Light Source Restoration Methods for Augmented Reality Systems

Maksim Sorokin[1[0000–0001–9093–1690]], Dmitriy Zhdanov[1[0000-0001-7346-8155]] and Andrey Zhdanov[1[0000-0002-2569-1982]]

[1] ITMO University, St. Petersburg, Russia

vergotten@gmail.com, ddzhdanov@mail.ru, andrew.gtx@gmail.com

Abstract. One of the main problems of augmented reality devices is the physically correct representation of the brightness distribution for virtual objects and their shadows in the real world. In other words, restoring the correct distribution of scene brightness is one of the key parameters that allows solving the problem of correct interaction between the virtual and real worlds, however, neural networks do not allow determining the position of light sources that are not in direct line of sight. The paper proposes methods for restoring the parameters of light sources based on the analysis of shadows cast by objects and on fully convolutional neural networks. The results of the proposed methods are presented, the accuracy of restoring the position of the light sources is estimated, and the visual difference between the image of the scene with the original light sources and the same scene with the restored parameters of the light sources is demonstrated.

Keywords: Augmented reality, Mixed reality, Fully convolutional network, Segmentation, Deep learning, Computer vision algorithms.

1 Introduction

Augmented reality (AR) - this is the environment that is created as a result of the imposition of information or objects on our perceived world in real time.

The fact is that all objects of augmented reality must comply with environmental conditions. And if we are talking about lighting, then all the objects of the virtual world should correspond to the lighting of the real world, should be properly lit and give shadows in the opposite direction from the light source. Therefore, this article addresses the problem of reconstructing optical light sources.

This work is focused on determining the real power of illumination of the light flux and its position in an environment, for this a manually synthesized sample of images with realistic optical parameters of the medium were used.

Although the sample consists of only 260 images (221 was used for training, and 39 for verification), the neural network at the output classifies with good accuracy the real optical parameters of the illumination of the medium, which were usually divided by the strength of illumination into 5 classes, where the first is 0 lumens, which means it is not lit at all, but grade 5 is the source of illumination of an ordinary room lamp. It's a segmentation approach.

Another approach – the shadow-based approach also consists of a neural network for detecting coordinates of the object and its shadow and the algorithm for beaming rays through these coordinates to determine the light source position where the most beam intersections are.

2 Related works

An analysis of outdoor lighting using a fully convolution network is presented in [1]. The following work [2] analyzes panoramic images of the environment in an open air as input and embeds the image under these environmental conditions. The work [3] also analyzes the environment and builds the shadows of objects as they should be. The convolution network is also used in [4], but to determine where the object is located: outdoors or indoors. The following article [5] presents its own architecture and solves three different problems: predicting depth, evaluating surface normals, and semantic marking. Many works [6,7,8,9] were aimed at detecting objects using convolutional neural networks.

The difference between the neural network described in this paper is that the data set is generated using a powerful renderer - “Lumicept” [10], which were used to train the neural network, restoring segmented sections of light similar to reference images with ground truth using the categorical cross entropy object function. The main task of the current work is to determine and classify the real illumination power of a real room and the light source position.

3 Segmentation method

The advantages of convolutional neural networks are that they can study and build complex feature maps based on data from previous convolutional layers that allow to recognize and create complex hierarchical features.

As a convolution network architecture, it was decided to use the VGG16 Net architecture, which was successfully used in the next work [11], it consists of 5 blocks with convolution, pooling, and “ReLU” activation function between layers, but in contrast to this work, instead of the standard method of “SGD with momentum” optimization, it was decided to use the “Nesterov” optimization method, it also was decided not to use the “dropout” regularization function.

The architecture of this network and its details are presented in figure 1.

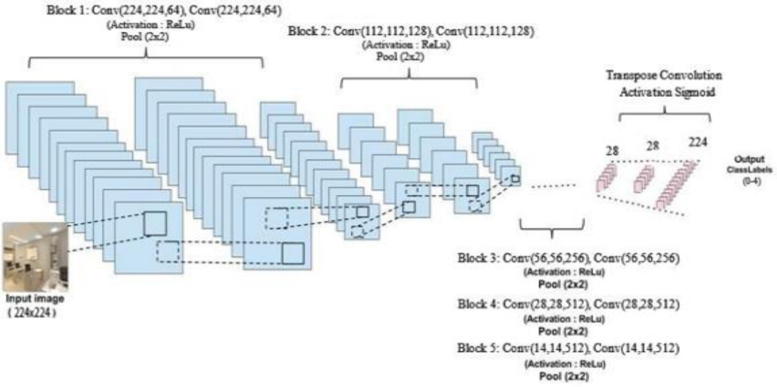

Fig. 1. The architecture of FCNN.

The task of a fully convolutional network is to classify each pixel of an image into one class. That is, passing through all convolutional layers, the network characterizes a certain area of the image to one class in accordance with the power of illumination.

The fully convolutional neural network was trained on 221 images with 200 epochs and was tested on 39 test images. To check the accuracy of the determination, the intersection over union method was used, which compares the original image and the predicted one. The average is 70 percent, which is a good result for such a small data set.

In addition, in order not to clutter up the image with unnecessary details, it was decided to use “Sobel” and “thresholding” noise reduction algorithms that emphasize the gradient edges of the image, leaving only the outlines of the interior. Figure 2 shows an example of a training dataset.

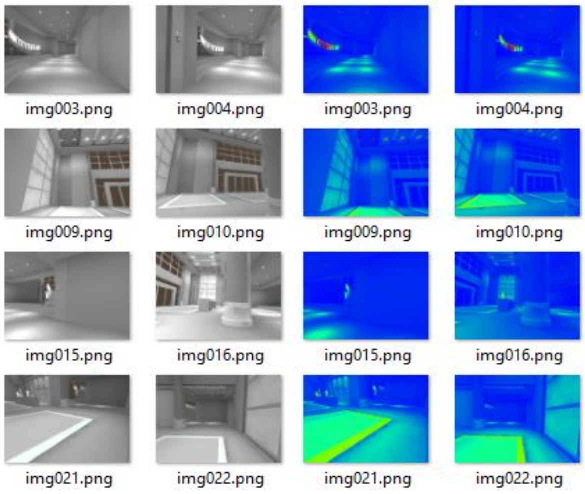

Fig. 2. Training dataset.

Figure 3 shows a working example of a trained fully convolutional neural network. Where on the left image is the original image, the right image is the reference (ground truth), and the middle one is the image predicted by the neural network.

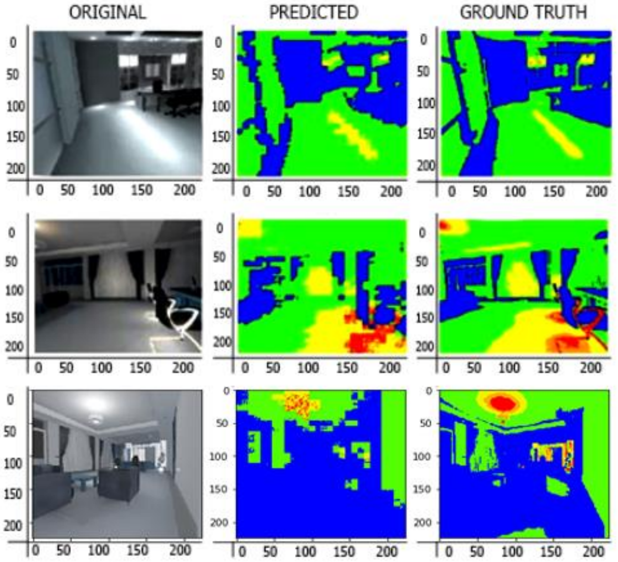

Fig. 3. An example of a trained fully convolutional neural network.

As you can see, the results are close to the ground truth images, but the network still does not capture the brightest areas very well. At this stage, the neural network works well, but there are some points that can be improved, such as adding an algorithm for determining light sources and developing an algorithm for determining diffuse surfaces.

4 Shadow based method

The objective of the current method is to determine the sources of illumination of the scene, knowing the coordinates of the shadows and the coordinates of objects casting these shadows. The method is based on the formation of narrow beams of rays connecting the coordinates of the object and the shadow, and in the region of the greatest intersection, the desired light source is more likely to be found.

The proposed method does not work with 3D models, but with images and depth maps that can be obtained using special tools and devices like 3D scanners and lidars, which can easily restore the distance to any point in the image, but working with images significantly faster than with 3D models.

Figure 4 shows a visual representation of the method of restoring light sources from the coordinates of objects and their shadows.

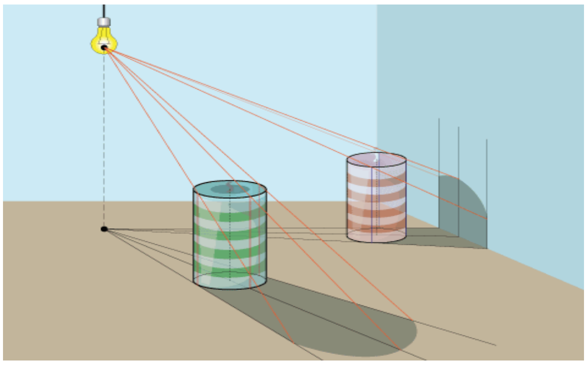

Fig. 4. Visual image of the restoration of light sources from the shadow of objects.

The algorithm of the developed method for reconstructing light sources consists of the following steps:

1. Receiving input data from the AR device: a depth map and the visible area of the scene;
2. Using a convolutional neural network, determining all the shadow areas in the image;
3. Using the Canny filtering algorithm in the ROI, defining objects that cast a shadow, selecting them with different colors and finding their boundaries;
4. Saving the coordinates of the boundaries of objects and shadows;
5. The procedure for the formation of rays. Letting narrow beams of rays through the coordinates of the points of the object and the shadow;
6. Getting results with coordinates where there is an intersection of rays.

The implementation of this method is graphically shown in Figure 5, where the rays passing through the points are highlighted in different colors, and the coordinates with the largest intersection are highlighted by the highlighted point.

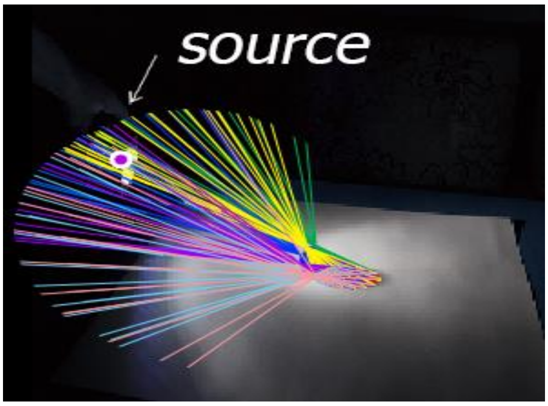

Fig. 5. Graphical representation of the method.

To restore the coordinate points of the object, an approach with a ROI (Region of Interest) relative to the shadow is used. Since the object in the image must be in contact with the shadow, the algorithm processes these areas (ROI) using the "Canny" filter and subtracts the already known coordinates of the shadows, thereby leaving only the coordinates of the object. The "Canny" filter also includes Gaussian smoothing to search for gradients and eliminate noise.

Using this approach allows us to separate the shadow area from the object itself and using the functions of the “NumPy” library makes it possible to obtain the coordinates of all non-zero pixels separately for the shadow and for the object.

This method is also based on a fully convolutional neural network and algorithms for reconstructing light sources. As a training data set, were used the SBU_Shadow data set [12], which consists of original images and their masks, where the shadow is highlighted in white and the shaded areas in black.

As the architecture of the fully convolution network, it was decided to take the U-Net architecture, because it is great for working with binary classification.

The architecture consists of 4 blocks of “downsample” layers for training classification, 4 blocks of “upsample” layers to get an array of the same dimension as the input, and also the so-called “bottleneck” block with a feature map value of 256, for deeper network learning.

U-NET architecture is shown in Figure 6.

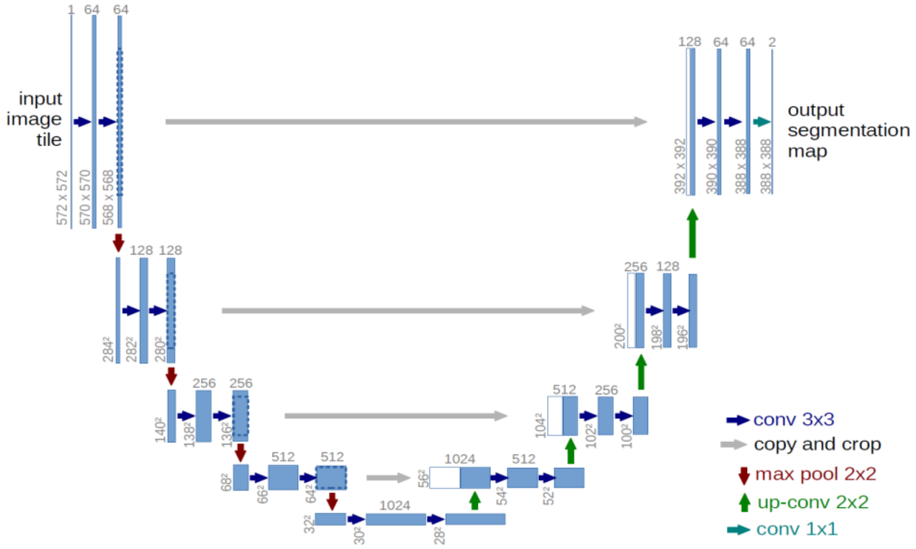

Fig. 6. U-Net architecture

An example of a pair of images of this dataset is presented in Figure 7.

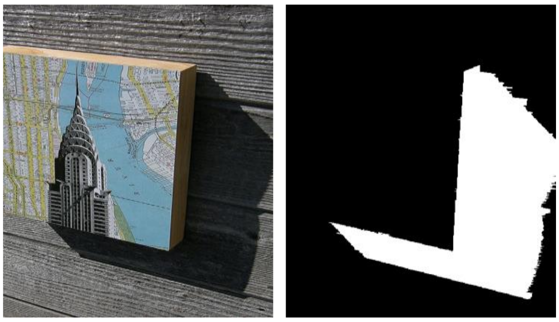

Fig. 7. Example of a dataset pair.

The training data set consists of 4085 pairs of images, and the test one consists of 638. To converge the neural network, it took only 6 epochs, which took about 3 minutes on the GeForce 1080Ti. The speed of the already trained neural network is 28 ms.

Sigmoid was used as layer activation functions, since it is non-linear, and a combination of such functions also produces a non-linear function. Another advantage of such a function is that it is not binary, which makes activation analog, in contrast to a step function. A sigmoid is also characterized by a smooth gradient.

This behavior allows finding clear boundaries in the prediction. Of the shortcomings, it is worth noting that when approaching the ends of the sigmoid, the values of Y tend to react weakly to changes in X.

This means that the gradient in such areas takes on small values. And this, in turn, leads to problems with the gradient of extinction.

Figure 8 shows the history of the training of a neural network in the form of a graph, where epochs are displayed horizontally, and the error of learning a neural network is displayed vertically. This chart is built for learning data (loss) and for test data (val_loss). The task of a neural network is to minimize the error of the objective function, which is what it achieves in the 5th epoch with values of loss: 0.2093 and val_loss: 0.2179. In this paper, were used 'binary_crossentropy' as an error function, and RMSprop with lr = 1e-4, decay = 1e-6 as the optimizer.

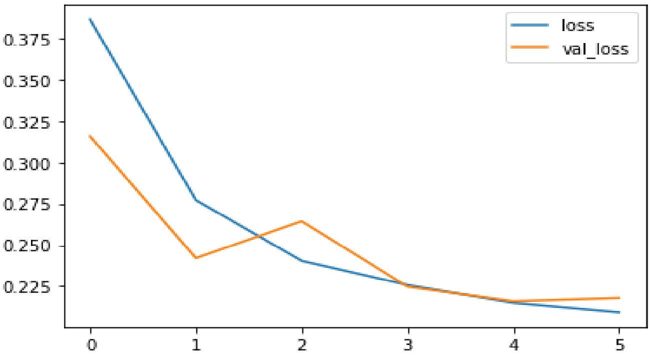

Fig. 8. Training history.

The classification accuracy on verification data was achieved 92 percent, and on test data - 94 percent using IoU (intersection over union). As can be seen in Figure 9, the results of a neural network are very close to ground truth ones, where the left images are the original images that are input, the right ones are the "ground truth", and in the middle are the result of the prediction.

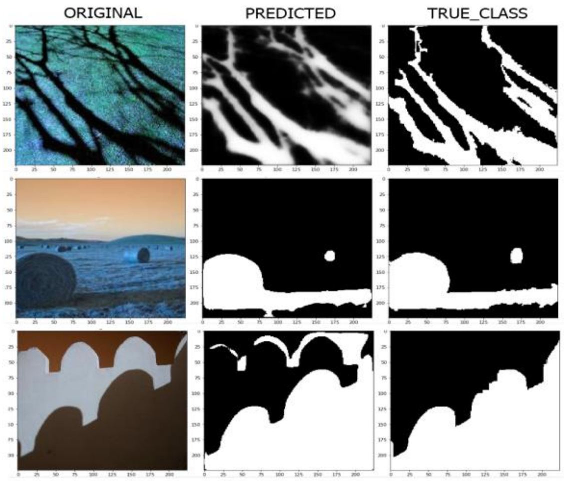

Fig. 9. The results of the neural network working.

After identifying areas with a shadow in the image, the ROI of this area is taken and the Canny operator is applied, and, knowing the coordinates of the area with the shadow, all uninteresting pixels are removed, leaving only areas with the contours of the object itself.

The result of the "Canny" operator is shown in Figure 10.

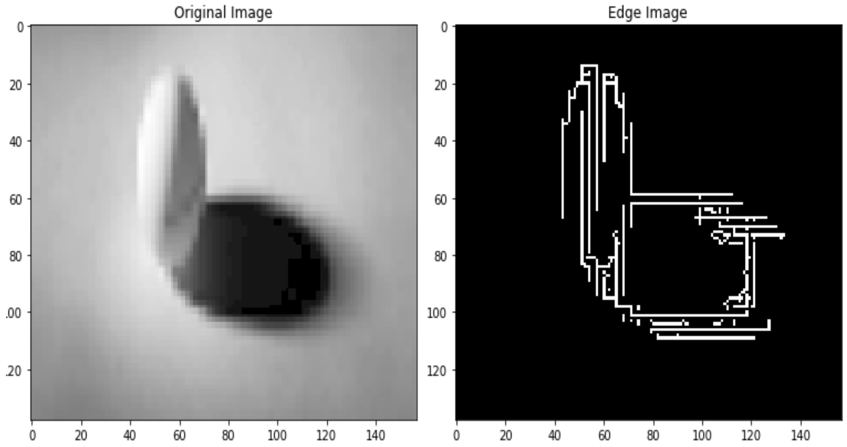

Fig. 10. Results of the Canny algorithm.

Now, knowing the coordinates of the object and the shadow, beams of rays are released connecting these points. This function is implemented as a numpy array that stores the coordinates of all lines. The numpy library is chosen because of the convenience of working with data. Now it remains only to find a greater number of intersections, i.e., the coordinate points that occur in the array the greatest number of times.

This method allows determining the light sources in a given coordinate system (i.e., relative to the current input image). Using this method with a 3D scanner will allow you to determine the coordinates of light sources in three-dimensional space.

Figure 11 shows an example of working on 5 real images with light sources. From left to right, the intermediate results of the operation of computer vision algorithms for determining the coordinates of objects and their shadows in the image are shown. The picture shows the original image, the image obtained by the Canny filter, the image with the shadow and the image with the object. The received image data allows determining the coordinates and cast rays to calculate the light sources through these points. Scenes are presented in order from 1 to 5 from top to bottom.

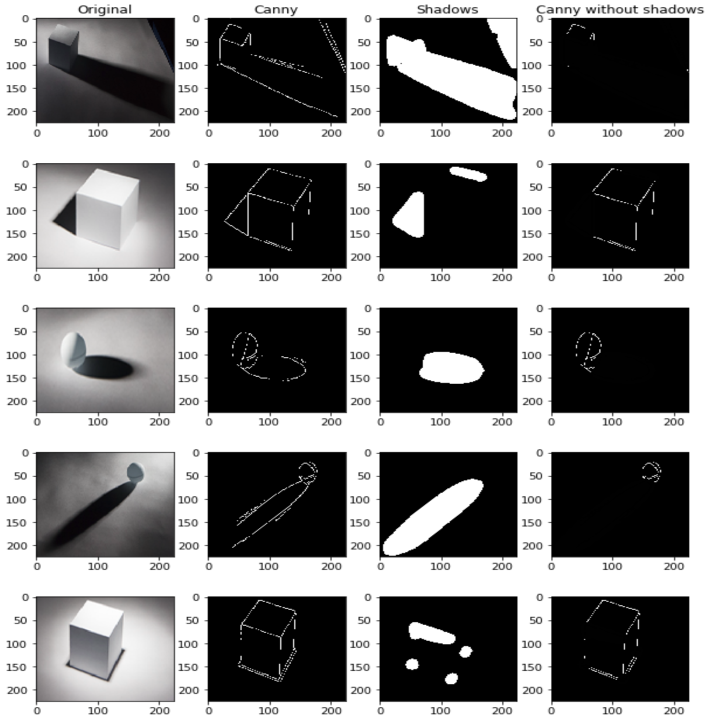

Fig. 11. An example of computer vision algorithms for determining the coordinates.

Figure 12 shows the graphical result of the algorithm to determine the light sources in the image based on the formation of rays.

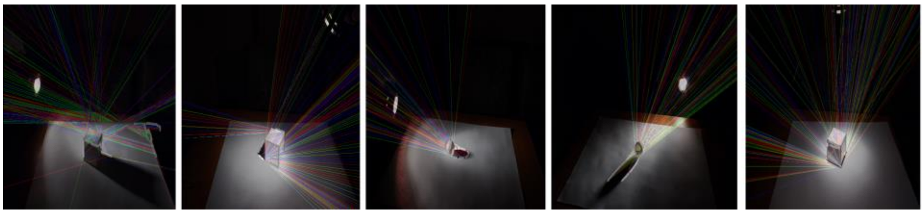

Fig. 12. The result of the algorithm for determining light sources (1 - 5 scenes).

In more detail, the result of work for 3 scenes is presented in Figure 13, where the original image is on the left and the result of work on the right, with the intersection of 3 points highlighted. The absolute error relative to the light source is 36 pixels, and the relative error is 13%.

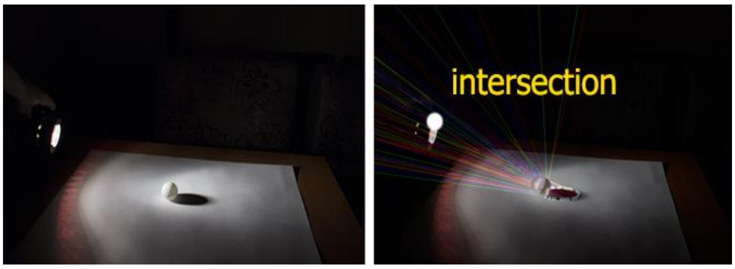

Fig. 13. The intersection of rays for scene 3.

The sizes of the original images of the scenes are 622x415 pixels. The sizes of the ROI of the image sections are 224x224 pixels so that it is possible to feed them to the neural network and classify the shadow areas.

All operations were performed on a computer with a Ryzen7-1700 processor and a GTX 1080Ti graphics card. The shadow recognition time on the image is 32ms, the object recognition is 25ms and the point search using the ray crossing method is 875ms.

In this work, were used the Python programming language and the libraries OpenCV, Keras, Numpy, and Scikit-learn.

Table 1. Accuracy of light source definition for 5 scenes.

n	Absolute error (in pixels)	Absolute error (in meters)	Relative error	Angular error (in radians)
1	12	0,036	2,58 %	0,0004
2	26	0,077	8,1 %	0,0084
3	37	0,109	13,9 %	0,0606
4	42	0,124	13,6 %	0,0412
5	19	0,056	6,7 %	0,0334

5 Conclusion

In this paper, it is shown that the developed method is suitable for working with augmented reality systems and copes with restoring the coordinate points of light sources relative to a given measurement system. In this work, we used a convolutional neural network with the U-Net architecture, after training the classification accuracy of which was almost 94 percent. The architecture of this network is excellent for binary data classification and can recognize even complex shadows in images, and the speed of operation allows it to be used in real-time systems. The recognition accuracy of light sources in some scenes proved to be quite good, in the future it is planned to improve the algorithm for restoring coordinates for better accuracy.

Acknowledgments

This work was supported by the Russian Science Foundation (project No. 18-79-10190).

References

1. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E. and Lalonde, J.-F.: Deep outdoor illumination estimation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
2. Lalonde, J.-F., Efros, A. A. and Narasimhan, S. G.: Estimating the natural illumination conditions from a single outdoor image. International Journal of Computer Vision, 98(2), 123–145 (2012)
3. Gardner, M.-A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C. and Lalonde, J.-F.: Learning to predict indoor illumination from a single image. ACM Transactions on Graphics. In: Proceedings of SIGGRAPH Asia, preprints (2017)
4. Lombardi, S. and Nishino, K.: Reflectance and Illumination Recovery in the Wild. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 129– 141 (2016)
5. Eigen, D. and Fergus, R.: Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. International Conference on Computer Vision (2015)
6. Girshick, R. B., Donahue, J., Darrell, T. and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR (2014)
7. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. ICLR (2013)
8. Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014)

9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going deeper with convolutions. CoRR, abs/1409.4842 (2014)
10. "Lumicept | Integra Inc.," Integra Inc., 2019, <https://integra.jp/en/products/lumicept> (April 12, 2019)
11. Long, J., Shelhamer, E. and Darrell T.: Fully Convolutional Networks for Semantic Segmentation. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431-3440 (2015)
12. T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras.: Large-scale training of shadow detectors with noisilyannotated shadow examples. In: Proceedings of the European Conference on Computer Vision (2016)