# Lighting Estimation from a Single Image of Multiple Planes

Pin-Cheng Kuo, Hsin-Yuan Huang, and Shang-Hong Lai
National Tsing Hua University
Taiwan

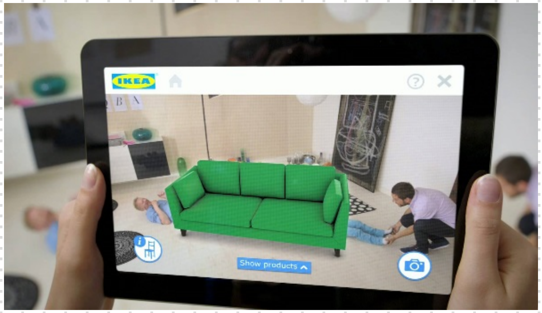
ACM MMSys, Klagenfurt, Austria, May 12, 2016

# Outline

- Introduction
- Related Works
- Proposed Method
- Experimental Results
- Conclusion

# Motivation

- Augmented Reality (AR) has attracted increasing attention in recent years.
- Delivering a visually coherent rendering plays an important role in the AR applications.
- However, relatively little work has been done for online lighting estimation from the scene images.



# Problem Description

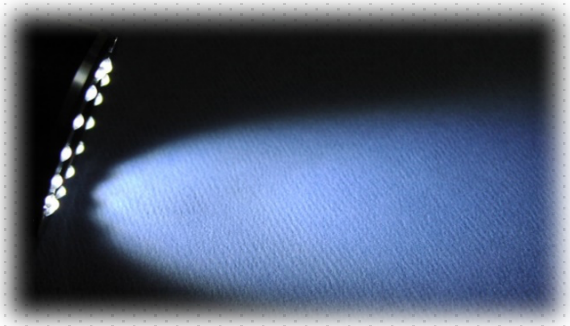
- In this paper, we aim to estimate the illumination conditions of near light source at indoor scene. And we render the lighting effect by using the estimated lighting parameters.

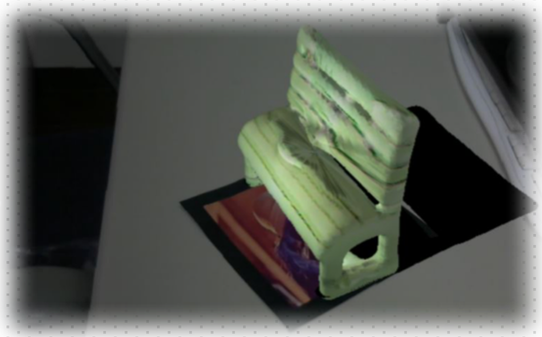
**Near Light source Estimation**

Estimating the lighting parameters from a single shaded image

**Augmented Reality System**

Render the lighting effect for virtual contents



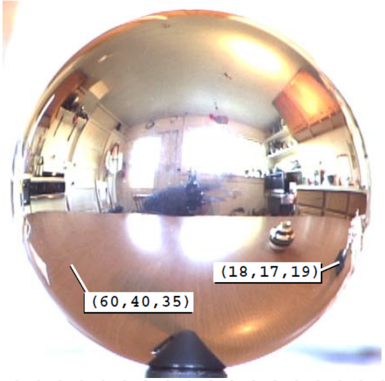


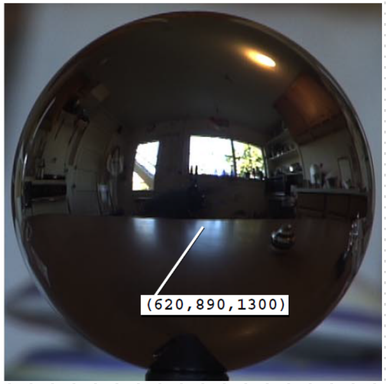
# Related works

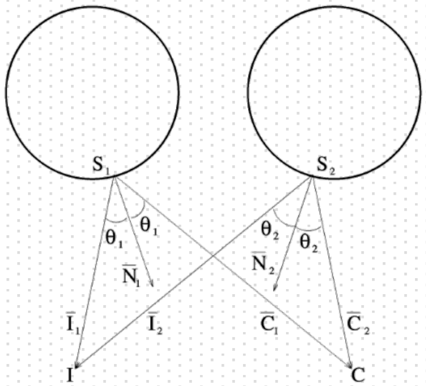
- In the following, there are five primary research directions related to lighting estimation problem.
  - **Light probes**
  - **Shadows**
  - **Outdoor images**
  - **HDR images**
  - **Arbitrary geometry**

# Lighting estimation from light probes

- Debevec [7] were among the first to estimate lighting by using a sphere. They capture the lighting environment map by photographing a mirror sphere, and relighting where all incoming distant illumination was modeled.
- Powell et al. [8] and Takai et al. [9] calibrated the near point light source by capturing images with two spheres.



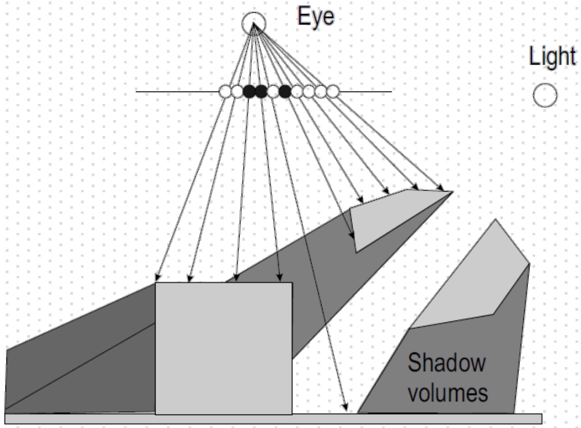




[7] Debevec, Paul. "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography." ACM SIGGRAPH 2008 classes. ACM, 2008.
[8] Powell, Mark W., Sudeep Sarkar, and Dmitry Goldgof. "A simple strategy for calibrating the geometry of light sources." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.9 (2001): 1022-1027.
[9] Takai, Takeshi, et al. "Difference sphere: an approach to near light source estimation." Computer Vision and Image Understanding 113.9 (2009): 966-978.

# Lighting estimation from shadows

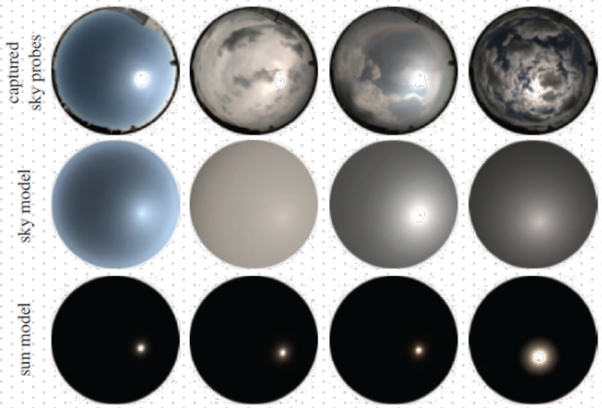
- The principle of this idea is based on the geometry of the shadow caster and correct segmentation for the shadows and background.
- The work of Haller et al. [14] is an example of using the geometry with known objects to analyze shadows.
- Wang and Samaras [15] presented a method for estimating multiple directional lights, from known geometry and Lambertian reflectance.




[14] Haller, Michael, Stephan Drab, and Werner Hartmann. "A real-time shadow approach for an augmented reality application using shadow volumes." Proceedings of the ACM symposium on Virtual reality software and technology. ACM, 2003.
[15] Wang, Yang, and Dimitris Samaras. "Estimation of multiple directional light sources for synthesis of augmented reality images." Graphical Models 65.4 (2003): 185-205.

# Lighting estimation from outdoor images

- Lalonde and Matthews [16] 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.




[16] Lalonde, Jean-Francois, and Iain Matthews. "Lighting Estimation in Outdoor Image Collections." International Conf. on 3D Vision (3DV), 2014.

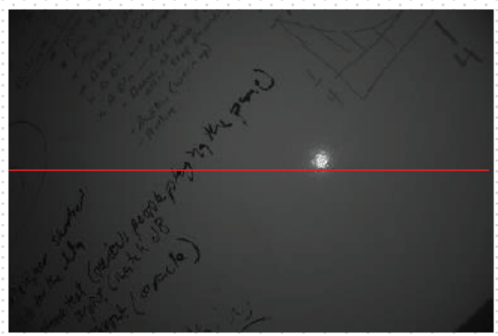
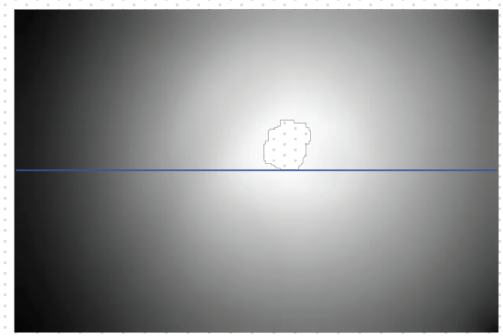
# Lighting estimation from HDR Cameras

- Meilland et al. [23] used an RGB-D camera as a dynamic light-field sensor, based on a dense real-time 3D tracking and mapping approach.
- The radiance map of the scene is estimated by fusing a stream of low dynamic range images (LDR) into an HDR image.

[23] Meilland, Maxime, Christian Barat, and Andrew Comport. "3D high dynamic range dense visual slam and its application to real-time object re-lighting." IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2013.

# Lighting estimation from arbitrary geometry

- Pilet et al. [18] presented a fully automated approach for geometric and photometric calibration by waving an arbitrary textured planar pattern in front of the cameras.
- Park et al. [22] focus on calibrating a near point light source rigidly attached to a camera using a single plane. They recover shading images by filtering high frequency gradients in the input image that correspond to albedo edges.

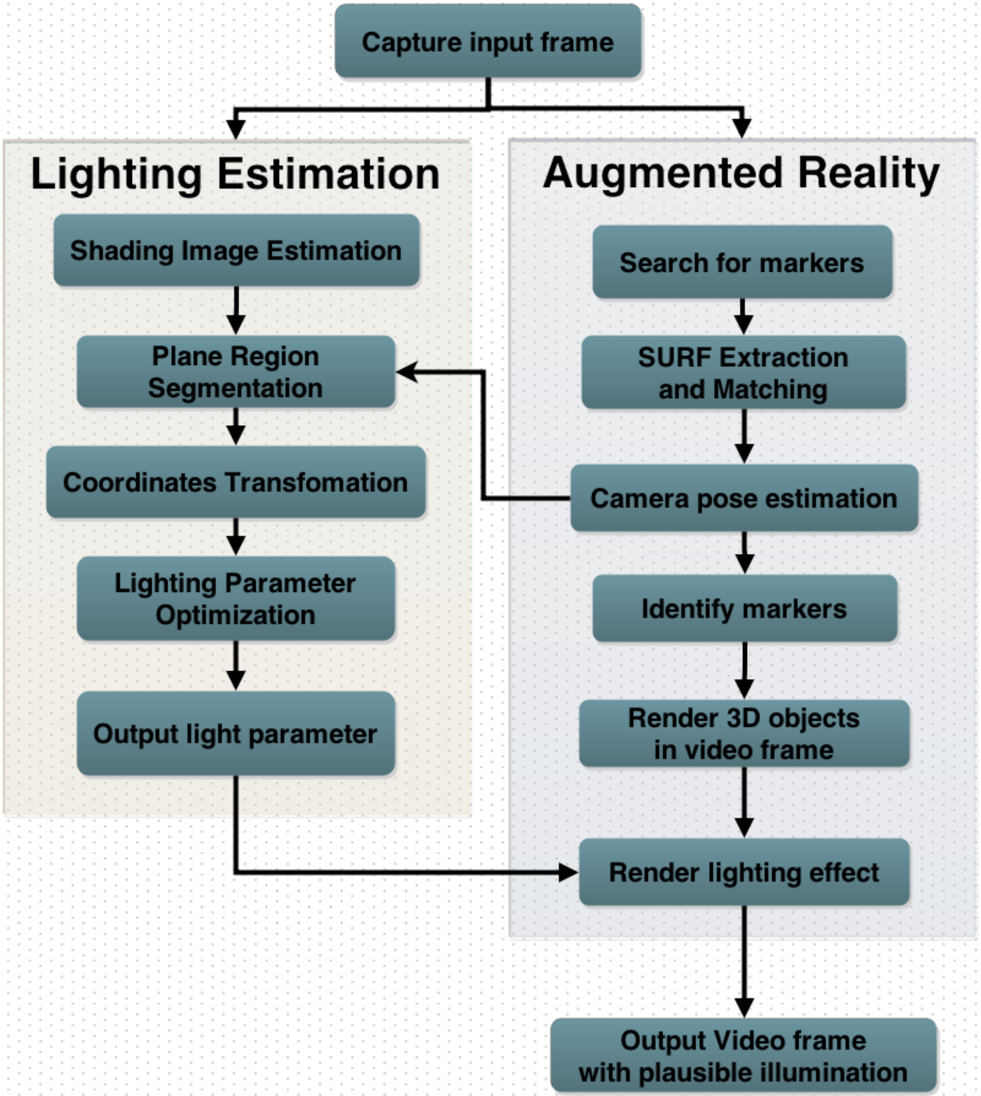




[18] Pilet, Julien, et al. "An all-in-one solution to geometric and photometric calibration." Mixed and Augmented Reality, 2006. IEEE/ACM International Symposium on ISMAR, 2006.
[22] Park, Jaesik, et al. "Calibrating a non-isotropic near point light source using a plane." Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

# Contributions

- We propose an image-based approach that estimates the illumination condition of a near point light source for indoor scene.
- We generalize the original lighting estimation algorithm for a 3D plane to 3D scenes containing two or more planes.
- We develop an Augmented Reality system which renders the virtual objects with plausible illumination after estimating the illumination conditions from real world.

# System Overview



# Shading Model



- Inspired by the work of Lalonde and Matthews [16], we employ a simple directional lighting model as follows. The intensity at the pixel $(x, y)$ in the image $I$ is given by

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

- To simplify the problem, we assume the ambient occlusion $o$ can be ignored in our method.
- The albedo $\rho$ can also be eliminated by replacing the input image by the shading image.

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

$$l(x,y) = X_l - X(x,y), \quad (2\text{-}1)$$

[16] , J.-F. Lalonde and I. Matthews. "Lighting Estimation in Outdoor Image Collections." 2nd International Conference on 3D Vision (3DV), 2014

# Shading Image Estimation (1/2)



- Since we attempt to eliminate the effect of the diffuse albedo $\rho$ in our shading model, we extract the shading image from the input image using gradient filtering.
- Inspired the work in Park et al. [22], the shading image $\hat{I}$ can be recovered by minimizing the following objective function.

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

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

[22] J. Park et al. "Calibrating a non-isotropic near point light source using a plane." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

# Shading Image Estimation (2/2)

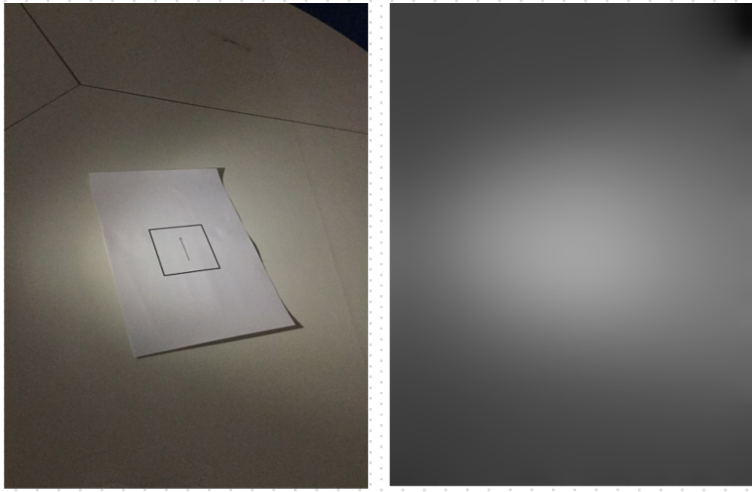


$$\hat{I} = \underset{I}{\operatorname{argmin}} \sum_{p \in P} \left\{ \left( \frac{\partial I_p}{\partial x} - f\left( \frac{\partial O_p}{\partial x} \right) \right)^2 + \lambda w_p \left( I_p - O_p \right)^2 \right\},$$

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

- The first term encourages the gradients of $I$ to match the clipped gradients of $O$.
- The second term makes the intensity of both images as similar as possible.
- The weight $w_p$ is defined by

$$w_p = 1 - \left| O_p - G * O_p \right|,$$



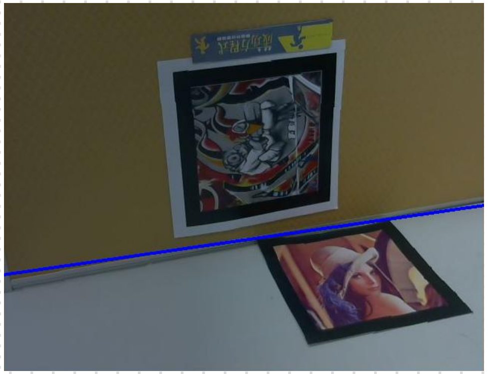
(a) (b)

**Figure 3.** (a) An input image. (b) The corresponding shading image computed by gradient

# Plane Region Segmentation



- Here, we use the marker-based 3D pose estimation technique commonly used in AR to segment the input image.
- A reasonably good image segmentation for plane regions can be obtained by projecting these rectangle from world coordinates to image coordinates by the projection matrix estimated from camera pose estimation.



# Coordinates Transformation (1/2)



- By searching four corners of the square marker, we can compute the homography matrix for each marker.

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

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

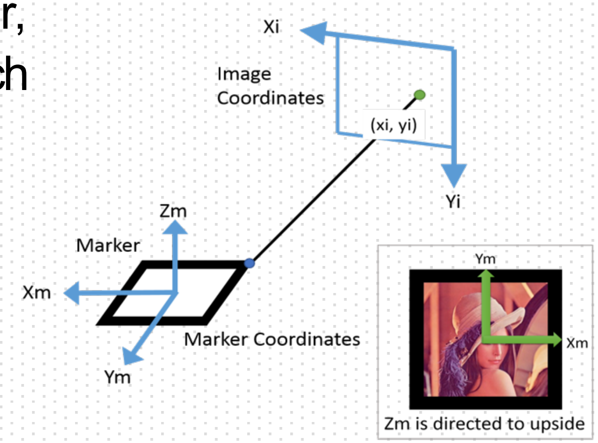



**Figure 5.** The relationship between marker coordinates and image coordinates observed from the screen. Here, (Xi,Yi) are the x axis and y axis respectively in image coordinates. Likewise, (Xm, Ym) are the axes in marker coordinates. We can transform the pixels by the homography computed from four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

- We transform each pixel $p \in P_s$ from image coordinates to the corresponding marker coordinates.
- $\mathrm{z}'_\mathrm{p}$ is assigned by the plane equation with the $s$-th surface normal $\mathrm{N_s} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$.

# Coordinates Transformation (2/2)



- After transforming to the marker coordinates, we select a marker as **the major marker**, whose coordinate system is regarded as the world coordinate.

$$X_j^{(1)} = R^{(i)} X_j^{(i)} + t^{(i)}, \qquad (7)$$

- The rotation matrix $R^{(i)}$ and translation vector $t^{(i)}$ can be computed previously since we know the layout of the markers.
- We can use Eq.(7) to transform the pixels from other marker coordinates to the coordinates of the major marker (world coordinates).

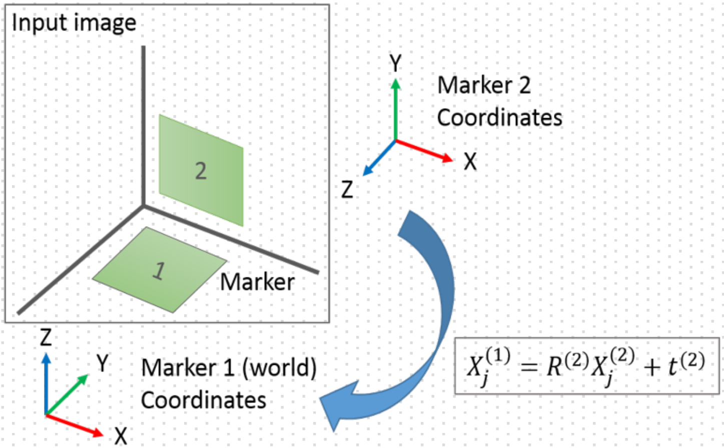



**Figure 6.** An example of transforming a point from marker coordinates to world coordinates. We assign that the marker 1 is the major marker, so the pixels which locate in the plane region $s_2$ should be transformed to world coordinates by Eq. (7).

# Lighting Parameters Optimization



- We define the error function:

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

$$L_p = X_l - X_p$$

where $\theta = [a, d, N, X_l]$ consists of the parameters that we want to optimize and $N = \{\mathrm{N}_s | s \in S\}$ is the set of the surface normal for all the regions $s \in S$.

- We estimate the lighting parameter θ by minimizing the above error function. It can be regarded as a nonlinear least square problem. Here, we employ the COBYLA (Constrained Optimization BY Linear Approximations) algorithm [27] to minimize it.

[27] Powell, Michael JD. "A direct search optimization method that models the objective and constraint functions by linear interpolation." Advances in optimization and numerical analysis. Springer Netherlands, 1994. 51-67.

# Searching for markers



- In this section, we 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.
- When the camera captures an image, the first step in our system is binarizing the image with a threshold $\sigma$. 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. The marker corners will be used to estimate the camera pose.

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

# SURF Feature Extraction and Matching



- To make pose estimation more accurate, we use Speeded Up Robust Feature (SURF) to detect the interest points to establish point correspondences between the marker pattern and the input image.
- Each pixel in marker pattern is regarded as a point represented in 3D world coordinates.
- After extracting SURF feature points, we match these features to find point correspondences between the input and marker pattern.
- As a result, we can obtain the 3D-to-2D point correspondences for estimating camera pose.

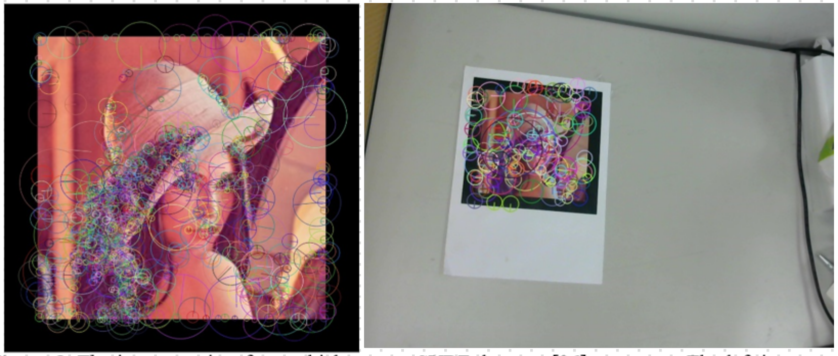


**Figure 8.** The interest points from which we use SURF detector [26] to extract. The left image is the marker pattern in recognized library, and the right image is input image captured by a video sequence.

# Camera Pose Estimation

- In this section, we are going to estimate the transform matrix which converts 3D world coordinates to 2D image coordinates.
- We can state this problem as a scene view is formed by projecting 3D points into the image plane using a perspective transformation.

$$\min_{R,t} \sum_{i} \|K[R|t]X_i - x_i\|^2$$

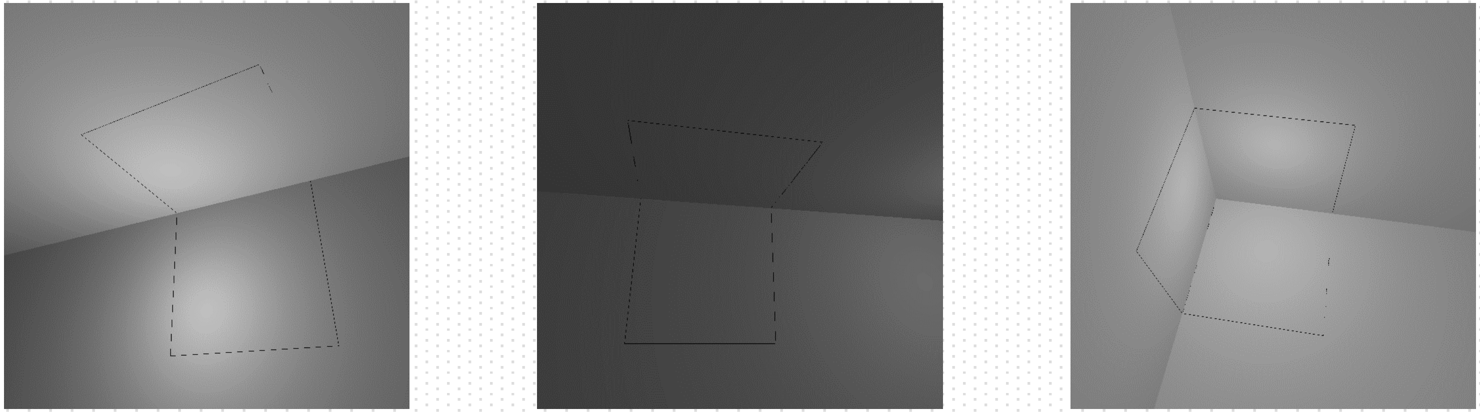
where $K$ is the camera matrix. It is a non-linear least-squares minimization problem, and we estimate the rotation matrix $R$ and translation vector $t$ while minimizing the reprojection error.

- Finally, the virtual objects are rendered at the corresponding positions in the real image.

# Evaluation with Synthetic Images



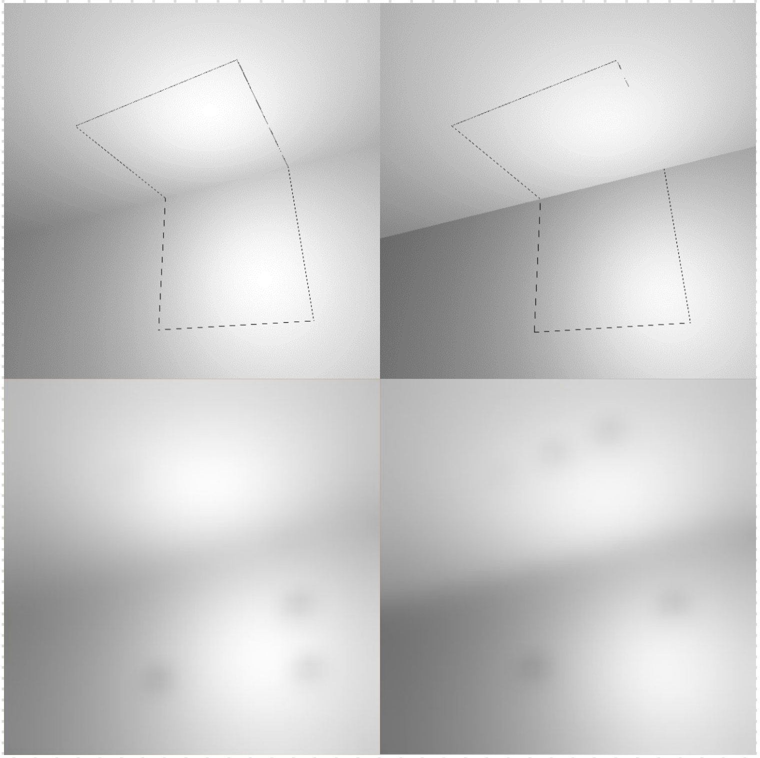
- We generated 20 synthetic images by OpenGL programming.
  - 600x600 resolution
  - Rendered by Blinn-Phong model [5]
  - The light intensity and position were generated randomly.



# Evaluation Metrics



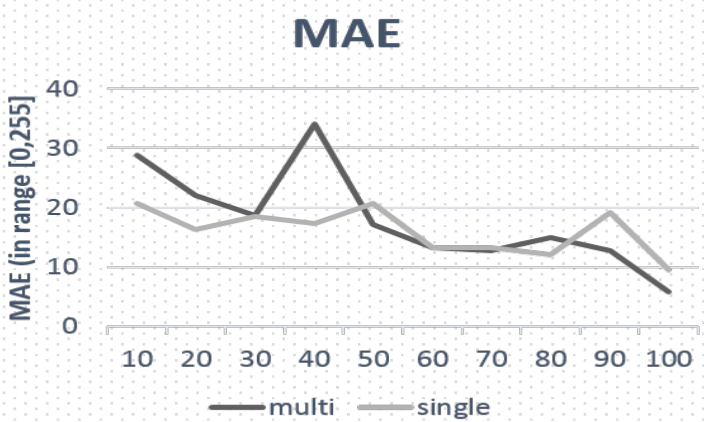
- Mean absolute error (MAE)
  - We generate the synthetic images using estimated lighting parameters, and compare them to the shading image recovered from the input image.
- Light position error
  - We directly compute the Euclidean distance between the estimated results and the ground truth of the synthetic images.

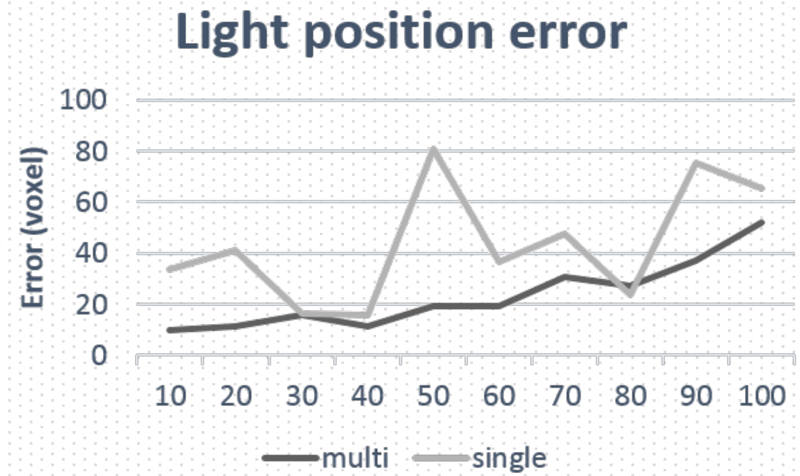


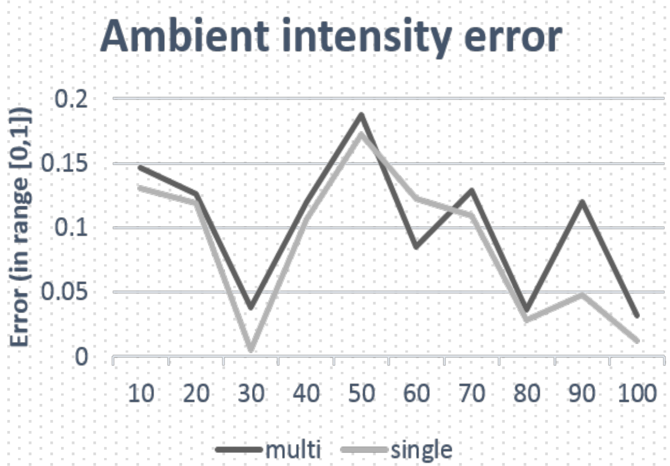

MAE: 10.9667

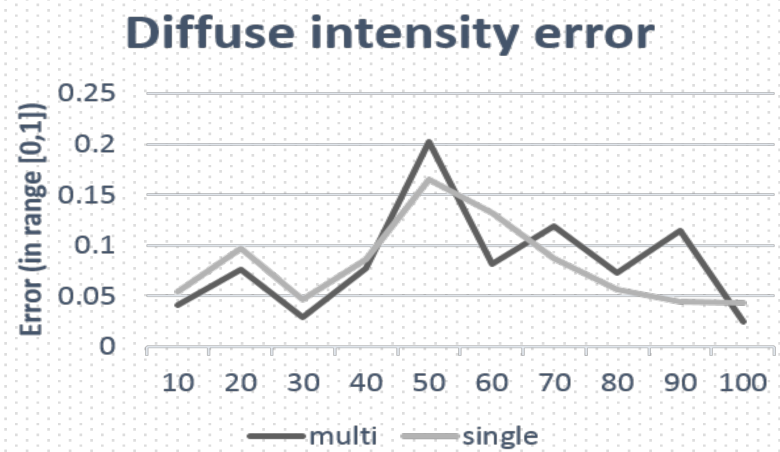
# Comparison of multi-plane and single-plane method





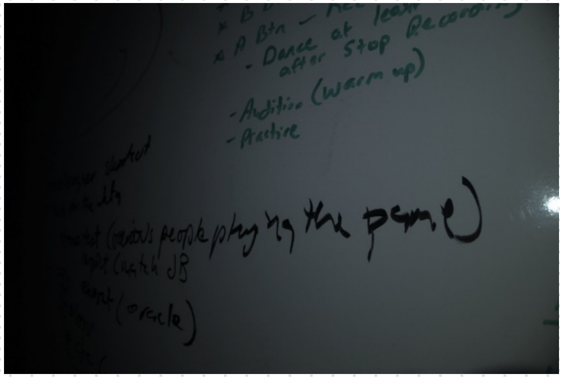



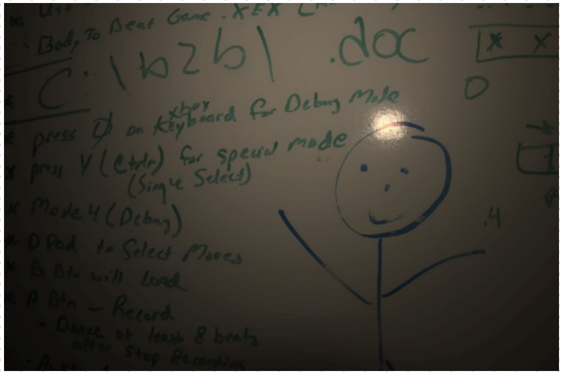



# Comparison with Park et al.'s work



- We evaluate with the dataset released from Park et al. [22].
- There are two datasets captured in different dark rooms.
  - CAMERA-LED has 42 images of whiteboards
  - SLR-FLASH has 31 images of whiteboards

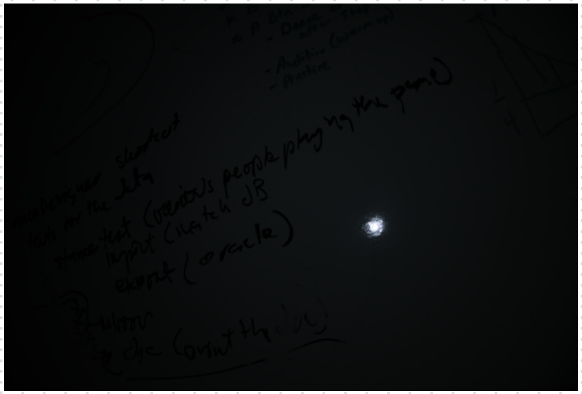




[22] Park, Jaesik, et al. "Calibrating a non-isotropic near point light source using a plane." Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

# Comparison with Park et al.’s work





Input image

**Table1**. The MAE comparison between Park et al.’s work [22] and our proposed method.

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

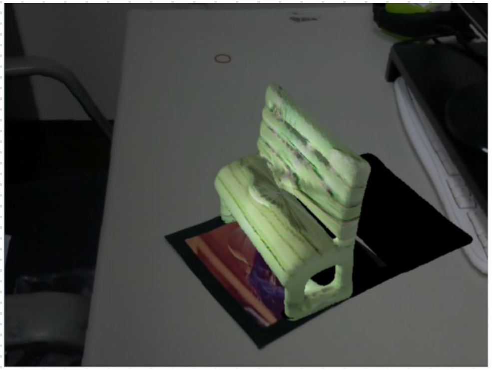
MAE : 4.32606

Input_shading

Relight_shading

# AR System Implementation

- Based on ARToolkit
- A simple framework for creating real-time augmented reality applications
  - Based on OpenGL



# AR demo video

# Conclusion

- We propose 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.
- We compare the proposed algorithm with the method by Park et al. [22] and our estimation results are considerably better than those of the previous method.
- We also 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.

**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 $X_l$.

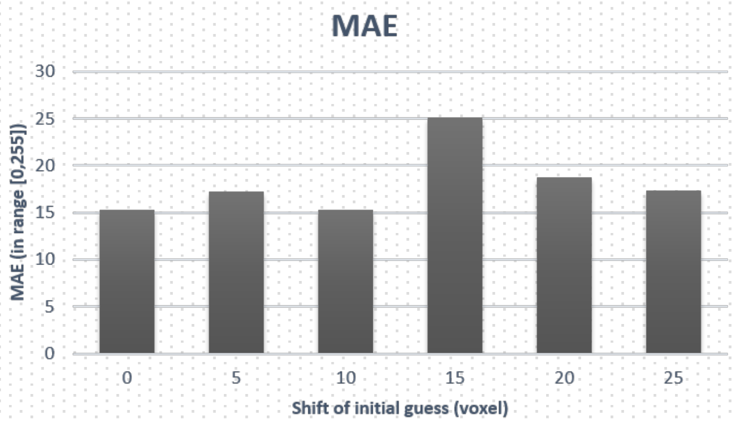
1: Estimate the shading image $B$ from the input image $I$ by minimizing the Eq. (3).
2: Segment plane region $S$ using the markers in the input image $I$.
  **for** each plane $s$ in the set of plane region $S$
3:   $C_s \leftarrow$ find the correspondences from $C$
4:   $H_s \leftarrow$ compute the homography matrix by $C_s$
    **for** each pixel x in the plane $s$
5:     $\mathrm{X}^{(s)} \leftarrow$ 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.(8-1)

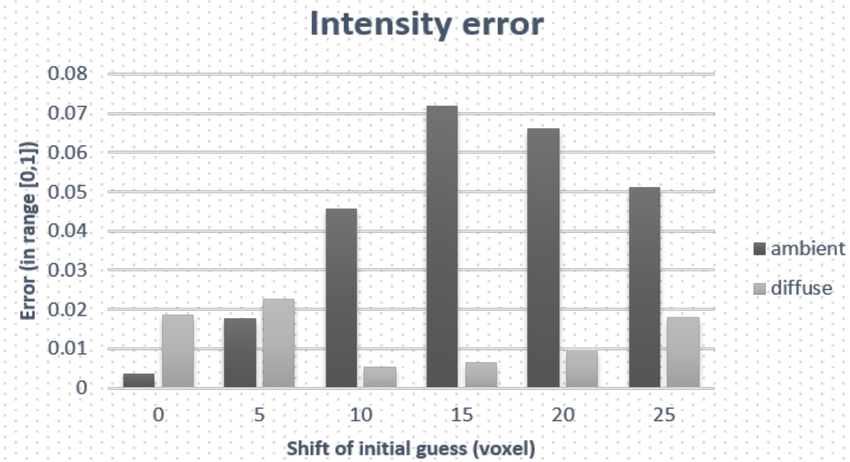
Lighting Estimation Algorithm

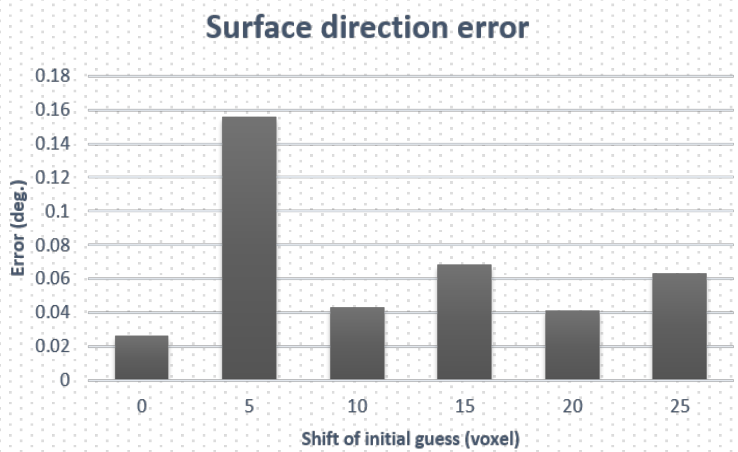
Augmented Reality System

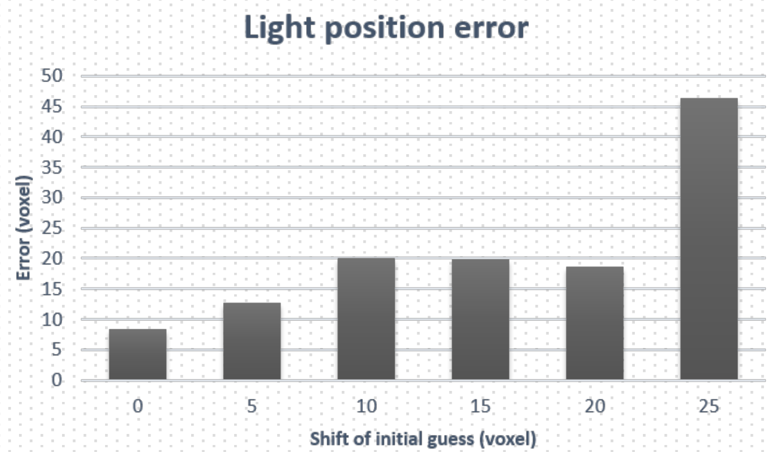
# Error function Verification





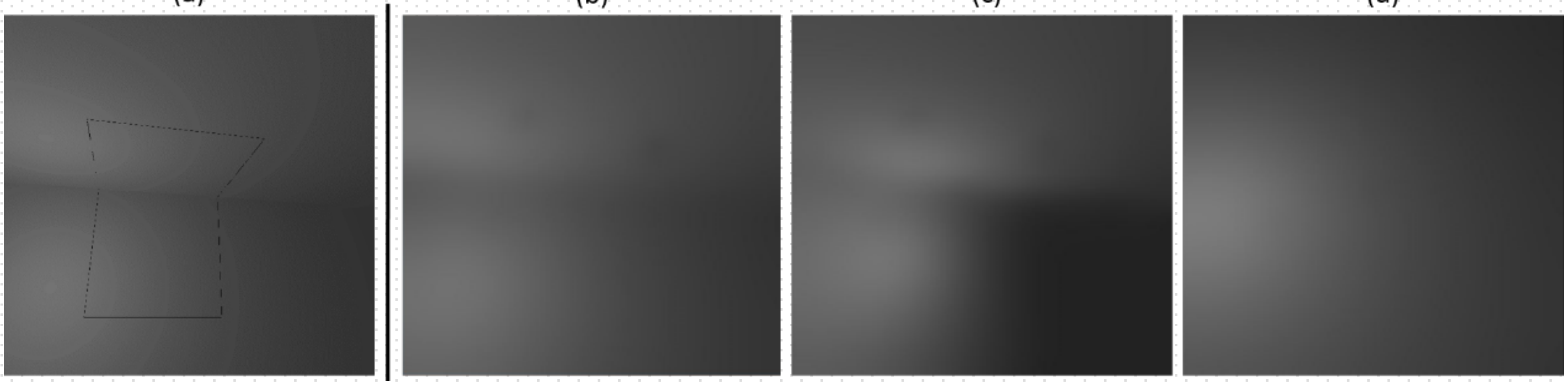



# Example of the comparison



| | MAE | Average cost | ambient | diffuse | n_x | n_y | n_z | n_x | n_y | n_z | pos_x | pos_y | pos_z | input |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GT | | | 0.10 | 0.35 | 0.00 | 0.00 | 1.00 | 0.00 | -1.00 | 0.00 | -30.00 | -12.00 | 30.00 | synth_25 |
| | 11.7793 | 0.000029532 | 0.135372 | 0.331123 | 0.459847 | 0.00234666 | 0.887995 | 0.149743 | -0.988699 | 0.0072146 | -9.52394 | -2.90508 | 19.0437 | synth_25 |
| | 11.6027 | 0.000049816 | 0.0873997 | 0.404178 | -0.0106057 | 0.00131071 | 0.999943 | X | X | X | -39.5887 | 9.10934 | 31.7505 | synth_25 |

(a) (b) (c) (d)

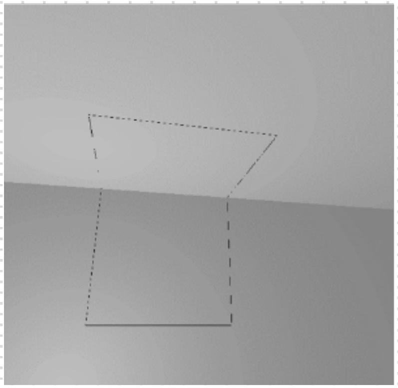


Input image | Input_shading | Multi-plane_shading | Single-plane_shading

# Example of the comparison



| | MAE | Average cost | ambient | diffuse | n_x | n_y | n_z | n_x | n_y | n_z | pos_x | pos_y | pos_z | input |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GT | | | 0.4553 | 0.2909 | 0.00 | 0.00 | 1.00 | 0.00 | -1.00 | 0.00 | -23.6197 | -35.4461 | 26.1856 | synth_28 |
| | 11.6825 | 0.000033115 | 0.216919 | 0.519694 | -0.146361 | -0.0921812 | 0.984927 | -0.0928225 | -0.935315 | 0.3414240 | -27.4901 | -34.13 | 38.9412 | synth_28 |
| | 17.1857 | 0.000079786 | 0.214364 | 0.477558 | 0.00216587 | -0.0018267 | 0.999996 | X | X | X | -3.6466 | 31.1282 | 73.042 | synth_28 |



(a) Input image

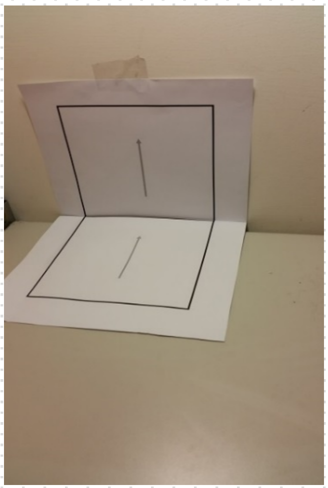
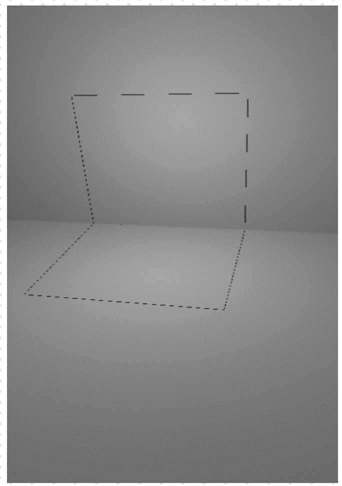
(b) Input_shading

(c) Multi-plane_shading

(d) Single-plane_shading

# Results of real scene including two planes

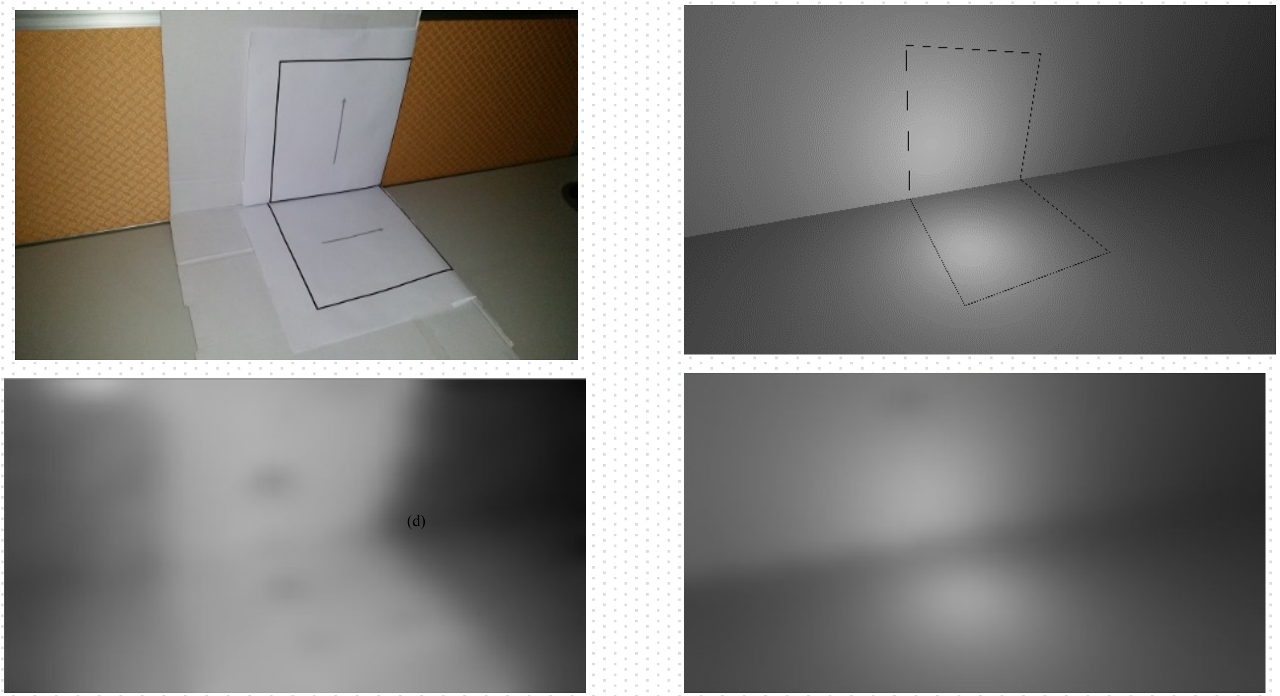


| Input name | Ambient $a$ | Diffuse $d$ | | X | Y | Z | MAE |
|---|---|---|---|---|---|---|---|
| input34 | 0.1865 | 0.4601 | **Pos.** $X_l$ | 5.3815 | -8.6364 | 34.2992 | 12.44 |
| | | | **Normal** $N_1$ | 0.0037 | -0.0116 | 0.9999 | |
| | | | **Normal** $N_2$ | 0.0000 | -0.9999 | 0.0122 | |

# Results of real scene including two planes





(c) (d)

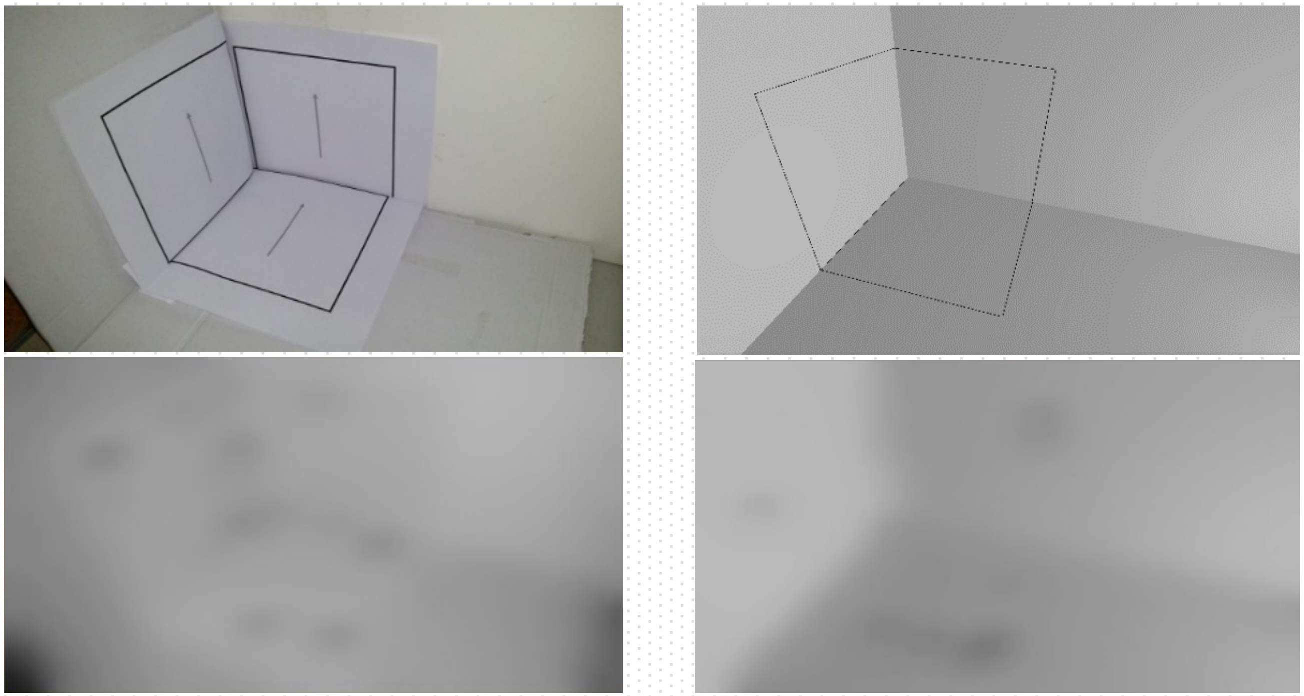
| Input name | Ambient $a$ | Diffuse $d$ | | X | Y | Z | MAE |
|---|---|---|---|---|---|---|---|
| input41 | 0.0943 | 0.5963 | **Pos.** $X_l$ | -12.8731 | -6.2155 | 15.4979 | 25.17 |
| | | | **Normal** $N_1$ | -0.0132 | 0.0106 | 0.9999 | |
| | | | **Normal** $N_2$ | 0.0168 | -0.9999 | 0.0004 | |

# Results of real scene including three planes



(a)

(c)



| Input name | Ambient $a$ | Diffuse $d$ | | X | Y | Z | MAE |
|---|---|---|---|---|---|---|---|
| input36 | 0.5000 | 0.2294 | Pos. $X_l$ | -12.8731 | -6.2155 | 15.4979 | 21.66 |
| | | | Normal $N_1$ | 0.0182 | -0.0027 | 0.9998 | |
| | | | Normal $N_2$ | -0.0004 | -0.9998 | 0.0184 | |
| | | | Normal $N_3$ | 0.9998 | 0.0148 | -0.0108 | |