Fast and Accurate Illumination Estimation Using LDR Panoramic Images for Realistic Rendering

Supplementary Material

1 Overview

In this supplementary material, additional results and algorithm details are represented to complement the main manuscript. In Sec. 2, we show more illumination calculation results in different real-world scenes using our method. In Sec. 3, we analyse the computational efficiency of the proposed method. In Sec. 4, we show the visual comparison of realistic rendering results in indoor and outdoor scenes. In Sec. 5, we show rendering effects of virtual vs. 3D printed models. In Sec. 6, details of false light source elimination are described. In Sec. 7, we describe the dataset and architecture of the proposed CNN model. Finally, we provide the comparison of the inverse CRF estimation in Sec. 8.

2 Visual Assessment of Illumination Estimation Results

Geometric structures, environment contexts and illumination conditions of realistic scenes are generally diverse and complex. In Figure 1 and 2, we show some indoor and outdoor scenes for testing. The proposed method can produce accurate lighting estimations with precisely calculated intensities, positions and boundaries of light sources in LDR images

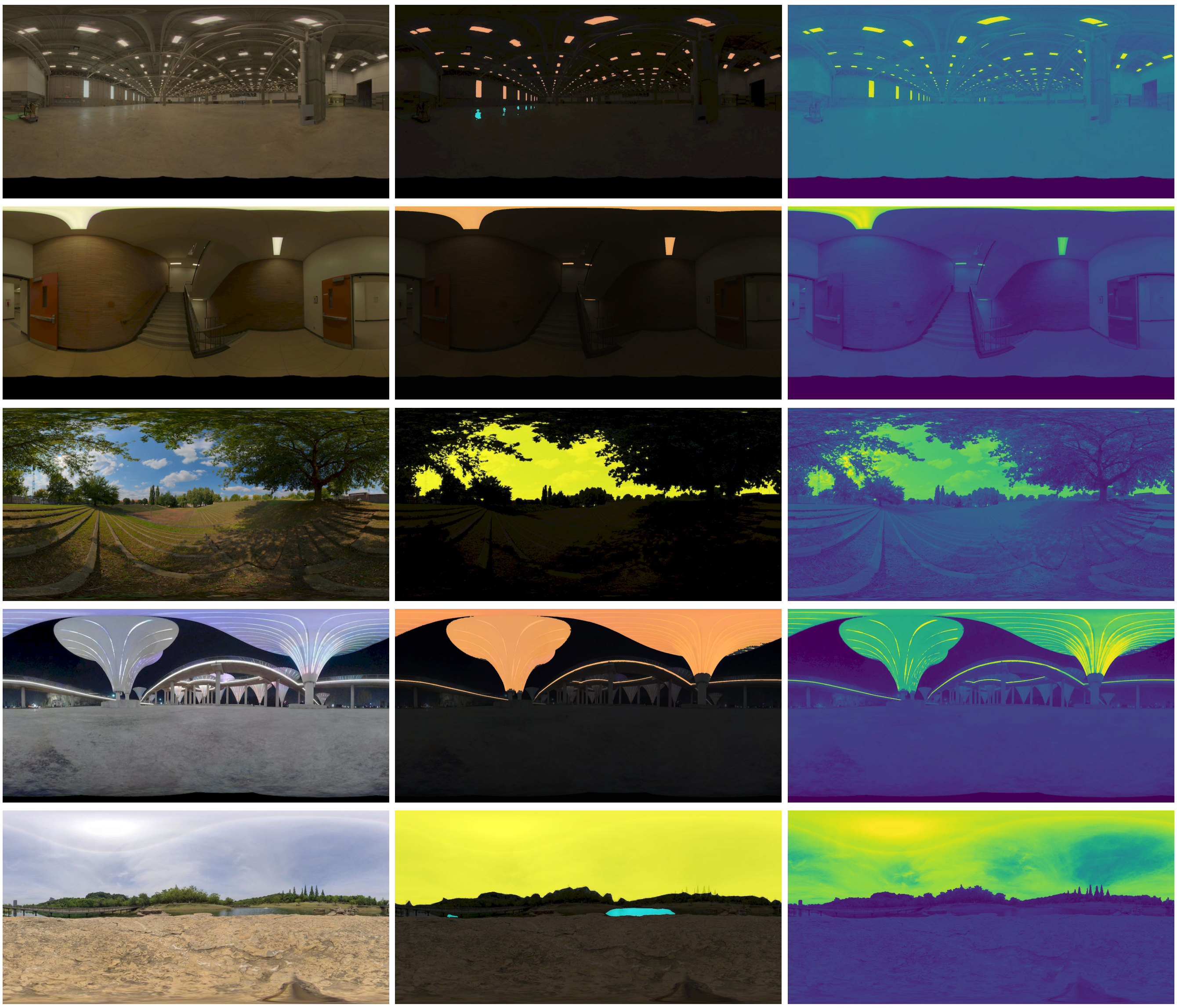
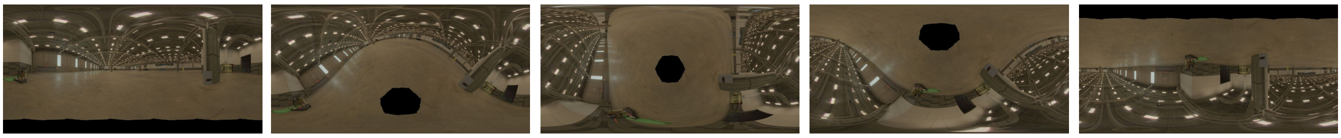


Fig. 1: Visual assessment of illumination calculation results in different real-world scenes using the proposed method. Images in the left column are input LDR images. Images in the middle column are results of light source detection with types information. Diffuse illuminants, directional illuminants and reflective objects are marked in orange, yellow and blue respectively. Images in the right column represent the intensity map of the predicted illumination.

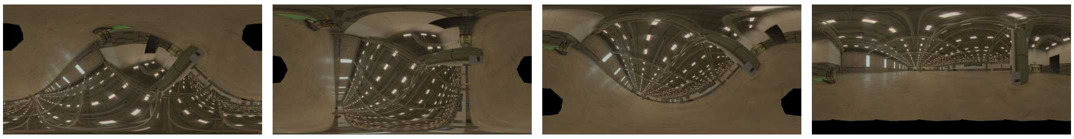
Fig. 2: Visual assessment of illumination calculation results in different real-world scenes using the proposed method. Images in the left column are input LDR images. Images in the middle column are results of light source detection with types information. Diffuse illuminants, directional illuminants and reflective objects are marked in orange, yellow and blue respectively. Images in the right column represent the intensity map of the predicted illumination.

3 Analysis of the Computational Efficiency

Fast lighting estimation not only can significantly help improve the user interaction, but also makes it possible to implement a dynamic illumination effect with light sources that move and change over time. According to the principle of the equirectangular image generation, different view directions of camera lenses can lead to different geometric structures of the panorama, even in the same scene. In our experiment, the pitch angle φ of each panorama were constantly changed to produce different geometric structure complexities in the warped panorama.



(a) $\varphi = 0°$ (b) $\varphi = 45°$ (c) $\varphi = 90°$ (d) $\varphi = 135°$ (e) $\varphi = 180°$



(f) $\varphi = 225°$ (g) $\varphi = 270°$ (h) $\varphi = 315°$ (i) $\varphi = 360°$

Fig. 3: A example of panorama warping. With the pitch angle changing, the number and geometric structure of light sources also gradually change.

Eight scenes in Figure 1 and 2 were chose for testing computational efficiency. With our GPU-accelerated implementation, the proposed method can achieve an average 39 FPS on a single NVIDIA GeForce GTX1060 GPU.

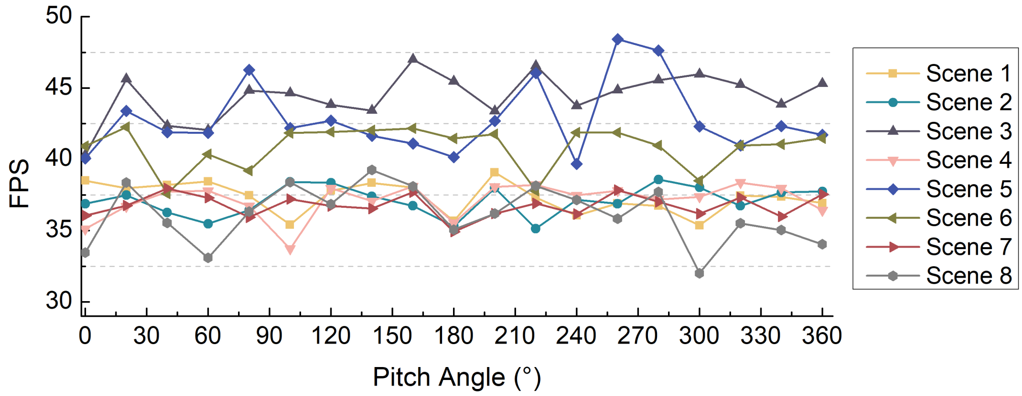
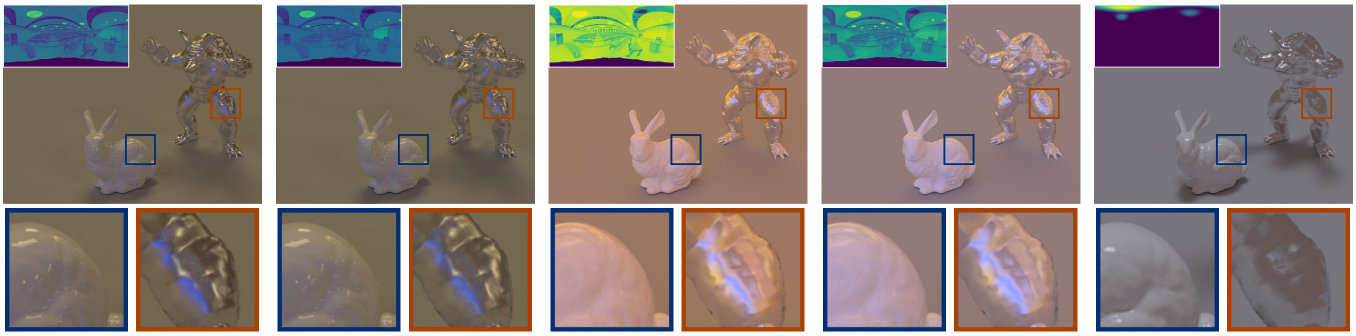


Fig. 4: The computational efficiency of lighting estimation in different scenes using our method. The measured elapsed time included operations of light source direction calculation, light source classification and illumination intensity calculation.

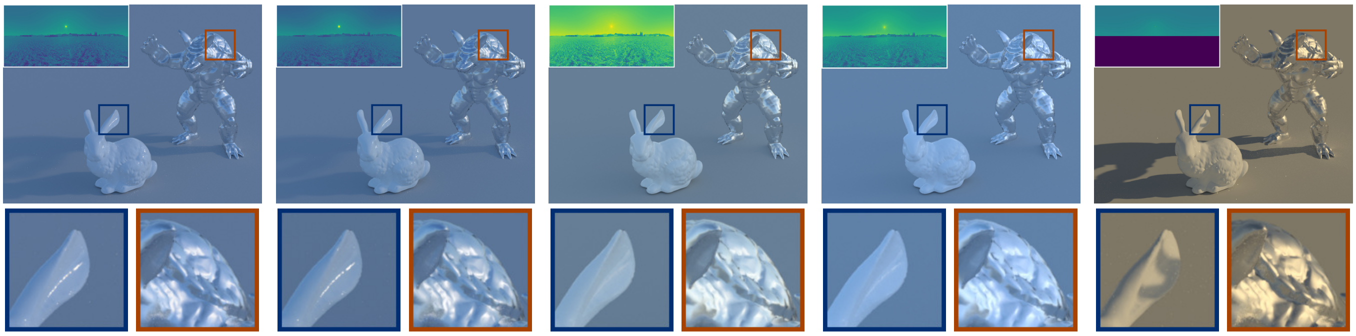
4 Visual Comparison of Realistic Rendering

In order to intuitively inspect the illumination estimation in realistic rendering, four real-world images (Figure 5) were selected as the environment map to render virtual objects. Rendering results of Eilertsen et al. [1] and Liu et al. [2] cannot produce realistic shadows on the ground. With information of the limited FOV, methods of Gardner et al. [3] and Hold-Geoffroy et al. [4] are difficult to satisfy the realistic rendering result with a high precision of the lighting estimation. In our rendering results, shadows on the ground and lights reflected on the model surface look much more identical to the ground truth.

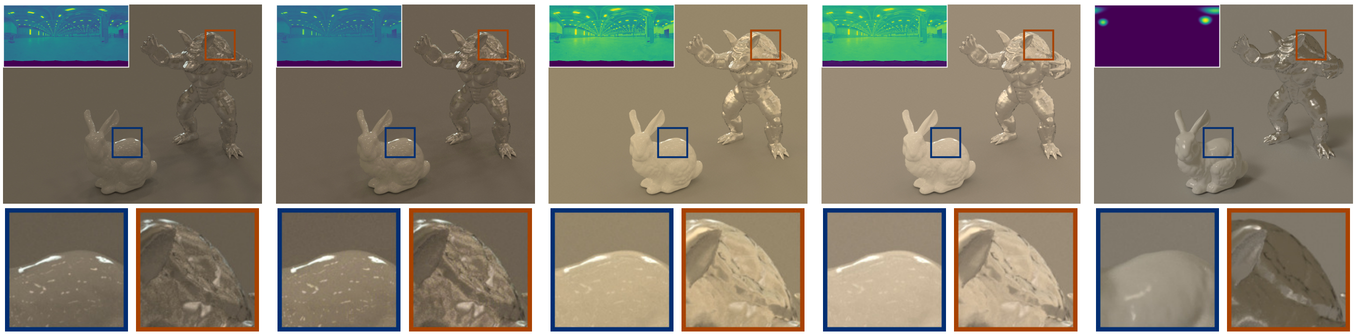
Fig. 5: Real-world images were used as environment map to render objects using different methods in Figure 6.



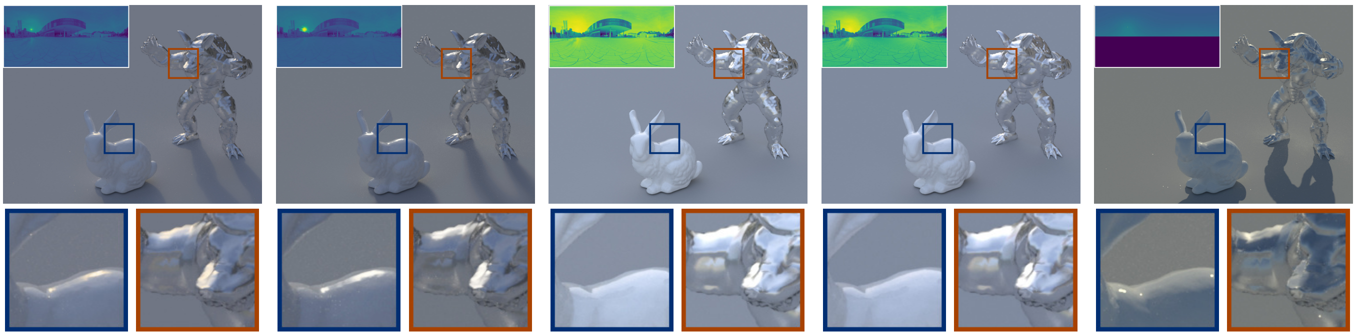
(a) Ground truth lighting (b) Proposed method (c) Eilertsen et al. [1] (d) Liu et al. [2] (e) Gardner et al. [3]



(f) Ground truth lighting (g) Proposed method (h) Eilertsen et al. [1] (i) Liu et al. [2] (j) Hold-Geoffroy et al. [4]



(a) Ground truth lighting (b) Proposed method (c) Eilertsen et al. [1] (d) Liu et al. [2] (e) Gardner et al. [3]



(f) Ground truth lighting (g) Proposed method (h) Eilertsen et al. [1] (i) Liu et al. [2] (j) Hold-Geoffroy et al. [4]

Fig. 6: Visual comparison of realistic rendering results between our method and existing state-of-the-art methods in indoor and outdoor scenes.

5 Rendering Effects of Virtual vs. Real Objects

To better evaluate the rendering quality between existing methods and our approach, 3D printed models and virtual models were photographed in the same place. Then, we conducted a subjective user study. In our comparison, we selected some real-world scenes and five algorithms to compare. Results in Figure 7, 8 and 9 show that our approach outperforms other existing methods in terms of shadings and shadows.

(a) Input panorama (b) Real object (c) Proposed method

(d) Eilertsen et al. [1] (e) Liu et al. [2] (f) Gardner et al. [3]

(a) Input panorama (b) Real object (c) Proposed method

(d) Eilertsen et al. [1] (e) Liu et al. [2] (f) Hold-Geoffroy et al. [4]

Fig. 7: User study images. Comparing virtually rendered objects in relighting results and real objects photographed in images of real-world scenes.

(a) Input panorama (b) Real object (c) Proposed method

(d) Eilertsen et al. [1] (e) Liu et al. [2] (f) Gardner et al. [3]

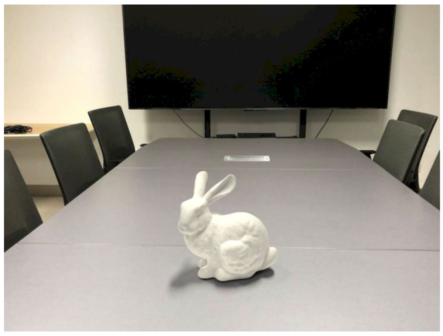
(a) Input panorama (b) Real object (c) Proposed method

(d) Eilertsen et al. [1] (e) Liu et al. [2] (f) Hold-Geoffroy et al. [4]

Fig. 8: User study images. Comparing virtually rendered objects in relighting results and real objects photographed in images of real-world scenes.


(a) Input panorama


(b) Real object


(c) Proposed method


(d) Eilertsen et al. [1]


(e) Liu et al. [2]


(f) Gardner et al. [3]


(a) Input panorama


(b) Real object


(c) Proposed method


(d) Eilertsen et al. [1]


(e) Liu et al. [2]


(f) Gardner et al. [3]

Fig. 9: User study images. Comparing virtually rendered objects in relighting results and real objects photographed in images of real-world scenes.

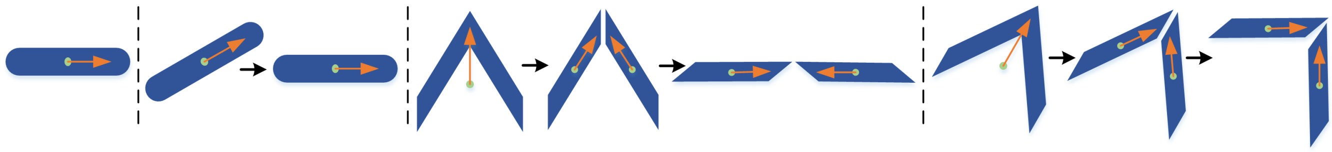
In the additional experiment, virtual objects were placed near real objects to compare. Could you guess which one is the real object in each image? The answer is given in the caption of Figure 10.

Fig. 10: Visual comparison of rendering objects and real objects in real-world scenes. (First row: the right object is the real one. Second and third rows: the left object is the real one.)

6 Details of Elimination of False Light Source

The Section 3.2.2 in the main manuscript aims to eliminate false light sources rather than detect precise positions of light sources. The process of pooling pixels of the detection to the 2×2 resolution (i.e. Figure 4(a) in the main manuscript) has two advantages. First, this step can reduce the camera quantization error [5]. Second, the image after pooling has small resolution, it also helps improve the computational efficiency of subsequent steps of our algorithm.

But for large light sources with irregular boundaries, the pooling process inevitably loses geometric structure information of the object $\mathbf{O}$ and may decrease the accuracy of false light source detection. To solve this problem, we develop the approach that splits the irregular region into some relatively regular regions based on the image moment [6]. The first-order moment of an image can be used to calculate the centroid p. The second-order moment of an image can be used to calculate major axis orientation θ. In our implement, four cases and the solution are shown in Figure 11. In Figure 11(a), $\mathbf{O}$ is a regular object and need not be adjusted. In Figure 11(b), $p \in \mathbf{O}$ and $\theta \notin \Omega$ (Ω represents angles of horizontal and vertical direction). $\mathbf{O}$ need to be simply rotated. In Figure 11(c) and 11(d), if $p \notin \mathbf{O}$, $\mathbf{O}$ need to be iteratively split into sub-regions until $p \in \mathbf{O}$ and $\theta \in \Omega$. More details of elimination of false light source are described in Algorithm 1.



(a) $p \in \mathbf{O}, \theta \in \Omega$ (b) $p \in \mathbf{O}, \theta \notin \Omega$ (c) $p \notin \mathbf{O}, \theta \in \Omega$ (d) $p \notin \mathbf{O}, \theta \notin \Omega$

Fig. 11: Splitting $\mathbf{O}$ into some relatively regular sub-regions. The green dot indicates the centroid of the blue region. The orange arrow indicates the major axis orientation of the blue region.

Algorithm 1: Elimination of False Light Source

Input : The region I of the object $\mathbf{O}$ cropped from a panoramic image
Output: $\mathbf{O}$ is a real light source or not

//Splitting irregular $\mathbf{O}$ into some regular sub-regions
Calculate the centroid p and major axis orientation θ of I using image moments;
if $p \in \mathbf{O}$ *and* $\theta \in \{0°, 90°, 180°, 270°\}$ **then**
    $\mathbf{O}$ is a regular object. The number $M = 1$;
else
    if $p \in \mathbf{O}$ *and* $\theta \notin \{0°, 90°, 180°, 270°\}$ **then**
        Rotate $\mathbf{O}$ to horizontal or vertical according to θ;
        $M = 1$;
    else
        repeat
            Split I into the sub-region I_M;
            Calculate a new centroid p_M and major axis orientation θ_M of I_M;
            $M = M + 1$;
        until $p_M \in \mathbf{O}$ *and* $\theta_M \in \{0°, 90°, 180°, 270°\}$;
    end
end
//Determination of luminous characteristics
for $m \leftarrow 1$ **to** M **do**
    m-th region is spatially pooled to an image with the 2×2 resolution;
    for $k \leftarrow 1$ **to** 4 **do**
        for *attenuation offset of k-th pixel* $\lambda_k \in (-1, 0]$ **do**
            Construct and update the global attenuation matrix of the m-th region $\mathbf{C}_m$;
        end
    end
    Reshape $\mathbf{C}_m \in \mathbb{R}^{2\times2\times N}$ to $\mathbf{D}_m \in \mathbb{R}^{4\times N}$;
    Calculate the projected matrix $\mathbf{S}_m = P \cdot \mathbf{D}_m$, where $P \in \mathbb{R}^{2\times4}$ (e.g. $[2, -1, -1, 0; 0, 1, -1, 0]$);
    Analysis distributions of light attenuation direction $\mathcal{T}$ according to the SVD solution;
    Compute the standard deviation of $\mathcal{T}$, i.e. $\mathcal{M}(\mathcal{T})$;
    if $\mathcal{M}(\mathcal{T}) \neq 0$ **then**
        I_m is not a real light source region. The number $L = L + 1$;
    end
end
if $L > 0$ **then**
    $\mathbf{O}$ is the false light source detection;
else
    $\mathbf{O}$ is the real light source;
end

7 Details of the Light Source Classification

Light sources of different types have different luminous characteristics. In our work, a CNN-based semantic segmentation model is designed to classify all light sources identified in the LDR image.

7.1 Dataset

Based on different object properties and the human experience, labels in the ADE20K dataset [7] were categorized into four classes (Figure 12), including directional illuminants (or infinite illuminants), diffuse illuminants (or non-directional illuminants), reflective objects and non-luminous objects. Normally, the total image area occupied by light sources is less than the image area covered by non-luminous objects for most scenes.

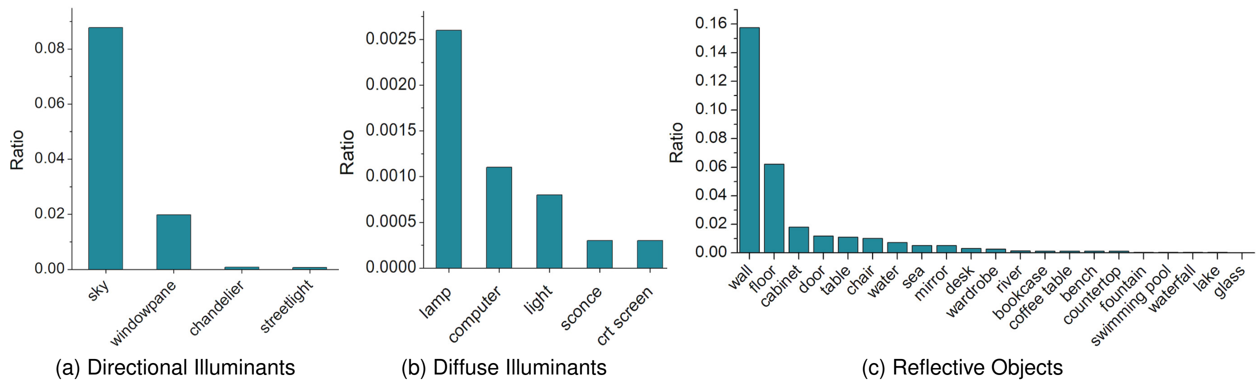


Fig. 12: The luminous labels of different objects in our dataset adjusted according to the ADE20K dataset. The stuff or object that is not shown in the bar chart is categorized into a non-luminous object.

7.2 The Proposed CNN Architecture

Our CNN model architecture is illustrated in Figure 13. The GhostNet [8] model is selected as the backbone of our CNN model structure to perform fast image feature extraction. In addition, we add the *channel attention module* and *global feature extraction module* to improve the performance of our network.

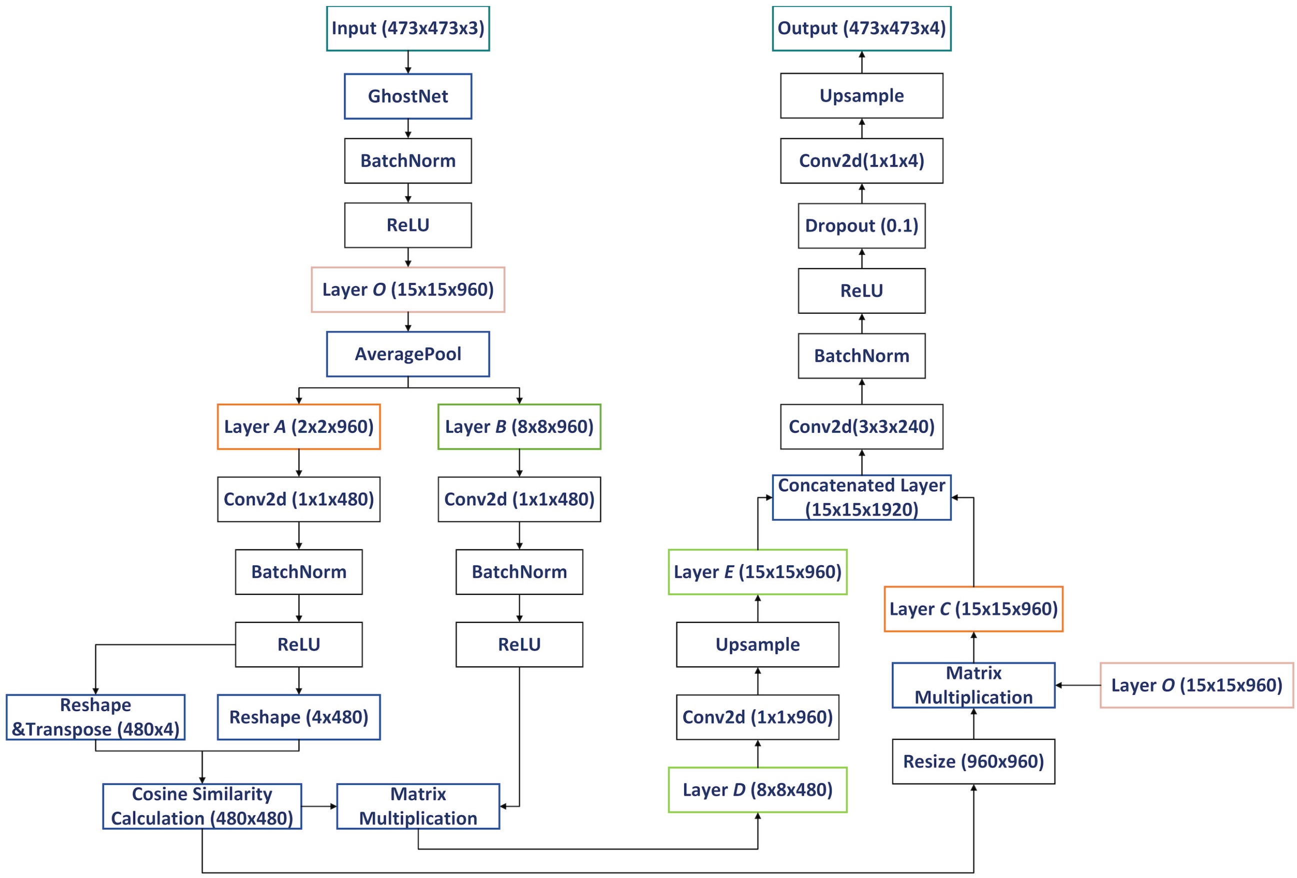


Fig. 13: The architecture of the proposed CNN model, including modules, setups for input and output dimensions, kernel sizes and channel numbers.

8 Comparison of Inverse CRF Estimation

To evaluate our inverse CRF estimation performance, we collect some real-world images captured using the Insta360 ONE X camera. Ground-truth CRFs were calculated by the algorithm of Debevec et al. [9]. In Sec. 5.1 of the main manuscript, we set $B \times B$ mesh grids to estimate final inverse CRF. B is a crucial factor to affect the accuracy of the inverse CRF estimation. In our experiment, B was set within $[5, 35]$. Figure 14 indicates that our approach can keep low RMSE than Sharma et al. [10] even if B is set small. In practice, we set $B = 15$ to trade off between a high FPS and a low RMSE. Then the proposed method was compared with other methods, i.e. Eilertsen et al. [1], Liu et al. [2] and Sharma et al. [10]. The comparison result shows that non-learning-based methods provide better generalization capability to predict CRFs beyond the training data (Figure 15).

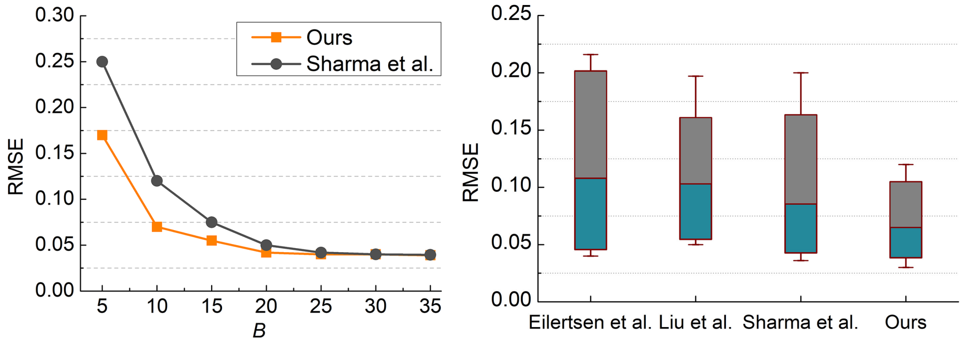


Fig. 14: Quantitative comparison on inverse CRF estimation. The RMSE was computed between predicted inverse CRFs by different methods and ground-truth inverse CRFs.

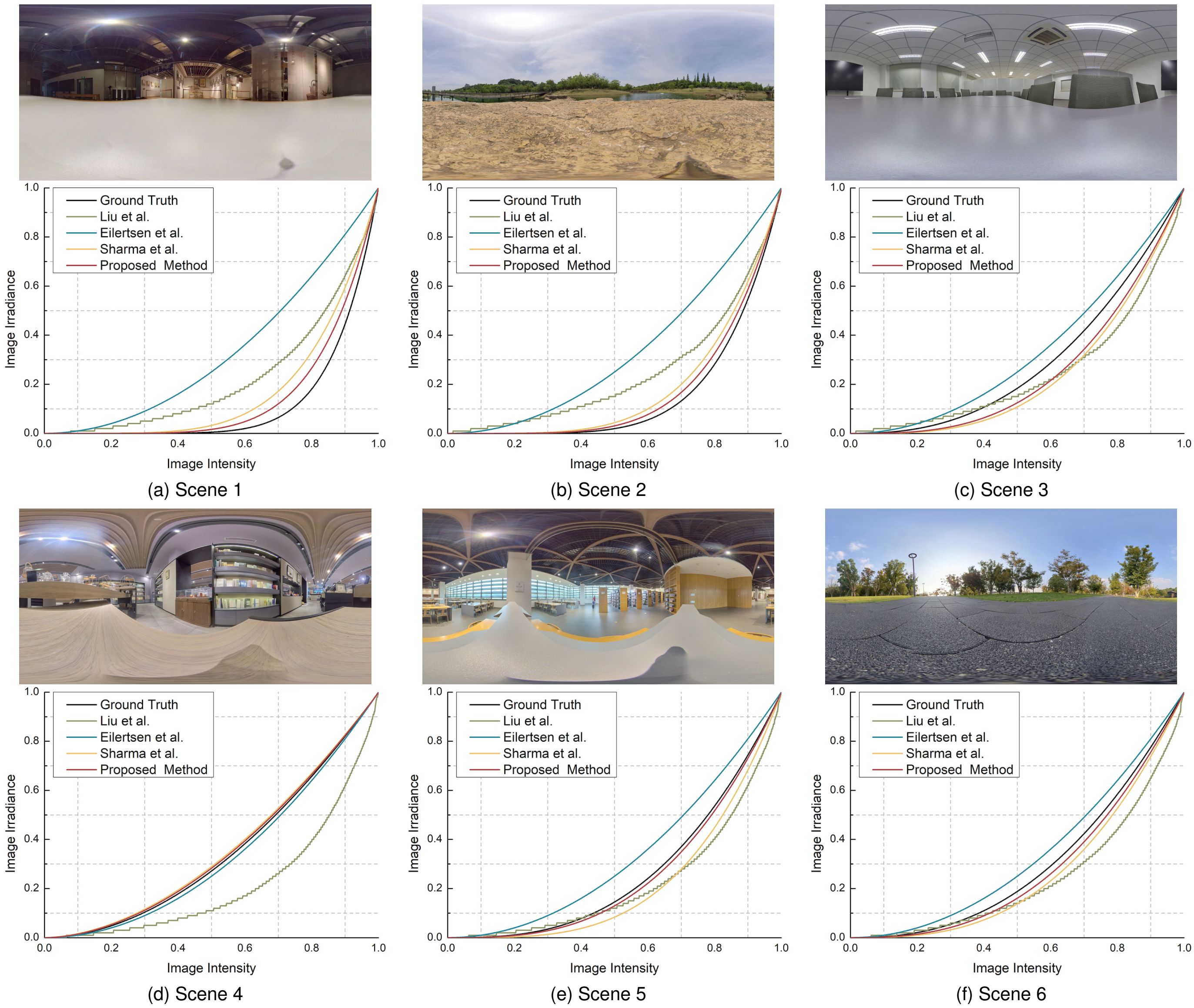


Fig. 15: The quantitative evaluation and comparison of our approach and other existing methods

References

[1] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR Image Reconstruction from a Single Exposure Using Deep CNNs," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–15, Nov. 2017.

[2] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 1651–1660.

[3] M.-A. Gardner, Y. Hold-Geoffroy, K. Sunkavalli, C. Gagne, and J.-F. Lalonde, "Deep Parametric Indoor Lighting Estimation," in *Proceedings of International Conference on Computer Vision*, Oct. 2019, pp. 7175–7183.

[4] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J. Lalonde, "Deep outdoor illumination estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 2373–2382.

[5] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan, "Algorithmic Principles of Remote PPG," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1479–1491, Sep. 2017.

[6] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis: theory and applications*. World Scientific, 1998.

[7] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene Parsing Through ADE20K Dataset," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 633–641.

[8] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features from Cheap Operations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 1577–1586.

[9] P. E. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," in *Proceedings of ACM SIGGRAPH*, Aug. 1997, pp. 1–10.

[10] A. Sharma, R. T. Tan, and L.-F. Cheong, "Single-Image Camera Response Function Using Prediction Consistency and Gradual Refinement," in *Proceedings of Asian Conference on Computer Vision*, Oct. 2020, pp. 19–35.