ISSN 0361-7688, Programming and Computer Software, 2020, Vol. 46, No. 3, pp. 207–216. © Pleiades Publishing, Ltd., 2020.
Russian Text © The Author(s), 2020, published in Programmirovanie, 2020, Vol. 46, No. 3.

Restoration of Lighting Parameters in Mixed Reality Systems Using Convolutional Neural Network Technology Based on RGBD Images

M. I. Sorokin[a],*, D. D. Zhdanov[a],, A. D. Zhdanov[a],***, I. S. Potemin[a],****, and N. N. Bogdanov[a],*******

[a] *St. Petersburg National Research University of Information Technologies, Mechanics, and Optics, St. Petersburg, 197101 Russia*

**e-mail: vergotten@gmail.com*

***e-mail: ddzhdanov@mail.ru*

****e-mail: adzhdanov@itmo.ru*

*****e-mail: ipotemin@yandex.ru*

******e-mail: bogdanov@k-remonta.com*

Received December 18, 2019; revised January 08, 2020; accepted January 21, 2020

Abstract—One of the main problems of mixed reality devices is the lack of universal methods and algorithms for the visualization of virtual world objects in real space. The key point of natural perception of virtual objects in the real world is the creation of natural lighting conditions for virtual world objects by light sources located in the real world, i.e. the formation of natural glares on virtual objects and shadows cast by these objects in the real world. The paper proposes a method for adequately determining the position of the main light sources of the real world in mixed reality systems. Modern technologies that combine the capability of forming 2.5D images created by depth cameras and their subsequent computer processing using neural networks make it possible to identify real-world objects, recognize their shadows, and correctly restore the light sources that create these shadows. The results of the proposed method 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.

DOI: 10.1134/S0361768820030093

1. INTRODUCTION

Currently, visual interactive technologies, such as virtual reality (VR), augmented reality (AR), and mixed reality (MR), undergo intensive development [1–3]. A lot of VR, AR, and MR devices [4–6] are available on the market, and many research groups [7–9] work on improving the quality of visual perception, including the restoration of real world light sources. There are a number of implemented pilot projects that use these technologies not only in the entertainment and game industries. The VR, AR, and MR technologies are used in medicine [10], architecture [11], military science [12], and in other fields.

Note that the MR technology is more complicated than the VR and AR technologies. While in a VR system the user is completely embedded in a virtual world and does not see the environment and in the AR system the user can observe auxiliary elements, such as navigational data or information messages, in a MR system the user sees virtual objects embedded in the real world space. In this case, the user must perceive the virtual objects as real world objects, and he or she should not have the visual perception discomfort caused, e.g., by the unnatural illumination of virtual objects. For an object to be perceived realistically, a number of conditions must be satisfied. First, the virtual object surface must have natural optical properties (reflection, transmission, and refraction, including proper textures). In most cases, the properties of virtual object surfaces (textures and optical properties) can be assigned by the computer system (e.g., using a library of materials) or by the user (on the basis of the personal experience). For this reason, the assignment of optical properties is a relatively simple task.

Another important condition for the realistic perception of virtual objects is the physical correction of their illumination. That is, the reflection spots, bright and shadowed regions in the image, and virtual objects must cast shadows corresponding to the illumination conditions. From the viewpoint of the visual perception, the reflection spots and the shadows cast by them must correlate with the reflection spots and shadows cast by real objects and be not uncomfortable for observing the mixed image.

In this paper, we propose an efficient method for restoring the location of real world light sources (LS)

given 2.5D images obtained from depth cameras that can be integrated into MR systems.

2. ANALYSIS OF RELATED LITERATURE

Earlier, we have considered the restoration of brightness distribution using the method of three spheres, which is based on determining the location of the light source in the scene [13]. In that paper, it was shown that the proposed method for determining the light source location on the basis of HDRI analysis can provide fairly accurate results. However, it has a number of drawbacks, among which are the following ones.

1. The method is valid only under the condition that the reflection properties of the surfaces are close to Lambert's law, i.e., if they are perfectly "matt" or diffusively reflecting, and the light source radiation pattern is close to Lambert's law.

2. For a local region in the image, the method makes it possible to find only one light source.

The method proposed in this paper does not have such drawbacks. It can deal with arbitrary diffuse surfaces (specified by their bidirectional scattering distribution function) and with several light sources.

The restoration of illumination conditions of real-life scenes is studied by a number of research groups. In [14], a method based on the high quality evaluation of illuminance using a convolutional neural network (CNN) is described. The authors train the CNN using synthesized images and then use it for the analysis of real-life images. To guarantee the accuracy and effectiveness of the method, the illuminance evaluation results obtained by a number of CNN instances are combined. The experimental results show that this method gives fairly accurate estimates in the analysis of real-life images.

In [15], a shadow visualization method using the lighting generated using images in AR applications is described. To approximate the illumination results and shading the environment. The system uses a dome with light sources of different colors. The color of each shadow is determined by an area of the environment behind the light source. As a result, it is possible to determine the influence of changes in the lighting conditions on shadow casting by virtual objects.

In [16], a concept of real-time shadow analysis for AR applications that use shadow volumes is proposed. This concept was implemented in the prototype shadowAReality and demonstrated promising results. The shadows significantly improve the real scene and present a more intuitive and realistic world to the user. As claimed in [17], the algorithm of shadow volume analysis can be improved using portals, occlusion, and view frustum culling techniques; all of them can improve performance by avoiding the rendering of unnecessary shadow volumes. Moreover, the shadow volume algorithm can be improved using the nVIDIA Cg shading language [18, 19].

In [20], a method of restoring the illumination and surface properties from randomly scanned geometry is proposed. This implies a fast and potentially "noisy" procedure of scanning unmodified and unstructured scenes using the standard RGB-D sensor. In distinction from the procedures for restoring the illumination parameters, which require thorough preparation in laboratory conditions, this method works with the data that can be acquired by users in field conditions. To obtain a reliable restoring procedure, the authors segmented the recontructed geometry into surfaces with uniform properties of the material and calculated the transport of radiation on these segments. Using these input data, the authors solved the inverse rendering problem—factorization of illumination and material properties—using optimization in the form of spherical harmonics. This makes it possible to take into account self-shadowing and to restore the reflective properties of objects. The results thus obtained can be used to generate a wide range of MR applications, including the rendering of synthesized objects with the corresponding illumination in the given scene, and to generate the image of a scene (or its part) under new lighting conditions. The reliability of this approach was demonstrated using real-life and synthesized images under various lighting conditions, and the results were compared with the input data.

Restoring the parameters of natural and artificial illumination from HDRIs was studied in [21–23]. The proposed methods can find bright light sources that create reflection spots and shadows. However, for MR headsets, a more natural approach for finding the parameters of natural lighting (the Sun position) must be based on the analysis of the parameters of the sensors that determine the headset's attitude in space, date, and time; this approach must associate these parameters with the sky model parameters and, correspondingly, to the position of the Sun on the sky.

In [24], a method for recovering shadow contours using the region of interest (ROI) and the analysis of adjacent pixels for contrast changes was proposed. As applied to simple objects and shadows, this method is fairly simple and reliable. In distinction from this method, convolutional networks make it possible to detect complex shadows in the entire image rather than in the region of interest only.

The studies just discussed propose effective approaches to the recovery of lighting distribution in MR systems, which in some cases gives acceptable results. However, these approaches perform well if the scene contains only one light source (the Sun or an artificial light source) or if the sources are far from the scene (practically, at infinity). In real life, there are multiple light sources. In this paper, we propose a novel approach that helps recover the illumination

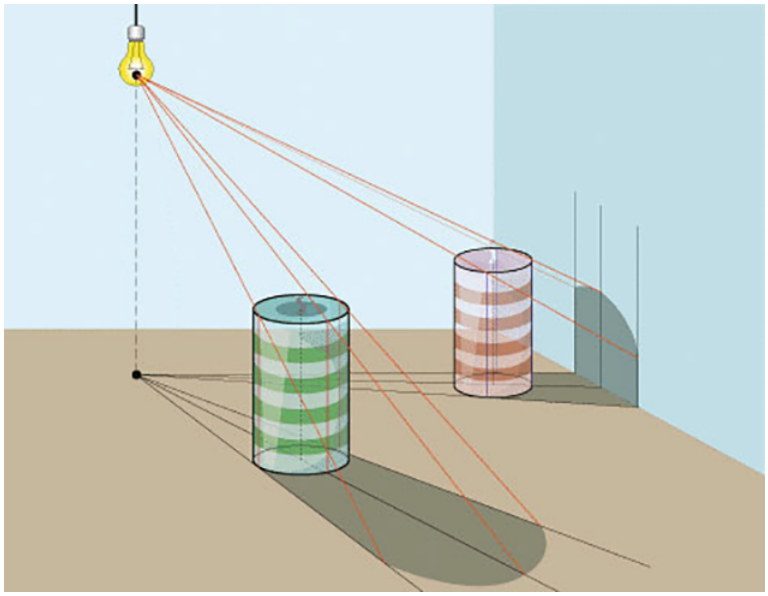

Fig. 1. The method of recovering the coordinates of light sources on the basis of shadows cast by objects.

parameters in complex scenes with multiple light sources.

3. METHOD

In this paper, we propose the method that determines the scene lighting sources on the basis of its RGBD image. This RGBD image makes it possible to calculate the coordinates of shadow boundaries and scene objects casting these shadows. The method is based on forming and analyzing light rays connecting the points on the shadow boundary with the object. The main point on which the proposed method relies is that, for each point on the shadow boundary, there exists a point of the object such that the line connecting it with the shadow point hits the light source. Therefore, the region of the maximum concentration of rays connecting the points of shadow boundaries with points on the object boundaries contains the light source. Moreover, the size of the ray concentration region can be used to estimate the size of the light source. This method is able to deal both with the scenes in which the light source is outside of the field of view and with the scenes containing multiple light sources, including extended light sources. It is clear that this method can only detects the light sources from which scene objects cast discernible shadows.

Note that the proposed method works with images and depth maps rather than with 3D scene models. Such images can be obtained using special devices, e.g. scanners or LIDARs, which are able to determine the distance to any image point and do it much faster than a full 3D model can be created.

Figure 1 illustrates the proposed method of recovering the coordinates of light sources given the coordinates of objects and their shadows.

The block diagram of the algorithm implementing the proposed method is shown in Fig. 2. It involves the following steps

1. Getting data from the MR devices—depth map and the corresponding image of the visible part of the scene.

2. Using a convolutional neural network, determine all shadow regions in the image and determine their boundaries.

3. Use the spatial filter and the Canny filtering algorithm in the ROI to detect the shadow casting objects and assign different colors to them. Detect the boundaries of objects.

4. Save the coordinates of boundaries of objects and shadows.

5. Form beams of rays. The beams of rays are emitted through the coordinates of shadow boundary points to the points on the object boundaries. The object boundary is fired from the points on the shadow boundary spaced five pixels apart

6. Find the region in which the rays from different points on the shadow boundary intersect (ray caustics). We assume that the light source can be in the region that contains the intersection of at least three rays.

The implementation of this method is graphically illustrated in Fig. 3; here the rays beginning at different shadow boundary points and passing through the same point on the object boundary have the same color, and the dot shows the coordinates with the

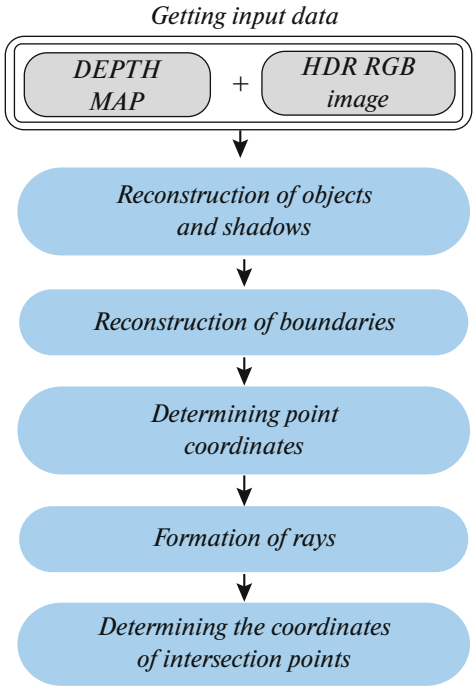

Fig. 2. The algorithm implementing the method of recovering the coordinates of light sources.

greatest caustic density, i.e., the most probable location of the light source.

To find the coordinates of the object points, we use the ROI approach relative to the shadow image. If we assume that the object touches its shadow (which is the most widespread case), then the algorithm processes these ROIs using the Canny filter and subtracts the known coordinates of shadows to obtain the coordinates of the object. The Canny filter also includes the Gaussian smoother for finding the gradients and removing the noise.

The use of the approach just described makes it possible to separate the shadow cast by an object from the object itself, and the use of the NumPy library makes it possible to obtain the coordinates of all nonzero pixels separately for the shadow and the object.

4. IMPLEMENTATION

The proposed method uses the fully convolutional neural network, which allows to detect in the original scene image objects and shadows cast by them, and it also uses algorithms that can restore the light sources by analyzing 2.5D images of objects and their shadows. As the training data set, we used the set SBU_Shadow [25], which consists of original images and their masks in which the shadows are marked by white and the unshaded regions are black.

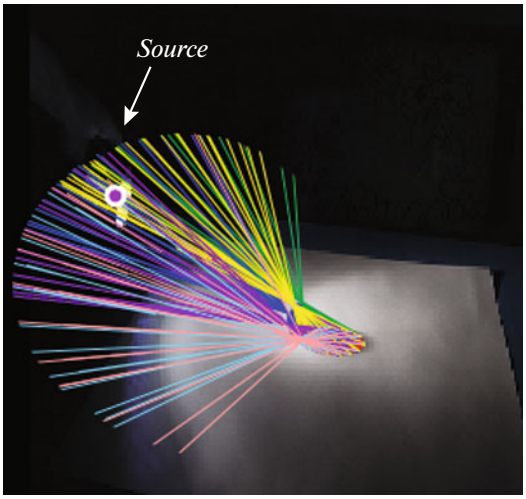

Fig. 3. Illustration of the method of restoring the coordinates of light sources.

We used the U-Net architecture of the fully convolutional network because it is best suited for dealing with the binary classification.

This architecture includes four blocks of downsample layers for classification learning, four blocks of upsample layers for obtaining the output array of the same dimension as at the input, and bottleneck blocks with 256 values of the depth map for deeper network learning.

The U-NET architecture is shown in Fig. 4.

An example of a pair of images from this set is shown in Fig. 5.

The training data set consists of 4085 pairs of images, and the test set consists of 638 images. Only six epochs were needed for the neural network to converge, which took about 3 minutes on GeForce 1080Ti. The speed of the learned neural network is 28 ms.

We used sigmoid as the layer activation function because it is nonlinear in nature, and any combination of such functions is nonlinear as well. Another advantage of sigmoid is its smoothness and, in distinction from the step function, it makes it possible to use analog activation. Moreover, the sigmoid function has a smooth gradient. Thus, sigmoid seems to be well suited for classification problems. It tries to guide values to one of the curve sides (e.g., to the upper limit at $x = 1$ and to the lower limit at $x = -1$). Such a behavior helps find clear boundaries in predictions. A drawback of this function is that, in the vicinity of the sigmoid endpoints, the values of Y weakly respond to variations of X. This means that the gradient in these regions is small, which, in turn, involves difficulties with the vanishing gradient.

Figure 6 gives a graphical representation of the history of the neural network learning, where the epochs are plotted on the horizontal axis and the decrease of

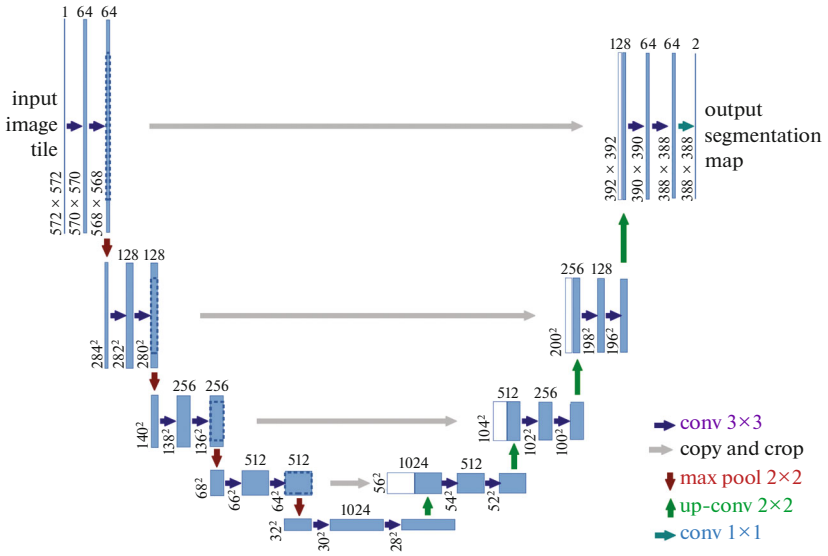

Fig. 4. U-Net architecture of the neural network.

losses is plotted on the vertical axis. This graph shows both the training data (loss) and the validation data (val_loss). The task of the neural network is to make the error of the objective function as small as possible, which is achieved at epoch 5 with the values of loss = 0.2093 and val_loss = 0.2179. In this work, we used binary_crossentropy as the error function, and RMSprop with the values lr = 1e–4 and decay = 1e–6 was used as the optimization (weight updating) function.

The accuracy of classification on validation data was 92%, and on the test data it was 94% in the IoU (intersection over union) metric. It is seen in Fig. 7 that the results produced by the neural network are close to the ground truth. The images on the left in Fig. 7 are the original images fed to the input of the neural network, the ground truth images are shown on the right, and the predictions of the neural network are shown in the middle.

After the regions containing shadows have been determined, the Canny operator is applied to the ROI of this domain. Using the coordinates of the region containing the shadow, all interior and exterior pixels are removed, and only the contours of the object remain.

The result produced by the Canny operator is shown in Fig. 8.

Given the coordinates of the object and its shadow, we emit rays connecting the points on the shadow contour with the points on the object contour. This algorithm was outlined in Section 2 of this paper. The function that stores the coordinates of all emitted rays is implemented as a NumPy array. The library NumPy

Fig. 5. Example of a pair of images used to learn the neural network.

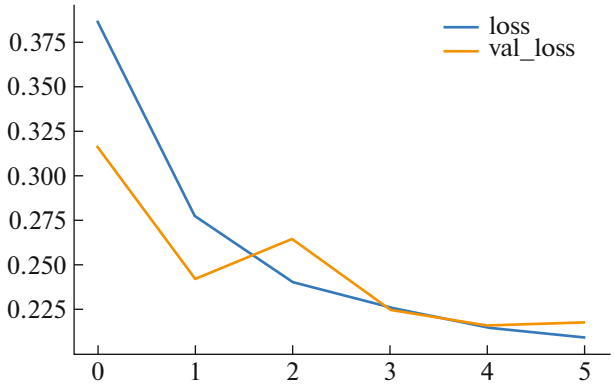

Fig. 6. History of the neural network learning.

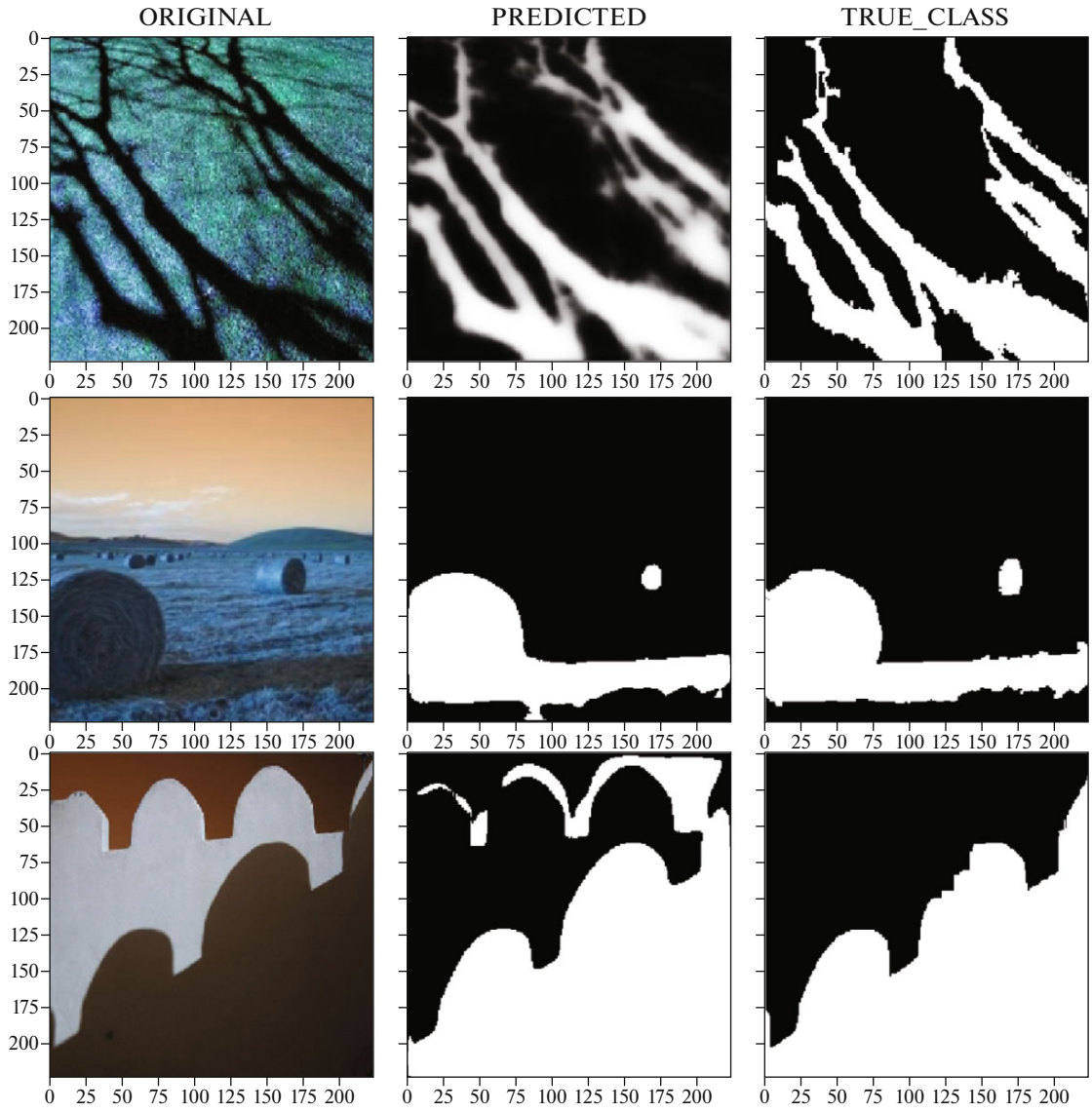

Fig. 7. Results produced by neural network.

was chosen because it provides convenient and efficient means for dealing with arrays. Next, for the groups of rays emitted from different shadow boundary points, the regions of caustics with the maximum concentration of rays are found. These regions contain the light sources that illuminate the scene.

The proposed method makes it possible to determine the light sources in a specified system of coordinates. In a simple case, the system of coordinates is formed relative to the current image, i.e., one axis is directed to the image center and the two others are directed along the image axes. The use of this method in combination with a 3D scanner or an MR headset that determines its location in space makes it possible to determine the coordinates of the light sources in the space of the scene.

Figure 9 shows examples of the algorithm operation for five real-life images. From left to right, there are the original images and intermediate results produced by the algorithms determining the coordinates of objects and their shadows in the original images. This figure contains the original images, the images processed by the Canny filter, the images with shadows, and the images with objects. Having all these images at our disposal, we can determine the coordinates of points on the objects and their shadows, then construct rays connecting the shadow and object boundaries, and determine the light sources. The scenes are numbered from the first one on top to the fifth on the bottom.

Figure 10 shows the results obtained using the proposed algorithm for determining the coordinates of light sources by analyzing the ray traces.

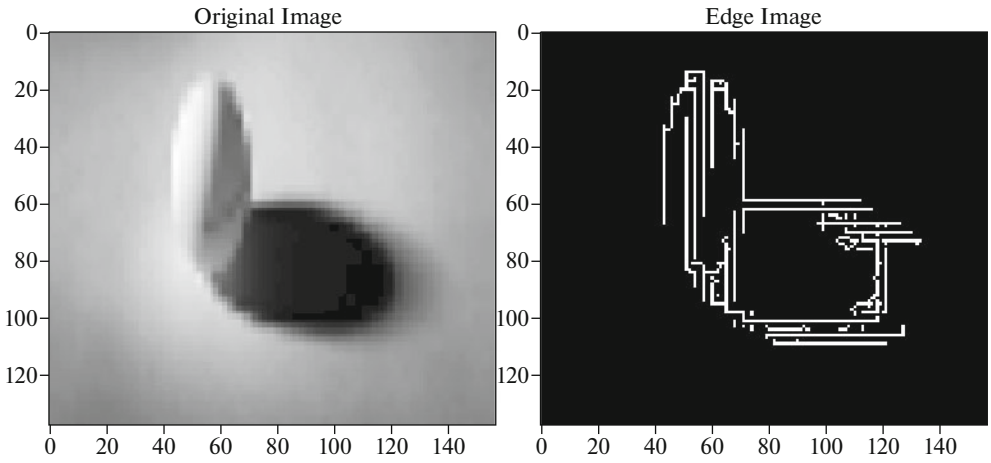

Fig. 8. Result produced by the Canny operator for contour enhancement.

The result of the algorithm operation for the third scene is illustrated in Fig. 11. The original image is shown on the left, and the result—the region of ray intersection—is on the right. The absolute error in determining the position of the the light source is 0.109 m (37 screen pixels), the relative error is 13.9%, and the angular error of determining the orientation of the light source relative to the center of the illuminated object is 0.0606 rad.

Table 1 presents the recovered coordinates of the light sources for the five scenes described above. The errors of recovering the light source coordinates are shown in absolute form as the deviation of the center of the original light source from the center of the recovered light source (in meters in the space of the scene and in pixels in the screen space); the difference between the directions from the object center to the original light source and from this center to the recovered light source is given in radians. In addition, the relative error of recovering the light source as the difference between the centers of the original and recovered light sources normalized by the distance from the object center to the original light source is shown on a percentage basis.

The size of the original images is 622 × 415 pixels. The size of ROIs in the images is 224 × 224 pixels, which enables us to feed them at the input of the neural network and classify the shadow regions.

All operations were performed on a computer with a Ryzen7-1700 processor and the graphics controller GTX 1080Ti. The time taken by detecting the shadow on the image was 32 ms, the object detection took 25 ms, and determining the coordinates of the light source center by finding the region of ray intersection took 875 ms.

For this study, we used Python and the libraries OpenCV, Keras, Numpy, and Scikit-learn.

5. CONCLUSIONS

A method for determining the coordinates of the light source centers is proposed; it is shown that it can be suitable for augmented reality systems and that it can determine the coordinates of light sources in the scene-related system of coordinates. The convolutional neural network with the U-Net architecture was used. After training this network, the classification accuracy turned out to be 94%. The architecture of

Table 1. Accuracy of finding the light sources

Scene no.	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

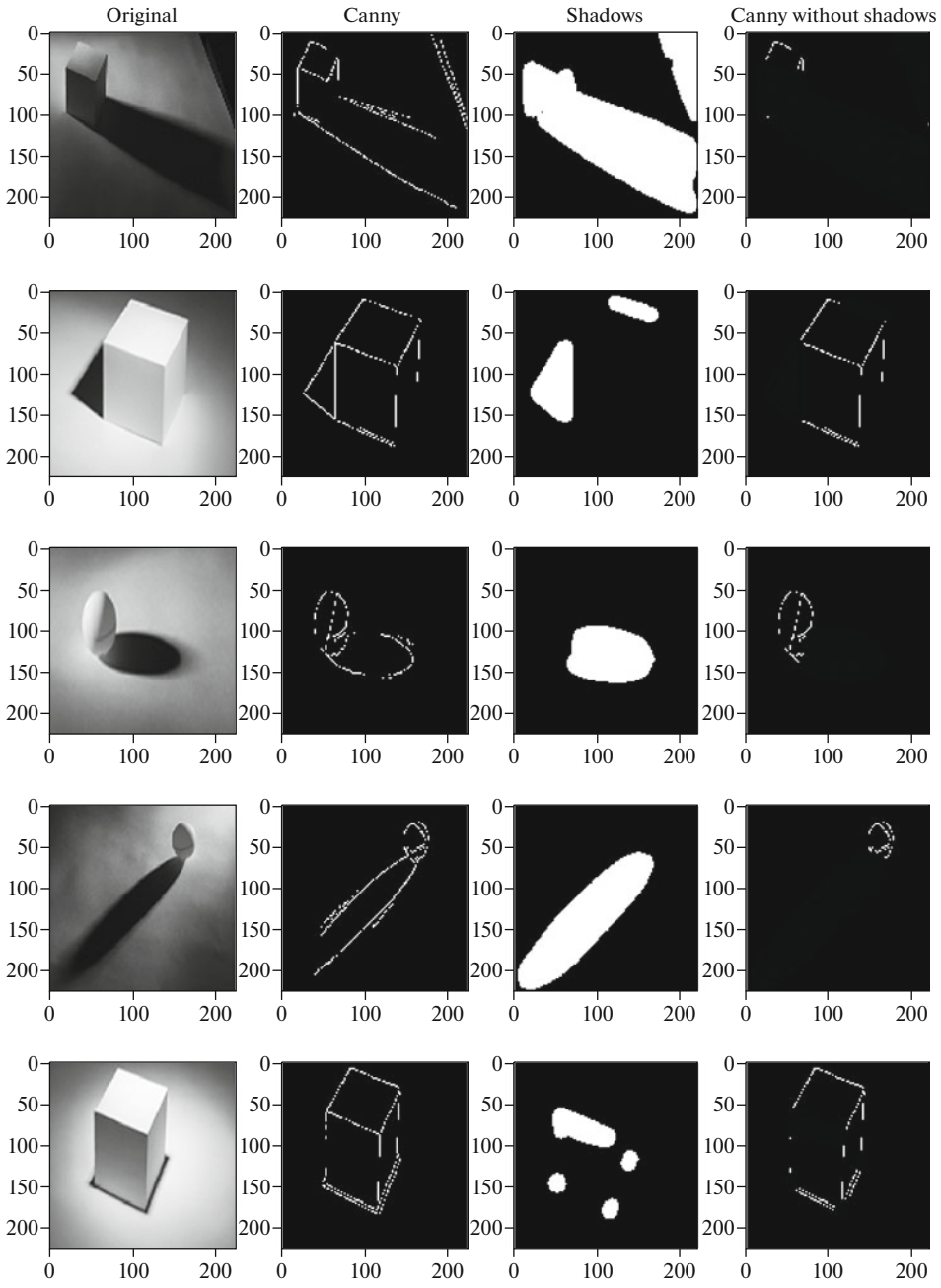

Fig. 9. Examples of determining the coordinates of boundaries of objects and shadows cast by them.

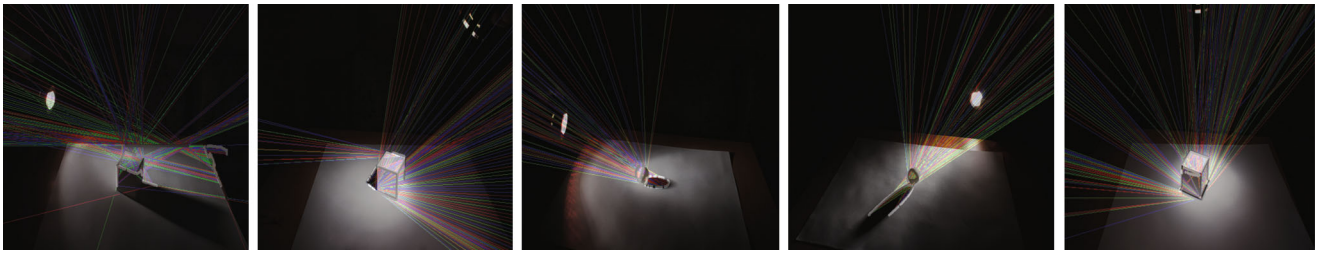

Fig. 10. Results obtained using the proposed algorithm for determining the coordinates of light sources (scenes 1–5 from left to right).

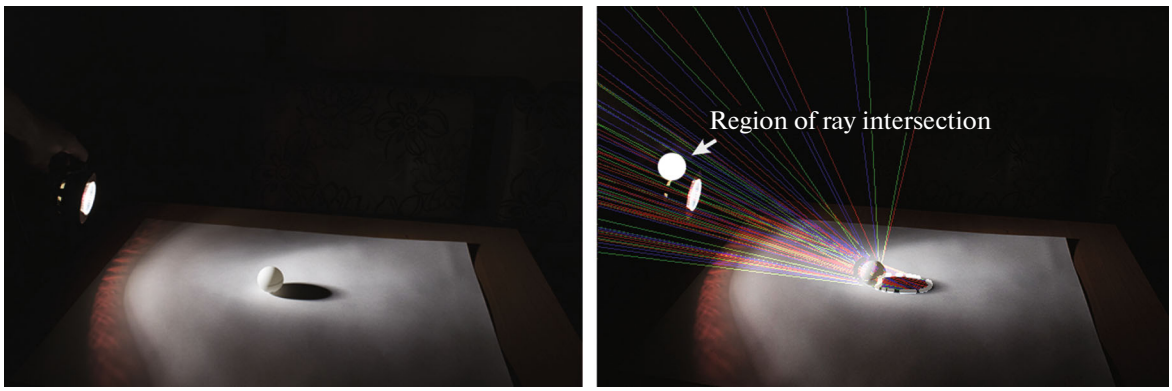

Fig. 11. The region of ray intersection relative to the original light source.

this network is suitable for binary data, and it can recognize complex shadows in images. The performance of the neural network makes it possible to use it in real-time systems, and the speed of recovering the light source parameters by the ray method also will allow to use it in real-time systems after implementing it on a GPU. The coordinates of light source centers were in most cases determined sufficiently accurately for the natural visual perception of the illumination of virtual images by real world light sources. In future, we plan to improve the accuracy of determining the coordinates of light sources, especially in the case of complex illumination, and implement the estimation of the size and shape of extended light sources.

FUNDING

This work was supported by the Russian Science Foundation, project no. 18-79-10190.

REFERENCES

1. Parsons, S. and Cobb, S., State-of-the-art of virtual reality technologies for children on the autism spectrum, *Eur. J. Special Needs Educ.,* 2011, vol. 26, no. 3, pp. 355–366.
2. Palmarini, R., A systematic review of augmented reality applications in maintenance, *Robotics Comput. Integ. Manufact.,* 2018, vol. 49, pp. 215–228.
3. Izadi, S., The reality of mixed reality, *Proc. of the 2016 Symposium on Spatial User Interaction,* 2016, ACM, pp. 1–2.
4. Oculus Rift. https://www.oculus.com. Cited April 10, 2019.
5. Epson Moverio. https://moverio.epson.com. Cited April 10, 2019.
6. Microsoft Hololens. https://www.microsoft.com/en-us/hololens. Cited April 10, 2019.
7. William, R. and Craig, A., *Understanding Virtual Reality: Interface, Application, and Design*, Kaufmann, 2018.
8. Iis, T.P., and Jung, T.H., and Claudia M.D., Embodiment of wearable augmented reality technology in tourism experiences, *J. Travel Res.,* 2018, vol. **57**, no. 5, pp. 597–611.
9. Trout, T., Collaborative mixed reality (MxR) and networked decision making, Next-Generation Analyst VI, Int. Soc. Optics, Photonics, 2018, Vol. 10653.
10. Sheena, B., Anandasabapathy, S., and Shukl, R., Use of augmented reality and virtual reality technologies in endoscopic training, *Clinical Gastroenter. Hepatol.,* 2018, vol. 16, no. 11, pp. 1688–1691.
11. Kiljae, A., Ko, D., and Gim, S., A study on the architecture of mixed reality application for architectural design collaboration, *Int. Conf. on Applied Computing and Information Technology,* Springer, Cham, 2018, pp. 48–61.
12. Livingston, M., Zhuming, A., and Decker, J.W., Human factors for military applications of head-worn augmented reality displays, *Int. Conf. on Applied Human Factors and Ergonomics,* Springer, Cham, 2018, pp. 56–65.
13. Wang, X., Zhdanov, D.D., Potemin I.S., Wang, Y., and Cheng, H., The efficient model to define a single light source position by use of high dynamic range image of 3D scene, *Proc. of SPIE,* 2016, vol. 10020, p. 100200I.
14. Mandl, D., Yi, K.M., Mohr, P., Roth, P.M., Fua, P., Lepetit, V., Schmalstieg, D., and Kalkofen, D., Learning lightprobes for mixed reality illumination, *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR),* IEEE, 2017, pp. 82–89.
15. Supan, P., Stuppacher, I., and Haller, M., Image based shadowing in real-time augmented reality, *Int. J. Virtual Reality,* 2006, vol. 5, no. 3, pp. 1–7.
16. Haller, M., Drab, S., and Hartmann, W., A real-time shadow approach for an augmented reality application using shadow volumes, *Proc. of the ACM symposium on Virtual reality software and technology,* ACM, 2003, pp. 56–65.
17. Everitt, C. and Kilgard, M.J., Practical and robust stenciled shadow volumes hardware-accelerated rendering, arXiv preprint cs/0301002, 2003.
18. Randima, F., and Kilgard, M.J., *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics,* Addison-Wesley, 2003.
19. Kirk, D., *CG Toolkit, User's Manual, Nvidia,* Santa Clara, CA, 2002.

20. Richter-Trummer, T., Instant mixed reality lighting from casual scanning, *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR),* IEEE, 2016, pp. 27–36.

21. Voloboi, A.G., Galaktionov, V.A., Kopylov, E.A., and Shapiro, L.Z., Computation of sunlight determined by a high dynamic range image, *Proc. 16th Int. Conf. on Computer Graphics and Its Applications, GraphiCon'2006,* Novosibirsk, 2006, pp. 467–472.

22. Voloboi, A.G., Galaktionov, V.A., Kopylov, E.A., and Shapiro, L.Z., Simulation of natural daylight illumination determined by a high dynamic range image, *Program. Comput. Software,* 2006, vol. 32, no. 5, pp. 284–292.

23. Valiev, I.V., Voloboi, A.G., and Galaktionov, V.A., A physically correct sunlight model specified by high dynamic range images, *Vestn. Komput. Inform. Tekhnol.,* 2009, no. 9, pp. 10–17.

24. Jiddi, S., Robert, P., and Marchand, E., Illumination estimation using cast shadows for realistic augmented reality applications, *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR-Adjunct),* 2017, Nantes, France, 2017.

25. Vicente, T.F.Y., Hou, L., Yu, C.-P., Hoai, M., and Samaras, D., Large-scale training of shadow detectors with noisily annotated shadow examples, *Proc. of the European Conference on Computer Vision,* 2016.

Translated by A. Klimontovich