DSNet: Deep Shadow Network for Illumination Estimation

Yuan Xiong[1], Hongrui Chen[1], Jingru Wang[1], Zhe Zhu[2], and Zhong Zhou*[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2]RaiLab, Duke University, Durham, North Carolina, United States

ABSTRACT

Illumination consistency has applications to modeling and rendering in virtual reality. In 3D reconstruction and Mixed Reality(MR) fusion, the appearance of a large-scale outdoor scene may change in response to lighting and seasons, for example. Since 3D reconstruction from scratch is costly, it is helpful to be able to update existing models with recently captured photographs. However, the illumination conditions of the captured photograph can be arbitrary, making it challenging to fit to the existing model. To tackle this problem, this paper proposes a novel approach that can precisely estimate the illumination of the input image. Our Deep Shadow Network (DSNet) collaboratively utilizes illumination-based data augmentation for sun position estimation, along with a dataset of illumination-based augmented renderings. Our run-time rendering and optimization strategy is also discussed. We show that accurate simulation of illumination can improve the performance of visual applications including place recognition and long-term localization. Experimental results validate the effectiveness of the proposed approach, and show its superiority over the state-of-the-art.

Index Terms: Computing methodologies—Modeling and simulation—Model development and analysis—Modeling methodologies; Artificial intelligence—Computer vision—Computer vision problems—Reconstruction

1 INTRODUCTION

Illumination consistency of large-scale outdoor scenes has been widely studied in modeling and rendering, with diverse applications such as virtual Olympics, traffic simulations and immersive telepresence. In all-element virtual scenarios, video streams with different illumination conditions are blended to produce a MR world. For visual mobile navigation, localizing outdoor photographs for augmented reality (AR) display is a basic need. In such applications, illumination in-consistency remains a long-standing problem. To accurately estimate and simulate illumination in real-time, a high quality 3D model is vitally important. In large-scale 3D reconstruction, urban appearance changes frequently, and its digital representation requires frequently updating. However, traditional structure from motion (SfM) methods bundle all inputs and perform global modeling. The same illumination conditions can rarely be achieved, due to changes of season, weather, time of day, vegetation and terrain appearance. Inconsistency of illumination causes defects in model fusion. Furthermore, illumination simulation plays an important role in MR rendering, in which the visual experience should be consistent with the user's environment. Thus, a novel framework is needed which can accurately estimate the primary illumination from user inputs. Existing illumination estimation methods discover solar parameters from multiple cues. Traditional methods directly calculate solar position in Earth coordinates but require an accurate geolocation and timestamp as additional inputs. Machine learning methods predict relative solar positions in query pictures, and are robust under challenging illumination conditions. Mainstream reconstruction methods [3, 19] create large-scale 3D models but dilute illumination cues after global color fusion during texture blending. To dynamically combine geometric information from the 3D model with learned features from deep neural networks, a renderer based on a GPU pipeline is required. In contrast to traditional shadow detection and structure of light approaches, this paper considers a more general scenario, in which unordered photographs are sampled randomly without viewport limitations. We present a deep learning based illumination estimation network; an overview of our pipeline is depicted in Fig. 1. We also consider enhancement of modeling applications using illumination-based data augmentation.

In summary, this paper makes the following contributions:

- DSNet, providing state-of-the-art illumination estimation with outstanding robustness and high precision under challenging conditions,
- illumination augmented datasets with free-viewport and random solar simulation, including 240K+ images with pixel-wise depth, shadow label and camera pose,
- experiments which demonstrate the usefulness of illumination-based data augmentation and an analysis of its enhancements to camera localization methods.

We believe we are the first to address the complete scope of illumination estimation using large-scale augmented 3D datasets. Extensive evaluations quantify resultant improvements to image retrieval and long-term localization.

2 RELATED WORK

2.1 Shadow detection

Detecting shadows in natural images has been widely investigated, using methods for extracting the scene geometry and light direction. Early works [7, 21] achieved high accuracy for cast shadows but failed in the presence of multiple light sources and soft shadows. Deep learning based shadow detection methods [9, 26, 28] are less sensitive to rich textures, and can extract indirect shadows. BEDSR-Net [13] is able to detect and remove shadows using a deep learning based network, but is limited to document images. Liu's method [14] places virtual 3D objects in the virtual scene for shadow comparison, but only works for a limited range of objects such as cars. Existing works still struggle with cases where the boundaries of the cast shadows are hard to find. Using them for sunlight direction estimation is far from their original purpose, where the gradient of shading information and the integral understanding of the illumination environment is more favorable. Direct shadow casting approaches perform poorly in special cases such as shadows with large area, shadows at sunset and shadows covered by specular reflections.

*e-mail: zz@buaa.edu.cn

2642-5254/21/$31.00 ©2021 IEEE
DOI 10.1109/VR50410.2021.00039

2021 IEEE Virtual Reality and 3D User Interfaces (VR) | 978-1-6654-1838-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/VR50410.2021.00039

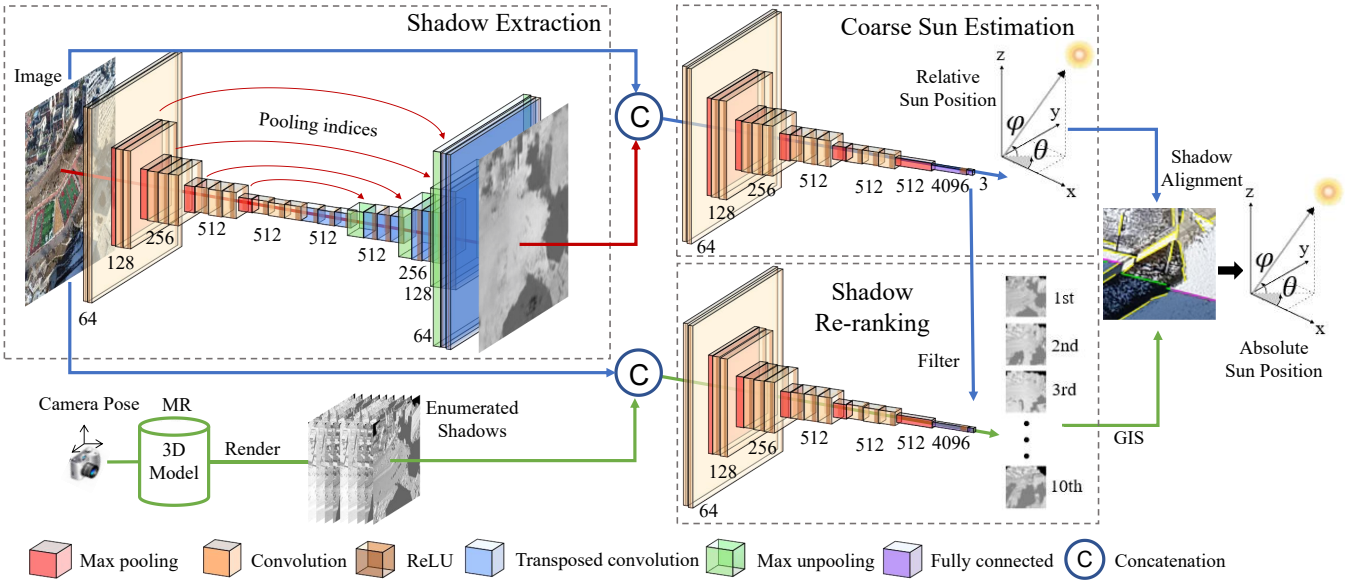


Figure 1: Coarse-to-fine framework of DSNet, our deep learning based illumination neural network for solar position estimation, trained on shadow datasets. Unlike other approaches which only predict imprecise results, our method has significantly increased precision of around 1°. Each major component can be trained and examined separately.

2.2 Indirect illumination estimation

Unlike direct cast shadow estimation approaches, indirect methods try to recover the illumination from implicit visual cues. Some methods reconstruct the shape of a 3D object under unknown illumination [22, 23, 25]. Such shading-based surface reconstruction can hardly be applied to large-scale outdoor scenes with complex details and irregular surfaces.

2.3 Deep learning based illumination estimation

Deep learning based approaches achieve better performance on datasets with ambiguous texture patterns. Sun-CNN [16] learns relative solar orientation from shadow cues using a deep AlexNet [12] framework trained on an embedded dataset. SunOriNet [10] slightly outperforms Sun-CNN by adding a branch layout focusing on patterns of intermediate size. Yannick [8] trains his network on a panorama dataset and removed pooling steps in his network for fast convergence. Illumination based approaches play helpful roles in assisting visual tasks like long-term localization. However, indirect deep learning based methods suffer from the problem of insufficient data. Both KITTI Sun [6] and Sun360 [24] provide only street view photographs with limited sun positions.

2.4 Machine learning localization

Machine learning based long-term localization approaches recover the pose of a query image based on machine learning models trained on datasets. Some end-to-end methods can directly predict camera translation and rotation for 2D to 3D registration. PoseNet [11] can acquire camera pose directly from an image input. SuperGlue [18] with SuperPoints [4] and HF-NET [17] achieve better robustness using deep learning based descriptors. Before localization, usually images with overlapping regions are retrieved for feature mapping, as in the NetVLAD [1] network for place recognition. This paper focuses on improved performance over these methods by using augmented datasets with illumination simulation.

3 Deep Illumination Estimation

In our scenario, given an existing 3D model, and few pictures of a new scene, we aim to determine illumination parameters for model update and blended rendering tasks. Unlike other reconstruction approaches with global color fusion for all photographs, we choose a primary photograph with robust features, in which we can estimate and simulate the illumination, to better process the other photographs.

3.1 Solar Position Estimation

When GPS, date and time are all accessible, the direction of the sun can readily be calculated. Without such prior knowledge, solar orientation can only be estimated from visual cues. Since earth orbits the sun at an average of about 1.5×10^8 km, the sun can be treated as a parallel light source. We combine both indirect analysis in structured light methods and direct shadow detection methods in our deep learning framework: see Fig. 1. Three major sub-networks collaborate to provide coarse to fine estimates of solar position. A rendering-based optimizer is responsible for final adjustment of sun parameters. In this Section, the approach is described in detail.

3.1.1 Sun-VGG sub-network for coarse estimation

Unlike Sun-CNN [16] which uses only color information from the image, we include another channel representing the shadow with shading in our network. We use the deeper VGG [20] network instead of AlexNet [12] because it can handle large scale images with high-level semantic features; in our case, these are the structured light information. It makes an important improvement over AlexNet by replacing large kernel-sized filters with smaller kernel-sized filters, partially solving the problem of overfitting on certain patterns. Traditional VGG networks have been shown to be efficient for classification and segmentation. However, in our scenario, we use them for parameter regression. Instead of using pre-trained models for classification, we train different models on our own datasets, which are generated by a virtual sampler randomly taking snapshots in a virtual environment. Unlike Sun-CNN which uses two angles to represent the solar position, we use a normalized vector in 3D Euclidean space. This is mainly because relative solar azimuth and altitude as used in Sun-CNN do not converge around 0°, 180°and 360°. For free viewport solar estimation, this is a common issue because no street-view style input is guaranteed and the sun can be located anywhere relative to the camera. We remove the softmax

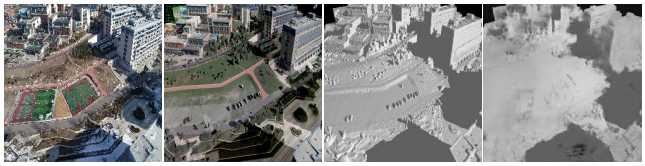
layer at the end for regression purposes. The input to the first layer is a $224 \times 224 \times 4$ RGBS image, where the S channel contains the shadow rendering. The purpose of adding this channel is to leverage existing shadow renderings reflecting structured 3D information of the scene without biased fitting to texture patterns. During training and evaluation steps, we include virtual shadows rendered by shaders. However, while testing, we can only extract shadows from the image. As a result, we are motivated to design a pre-trained shadow extraction network, as described in the next section. Each layer of data in a convolution group is a four-dimensional tensor of size $b \times w \times h \times d$, where b is the batch size, w and h are spatial resolutions, and d is the number of convolutional filters or channels listed in the figure. All convolution groups use 3×3 kernel filters with 1-pixel padding and stride, followed by 2×2 max pooling with stride 2. We use the standard ReLU activation function. The output vector represents the normalized direction of the sun, relative to the viewport. To measure squared Euclidean distance in the tensor-based machine learning framework, we employ the equivalent MSELoss function. The three channels of the output vector give the solar position, with normalized coordinates (x, y, z) defined by:

$$\begin{cases} x = \sin(\theta)\cos(\phi), \\ y = \cos(\theta)\cos(\phi), \quad -\frac{\pi}{2} \le \phi \le \frac{\pi}{2}, \quad -\pi \le \theta < \pi \\ z = \sin(\phi), \end{cases} \tag{1}$$

where ϕ is the relative elevation angle and θ is the relative azimuth angle, as shown at the upper right corner of Fig. 1. Unlike in the KITTI dataset [6] in which the sun is always overhead, the scenario in this paper is more general, and the relative solar angles ϕ, θ can be negative, depending on the yaw, pitch and roll of the observer. Since no open source dataset provides such data, we created our own with the help of the MR snapshot sampler, a virtual camera randomly taking snapshots in a virtual 3D world.

3.1.2 Shadow-VGG sub-network for shadow extraction

The basic flow of our Shadow-VGG sub-network is illustrated in Fig. 1(left). Two major parts, the encoder and the decoder, are illustrated. The encoder has 13 convolutional layers in five groups; its role is to extract features from the input image. The decoder has a symmetric architecture and is responsible for assembling feature fragments into the output. We use an indexed unpooling and deconvolution strategy similar to those in SegNet [2] and FCN [15]. Most of the network convolutional groups are similar to those in the VGG network, with the same dimensions, layers, kernel size and activation function. The fully connected layers after the encoder to allow connection to the decoder. The output image has the same resolution as the input, and is a prediction of a grayscale shadow rendering with proper shading. To minimize pixel-wise differences, we use the 2D MSELoss function. Our novel contribution lies in the training step, where virtual snapshots with random shadows are dynamically generated from different scenes. The shadow images are rendered by shaders which project random sunlight onto coarse grayscale models. We aim to reduce the negative impact of textures so that the real structured light information can be extracted. The 2D MSELoss function is applied to compute the pixel-wise difference between the labeled shadow and the output prediction, to maximize their difference. As shown in Fig. 2, the predicted shadow retains primary shading information and removes ambiguous patterns that are harmful to solar direction estimation, including but not limited to textures on windows, cars, stairs, trees and playgrounds. Compared with a visualization of CNN layers used in Sun-CNN [16], our output shadow renderings contain more detailed information with higher resolution. Our experiments show that the additional shadow channel created this way and added to the input of the Sun-VGG network significantly improved its performance, more so than any other means of adjusting network parameters and layouts. See the experimental results in Fig. 8.



(a) Photo with appearance change (b) Ground-truth MR rendering (c) Ground-truth shadow (d) Predicted shadow of the photo

Figure 2: Shadow extraction result. Despite the appearance change, we predict a grayscale shadow appearance (d) from the given photograph (a). The ground-truth rendering and shadow (b,c) are generated by the shadow map rendering program using geolocation and timestamp. The depth information included by the shader is used for masking out pixels in unmodeled areas(black).

3.1.3 Shadow-Rank re-ranking sub-network

To further optimize the solar position estimate by verification of the rendering result, the Shadow-Rank sub-network compares the current image with multiple rendered shadows. The structure of the Shadow-Rank sub-network is given in Fig. 1(lower right). It has the same convolutional layers as other VGG networks. Inspired by NetVLAD [1], we designed the Shadow-Rank network for ranking and recommendation. Differing from the scenario in NetVLAD, our shadow images are selected and rendered in real-time according to the camera pose. As a result, feature vectorization, indexing and retrieval steps are skipped. For training, we include negative samples so that the model can distinguish between likely and unlikely pairs. For each query image with free-viewport camera pose, we set the sampling interval to 5° and render $12 \times 72 = 864$ shadows, where 12 is the number of elevation angles in the range [15°, 70°], and 72 is the number of azimuth angles in the range [-180°,180°). These daytime ranges work for most places on Earth regardless of date. For places close to the ecliptic plane, the solar elevation angle may exceed 70° at noon, and for some northern cities, it may below 15° in the morning or evening. In these cases, the proposed method is inapplicable. For each training epoch, we use an undersampling strategy by assembling one positive pair and k randomly chosen negative pairs respectively. To make the model trainable and the result reasonable, we assign cosine loss between the predicted solar position and the ground-truth to the labels. The loss function is:

$$\mathscr{L}^* = \beta(\mathscr{L}(p, q^+))^2 + \sum_{i=1}^{k}(\mathscr{L}(q^+, q_i^-) - \mathscr{L}(p, q_i^-))^2 \tag{2}$$

where p is the predicted solar position, q^+ is the positive pair, q_i^- is the ith negative pair, k is the undersampling factor for picking negative samples, usually set to 50 for training, and β is the compensation factor for positive learning, usually set to 10. The cosine loss function $\mathscr{L}$ is defined as:

$$\mathscr{L}(p, q) = (1 - \frac{p \cdot q}{|p||q|}). \tag{3}$$

The cosine loss is used as we want the network to learn the angular difference between different samples, especially when they are organized in 3D coordinates. We do not follow Yannick's [8] method of using Kullback-Leibler (KL) divergence loss, because we treat each negative sample differently according to its relative solar position. In our generalized scenario, the distribution of relative solar positions is static and isotropic, and the KL divergence loss does not meet our requirements. The equation $\mathscr{L}^* = 0$ has a unique solution $p = q^+$. Notice that there is a learnable constant term in Eq.(3) representing the cosine loss between the negative solar position and the positive solar position. Using this loss function and undersampling strategy, our model converges rapidly, so that we can use a smaller undersampling factor k and larger batch size.

We include the direct output of the Sun-VGG sub-network as a filter to re-rank the Shadow-Rank result, using the aggregated score:

$$S_i = \mathscr{L}^*(q_i) + (\mathscr{L}(p, q_i))^2 \tag{4}$$

where S_i is the new score of the ith enumerated solar position q_i. It adds the output loss in Eq. 3 to a filter term, the square of the cosine loss between q_i and the coarse prediction p. We use the total loss to re-rank the enumerated solar positions in ascending order. In Fig. 3 we show an example of such re-ranking; here, a mask determined by the model boundary is included. Prior knowledge of the geolocation and knowledge that it is afternoon are applied to optimize the enumeration interval.

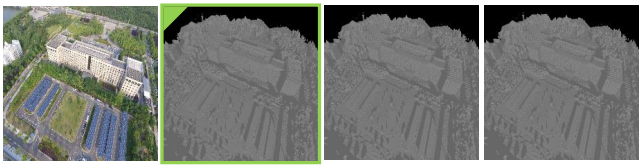


Figure 3: Example of Shadow-Rank re-ranking. The top 3 recommended shadow enumerations are presented and the ground-truth is highlighted in green. Angular errors, absolute and relative sun angles are provided for reference. Besides the top result, other recommendations are also acceptable for using in shadow alignment.

3.1.4 Shadow Alignment

Having obtained the coarse prediction from the Sun-VGG and the re-ranked result from Shadow-Rank, we can further improve the solar orientation determination with the shadow alignment optimizer, which iteratively maximizes the total length of corresponding shadow boundaries between snapshots and shadow masks: see Fig. 4. The shadow mask is a binarized image generated by a shader which compares the pixel-wise distance to the sun with the rasterized depth of the virtual solar projector. The shadow boundary can be extracted by line segment detection (LSD). For each line segment in the input image, we look for a corresponding match in the shadow mask with minimum distance, and similar orientation, in a certain range. We enumerate every possible solar azimuth and elevation angle to find the best estimate, the one with the largest total length of corresponding shadow boundaries. In the first round of enumeration, we try different solar angles at an interval of 5°, starting from recommended results of Shadow-rank and the prediction of Sun-VGG, and decrease this to 0.5° in the second round, starting from the best result previously found. We apply an early stop if a pair of solar angles is better than others within 30°, and start a new round of enumeration if the current guess is close to the boundary of the enumeration interval. As a result, the initial guess determined by previous predictions is vitally important: it not only can speed up enumeration, but also reduce the chance of failure caused by wrong shadow cues. The number of recommendation adopted from Shadow-rank is determined by the performance of the primary photo. Without prior distribution knowledge of the dataset, it can be arbitrarly set to 20.

3.2 Rendering and Optimization

To simulate the real-world illumination in virtual scenarios comprising 3D meshes, textures, geographical information, registered videos and other formatted semantic data, we implemented our own rendering engine using OpenGL. As shown in Fig. 5, we integrated shader programs, textures, and buffers in our rendering engine. For shadow rendering, we use a shadow mapping strategy, storing the

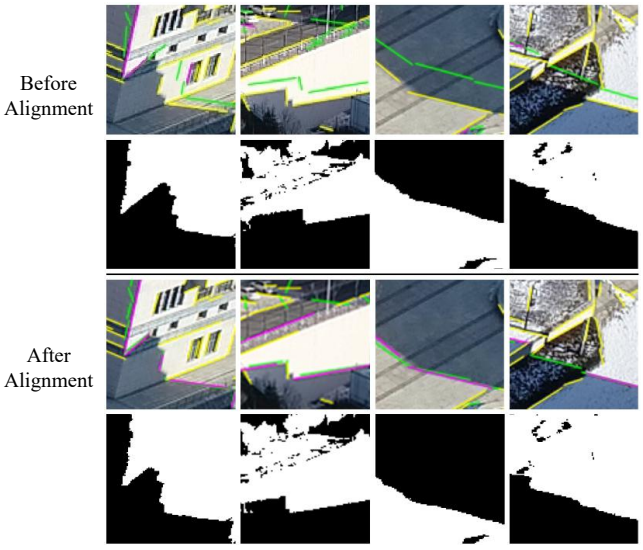


Figure 4: Final solar position estimation with automatic shadow alignment optimization. Above: snapshot and shadow masks before shadow alignment. Below: results after shadow alignment. Yellow: LSD features of images. Green: LSD features of shadow masks. Purple: matched LSD features. The shadow alignment optimizer reuses solar position estimations from previous layers and calls the renderer to dynamically generate shadows. The shadow with strongest shadow edge correspondence is recorded.

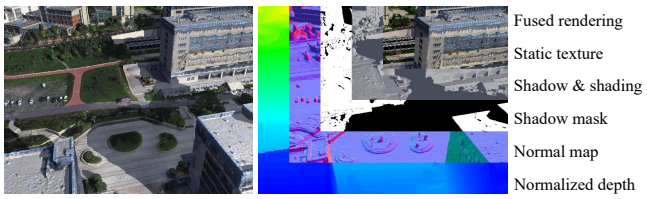


Figure 5: For real-time performance, we implement GLSL shaders for 3D model rendering. Left: a fused rendering of an illumination augmented 3D scene. Right: decomposed shader outputs stored in GPU buffers. We combine different shaders for different rendering tasks. Texture and frame buffers are used for data exchange.

depth buffer of a sunlight projector into an 8K texture for shadow detection. After placement of the sun as a directional light source, we use shaders to optimize other shading parameters, and iteratively render and evaluate snapshots. We use a modified reflective Blinn-Phong model for shading, in which objects are made of intermediate materials between perfectly diffuse and mirror-like surfaces. Ambient color and attenuation are ignored. For each fragment, the output color is computed by:

$$C = c_o(c_{\text{sun}}(n \cdot l + (n \cdot h)^s) + c_{\text{sky}}) \tag{5}$$

$$h = \frac{(l+v)}{|l+v|} \tag{6}$$

where c_o is the object color obtained from the model texture. We only consider colors from two light sources, the sunlight color c_{sun} and the skylight color c_{sky}. The surface normal n, sunlight direction l, viewing direction v and specular exponent s are also used for shading; the negative dot product between them being ignored. We assume that the model scatters incident skylight equally in all directions, and the color of the sky is globally unique. To further optimize the illumination, we minimize the mean squared error of pixels in a

least-square sense:

$$\arg\min_{c_{\text{sky}},c_{\text{sun}},s} \sum_{p\in\Omega} \left(C(p) - C'(p, c_{\text{sky}}, c_{\text{sun}}, s)\right)^2 \quad (7)$$

where C and C' are RGB colors for each pixel p in image coordinates Ω. We use C in the inpute images and C' in the augmented snapshots from the renderer, constrained by parameters including c_{sky}, c_{sun} and the specular exponent s. Illumination parameters can be optimized by standard gradient descent iteratively evaluating the difference between the photograph and the rendered image.

3.3 Augmented Dataset

Our datasets come with 3D meshes, textures and pixel-wise depth captured from the MR. With the help of an SfM pipeline and multiview-stereo methods, we are able to reconstruct models using images from oblique photography, sampled from the camera on a DJI Phantom 4 RTK (84° FOV, mechanical shutter, 1 inch CMOS with resolution 4000×3000 for photography and 1080P for video). Illumination ground-truth is labeled dynamically by the MR renderer in the virtual sub dataset. Real photographs and video frames are manually labeled. We currently provide 10 urban scenes sampled from 4 cities. In each dataset, we provide 24,000 renderings for training, 3,000 for validation and 3,000 for testing. We also provide labeled real photographs and videos for additional testing. Each test case contains pixel-wise depth and shadow which could also be of use in other virtual reality applications. Unlike the KITTI [6] and Sun360 [24] datasets, our dataset is not restricted in terms of camera pose and solar position. A detailed comparison is shown in Table 1.

Table 1: Dataset comparison.

Features	KITTI-Sun [6]	Sun360 [24]	DSNet
Photographs	3314	38814	6417
Simulations	0	0	240K+
Viewport	street view	panorama	free
Scene	highway	urban	urban
Sun positions	limited	limited	unlimited
6-DOF Camera	no	no	yes
Depth	sparse	no	dense
Video	yes	no	yes
3D Mesh	no	no	y es
Pixel-wise shadow	no	no	yes

3.4 Illumination Enhancement

While illumination estimation benefits from accurate camera pose in terms of rendering-based optimization, it can also enhance camera pose estimation through data augmentation.

3.4.1 Place recognition

NetVLAD tackles large scale visual recognition by accurately recognizing a query photograph using a deep neural network. Although we do not apply NetVLAD [1] during shadow ranking, we use it during model updating to perform global retrieval of images similar to the query image is a necessary precursor to localization, so that feature mapping can be deployed, inspired by [17].

We enhance NetVLAD by augmenting the training datasets with illumination-based data. Augmented shadows can not only provide robust local features, but can also guide clustering according to the sunlight direction. We note that the 2D convolution kernels used by NetVLAD are not perspective invariant. However, the simulated illumination provides approximately perspective invariant patterns like edges, shading on eaves and pole shadows. Experimental verification of this assertion is given later.

3.4.2 Long-term localization

In model updating and MR rendering, accurate localization of the input images is extremely important. However, due to significant appearance change and illumination inconsistency, localization algorithms like SuperGlue [18], which is based on SuperPoint [4] deep features, can no longer achieve the desired performance. The reasons are mostly similar to those in place recognition, with respect to features, limitation of convolution receptive field, datasets and evaluation metrics. SuperGlue has different characteristics from NetVLAD. First, SuperGlue uses a synthetic shape dataset for pretraining; it consists of rendered patterns that are integrated into the MagicPoint-base detector. Illumination simulation here plays a similar role, by feeding more augmented features into the network. In extreme cases where the majority of the scene is covered by shadows, like Fig. 6(d), augmented ground patterns can still be reliably matched. SuperGlue establishes pointwise correspondence with a graph neural network. With accurate illumination-based data augmentation, we can strengthen such bundled correspondences by providing more illumination consistent keypoints.

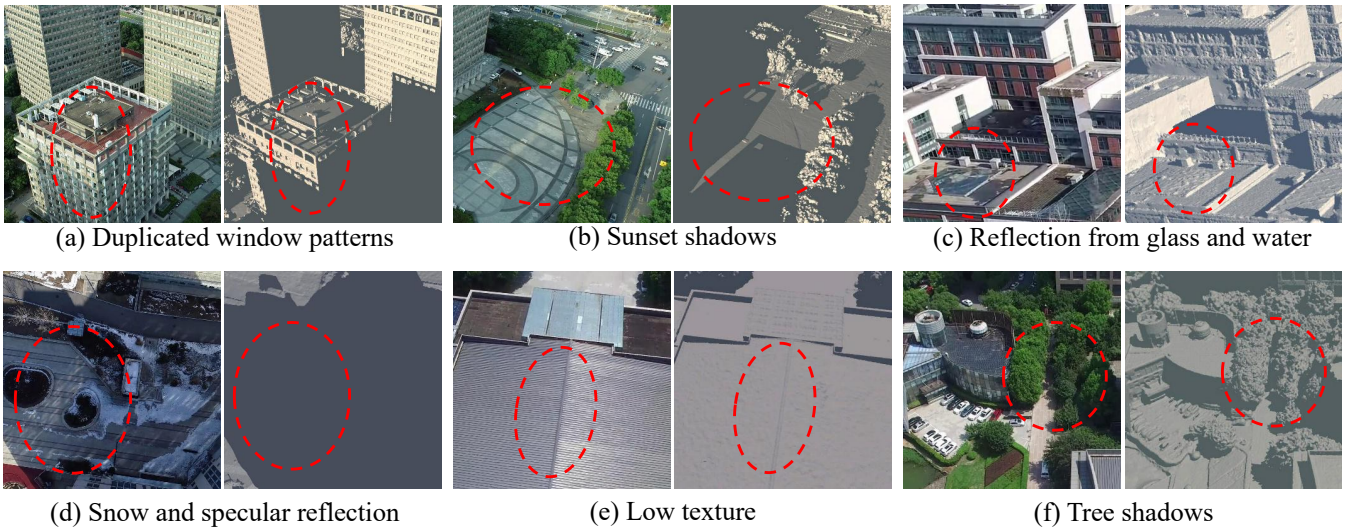
4 EXPERIMENTS

This section expounds on detailed experiments of our framework. We present experimental evaluations on illumination estimation and data augmentation. We aim to prove that our approach can handle large-scale illumination estimation tasks with flexible applicability, and its enhancement of localization is remarkable. Our self-collected datasets with free-viewports and random sun positions are presented. All experiments are deployed on a personal workstation with 6 cores 3.2 GHz processors, 32 GB RAM and a single NVIDIA GeForce RTX 2080Ti GPU. The MR rendering engine is working on an OpenGL pipeline with GLSL shaders. The procedure we use to train the deep neural network is supported by PyTorch1.6 framework with Python 3.6 runtime environment. For sun position evaluation, the cumulative prediction errors of relative solar angles are considered, which is the same metric used in Sun-CNN [16]. For enhancement of the illumination-based data augmentation, statistical and visualized evaluation are presented.

4.1 Illumination Estimation Experiments

4.1.1 Results

For evaluation purposes, we compare the DSNet result with deep learning based solar position estimation networks, including Sun-CNN [16], SunOriNet [10] and Yannick's method [8]. Prediction errors of these different approaches is summarised in Table 2, and visualized in Fig.8. Our method clearly outperforms others on all metrics. In all tests, the VGG [20] network surpasses others because of its fixed kernel size and deeper convolutional layers, at the expense of much slower convergence. After training on our self-collected datasets with illumination-based data augmentation, the Sun-VGG sub-network showed improved performance by leveraging both color and shadow information. We also demonstrate that the re-ranking network filtered by the coarse solar position estimate works as expected. The angular errors of the shadow ranking system are evaluated by comparison between the top ranked prediction and the ground-truth solar position. The recall@1 rate is 37.71%; here the prediction error of the top ranked result is only related to the enumeration precision, which is set to 5° in production. However, due to the cosine loss function used in training, other top ranking results are still close to the ground-truth, as shown in Fig. 3. The average ranking of the ground-truth is 7.03, which means that for the run-time shadow alignment optimization, we should search a range of solar angles to find the globally best match. However, when the error from the Sun-VGG filter is larger than 52.6°, the performance of the re-ranking network suddenly drops, possibly due to failure to pair shadow enumerations. The final optimized result after shadow



(a) Duplicated window patterns (b) Sunset shadows (c) Reflection from glass and water

(d) Snow and specular reflection (e) Low texture (f) Tree shadows

Figure 6: Examples with challenging patterns (red) for primary shadow extraction and illumination estimation. Images are cropped for better visualization. For every input image (left), our method is able to estimate the solar position, then simulate it in a 3D MR world (right). Shading parameters are also optimized.

alignment can be directly integrated into the MR renderer for immediate simulation and optimization. Next, the sunlight color and skylight color for Blinn-Phong shading are estimated by the minimization of the pixel-wise visual difference using the render-based optimization strategy. Final estimation results are shown in Fig. 6, in most cases for examples for which traditional shadow detection methods are inapplicable or give inaccurate results. Our model is able to estimate accurate sun positions in these hard cases. Yannick's method [8] fits a panorama dataset with fixed camera height, because it looks for cues from the sky and sun directly in the image. It generates implausible camera positions when the sun is behind the camera. SunCNN [16] and SunOriNet [10] can handle street-view data but lose the sun in a bird view, which includes many ambiguous shadow patterns. However, after training on massive datasets created by the MR renderer, our model is able to recognize the scene and the structured light information inside it, which is extremely beneficial to illumination estimation.

Table 2: Angular errors in solar position prediction.

Method	Min (°)	Max (°)	MAE (°)
Sun-CNN [16]	1.07	176.57	52.69
Yannick's [8]	1.90	173.73	51.57
SunOriNet [10]	0.47	166.81	52.92
DSNet-coarse	1.27	158.02	43.81
DSNet-reranked	0.18	141.12	32.91
DSNet-aligned	**0.03**	**8.81**	**1.21**

4.1.2 Ablation study

To evaluate our design decisions and examine performance of each sub-network, we designed a group of ablation experiments, with results in Table 3. We first examined the improvement provided by the Shadow-VGG sub-network by removing the shadow channel from the input to the Sun-VGG sub-network, denoted Sun-VGG-RGB. Its result is slightly better than Yannick's, because of its use of a deeper VGG network. We then tested the performance of different filters; these provide the majority of the gains. By comparing different combinations, we draw the conclusion that a better filter is more likely to remove more outliers in the ranking, resulting in lower prediction errors. The optimal combination is our Sun-VGG filter applied to our Shadow-Rank network. We also tested the Shadow-Rank component independently without a filter as a control. However, Shadow-Rank can be enhanced by prior knowledge of geolocation, time period and data, as shown in Fig. 3.

Table 3: Ablation study on DSNet components.

Method	MAE (°)
Sun-VGG-RGB	50.27
Sun-CNN [16] + Shadow-Rank	50.92
Yannick's [8] + Shadow-Rank	42.02
SunOriNet [10] + Shadow-Rank	49.25
Shadow-Rank(no filter)	67.1
Ours(Sun-VGG + Shadow-Rank)	**32.91**

4.2 Illumination Enhancement Experiments

We next tested how our illumination-based data augmentation can enhance two major applications related to modeling and fused rendering: place recognition and camera localization. We used our own datasets, and organized them according to solar position used for illumination simulation. We first collected a query group with 240 images from different viewports, then rendered $12 \times 7 = 84$ search groups using different relative solar elevation and azimuth angles. The interval of elevation angle was set to 5° and azimuth angle, 20°, within different ranges accordingly. This setup was based on real geolocation of the query photographs, taken in cities in the northern hemisphere. In each search group representing different illumination, we randomly sampled 2400 snapshots from random viewports, with color, shadow, depth and pose information given. We queried 240 images from 204K candidates to determine which group performs better by evaluation of the recall rate and localization accuracy.

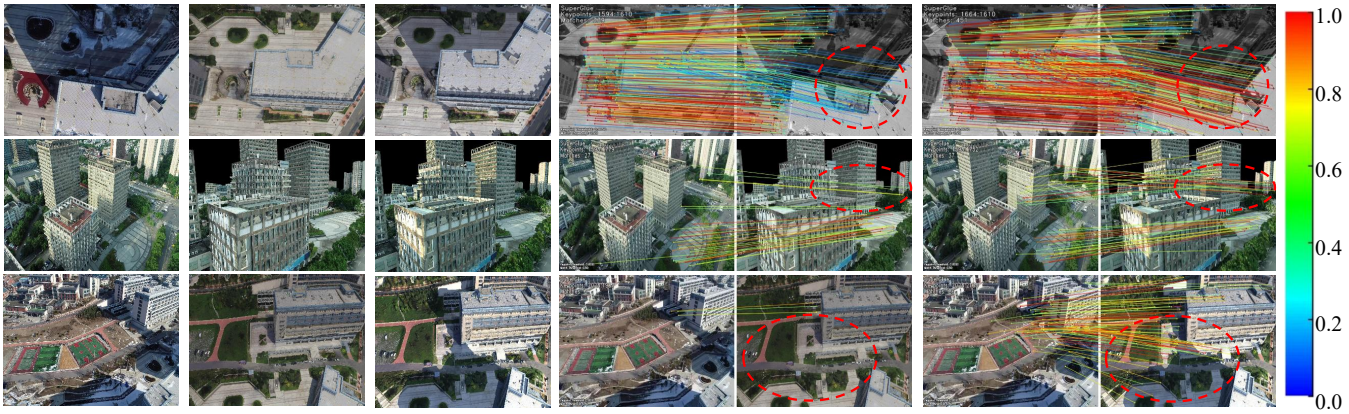


Figure 7: Enhancement of SuperGlue, showing how query photographs (a) can be better localized by illumination simulation. Due to the significant difference between viewport and appearance, SuperGlue fails to find acceptable matches in original renderings (d). To solve this problem, we estimate the global illumination and simulate it in augmented renderings (c). SuperGlue successfully discovers more robust features with higher confidence (e). Red highlights: regions with obvious improvement.

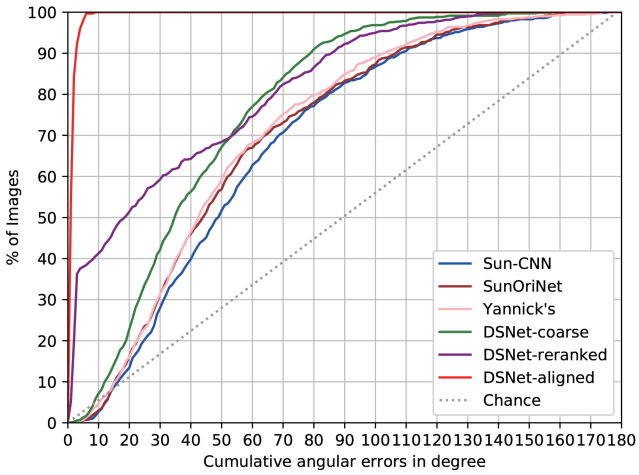


Figure 8: Cumulative angular error from solar position estimation methods. Our final result after shadow alignment (red) far exceeds the prior state-of-the-art: 99.5% prediction errors are smaller than 6°.

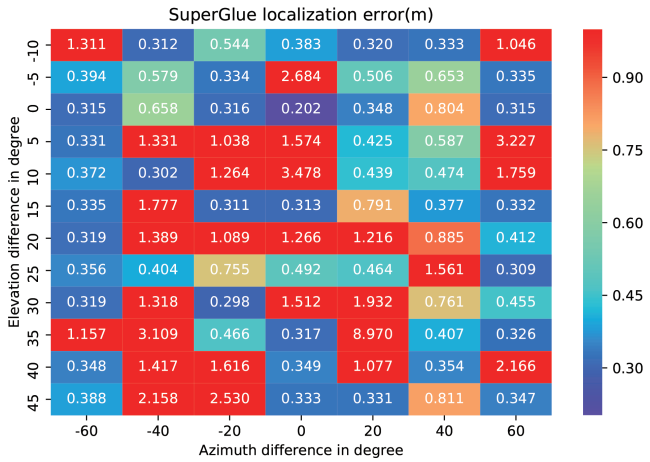


Figure 9: Enhancement heatmap for SuperGlue and different illumination groups. Each cell represents average localization error for a certain illumination dataset. The lowest average localization error is in group (0, 0) without illumination difference. The similar heatmap reflecting the rotation error is omitted.

4.2.1 NetVLAD place recognition

The first enhancement experiment tested the NetVLAD network's recognition ability for query images. Photographs in the test group were not part of the training or evaluation data. We used 10-fold cross-validation for training. To label the ground-truth in each illumination group, for each query photo, we inserted a virtual rendering with the same camera pose. In all test groups, NetVLAD is able to recognize the insertions with a different recall rate. The result is shown as a heatmap in Fig.10. We see that the group with most similar illumination as the query photographs performs best, especially on recall@5 metrics: see Fig. 10. An example can be seen in Fig. 11, in which we show how varying data augmentation can effect NetVLAD retrieval. From these results we conclude that performance of NetVLAD is linearly related to illumination change, emphasising the need for inconsistent illumination to be avoided in place recognition. For 3D model reconstruction, in most cases, it desirable to have little or no illumination change between the original group of photographs and the update group.

4.2.2 SuperGlue localization

The second enhancement experiment tested the SuperGlue network's camera pose estimation. By giving query images, we first applied NetVLAD to retrieval images. Then we looked for matching SuperPoint keypoints and found the SuperGlue mapping. With the pixel-wise depth information extracted from the render buffer, we registered each 2D keypoint to the virtual 3D world through the coordinate transformation. Finally we used RANSAC PnP [27] to acquire the 6-DOF camera pose. To eliminate the influence brought by NetVLAD and test SuperGlue individually, we use the same retrieval result for all groups. Results are shown as a heatmap in Fig.9. It clearly demonstrates that correct illumination simulation enhances the representation power of the network by reducing the localization error to a global minimum (in purple). However, we do not find a linear trend like that in the NetVLAD experiment. In circumstances with specific illumination, the performance of SuperGlue drops significantly: SuperGlue can be misguided by ambiguous shadow

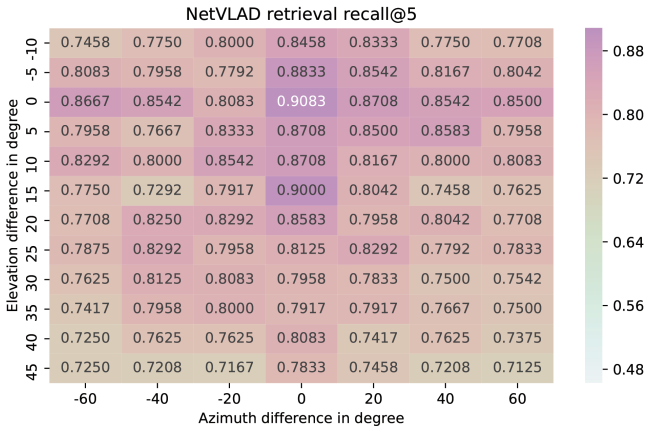


Figure 10: A heatmap showing the enhancement of the NetVLAD in different control groups rendered using different illumination. Each grid represents an average recall rate(the higher the better) of NetVLAD retrieval. The best performance is achieved by the rendering group (0, 0) without illumination difference. On other heatmaps using different recall metrics, a similar linear trend is observed.

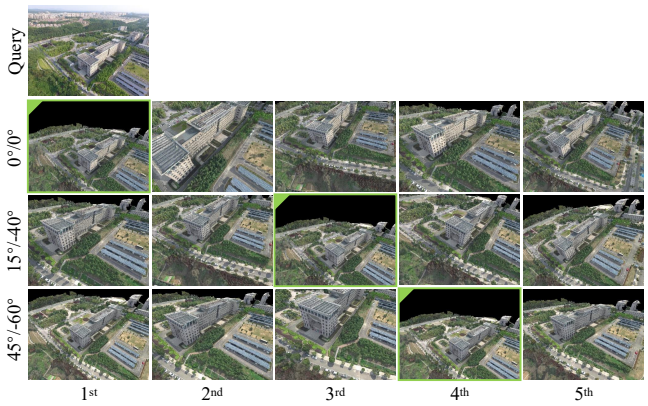


Figure 11: Enhancement of NetVLAD. For a given query photograph, we look for matches from control groups with illumination augmented renderings, in the form of relative elevation/azimuth angles. Green: ground-truth images with closest camera poses in each group. The group without systematic illumination errors provides the best retrieval performance. Camera pose and depth information are not available in the retrieval step. Query photographs are directly indexed and compared with virtual renderings without masking.

patterns with a systematic shift. However, in datasets with illumination differences larger than 60°, SuperGlue is observed to have the ability to filter out wrong augmented patterns. SuperGlue achieves best performance on the dataset without illumination differences between the query images and the test group. An intuitive visualization of the result is presented in Fig. 7. With accurate solar parameters acquired from DSNet, the run-time renderer can augment the entire 3D scene using the optimized illumination simulation. SuperGlue applied to the augmented data is able to increase the number of correct matches with higher confidence. It also improved the spacial distribution of keypoints, especially on none-planar objects. Hence, the diversity of the data is improved. As a result, the following PnP algorithm can predict more accurate camera parameters with stronger epipolar constrain.

In production, we first arbitrarily choose a primary photograph from the NetVLAD retrieval set for long-term localization. Without illumination-based data augmentation, SuperGlue can still predict a coarse camera pose. We then acquire the solar parameters from DSNet, and apply SuperGlue iteratively. This recursive optimization can be repeated until the angular change in the solar position is less than 1°. The final result is that both the camera pose and the solar parameters can be accurately acquired. Then the solar parameters can be fixed for other photographs in the same batch.

4.3 Discussion

Experimental results for illumination estimation and its improvement to camera localization have both been given. They indicate that both benefit from each other in different ways. However, their tolerances to systematic errors are different. In DSNet, solar position estimations are sensitive to camera pose errors. NetVLAD is sensitive to illumination and its performance depends linearly on illumination consistency. SuperGlue is more robust but can be misled by certain shadow patterns, depending on terrain appearance. Both achieve best performance on the dataset without solar angle errors.

For real production cases, systematic illumination differences between the photograph and the original scene should be avoided. For the first run of retrieval and localization without data augmentation, it is recommended to approximately estimate the recall rate and error distribution. Then, it is suggested to iteratively use localization networks and DSNet to obtain both accurate camera poses and solar parameters, from coarse to fine. Once the primary photograph is fully calibrated, we can fix the solar parameters for the rest of the photographs in the same batch.

5 Limitations

The proposed method relies on high quality 3D models, which can be obtained from oblique photography. However, it is not limited to aerial scenarios. In early work, experiments on models reconstructed from other types of photographs were also conducted, such as the open Lund [5] dataset. Models that already have strong shadows are not suited to this method. This issue becomes less critical when global illumination averaging is applied using reconstruction tools. The initial camera pose is required, and a coarse estimation of the primary photo is necessary. Our method is computationally intensive. The highest rendering resolution is 1920×1080 at 60 fps for visualization and optimization, and 224×224 for data augmentation and indexing. Depending on the GPU, the default maximum number of allowed triangles for smooth rendering is 10 million.

6 Conclusions

In this paper, we have presented an illumination estimation method for model updating by solving the illumination inconsistency problem. Our method first extracts shadows from the image with shading information, using them to obtain a coarse solar position estimate; it employs a re-ranking network for optimization. The final output of the shadow alignment optimizer provides pixel level estimates and visual accuracy. Our results surpass those of existing approaches; comprehensive experiments demonstrate the usefulness of our method in virtual reality applications. We show that a well-integrated illumination simulation can enhance modeling and rendering performance. The key contribution is the organic integration of MR rendering with deep learning based geometry understanding. Our self-collected dataset is available at `http://nave.vr3i.com/`.

Acknowledgments

The authors thank Ralph Martin for his help in language editing and proofreading, and anonymous reviewers for their insightful feedback and comments. This work is supported by the National Key Research and Development Program of China under Grant No. 2018YFB2100601 and the National Natural Science Foundation of China under Grant No. 61872023.

REFERENCES

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5297–5307, 2016.

[2] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[3] Contextcapture. `https://www.bentley.com/en/products/brands/contextcapture`.

[4] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 224–236, 2018.

[5] O. Enqvist, F. Kahl, and C. Olsson. Non-sequential structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 264–271, 2011.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361. IEEE, 2012.

[7] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967, 2012.

[8] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde. Deep outdoor illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7312–7321, 2017.

[9] X. Hu, Y. Jiang, C.-W. Fu, and P.-A. Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2472–2481, 2019.

[10] X. Jin, X. Sun, X. Zhang, H. Sun, R. Xu, X. Zhou, X. Li, and R. Liu. Sun orientation estimation from a single image using short-cuts in DCNN. *Optics & Laser Technology*, 110:191–195, 2019.

[11] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[13] Y.-H. Lin, W.-C. Chen, and Y.-Y. Chuang. Bedsr-net: A deep shadow removal network from a single document image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12905–12914, 2020.

[14] Y. Liu, T. Gevers, and X. Li. Estimation of sunlight direction using 3D object models. *IEEE Transactions on Image Processing*, 24(3):932–942, 2014.

[15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[16] W. Ma, S. Wang, M. A. Brubaker, S. Fidler, and R. Urtasun. Find your way by observing the sun and other semantic cues. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, 2017.

[17] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12716–12725, 2019.

[18] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4938–4947, 2020.

[19] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] J. Tian, X. Qi, L. Qu, and Y. Tang. New spectrum ratio properties and features for shadow detection. *Pattern Recognition*, 51:85–96, 2016.

[22] C. Wu, Y. Liu, Q. Dai, and B. Wilburn. Fusing multiview and photometric stereo for 3D reconstruction under uncalibrated illumination. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1082–1095, 2010.

[23] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 969 – 976, 2011.

[24] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2695–2702, 2012.

[25] D. Xu, Q. Duan, J. Zheng, J. Zhang, J. Cai, and T.-J. Cham. Shading-based surface detail recovery under general unknown illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):423–436, 2017.

[26] Q. Zheng, X. Qiao, Y. Cao, and R. W. Lau. Distraction-aware shadow detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5167–5176, 2019.

[27] H. Zhou, T. Zhang, and J. Jagadeesan. Re-weighting and 1-point RANSAC-based PnP solution to handle outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3022–3033, 2019.

[28] L. Zhu, Z. Deng, X. Hu, C.-W. Fu, X. Xu, J. Qin, and P.-A. Heng. Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 121–136, 2018.