The Visual Computer (2019) 35:873–883
https://doi.org/10.1007/s00371-019-01666-x

ORIGINAL ARTICLE

DeepLight: light source estimation for augmented reality using deep learning

Peter Kán[1] · **Hannes Kafumann**[1]

Published online: 7 May 2019

Abstract
This paper presents a novel method for illumination estimation from RGB-D images. The main focus of the proposed method is to enhance visual coherence in augmented reality applications by providing accurate and temporally coherent estimates of real illumination. For this purpose, we designed and trained a deep neural network which calculates a dominant light direction from a single RGB-D image. Additionally, we propose a novel method for real-time outlier detection to achieve temporally coherent estimates. Our method for light source estimation in augmented reality was evaluated on the set of real scenes. Our results demonstrate that the neural network can successfully estimate light sources even in scenes which were not seen by the network during training. Moreover, we compared our results with illumination estimates calculated by the state-of-the-art method for illumination estimation. Finally, we demonstrate the applicability of our method on numerous augmented reality scenes.

Keywords Light source estimation · Augmented reality · Photometric registration · Deep learning

1 Introduction

Visual coherence plays an important role in augmented reality (AR) applications. One of the key factors for achieving visual coherence between virtual and real objects is consistent illumination. In order to render virtual objects with consistent lighting, we need to have information about real-world light sources. Therefore, the estimation of real-world illumination is of high importance for AR.

Typically, real-world illumination can be estimated using a passive or active light probe positioned in a scene [19,22]. Ideally, light sources would be estimated without the light probe to avoid the necessity of undesirable objects in the scene [9,11]. However, the estimation of real illumination from one image of the scene is a challenging problem, especially if the light sources are not directly visible in the image. Past research showed that if no priors are used in light source estimation from a single image, it is an ill-conditioned problem [6,25].

Our method is based on an assumption that prior information about lighting can be learned from a large dataset of images with known light sources. We show that this learned information can be encoded in a neural network. Such a trained network can be then used to estimate light sources during runtime in an AR scene which was not previously seen by training. We demonstrate that the neural network can achieve sufficient generality to estimate light in various scenes. This generality can be achieved by increasing the complexity of the network. In order to maintain the convergence of training with increasing network depth and to avoid a vanishing/exploding gradients problem, we design our network using residual blocks of convolutional layers [12]. Previous research showed that it is possible to calculate diffuse lighting in form of an omnidirectional image by a neural network [9,22]. In the previous work, a neural network was used to calculate the image-to-image relationship between an input image and the estimated illumination. In contrast to that, we demonstrate that a neural network can be trained to directly regress a dominant light direction from an input RGB-D image.

The varying camera poses in an AR scenario cause problems for light estimation by a neural network. These problems are caused by high dimensionality and complexity of input if a network should handle distinct camera poses in world

✉ Peter Kán
peterkan@peterkan.com

[1] Institute of Visual Computing and Human-Centered Technology,TU Wien, Vienna, Austria

space. We address this problem by regressing a dominant light direction in form of relative Euler angles ϕ and θ. These angles are always relative to a camera pose and therefore are independent of camera view angle. A dominant light direction in world space can be then calculated by adding relative Euler angles of a light source to Euler angles of a camera.

Once light sources can be estimated from each image of an AR video stream, discontinuities in the temporal domain may appear. In order to address this problem, a filtering or outlier removal needs to be applied in the temporal domain. In this paper, we propose an efficient method for outlier removal from a low amount of subsequent samples in the temporal domain. This method is based on previous research of outlier removal in the spatial domain [4], and it is adapted to the problem of outlier removal from light source estimation data.

We demonstrate the capabilities of deep learning for light source estimation in AR by integrating the presented method into a real-time AR rendering system based on ray tracing. We also evaluated the results of our method and compare them to the results of a state-of-the-art method for illumination estimation [9]. Our results indicate that a deep neural network can be used to estimate light sources on scenes which have not been previously seen in the training process.

The main contributions of this paper can be summarized as follows:

- A novel method for probe-less light source estimation in AR scenes,
- A novel method for outlier removal in the temporal domain,
- Evaluation of the proposed methods on multiple real-world scenes,
- Integration of the proposed methods in an AR rendering system based on ray tracing.

2 Related work

Light source estimation has been a challenging problem for researchers in computer graphics for decades. Knowing light position in the 3D world is required for many fields of research including computer vision, image processing and augmented reality. In augmented reality, we can see two main approaches for obtaining information about the real illumination in order to achieve consistent light: (1) inserting active or passive light probes into a scene and (2) estimating the illumination from the image of the main AR camera.

Methods based on light probes use either an active camera with a fish-eye lens or a passive object with known reflectance properties to capture environmental illumination in real time. The hemispherical image from the camera with the fish-eye lens can be used to reconstruct HDR panorama. This image can be utilized directly for image-based lighting in AR [16,19,27,29]. The image can be also processed by image processing methods to identify dominant light sources [8,34]. In case of passive light probes, illumination is captured by the main camera from the object of known geometry and reflectance which is inserted into a scene. The most common passive light probe is a mirror sphere [1,5]. We can also use a human face as a light probe to capture illumination from the front-facing camera of a mobile phone [20]. Recently, Mandl et al. showed that it is possible to utilize an arbitrary object as a light probe [22]. In their method, a series of neural networks are trained for a given light probe object. These networks are then employed to estimate light from a scene which contains a given light probe object.

The second category of methods (probe-less methods) can estimate illumination from a main AR camera image without the need of having an arbitrary known object in the scene. These methods typically use image features which are known to be directly affected by illumination. Examples of such features are shadows [28], gradient of image brightness [2,3,18] and shading [10,11,14,21,26,31,32]. Real-world illumination can be also reconstructed from RGB-D images by utilizing the estimation of surface normals and albedo [33]. Recent research showed data-driven approaches to address the problem of light source estimation. These methods typically use a large datasets of panoramas to train an illumination predictor. The predictor estimates surrounding lighting (also represented as a panorama) from a single input image. The predictor is typically based on finding similarity between an input image and one of the projections of individual panoramas [17]. The predictor can be also automatically learned from a large dataset and encoded into a neural network [9,13]. In our method, we also use a deep neural network to encode a relation between the input image and a dominant light direction. In contrast to prior work, we focus on delta directional light sources which cause hard shadows and strong directionality of the light in the scene. Additionally, we demonstrate direct applicability of our method into an AR scenario and we also focus on temporal coherence of the estimated light.

The light source estimation by neural networks can be also posed as a classification problem. In this case, the space of light directions is discretized into the set of N classes and the network classifies an image as one of these classes [23]. Previous research also showed evidence that dominant light direction can be directly regressed from an input image by a neural network [7]. Our research is based on a similar methodology while we aim at higher complexity of a scene, temporal coherence and direct application of the network to an augmented reality scenario.

3 Light source estimation using deep learning

Our method for light source estimation uses a deep neural network to learn a functional relationship between the input RGB-D image of the scene and a dominant light direction. This network needs to be trained only once on a variety of scenes, and then, it can be applied in a new scene with an arbitrary geometry. We trained our network with an assumption of one dominant light direction in a scene. Therefore, it works the best on scenes with delta light sources. The presented method for light source estimation was integrated into an AR rendering framework and evaluated on several real scenes (which were not used during training).

An important problem arises when the network needs to deal with various camera poses for the estimation of light sources. In this case, the burden of proper registration and spatial alignment of light sources with the coordinate space of a camera is posed on the network. During our research, we found out that including a camera pose into the computation makes the problem intractable for the network due to the increased dimensionality and complexity of the problem. Therefore, we decided to make light source estimation by the network independent of camera pose and calculate a transformation to world space after estimation. This can be achieved by estimating light sources in a coordinate space aligned with the camera. For this purpose, we model a dominant light direction in terms of relative Euler angles. These Euler angles are being calculated in a camera coordinate space to make the light source estimation independent of a camera pose. We need only two Euler angles (ϕ and θ) to define the direction of a light source. In our design of the neural network, these two angles are being directly regressed by the network. As ϕ and θ are being estimated in the camera coordinate space (i.e., they are relative to camera pose), we still need to transform them into the world space after estimation. For this purpose, we express the direction of a camera also in terms of Euler angles in world space. Then, we sum up the camera ϕ_c angle with the light source ϕ angle and camera θ_c angle with light θ angle to calculate absolute Euler angles of the light source (ϕ_l and θ_l) in world space. Finally, we transform the Euler representation of the dominant light direction into the vector representation (x, y, z) which is directly used for rendering.

For training of a neural network, we also need to transform the ground-truth light direction from world space into relative Euler angles. For this purpose, we first calculate a camera direction in Euler angles ϕ_c and θ_c and subtract those camera angles from absolute Euler angles of the light direction ϕ_l and θ_l (calculated in the world space). As a result, we get the dominant light direction represented as relative Euler angles. These relative values are used for training of a neural network. The calculation of relative Euler angles is depicted in Fig. 1. More formally, a relation between the relative Euler angles of light (ϕ and θ) to the absolute Euler angles of the camera (ϕ_c and θ_c) and light (ϕ_l and θ_l) can be written in the following equations:

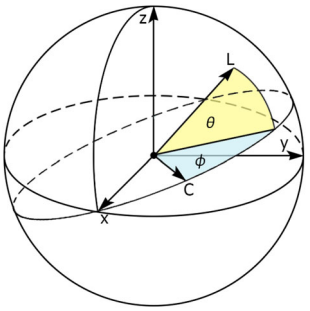

Fig. 1 Relative Euler angles ϕ and θ of dominant light direction which are regressed by our neural network from an input RGB-D image. The angles ϕ and θ are relative to the camera pose C. L denotes a dominant light direction

$$\begin{aligned} \phi_l &= \phi_c + \phi \\ \theta_l &= \theta_c + \theta \end{aligned} \tag{1}$$

Network structure Our network for light source estimation is using residual blocks of convolutional layers to avoid the problem of vanishing or exploding gradients [12]. These blocks use a shortcut connection from the beginning to the end of a block to let the network learn only a residual value from an original input. The shortcut connection and the result of a block are merged by an addition operation. The structure of our network for light source estimation is depicted in Fig. 2. The network starts with an input image of size $160 \times 120 \times 4$. Four dimensions represent RGB-D image channels. The input layer is followed by a convolutional layer with 64 kernels of size 7×7. This layer also uses strides to halve image size. The network continues with a max pooling layer which again halves the resolution. Then, the network contains 48 convolutional layers organized into 16 residual blocks. These residual blocks have an increasing number of kernels (denoted by dotted connections in Fig. 2. The convolutional layers are followed by an average pooling and by four fully connected layers with a decreasing number of neurons. All layers in the network have ReLu [24] activation function except the last two dense layers. The last layer regresses directly relative Euler angles of light direction ϕ and θ.

Training data Deep neural networks require large amount of data to be able to accurately regress a target function. During our research, we trained our network on a synthetic dataset which was rendered using Monte Carlo path tracing. Synthetic data contain five simple scenes which were rendered with a random light source position and a random camera position. A camera viewing direction was rotated toward the

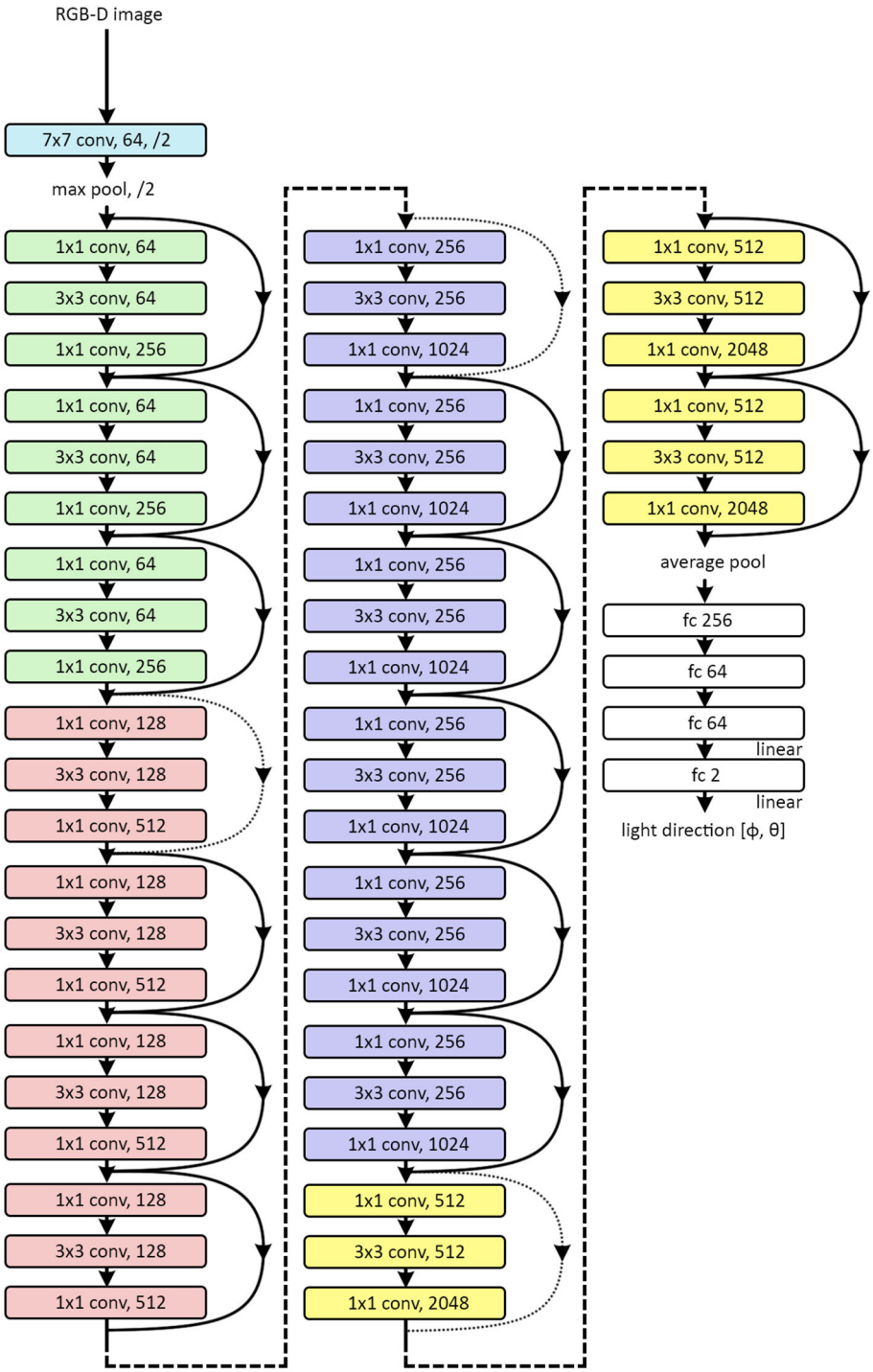

Fig. 2 The structure of the used residual neural network. Shortcuts for residual blocks [12] are indicated by curved arrows. Shortcut connections from the beginning to the end of blocks ensure that inner convolutional layers will compute a residual value. Dotted shortcuts mark the increase in dimensionality. Blocks with different dimensions are highlighted by different colors. All activation functions are ReLu except the last two layers which contain linear activations. Each layer indicates the size of a kernel for convolution as well as the number of kernels. Fully connected layers (fc) indicate the number of neurons

center of a scene. The synthetic dataset consists of 23,111 images. 3D objects used for the creation of the synthetic dataset are shown in Fig. 3.

In addition to synthetic data, we experimented with a real-world dataset which was captured in multiple indoor spaces using multiple measured light source positions and a tracked RGB-D camera. This real dataset contains 5650 images from six real scenes. During our experiments, we found out that the network converged much better on the synthetic dataset than on the real one. Moreover, a very interesting finding was that the network trained on the synthetic dataset performs also better in a real-world AR scenario than the network trained on the real dataset. The network trained on the real dataset did not converge properly and performed poorly in AR. We hypothesize that the amount of noise present in depth images was too high for the training process. Due to bad performance

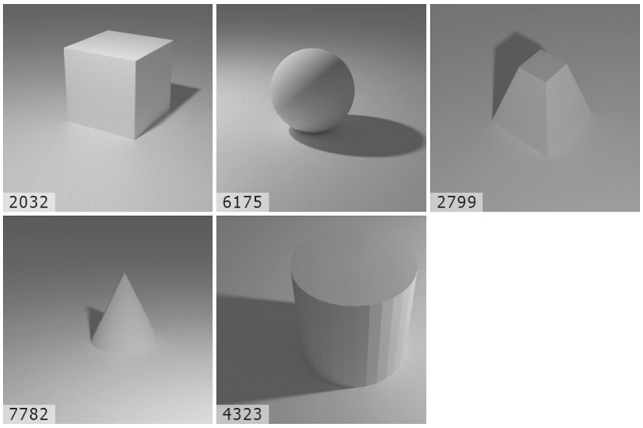

Fig. 3 3D models used for generation of synthetic training images. Each scene was rendered many times with variable light source direction and camera position. The numbers of rendered images per each scene are indicated in lower left corners of the images

of real data in the training and test scenarios, we decided to use only the synthetic dataset for training of our neural network. Moreover, we performed an experiment of training the network with RGB data only while omitting depth data. In this experiment, the network performed poorly in an AR test scenario. Therefore, we decided to use RGB-D data for all subsequent experiments.

Training A stochastic gradient descent optimizer was used for training our neural network. We trained multiple networks with varying learning rates, and finally, we selected the one with the lowest training error. In case of our light source estimation network, the learning rate 0.00004 led to the lowest error and the best performance. A loss function used for an optimization was mean squared error between a ground-truth light direction and an estimated light direction. We used zero-mean normalization on the whole dataset before training. Training was done with batch size 32. We used 200 epochs to train the network. The order of input data for training was randomized. We evaluated the results of the network in terms of both mean squared error and the real-time AR application to assess applicability to the light source estimation in real conditions.

4 Temporal coherence

In order to apply single-image light source estimation to an AR system with live video stream, we need to ensure that the estimated illumination will be coherent in the temporal domain. As the neural network estimates light for each frame separately, we need to filter the estimated light direction in real time. To achieve temporal coherence, we apply outlier removal and temporal smoothing. These two methods are both combined into one filtering algorithm. Our algorithm is inspired by previous work in outlier removal from 2D vector fields by utilizing neighborhood constraint and smooth-change constraint [4]. We adapt this method to operate in the 1D temporal domain instead of the 2D spatial domain. We observe that neighborhood constraint is represented by the first derivative and the smooth-change constraint is represented by the second derivative of the vector field. As we operate in the 1D temporal domain, the first derivative can be approximated as a difference between subsequent frames and the second derivative as difference in the first derivatives (Eqs. 2 and 3).

$$\frac{\partial l(i)}{\partial t} \approx l(i) - l(i-1) \tag{2}$$

$$\frac{\partial^2 l(i)}{\partial t^2} \approx \frac{\partial l(i)}{\partial t} - \frac{\partial l(i-1)}{\partial t} \tag{3}$$

$l(i)$ represents a dominant light direction in the ith frame. ∂t stands for a derivative in temporal domain. Our algorithm records N last light source estimates in time and uses them for filtering. Firstly, the first and second derivatives are calculated (Eqs. 2, 3) on these N estimated directions. Then, we classify a light source estimate as an outlier if the first derivative is higher than the neighborhood threshold and if second derivative is higher than the smooth-change threshold. Both thresholds must be exceeded to report an outlier. In our implementation, we empirically set both neighborhood and smoothing thresholds to value 0.1. We found these values to work best during our experiments.

After each light direction from the last N frames is classified as inlier or outlier, we calculate a resulting light source direction as an average of all inliers. This averaging enables temporal smoothing of the estimated light direction. We empirically set N to value 6 in our experiments.

5 Rendering

When the light source of a real scene can be estimated for each frame, we need to integrate this algorithm into an AR scenario. In our experiments, we used an RGB-D camera Microsoft Kinect and we integrated it into an AR rendering system based on real-time ray tracing [15]. ARToolKitPlus [30] marker-based tracking is used to track the RGB-D camera in a real scene. The light source estimation runs asynchronously in a separate thread, and it always uses the last frame from the RGB-D camera to estimate a dominant light direction. This estimated light direction is then used in the ray-tracing system to illuminate virtual objects. As both rendering and light source estimation run in interactive time, the light reflections, shadows and caustics are always adapted to the light direction in the real world. Therefore, consis-

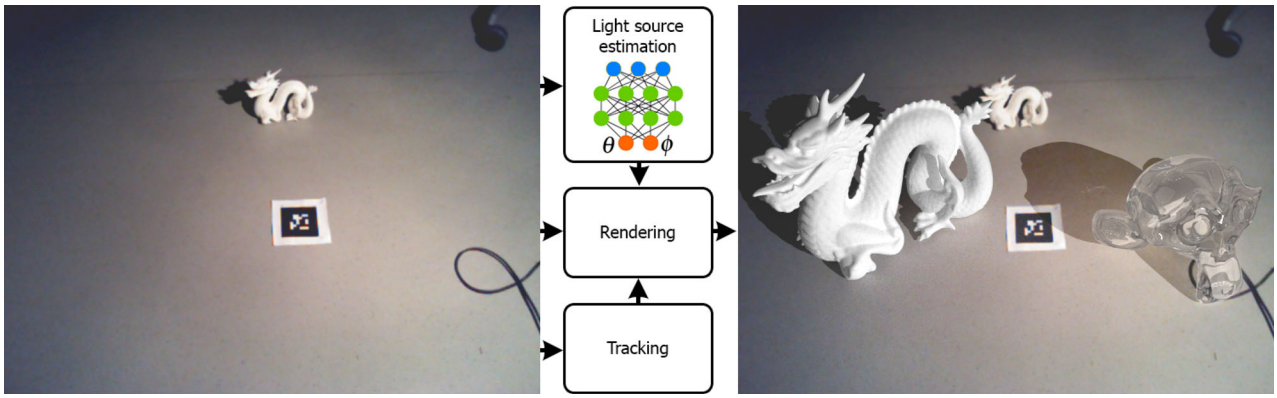

Fig. 4 Our AR system uses an input RGB-D image (left) to estimate light sources by neural network and to track a camera pose with respect to an AR marker. The estimated light source and camera pose are used in a rendering system to calculate the final AR image (right) with consistent illumination between real and virtual objects. A 3D printed model of the dragon was used in the real scene. Please note the consistent direction of shadows cast by the real dragon and virtual objects

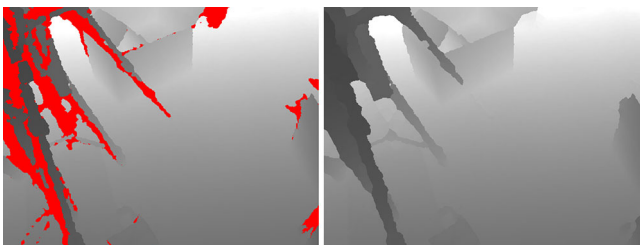

Fig. 5 Depth data from RGB-D contain holes (red pixels) around edges of objects (left). The holes are filled by using information from neighborhood valid pixels (right). The scene contains a tripod and a box

tent illumination between real and virtual objects is achieved (Fig. 4).

Depth image processing Depth data from a Kinect camera typically contain holes (with no information) around the edges of objects or in areas with specular materials. In order to improve input data for light source estimation during runtime, we utilize a hole-filling algorithm. The holes of each depth frame are filled by information from neighborhood valid pixels. For this purpose, we employ a flood-fill algorithm. Using this algorithm, we achieve a depth image fully populated with depth data. The comparison of a depth image without and with the post-processing is shown in Fig. 5. The hole-filling algorithm improves the results of light source estimation by our neural network.

6 Evaluation and results

We evaluated our method for light source estimation on the set of test scenes. These scenes were not seen by the network in the training process. All test scenes contain data from real world captured by an RGB-D camera. The results of estimated dominant light were compared to a state-of-the-art method for light source estimation by deep learning [9] and to a ground truth. The ground-truth light direction was measured by physical measurements of distances in x, y and z coordinates to the reference point of coordinate space (the AR marker). Despite the fact that our network was only trained on the synthetic dataset, we hypothesize that it can be used on real-world data. The test dataset for evaluation of our method contains five real scenes (Fig. 6). The comparison with the state of the art [9] was made by calculating the angular error to the ground-truth light direction and by rendering virtual objects in AR using an estimated light source. The method of Gardner et al. [9] estimates the illumination in form of a light probe. In order to compare their results with ours, we need to extract a dominant light direction from the resulting light probe image. Typically, this image contains one or more areas with overexposed pixels. These areas correspond to the directions of dominant light sources. In order to get a single direction, we first calculate a centroid for each of these overexposed areas and then we select the one which is the closest to the ground-truth light direction. We compare this extracted direction with the result of our method in terms of angular error to the ground truth.

The results of the evaluation are shown in Fig. 6 and Table 1. The results indicate that our method achieves higher accuracy of an estimated light direction than the compared state-of-the-art method in four scenes and the compared method is better in one of the tested scenes. The results of AR renderings with light direction estimated by our method, by the compared method and with ground-truth light direction, are shown in Fig. 6. We can see that for all scenes our method estimates light direction which is visually acceptable and comparable with a real scene illumination (i.e., the shadows of virtual and real objects are consistent). The results

Fig. 6 The comparison of AR renderings with illumination estimated by our method (left column) and the state-of-the-art method from Gardner et al. [9] (middle column). Rendering with ground-truth light direction is shown in the right column. Scenes contain the following real objects from top to bottom: cactus, ping-pong racket, 3D-printed dragon, fan and humidifier. In the scene with fan, the method of Gardner at al. estimated a dominant light direction with negative z coordinate. Therefore, the dragon appears dark

Table 1 The comparison of our method to algorithm from Gardner et al. [9]

Scene	Our method Angular error	Gardner et al. Angular error
Cactus	26.8	31.6
Ping-pong racket	25.1	43.1
Dragon	27.1	36.8
Fan	31.9	97.2
Humidifier	31.3	25.7
Average error	28.4	46.9

Numbers represent angular errors to the ground-truth light directions (i.e., the angle between estimated light direction and measured light direction), measured in degrees. The method of Gardner et al. uses only an RGB image, while our method utilizes RGB-D data

of our evaluation support our hypothesis that the network trained on synthetic dataset can successfully estimate light direction also on real-world data. Additionally, the results show that a deep neural network can estimate light direction in scenes which were not seen in a training process.

In our evaluation, we also measured the performance of our method on a synthetic dataset. For this purpose, we rendered a new dataset which consists of 7097 images. The scene with a cone model was used in this case. A similar scene was also used in training, but the test data were rendered with new viewpoints and light source positions which were not used during training. We calculated the average angular error of estimated light direction. All images from the new rendered dataset were used for this evaluation. The resulting error is 20.4°. This result indicates that the trained neural network performs well also on synthetic data under new viewpoints.

The above-discussed evaluations were performed in an environment with controlled light. We were also interested to investigate the performance of our neural network in a scene with natural (uncontrolled) illumination. For this purpose, we ran an experiment in an office scene lit by sunlight through a window. In this scenario, we measured the position of the window with respect to an AR marker to represent a reference light direction. We compared our result with the method of Gardner et al. [9]. The results of this experiment are shown in Fig. 7. The rendering with the estimated light direction indicates correct estimation of light by our method. (Virtual shadows are consistent with the real ones.) This positive result is also supported by the projection of the estimated light direction into the captured light probe (Fig. 7 red dot in the ground-truth environment). The light probe image was captured by a camera with fish-eye lens (185° field of view), and it represents the upper hemisphere of incoming light. Our method calculated the light direction with an angular error of 21.7°. The compared method performed better in this scene (Angular error of 12°). Nevertheless, the result of this experiment suggests that our method performs well also in scenes with arbitrary uncontrolled lighting.

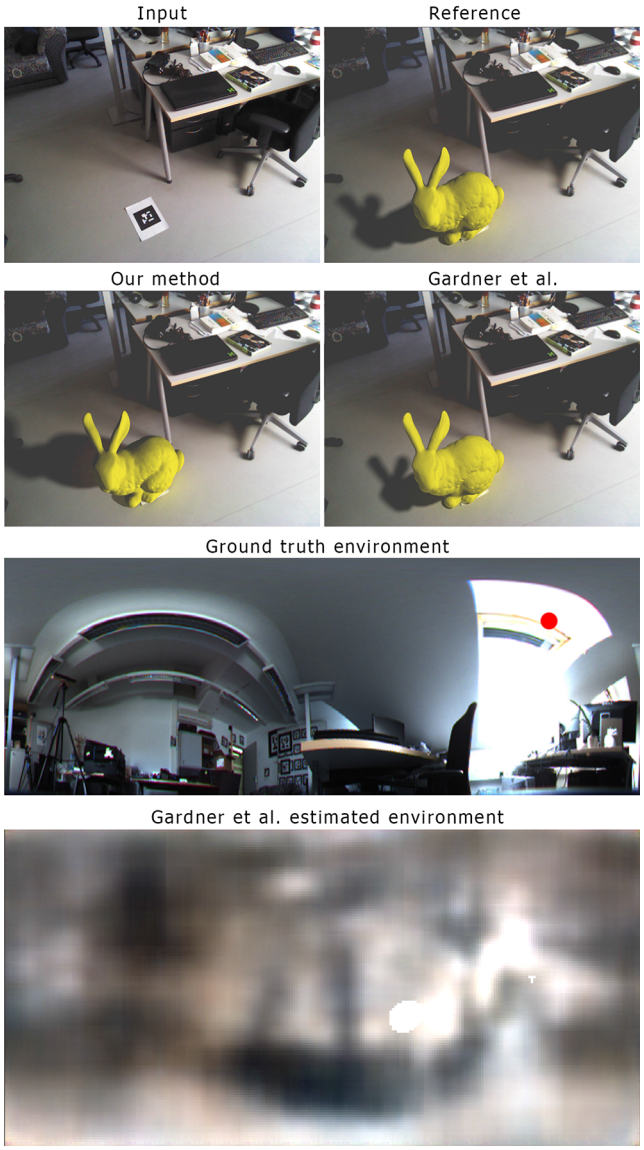

Fig. 7 The evaluation of our method in a scene with uncontrolled natural illumination. The image shows an input image, AR rendering using a reference light direction, the result of our method and the result of Gardner et al. [9]. The ground-truth environment map shows the surrounding environment (upper hemisphere) in the scene. The red dot indicates the light direction estimated by our method. The environment map estimated by Gardner et al. correctly identifies the light from the window. Please note that two environment maps are not aligned in elevation angle because the ground truth was captured in world space, while Gardner et al. estimates the environment in camera space

In addition to single-image light source estimation, we evaluated our method for temporal filtering and outlier removal on a live AR stream. We captured the video of a scene with a moving light source, and we evaluated our light source estimation with and without temporal filtering in comparison with ground-truth data. In this evaluation, an average angular

error was 41° without temporal filtering and 38.3° with temporal filtering. The results suggest that our method achieves higher temporal coherence (and lower average error) when temporal filtering is used. Additionally, the example of our method for light source estimation with temporal filtering is given in supplementary video.

Finally, we measured the calculation time of our method. The neural network processing and whole light source estimation with AR integration were measured separately to provide detailed analysis. Calculation times were calculated as averages of multiple measurements. A computer with a hexa-core 3.2 GHz processor and NVIDIA Titan Xp graphics card was used for time measurements. Light source estimation by the neural network was executed on the CPU because the GPU was fully utilized by ray tracing. The average time of light source estimation by the neural network was 380 ms. In our implementation, light source estimation and AR rendering were represented by two services. Therefore, a communication overhead between them also influences the update rate of estimated light. The average time for communication between the rendering and illumination estimation was 50 ms. In the future, this overhead can be reduced by integration of both algorithms into one standalone system. The AR rendering is running asynchronously and therefore is independent of the light estimation speed. With ray-tracing-based rendering, we achieved an average rendering time of 58ms. The results indicate that processing by the neural network achieves interactive speed suitable for AR applications.

7 Discussion

The results of our evaluation show that the trained neural network is capable of estimating illumination from real scenes which were not used during training. Moreover, an interesting finding was that training on a synthetic dataset leads to better convergence than on real-world data and to better performance in AR. We hypothesize that bad performance of training with real-world data was caused by insufficient variability of light positions and by a high level of noise in depth data.

The results also indicate that in average our method performs better in the estimation of delta light sources than the method of Gardner et al. [9]. The average error of our method to ground-truth light direction was 28.4°, while the average error of the compared method was 46.9°. Nevertheless, it is important to note the compared method is also capable of estimating an omnidirectional light source which is important for scenes with diffuse lighting. Therefore, we see both methods rather as complementary than competitive.

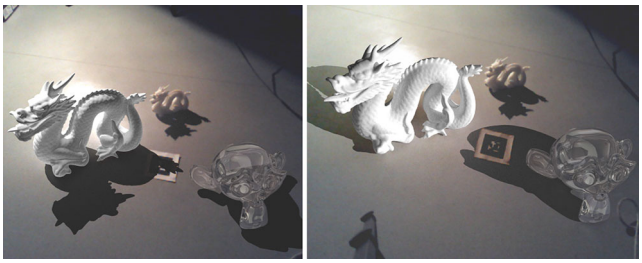

Fig. 8 Light source estimation by our method when light comes from a direction opposite to a camera. Left image shows correct estimate of light direction by our method. In some cases, the method estimates light incorrectly (right)

Limitations and future work Our method works well in many tested AR scenes. However, in some special cases the network does not estimate a light direction correctly. This is often the case if a light source is positioned opposite to a camera. In this case, the uncertainty of the network can be observed (Fig. 8). We hypothesize that this uncertainty is caused by discontinuity in yaw angle on the direction opposite to the camera. This angle can be represented as both π and $-\pi$. Therefore, the network cannot find a continuous transition from one side to another. In the future, this problem can be addressed by using a different representation of the light direction. For example, the relative direction (x, y, z) in the camera coordinate space can be used.

Another limitation of the network, trained on synthetic data, can be caused by a domain gap between real-world and simplistic synthetic data. As a consequence, the network might not operate properly in complex real scenes. This problem can be addressed in the future by two solutions: The first one is to create complex synthetic scenes which mimic the real world as close as possible. The second direction is to improve quality of the capturing process and use high-quality real-world data to do additional training of the network.

An interesting direction for future work will be training of a network which will operate in both the spatial (2D image) and temporal domain. Such a network might calculate light estimates which are already temporally coherent and therefore additional filtering in the temporal domain would not be needed. Additionally, as real scenes often contain more than one light source, we also aim in future work at training a deep neural network which can estimate multiple light sources.

Finally, the exploration of a wide space of various network designs would be vital for finding the best design for a given problem. We explored many possible network designs during this research, and we found the network in Fig. 2 to work the best in our experiments. Nevertheless, the automatic exploration of design space of neural networks would be beneficial for finding the most appropriate model for various problems.

8 Conclusion

This paper presents a novel method for delta light source estimation in AR. An end-to-end AR system is presented which estimates a directional light source from a single RGB-D camera and integrates this light estimate into AR rendering. The rendering system superimposes virtual objects into a real image with consistent illumination using the estimated light direction. Moreover, temporal coherence of light source estimation is achieved by applying outlier removal and temporal filtering. We evaluated the proposed methods on various AR scenes. The results indicate that the proposed neural network can estimate a dominant light direction even on scenes which were not seen by the network during training. Finally, our evaluation shows that our method can be a beneficial complement to the methods estimating diffuse lighting to faithfully estimate all frequencies of illumination in AR.

Acknowledgements Open access funding provided by TU Wien (TUW). This research was funded by the Austrian research project WWTF ICT15-015. We thank Marc-Andé Gardner for providing us with the results of his algorithm through a Web service and for the kind explanations of details of the algorithm. We are also thankful to Alexander Pacha for advices about deep learning. We would like to thank NVIDIA Corporation for the donation of a Titan Xp graphics card and the Center for Geometry and Computational Design for access to a multi-GPU PC for training our neural networks. We also thank Min Kyung Lee, Iana Podkosova and Khrystyna Vasylevska for their support with controlled light experiments.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical standard This research did not involve human participants.

References

1. Agusanto, K., Li, L., Chuangui, Z., Sing, N.W.: Photorealistic rendering for augmented reality using environment illumination. In: IEEE ISMAR, pp. 208–218 (2003)
2. Boom, B., Orts-Escolano, S., Ning, X., McDonagh, S., Sandilands, P., Fisher, R.B.: Point light source estimation based on scenes recorded by a RGB-D camera. In: British Machine Vision Conference (2013)
3. Boom, B.J., Orts-Escolano, S., Ning, X.X., McDonagh, S., Sandilands, P., Fisher, R.B.: Interactive light source position estimation for augmented reality with an RGB-D camera. Comput. Animat. Virtual Worlds **28**(1), 5149–5159 (2015)
4. Dante, A., Brookes, M.: Precise real-time outlier removal from motion vector fields for 3D reconstruction. In: International Conference on Image Processing, vol. 1 (2003)
5. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: SIGGRAPH, pp. 189–198. ACM, New York (1998)
6. Dong, Y., Chen, G., Peers, P., Zhang, J., Tong, X.: Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting. ACM Trans. Graph. **33**(6), 193:1–193:12 (2014)
7. Elizondo, D.A., Zhou, S.M., Chrysostomou, C.: Light source detection for digital images in noisy scenes: a neural network approach. Neural Comput. Appl. **28**(5), 899–909 (2017)
8. Frahm, J.M., Koeser, K., Grest, D., Koch, R.: Markerless augmented reality with light source estimation for direct illumination. In: CVMP, pp. 211–220 (2005)
9. Gardner, M.A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., Lalonde, J.F.: Learning to predict indoor illumination from a single image. ACM Trans. Graph. **36**(6), 176:1–176:14 (2017)
10. Gruber, L., Richter-Trummer, T., Schmalstieg, D.: Real-time photometric registration from arbitrary geometry. In: ISMAR, pp. 119–128 (2012)
11. Gruber, L., Ventura, J., Schmalstieg, D.: Image-space illumination for augmented reality in dynamic environments. In: 2015 IEEE VR, pp. 127–134 (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR arXiv:1512.03385 (2015)
13. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., Lalonde, J.: Deep outdoor illumination estimation. CoRR arXiv:1611.06403 (2016)
14. Jiddi, S., Robert, P., Marchand, E.: Reflectance and illumination estimation for realistic augmentations of real scenes. In: IEEE ISMAR, pp. 244–249 (2016)
15. Kán, P.: High-quality real-time global illumination in augmented reality. Ph.D. thesis, TU Wien (2014)
16. Kán, P., Unterguggenberger, J., Kaufmann, H.: High-quality consistent illumination in mobile augmented reality by radiance convolution on the GPU. In: ISVC 2015, Part I, LNCS 9474, pp. 574–585. Springer (2015)
17. Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., Forsyth, D.: Automatic scene inference for 3D object compositing. ACM Trans. Graph. **33**(3), 32:1–32:15 (2014)
18. Kasper, M., Keivan, N., Sibley, G., Heckman, C.R.: Light source estimation with analytical path-tracing. CoRR arXiv:1701.04101 (2017)
19. Knecht, M., Traxler, C., Mattausch, O., Purgathofer, W., Wimmer, M.: Differential instant radiosity for mixed reality. In: IEEE ISMAR, pp. 99–107 (2010)
20. Knorr, S.B., Kurz, D.: Real-time illumination estimation from faces for coherent rendering. In: IEEE ISMAR, pp. 113–122 (2014)
21. Lopez-Moreno, J., Garces, E., Hadap, S., Reinhard, E., Gutierrez, D.: Multiple light source estimation in a single image. Comput. Graph. Forum **32**(8), 170–182 (2013)
22. Mandl, D., Yi, K.M., Mohr, P., Roth, P.M., Fua, P., Lepetit, V., Schmalstieg, D., Kalkofen, D.: Learning lightprobes for mixed reality illumination. In: IEEE ISMAR, pp. 82–89 (2017)
23. Marques, B.A.D., Drumond, R.R., Vasconcelos, C.N., Clua, E.: Deep light source estimation for mixed reality. In: VISIGRAPP, pp. 303–311 (2018)
24. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML, pp. 807–814. Omnipress, USA (2010)

25. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH, pp. 117–128. ACM, New York, NY, USA (2001)
26. Richter-Trummer, T., Kalkofen, D., Park, J., Schmalstieg, D.: Instant mixed reality lighting from casual scanning. In: IEEE ISMAR, pp. 27–36 (2016)
27. Rohmer, K., Bschel, W., Dachselt, R., Grosch, T.: Interactive near-field illumination for photorealistic augmented reality on mobile devices. In: IEEE ISMAR, pp. 29–38 (2014)
28. Sato, I., Sato, Y., Ikeuchi, K.: Illumination from shadows. IEEE Trans. Pattern Anal. Mach. Intell. **25**(3), 290–300 (2003)
29. Supan, P., Stuppacher, I., Haller, M.J.: Image based shadowing in real-time augmented reality. IJVR **5**, 1–7 (2006)
30. Wagner, D., Schmalstieg, D.: ARToolKitPlus for pose tracking on mobile devices. Technical Report, TU Graz (2007)
31. Weber, M., Cipolla, R.: A practical method for estimation of point light-sources. BMVC **2001**(2), 471–480 (2001)
32. Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: Elasticfusion: real-time dense slam and light source estimation. Int J Robot Res **35**(14), 1697–1716 (2016)
33. Yu, L., Yeung, S., Tai, Y., Lin, S.: Shading-based shape refinement of RGB-D images. In: IEEE CVPR, pp. 1415–1422 (2013)
34. Zhou, W., Kambhamettu, C.: Estimation of illuminant direction and intensity of multiple light sources. In: ECCV, pp. 206–220. Springer (2002)

Peter Kán is a postdoctoral researcher at the Institute of Visual Computing and Human-Centered Technology at Vienna University of Technology. He conducted his research in the areas of photorealistic rendering, augmented reality, virtual reality and automatic 3D content generation. He received his doctorate from Vienna University of Technology in 2014 with his PhD thesis about high-quality real-time global illumination in augmented reality.

Hannes Kafumann is associate professor at the Institute of Visual Computing and Human-Centered Technology at Vienna University of Technology and head of the VR group since 2005. He participated in projects and conducted research in the areas of virtual reality, tracking, mobile augmented reality, training spatial abilities in AR/VR, tangible interaction, medical VR/AR applications, real-time ray tracing, redirected walking, geometry and educational mathematics software. His habilitation (2010) was on "Applications of Mixed Reality" with a major focus on educational mixed reality applications. He managed over 20 national research projects and published more than 90 scientific papers.