to provide the proper lighting parameters for generating a plausible illumination condition in the Augmented Reality system (section 3.3). As a result, according to our modified shading model in Eq. (2), we define the following objective function:

$$\mathrm{E}(\theta) = \sum_{s \in S} \sum_{p \in P_s} \left(I_p - a - \frac{d \langle N_s \cdot L_p \rangle^{+}}{\|N_s\| \|L_p\|} \right)^2, \qquad (10)$$

where

$$L_p = X_l - X_p, \qquad (11)$$

and $\mathrm{X}_p \in \mathbb{R}^3$ is the 3D point represented in the world coordinate which is transformed from the pixel $\mathrm{p} = [\mathrm{x}, \mathrm{y}]$ in section 3.2.3. The unknown parameters $\theta = [a, d, N, X_l]$ would be estimated by minimizing the error function Eq. (10). $N = \{\mathrm{N_s} | \mathrm{s} \in \mathrm{S}\}$ is the set of the surface normal of all the regions $s \in S$.

We estimate the lighting parameter θ by minimizing Eq. (10). This equation can be regarded as a nonlinear least square problem. Here, we employ the COBYLA (Constrained Optimization BY Linear Approximations) algorithm [25] to minimize it. In our implementation, we use the open source library NLopt [22] to optimize it. This procedure is repeated until convergence, which is obtained when the mean difference in errors between two subsequent iterations is less than 0.01%. Typically, the minimization procedure converges within 2~3 seconds for a 640x480 image with an off-the-shelf PC.

3.3 Augmented Reality System

In this section, we describe an augmented reality system that includes the proposed lighting estimation to achieve illumination-coherent rendering of virtual objects based on the ARToolkit framework [23]. In this AR system, we relight the illumination on the virtual objects by using the lighting parameters estimated from section 3.2. We developed this system based on the ARToolkit framework [23]. However, we improve the camera pose estimation by using the additional feature point correspondences obtained by extracting the SURF interest points [24] from the marker's pattern and the input image, respectively. From the SURF interest point detection, we obtain the 3D-to-2D point correspondences for camera pose estimation. The following is an overview of our AR system. First, the marker corners are extracted from the input image, and then we compute the point correspondences for camera pose estimation by SURF matching and refine them from the marker corners extracted from the first step. Next, we estimate the 3D camera pose from the point correspondences. Finally, virtual objects are rendered according to the estimated 3D pose with the illumination condition estimated by the proposed algorithm described in section 3.2.

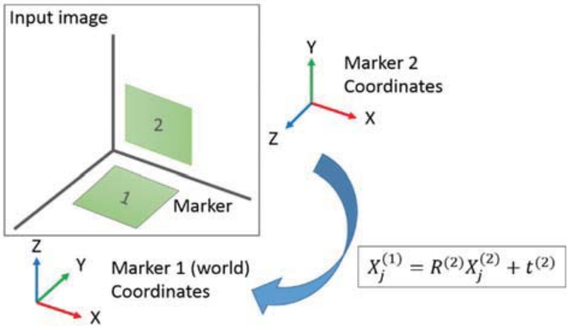
Input image
2
1
Marker
Y
Marker 2 Coordinates
X
Z
Z
Y
Marker 1 (world) Coordinates
X
$X_j^{(1)} = R^{(2)} X_j^{(2)} + t^{(2)}$

Figure 6. An example of transforming a point from marker coordinate to world coordinate. We assign marker 1 as the major marker, so the pixels which locate in the plane region $\boldsymbol{s_2}$ should be transformed to world coordinates by Eq. (9).

3.3.1 Searching for Markers and Finding Point Correspondences

In the AR system, we first search the markers in the input image and extract their four corners for camera pose estimation. To make the searching robust, we use the square marker surrounded with a black rectangle. Some examples of the markers used in our experiments are shown in Figure 7. The pattern in the middle of the rectangle can be replaced arbitrarily.

When the camera captures an image, the first step in our system is binarizing the image with a threshold σ. The region of the black rectangle would stand out in the binary image. Therefore, we can find the connected components and extract the marker edges and corners from the captured image. The marker corners will be used in conjunction with the interest points extracted from the marker region to estimate the camera pose.

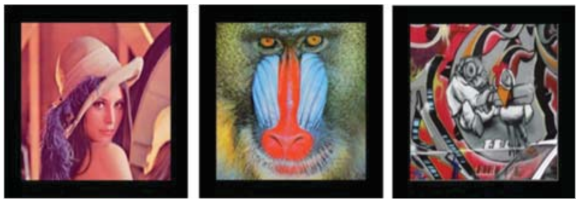

Figure 7. Some examples of the square marker with a black rectangle.

For AR applications, the image features to be used for matching should provide robust matching across a substantial range of changes in 3D viewpoints, noise levels and illumination conditions. In this paper, we use the SURF method [24] to detect the interest points to establish point correspondences between the marker pattern and the input image as depicted in Figure 8.

After extracting the SURF feature points, we match these features to find point correspondences between the input and marker pattern. Each pixel in marker pattern is regarded as a point represented in 3D world coordinates. As a result, after obtaining the 3D-to-2D point correspondences with SURF, we utilize these correspondences to estimate the camera pose in the next section.

3.3.2 Camera Pose Estimation

It is essential to find the relationship between the marker coordinate and the image coordinate, which is shown on screen, in an AR system. In other words, given the camera intrinsic parameters, we estimate the 3D pose of the marker with respect to the camera from the 3D-to-2D point correspondences by minimizing the re-projection error given by

$$\min_{R,t} \sum_i \|K[R|t]X_i - x_i\|^2 \tag{12}$$

where K is the pre-calibrated camera matrix, and R and t are the rotation matrix and translation vector, respectively. Here, we apply the Levenberg-Marquardt optimization algorithm to minimize the re-projection error function. Subsequently, we align virtual objects with markers by using the estimated 3D pose. Finally, the virtual objects are rendered with the 3D pose and lighting condition estimated from the real image.

4. EXPERIMENTAL RESULTS

In this section, we present some experimental results by using the proposed lighting estimation method. We evaluate our method with both synthetic and real data. All of our experiments were performed on a PC equipped with Intel Core i7-4770 3.4 GHz CPU and 16GB RAM.

4.1 Evaluation with Synthetic Images

In this experiment, we generated 20 synthetic images under different lighting conditions by OpenGL. These images were rendered at 600×600 resolution by using a Blinn-Phong model [3] to generate the illumination effect in the virtual scene. We rendered two or more large rectangles as the surface we observed by using different lighting parameters.

We first present the results of the error function in Eq. (10). The goal of this experiment is to verify if the proposed error function is converged to the optimal solution. We set different initial solutions and optimize it. The estimated lighting parameters $\theta = [a, d, N, X_l]$ are compared with the ground truth. Figure 9 shows that the results of the lighting estimation with different initial guesses. Here, we use the Mean Absolute Error (MAE) as one of the accuracy metrics. We generate the synthetic images using the estimated lighting parameters, and compare them to the shading image recovered from the input image. The result of comparison is shown in Figure 9(a). We can see that the differences in the MAE are not large for different initial guesses. It implies that the proposed method would optimize the lighting parameters to almost the same solution. Figure 9(b) compares the estimated ambient a and diffuse d intensity which lie in the range [0,1]. Most errors of the diffuse intensity are lower than ambient intensity. Figure 9(c) shows the errors in the estimated surface normal N for different initial guesses of lighting parameters. The errors of the estimated light source positions are shown in Figure 9(d). It roughly has an upward trend while increasing the deviation to the ground truth in the initial guess.

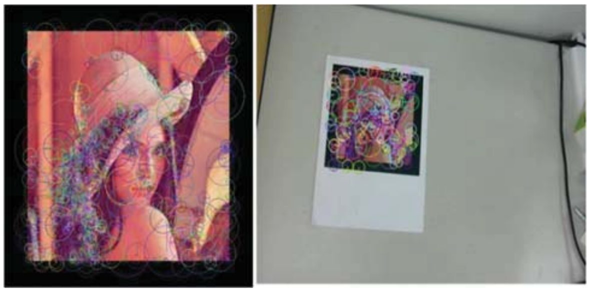

Figure 8. The interest points extracted by using SURF detector [24]: the left image is the marker pattern, and the right image is an input image captured by a camera.

Our second experiment is the comparison of the lighting estimation accuracy for the single-plane method and the multi-plane method. The single-plane method assumes all pixels in the image lying on a single plane and makes their estimated surface normal N be the same. In contrast, the multi-plane method segments the plane regions s∈S in the input image by using placing markers on all of the planes in AR (section 3.2.2). The pixel in the plane region s has the surface normal N_s. We estimate each surface normal N_s by minimizing the function in Eq. (10). In this experiment, we adjust different light source positions of the ground truth and assign them to a pair of random ambient and diffuse intensity.

Figure 10 shows the results of the comparison with these two methods. For the visual viewpoint, in spite of the MAE of multi-plane method is slightly higher than the one of single plane method, as shown in Figure 10(a), there are a significant improvement for the error of the light source position shown in Figure 10(b). Moreover, the intensity errors of these two methods are the same roughly. Some example are shown in Figure 11. For inputs on the first row, the MAE of column (c) and (d) are 11.78 and 11.60, respectively. For inputs on the second row, the MAE of column (c) and (d) are 11.68 and 17.19, respectively. We can see that the multi-plane method can model the lighting condition more accurately.

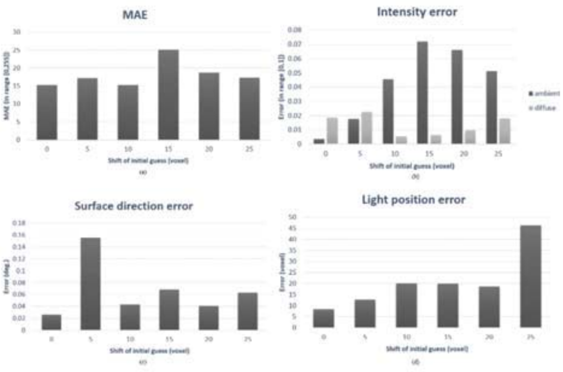

Figure 9. The results of lighting estimation with different initial guesses. Note that x-axis in all figures denotes the value of initial guess deviated from the ground truth.

4.2 Evaluation on Real Datasets

For the real data, we test the proposed lighting estimation algorithm on the datasets released from Park et al. [20]. There are two datasets with 42 images and 20 images of whiteboards captured in two different rooms. The first dataset CAMERA-LED was captured using a bright Cree XM-L LED rigidly attached to the top of a point-and-shoot camera (Panasonic DMC-LX5). The second dataset SLR-FLASH was captured with a Canon Mark III-1D camera with an auxiliary flash.

Table 1 shows the estimation results on the CAMERA-LED and SLR-FLASH datasets. The proposed method is compared with Park et al.'s work by using MAE as the accuracy metric. The MAE of our proposed method in CAMERA-LED dataset is 6.77 which is lower than 8.97, the result of Park et al. If we set the initial guess for each image, the MAE in this dataset can be decreased into 5.61. For another case SLR-FLASH, the result of our proposed method is 6.76, which is much lower than the value 15.35 produced by Park et al. [20]. While the work by Park et al. suffered from the case that the flash violates the assumption of a point light source, our proposed method is not affected by this case. The best result of our proposed method in this dataset is 5.11. There is an example shown in Figure 12.

Table 1. The MAE comparison between Park et al.'s work [20] and our proposed method. In the second row, we fix the initial guess to [0.01, 0.3, 0, 0, 30]. In the third row, we use the manually selected initial guess.

	CAMERA-LED	*SLR-FLASH*
Park et al. [22]	8.97	15.35
Ours	6.77	6.76
Ours *	5.61	5.11

Furthermore, we also test our method on real scenes containing two or more planes. Figure 13 depicts some estimation results. However, because of the evaluation with MAE, we have to make the camera pose of both images be the same. Therefore, we stick the markers close to each other for simplifying the implementation of the procedure which renders the image using the estimated lighting parameters. In this experiment, we test 20 images captured in different scenes containing two or three planes. The average MAE is 24.81. Even though the MAE is higher in these cases, we can obtain reasonable light position estimation by using the proposed method.

In Figure 14, we depict an example of the AR system with the virtual 3D objects rendered with the estimated illumination parameters by using the proposed algorithm and imposed into the original image.

5. CONCLUSION

In summary, we proposed a novel lighting estimation method from a single image of 3D planes. To improve the performance for the case of the images containing two or more planes, we utilize the planar markers to estimate the simple layout of the 3D scene easily. The estimation of the lighting parameters is accomplished by minimizing the error function derived in this work. In our experimental evaluation, we compare the proposed algorithm with the method by Park et al. [20] and our estimation results are considerably better than those of the previous method. Furthermore, we developed an augmented reality system that renders virtual objects with plausible illumination by using the lighting parameters estimated by the proposed algorithm from the input image. The proposed lighting estimation algorithm can provide improved accuracy by jointly optimizing all model parameters in an optimization framework.

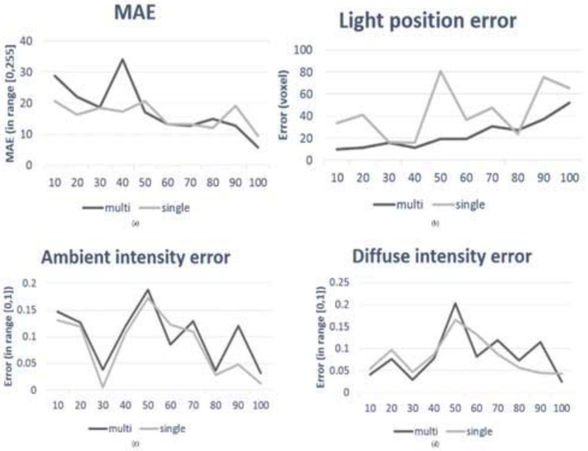

Figure 10. The errors in the lighting estimation with the multi-plane and single-plane methods. Note that x-axis in all figures denotes the distance between the center of world coordinates and the light position of the ground truth.

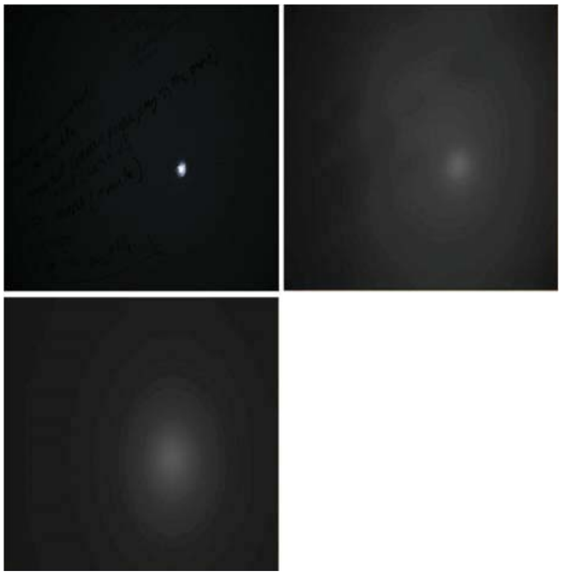

Figure 12. An example of the result of our proposed method in CAMERA-LED dataset. The top left image is the input image. The top right image is the shading image recovered from the input image. The bottom image is the shading image recovered from the rendered image using the estimated light parameters. Image contrast is stretched for better visualization.

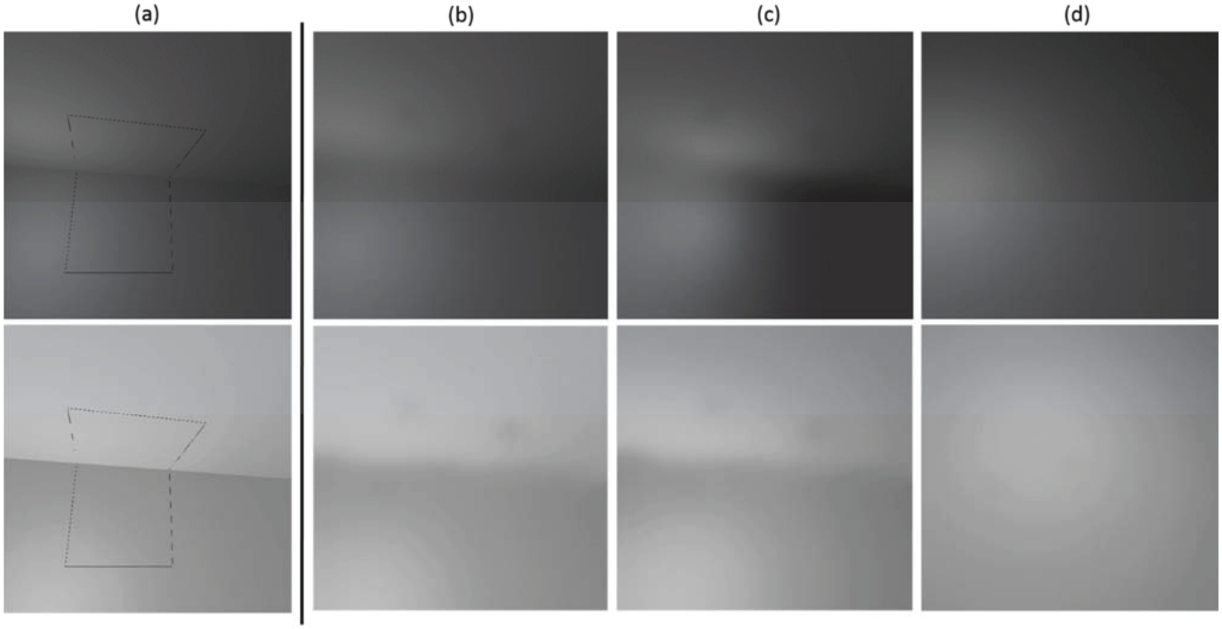

Figure 11. Some examples in the comparison for the two methods. Column (a) is the input synthetic image. Column (b) is the shading image recovered from the input image. Column (c) and (d) are the shading image recovered from the rendered image using estimated lighting parameters by the multi-plane method and the single-plane method, respectively**.**

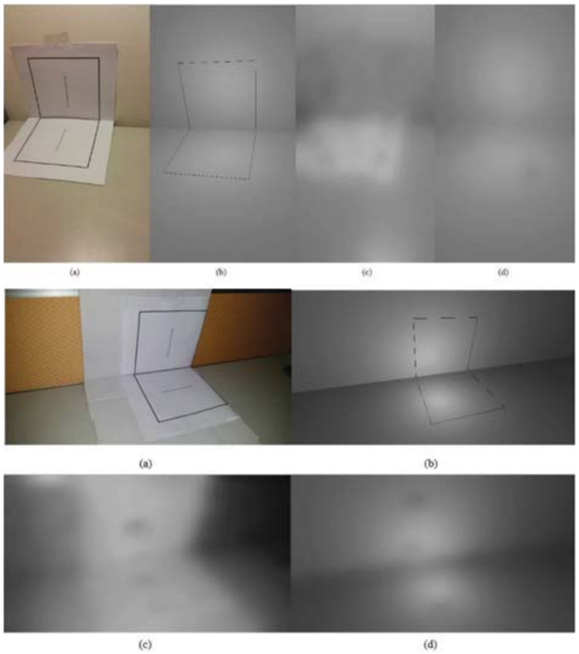

Figure 13. Evaluation for the scene containing two planes: (a) input image, (b) rendered image using the estimated lighting parameters, (c) shading image recovered from the input image, (d) shading image recovered from (b).

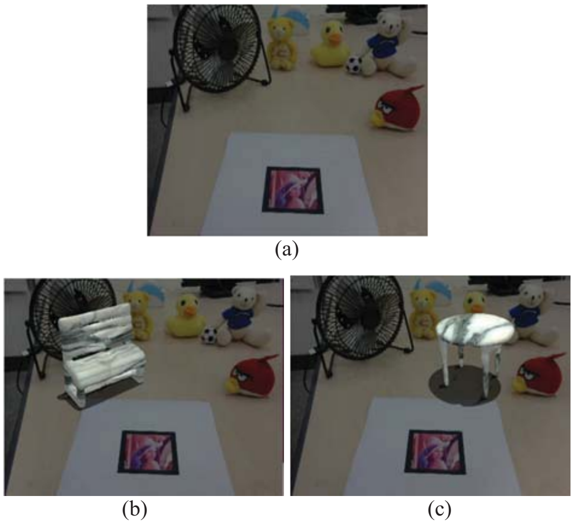

Figure 14. An AR example with the lighting condition of the scene estimated by the proposed algorithm: (a) original image, (b) and (c) virtual 3D models rendered with the estimated lighting condition and imposed into the original image.

6. REFERENCES

[1] Oxholm, G. and Nishino, K. 2014. Multiview shape and reflectance from natural illumination. *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* CVPR'14, 2163–2170.

[2] Johnson, M. K. and Adelson, E. H. 2011. Shape estimation in natural illumination. *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* CVPR'11, 2553–2560.

[3] Blinn, J. F. 1977. Models of light reflection for computer synthesized pictures. *Proc. of SIGGRAPH'77*, 11, 2, 192–198.

[4] Ackermann, J., Fuhrmann, S, and Goesele, M. 2013. Geometric point light source calibration. *Proceedings of the Vision, Modeling and Visualization Workshop*, 161–168.

[5] Debevec, P. 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Proceedings of SIGGRAPH'98,* 189-198.

[6] Powell, M. W., Sarkar, S, and Goldgof, D. 2001. A simple strategy for calibrating the geometry of light sources. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23, 9 (Sep. 2001), 1022–1027.

[7] Takai, T., Maki, A., Niinuma, K., and Matsuyama, T. 2009. Difference sphere: an approach to near light source estimation. *Computer Vision and Image Understanding*, 113, 9, 966–978.

[8] Kato, K., Sakaue, F., and Sato, J. 2010. Extended multiple view geometry for lights and cameras from photometric and geometric constraints. *Proc. International Conference on Pattern Recognition.* (Istanbul, Turkey, Aug. 23-26, 2010). ICPR'10, 2110–2113.

[9] Wong, K. Y. K., Schnieders, D., and Li, S. 2008. Recovering light directions and camera poses from a single sphere. *Proc. European Conference on Computer Vision*, 631–642.

[10] Yu, L. F., Yeung, S. K., Tai, Y. W., Terzopoulos, D., and Chan, T. F. 2013. Outdoor photometric stereo. *Proc. International Conference on Computational Photography,* 1–8.

[11] Schnieders, D. and Wong, K. Y. K. 2013. Camera and light calibration from reflections on a sphere. *Computer Vision and Image Understanding*, 117(10),1536–1547.

[12] Haller, M., Drab, S., and Hartmann, W. 2003. A real-time shadow approach for an augmented reality application using shadow volumes. *Proc. ACM symposium on Virtual reality software and technology*, 56–65.

[13] Wang, Y. and Samaras, D. 2003. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models*, 65(4),185–205.

[14] Lalonde, J. F. and Matthews, I. 2014. Lighting estimation in outdoor image collections. *Proc. International Conference on 3D Vision.* 1, 131–138.

[15] Wu, T. P. and Tang, C. K. 2005. Dense photometric stereo using a mirror sphere and graph cut. *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* 1, 140–147.

[16] Pilet, J., Geiger, A., Lagger, P., Lepetit, V., and Fua, P. 2006. An all-in-one solution to geometric and photometric calibration. *Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality,* 69–78.

[17] Chen, X., Wang, K. and Jin, X. 2011. Single image based illumination estimation for lighting virtual object in real scene. *Proc. International Conference on Computer-Aided Design and Computer Graphics*, 450–455.

[18] Lombardi, S. and Nishino, K. 2012. Reflectance and natural illumination from a single image. *Proc. European Conference on Computer Vision*, 582–595.

[19] Gruber, L., Richter-Trummer, T., and Schmalstieg, D. 2012. Real-time photometric registration from arbitrary geometry. *Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality*, 119–128.

[20] Park, J., Sinha, S. N., Matsushita, Y., Tai, Y. W. and Kweon, I. S. 2014. Calibrating a non-isotropic near point light source using a plane. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2267–2274.

[21] Meilland, M., Barat, C., and Comport, A. 2013. 3D high dynamic range dense visual slam and its application to real-time object re-lighting. *Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality*, 143–152.

[22] NLopt. http://ab-initio.mit.edu/wiki/index.php/Main_Page

[23] Kato, H. and Billinghurst, M. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *Proc. International Workshop on Augmented Reality*, 85–94.

[24] Bay, H., Tuytelaars, T., and Van Gool, L. 2006. Surf: Speeded up robust features. *Proc. European Conference on Computer Vision*, 404–417.

[25] Powell, M. J. 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. *In Advances in optimization and numerical analysis*, 51–67.