Journal of Real-Time Image Processing (2021) 18:271–281
https://doi.org/10.1007/s11554-020-01022-6

SPECIAL ISSUE PAPER

Detection of real-time augmented reality scene light sources and construction of photorealis tic rendering framework

Taile Ni[1] · Yingshuang Chen[1] · Shoupeng Liu[2] · Jinglong Wu[3]

Received: 22 March 2020 / Accepted: 18 September 2020 / Published online: 30 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

In this paper, the main network of multi-channel light sources is improved, so that multi-channel pictures can be fused for joint training. Secondly, for high-resolution detection pictures, the huge memory consumption leads to a reduction in batches and then affects the model distribution. Group regularization is adopted. We can still train the model normally in small batches; then, combined with the method of the regional candidate network, the final detection accuracy and the accuracy of the candidate frame regression are improved. Finally, through in-depth analysis, based on image lighting technology and physical-based rendering theory, the requirements for lighting effects and performance limitations, combined with a variety of image enhancement technologies, such as gamma correction, HDR, and these technologies used in Java. Real-time lighting algorithms that currently run efficiently on mainstream PCs. The algorithm can be well integrated into the existing rasterization rendering pipeline, while into account better lighting effects and higher operating efficiency. Finally, the lighting effects achieved by the algorithm are tested and compared through experiments. This algorithm not only achieves a very good light and shadow effect when rendering virtual objects with a real scene as the background but also can meet the realistic rendering of picture frames in more complex scenes. Rate requirements. The experimental results show that the virtual light source automatically generated by this algorithm can approximate the lighting of the real scene, and the virtual object and the real object can produce approximately consistent lighting effects in an augmented reality environment with one or more real light sources.

Keywords Real-time enhancement · Realistic scene · Light source detection · Photorealistic rendering · Real-time augmented reality

1 Introduction

Augmented reality is an emerging field of interdisciplinary research. It is a technology that uses computers, sensors, displays, and other devices to enhance or expand the real-world additional information seen by users. Augmented reality can realize the blending of virtual and real in the image or video stream, that is, the virtual information is dynamically superimposed into the real world by sensing and analyzing the objects and environments in the real world [1–3]. Among them, the three-dimensional registration technology for accurately registering virtual information is the basis for achieving a seamless fusion of virtual and real, and directly affects the user experience [4]. In the past two decades, many related theoretical algorithms and hardware devices have emerged in the field of augmented reality [5]. Hardware-based 3D registration mainly relies on the performance of hardware sensors and does not need to calculate

✉ Shoupeng Liu
nremington@sina.com

Taile Ni
nitaile@sina.com

Yingshuang Chen
chendashu88@sina.com

Jinglong Wu
wujinlong6868@sina.com

[1] School of Literature, Journalism and Communication, Xihua University, Chengdu 610000, Sichuan, China

[2] School of Packaging, Zhongshan Torch Polytechnic College, ZhongShan 528436, Guangdong, China

[3] School of Art, Nanjing Normal University Taizhou College, Taizhou 225300, Jiangsu, China

complex algorithms to obtain positioning. Such as the long-term stability system complex inertial navigation system, GPS, and high precision gyroscopes, speed sensors, optical or ultrasonic trackers that are often used in mobile terminals. These hardware technologies have large positioning errors or severe working environment conditions and have limited scope of action. In contrast, the positioning error based on vision 3D registration is small, there are many applicable scenarios, the system structure is simple, and the cost is low. Because of these advantages, vision-based 3D registration has been highly valued and developed [6–8]. Especially with the popularization of mobile terminals and the need for the integration of life scene applications, the technology is increasingly used in AR systems.

Each of the current mainstream methods has its own advantages and disadvantages [9, 10]. When the object has enough texture, the method based on feature point matching shows good results, but when there are a lot of cluttered objects in the scene, it will produce many feature mismatches. The model-based method performs well when dealing with untextured objects, but the effect will be affected when occlusion and lighting conditions change, and the performance will decrease when the object is the background texture [11]. The method based on deep learning can make end-to-end predictions. The accuracy is very high in scenarios the environment of the training set, but it is insufficient in generalization ability. Some methods use random forest classifiers to train with input image blocks or simplified pixel-based features. Although they work well, they rely on manually designed features that are difficult to fine-divide everyday items, and speed slower.

Lowe uses a combination of SFT descriptors and clustered images from similar viewpoints for pose prediction of a single model [12]. Martinez combined SFT features to propose a fast and scalable multi-target registration system for object recognition and pose estimation. In addition to feature point descriptors, sparse features of key point regions can also be learned as descriptions [13]. Lepetit and others used a random forest as a classifier to collect color appearance samples in key point areas to generate a training set, so that all possible appearance sets of each key point of an object are grouped into one class, and key point matching is completed by classifying pixel blocks in the point domain [14]. Prisacari and others performed global probability statistics on the color information of the front background and minimized the posterior probability error pixel by pixel to achieve region segmentation and pose parameter estimation. The improved algorithm enhances the pose optimization strategy, and the local histogram model is used to improve the robustness of the algorithm [15, 16]. However, when the color information of the foreground is close to the background, it is difficult to accurately segment the object and affect the attitude estimation.

In this paper, a self-coding convolutional neural network is constructed to predict the complete six-degree-of-freedom pose [17]. First, the self-encoder is used to reconstruct the target to suppress unfavourable factors such as background, lighting, noise, extract the main features related to the target, classify the viewpoint, and initially predict the rotation component. Secondly, the position and contour of the object in the plane are obtained according to the reconstructed map output from the encoder. The translation component is predicted using the proportional relationship of the camera imaging principle, and the complete rotation component is predicted using the offset angle of the bounding box. According to the requirements of photorealistic rendering, based on the existing hardware foundation, a reasonable and efficient lighting algorithm is designed and applied to photorealistic rendering.

2 Optimization framework construction

2.1 Detection area training parameter extraction

First zoom the picture four times to get three pictures including the original picture, and then train an RPN network at the front of the entire network to simply distinguish positive and negative samples, and then select some fixed-size ROIS, such as $512\times512\times40$ to ensure that these ROIS contain as many candidate frames as possible [18–20]. At the same time, many ROIS generated in the RPN are removed from large and small boxes to ensure that the scales are approximately the same. At the same time, all these ROIS must include all candidate boxes without omissions. These uniformly sized ROIS are then sent to the detection network. After the pictures of all sizes are detected, the test results are restored to the pictures to form a result [21].

As shown in Fig. 1, the images of three scales obtained a large number of ROIS after passing through the RPN network, and then continued filtering to delete incomplete candidate frames and candidate frames, that are too large or too small to obtain all the set of ROIS for the candidate box. Ensure that all targets can be trained without wasting data.

From Table 1, it can be clearly found that the effect of the SNIPER structure on the baseline produces a more obvious effect, and the more input picture scales, the better the effect, but as more and more pictures, the increasing effect tends to be saturated. Obviously, when the scale of the input picture is enough, the picture entering the main network has almost covered all the candidate boxes. At the same time, each candidate box can be enlarged to the same value. Even if the number of scales continues to increase, the target cannot be reduced the scale is different, so the later the scale, the smaller the scale effect is.

Fig. 1 Multi-scale training ROI generation method

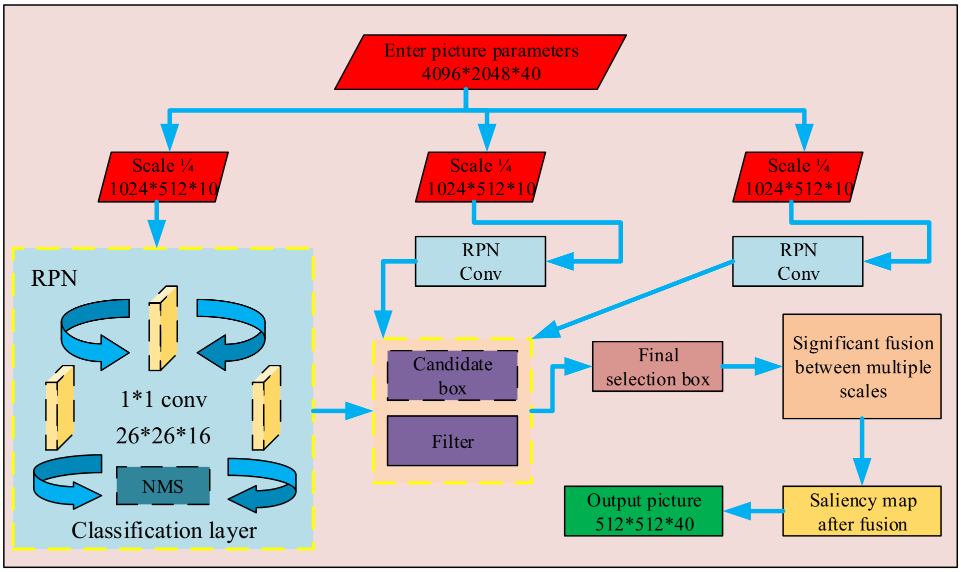
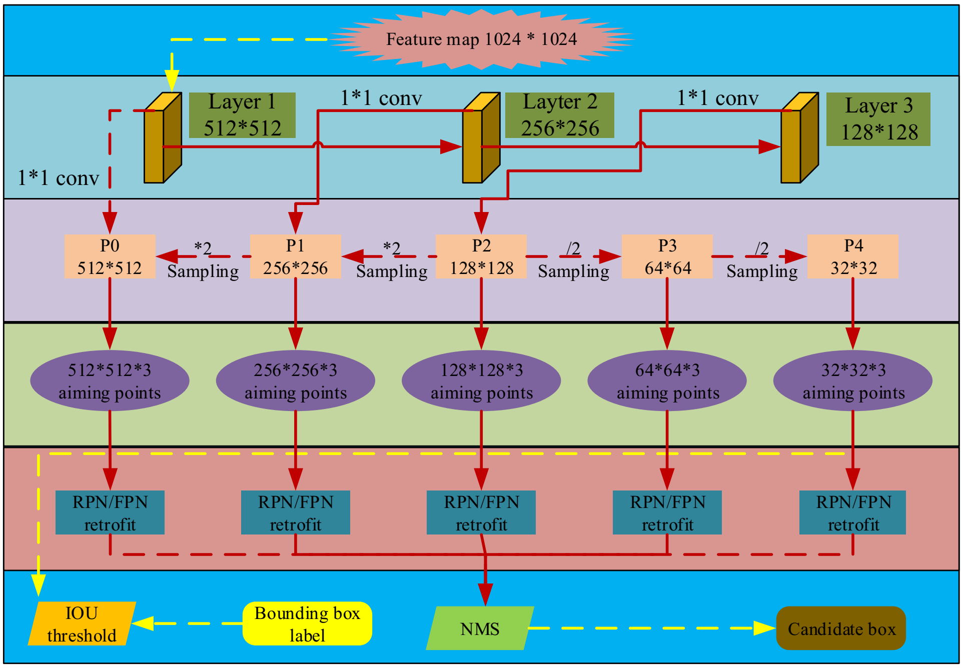


Table 1 Impact of data on SNIPER structure on test results

Method	mAP25	AP45	AP65
Faster	0.7526	0.8921	0.8627
SNIPER + faster (1 scale)	0.7918	0.9184	0.9054
SNIPER + faster (3 scale)	0.8129	0.9408	0.9128
SNIPER + faster (5 scale)	0.8271	0.9556	0.915
SNIPER + faster (7 scale)	0.8226	0.9572	0.9171

As shown in Fig. 2, a feature map is output for operations from bottom to top, top to bottom, and horizontal connection. However, the difference is that in this part, instead of extracting only the features of the last layer, all the intermediate feature maps of all the right branches are extracted, and then these feature maps are used for RPN operations. It is obvious that the ratio of each feature map to the original image after this operation is different. Similarly, the receptive field of each anchor point on the feature map is also different, which corresponds to the size of the candidate frame. So extracting candidate frames on these feature maps does not need to select three areas in addition to the three aspect

Fig. 2 FPN in RPN



ratios like the Faster R-CNN mentioned above. Each RPN part in this network only needs to select 3 candidate frames of aspect ratio, even if there are feature maps of five scales, there are only 15 candidate frames of size. Not much more computation than the original RPN. In these feature maps, $(512 \times 512 + 256 \times 256 + 128 \times 128 + 64 \times 64 + 32 \times 32) \times 3$ candidate frames can be extracted in total.

The lighting effect is not only determined by the light source, the fineness of the object model also plays a large role. To make the rendered scene look more realistic, it is necessary to use a physical-based rendering method to give the surface material of the object more finer parameters. This article chooses to use the following four parameters to control.

The reason that the method based on feature point matching shows good results is mainly due to the advantages of three aspects: small calculation amount, good robustness, and complex geometric shadow insensitivity.

Albedo The albedo map assigns a color or basic reflectance to each Texel pixel on the surface of an object. It represents the basic color of the object. It uses a texture to store the color RGB vectors.

Normal A normal map is a special texture that stores the normal vector of the object's local coordinate system (tangent space), which is expressed in RGB colors. In the traditional lighting model, the normal of each segment are obtained by interpolating the vertices of the triangles, so that the surface of the object will appear flat when calculating lighting, without levels and details. In actual life, the surface of the object is often uneven, because the normal arrangement of the surface of the object is not consistent. Normal mapping technology can give each segment a unique normal, which greatly enhances the surface details and enhances the bump feeling. Normal maps are generally generated by mapping the surface normal from the high-precision models and storing them in the maps. This eliminates the need to calculate triangle meshes of several orders of magnitude like high-precision models, which can greatly improve rendering efficiency.

Coarseness The Roughness map can specify the roughness of the surface for each Texel. It is used to control the normal distribution function and geometric occlusion function of the BRDF, so that the specular reflection range of the rough surface is larger but blurred, while the specular reflection of the smooth surface appears concentrated and sharp. It stores a floating-point value ranging from 0.0000 to 1.0000.

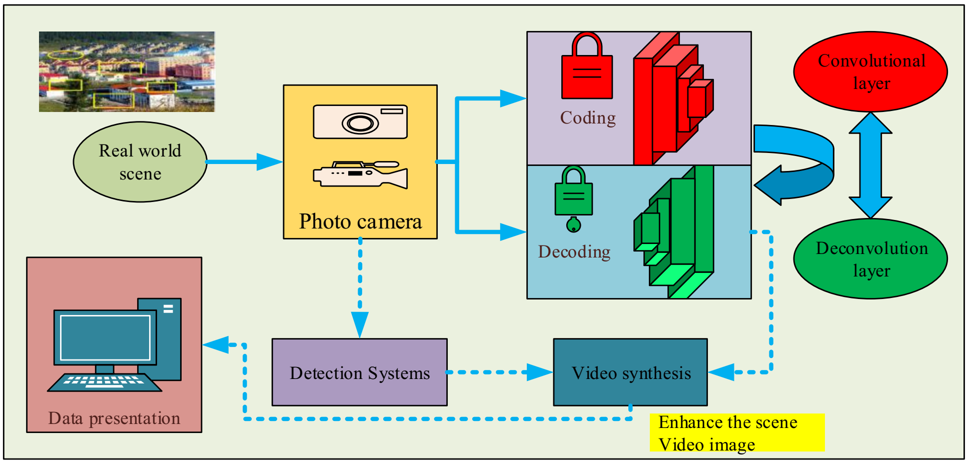
We set the light occlusion factor for dark areas on the surface of the object through the "Ambient Occlusion" map. For example, on the surface of a brick, the crack of the brick on the albedo map does not contain any shadow information, and the AO map can specify the crack, because the light in this area is easily blocked, which can significantly improve the reality on site. The floating-point values stored in the map range from 0.0000 to 1.000.

2.2 Detection algorithm of light source in real-time augmented reality scene

The encoder and decoder included in the reconstructed self-encoder are composed of a convolution layer and a deconvolution layer, respectively. The network structure is shown in Fig. 3. The convolution layer is responsible for extracting multi-scale feature maps from top to bottom to complete the goal. The dimension reduction indicates that the deconvolution layer improves the resolution of the feature map to restore the target size [22]. To the general network structure, we also stack the convolutional layer / deconvolution layer and the RELU activation layer to form the entire network. The training goal of the reconstructed autoencoder is to reproduce the input samples, and the loss of each sample is simply expressed as the average Euclidean distance of the pixels.

The training goal of the reconstructed autoencoder is to reproduce the input samples, and the loss of each sample is

Fig. 3 Network structure



simply expressed as the average Euclidean distance of the pixels:

$$D_{\mathrm{p}} = \sum\nolimits_{m \in D} \left\| x_m - \overline{x_m} \right\|_p. \tag{1}$$

Add random noise to the input image for enhancement, while the reconstruction target remains intact. The autoencoder pays attention to the target object and suppresses the influence of background, lighting, occlusion and other factors. We implement a random enhancement function $f_{\mathrm{aug}m}$ (*) for the input x, and the reconstruction target can be expressed as

$$\overline{x_m} = \left(\varphi * \phi * f_{\mathrm{aug}m} \right)(x). \tag{2}$$

The encoder extracts the features of the shape of the object. To output the prediction of the object's pose, we implement the pose classification immediately after the encoder with the fully connected layer. Considering the non-linear mathematical relationship of the pose estimation problem in the geometric mapping, two layers of full connections are used, the dimensions of which are 1024 and N, and N is the number of sampling viewpoints. The feature vector of the feature extraction network is further reduced in dimension, and the probability distribution vector of the sampling viewpoint is output. Generally, in a classification network, the fully connected layer followed by the SOMTEX layer turns the output of the neuron into a probability distribution vector. We use a cross-entropy loss function to determine the classification category:

$$d_{r-\mathrm{cla}} = -\sum\nolimits_{m}^{N} y_m \log s_m. \tag{3}$$

The network is roughly divided into two subjects, one is an autoencoder G that is responsible for generating the image, and the other is a full convolutional network D that is responsible for judging the authenticity of the data. The CGAN-based discriminant network has two inputs, one is the real image and the original image $\{x, y\}$, and the other is the generated image and the original image $\{x, G(x)\}$. The training cost function is

$$\min_{G} \max_{D} l_{\mathrm{adv}} = \left(M_{x,y}[\log D(x,y)] \right) \\ + M_x(1 - \log D(x,y)). \tag{4}$$

Define multi-task objective function for pose classification and target segmentation:

$$N = \alpha L_p + \beta L_p + \gamma L_p. \tag{5}$$

Generally, $\alpha = 1$, $\beta = 1$, and $\gamma = 1$. For pose estimation, pose classification is based on the reconstruction of the target model. Adversarial training has a good reconstruction effect on objects with complex textures. It helps to locate objects in the early stage and accelerates the convergence speed of the autoencoder.

Define the random affine transformation function f_{aff} (*) and random enhancement function $f_{\mathrm{aug}n}$ (*), then the input x and the reconstruction target are expressed as

$$X = f_{\mathrm{aff}}(x_{\mathrm{view}}) \tag{6}$$

$$\overline{x_n} = \left(\varphi * \phi * f_{\mathrm{aug}n} \right)(x). \tag{7}$$

2.3 Photorealistic rendering algorithm

The algorithm mainly contains two core points, there are voxel and cone tracking [23]. The first is the voxel, which is the pixel concept. It divides the three-dimensional space into unit cubes, and each cube has information such as position, normal, and material. Traditional scenes are represented using triangle primitives as the basic unit, while voxels are used as basic primitives in the new scene. Using voxels can greatly simplify the calculation of the intersection of light and objects. The process of turning triangle primitives into voxels is called voxelization.

After vowelizing the model, you can obtain the scene information stored in the three-dimensional texture, that is, the leaf nodes of the octree. An octree is a tree structure that is extended from the root node. Generally, an octree is created from the bottom up, and the structure of the entire tree is obtained by recursively merging the leaf nodes. The adjacent eight voxels are merged into a cube, and iteratively iterates until the top root node ends. Using sparse octrees has the following advantages. The first is the computational complexity of global illumination is independent of the complexity of the scene. The second is that you can avoid storing empty areas in the scene, which greatly saves memory consumption. Finally, with the hierarchical structure, the traversal speed will be much faster.

The basic process of real-time rendering is as follows (a) Set the rendering state required to render the current model, (b) Set the vertex data of the current model, (c) Set the texture and texture data of the current model, (d) Render the current model, (e) Switch to step (a) and repeat the rendering, Until all models are rendered.

On the CPU side, there are mainly rendering state management functions, and on the GPU side, various functions are mainly used to facilitate the calculation of lighting in the fragment shader [24, 25]. Sender State function: Enter Boolean parameters to control the state of OpenGL when rendering, such as depth test, template test, clear color, and blend on and off. Fresnel equation: It inputs an included angle cosine and roughness coefficient, and calculates the ratio of reflected to refracted light in the incident light according to the formula. It should return a vector that represents the reflectance of the three colors of the RGB.

During training, we first need to randomly select CL and CR from the captured image sequence as the input to the network. Then select another intermediate image Ct for supervised learning ($0 \leq L < t < R \leq N$). Since the image sequence is captured by a camera that moves uniformly around the object, the blending coefficient α can be determined in the following ways:

$$\alpha = \frac{T - L}{R - L} \tag{8}$$

Assuming that the depth features in the last six encoder layers are represented as F_{Lk} and F_{Rk}, the hybrid depth features used for decoding should be written as

$$F_{\mathrm{k}}^{\mathrm{b}} = (1 - \alpha)F_{\mathrm{k}}^{\mathrm{l}} + \alpha F_{\mathrm{k}}^{\mathrm{r}}. \tag{9}$$

The proposed network is trained in a supervised manner, including mask loss L_{m}, attenuation loss L_{a}, and refraction flow loss L_{b}. To improve the quality of the composite image in a new perspective, we added composition loss L_{c} and perception loss L_{d} to achieve this. Therefore, train the network by minimizing the loss function as follows:

$$L = w_{\mathrm{m}}L_{\mathrm{m}} + w_{\mathrm{a}}L_{\mathrm{a}} + w_{\mathrm{b}}L_{\mathrm{b}} + w_{\mathrm{c}}L_{\mathrm{c}} + w_{\mathrm{d}}L_{\mathrm{d}}, \tag{10}$$

where ω represents the equilibrium weight of the corresponding loss term. We use an additional SOFTMAX layer to normalize the output and use the BCE (binary cross entropy) function to calculate the loss as follows:

$$L_{\mathrm{m}} = \frac{1}{HW}\sum_{ij}(M_{ij} - m_{ij})^2, \tag{11}$$

where H and W represent the height and width of the input image, and M_{ij} and P_{ij} represent the pixel values of the true binary mask and normalized output at position (i, j), respectively. We use the MSE (mean square error) function to measure this loss:

$$L_{\mathrm{a}} = \frac{1}{HW}\sum_{ij}(P_{ij} - \overline{P_{ij}})^2(M_{ij} - 1)M_{ij}. \tag{12}$$

We normalize the output by an activation function, and then scale the output value using the size of the input image. Using the average endpoint error (EPE) function, this loss function is expressed as

$$L_{\mathrm{b}} = \frac{1}{HW}\sum_{mn}(P_{mm} - \overline{P_{mn}})^2. \tag{13}$$

To minimize the difference between the reconstructed image and the real image, the L_{c} function metric is used as follows:

$$L_{\mathrm{c}} = \frac{1}{HW}\sum_{am}(P_{am} - \overline{P_{am}})^2. \tag{14}$$

The introduced perceptual loss can better preserve the details and reduce the blur, while increasing the clarity of the reconstructed image, as follows:

$$L_{\mathrm{d}} = \frac{1}{HW}\sum_{bm}(P_{bm} - \overline{P_{bm}})^2. \tag{15}$$

3 Framework analysis

3.1 Detection algorithm analysis

To the method in this paper, the algorithm proposed by LINE2D is also predicted on a single RGB image. Figure 4 shows the comparison results with LINE2D on the Line mode data. LINE2D is able to detect objects relatively well (see 2D.Bonding.Box) but cannot reliably estimate the correct pose. Without depth information, it mainly relies on gradient features on the contour of the object, which makes it very difficult to estimate the rotation accurately. It can be seen from Fig. 4 that the method based on deep learning networks has shown excellent potential and achieved good results in various aspects.

Through Fig. 5, we show the comparison of the tracking accuracy of the algorithm on BUNQ, CAT and DUCK sequences. Finally, it is found that there is interference of complex background in the video sequence. Since the previous research will not filter the trusted edges, when the contour points match, it is inevitable that the false edge points in the front background will not match, resulting in inaccurate tracking optimization results. The improved algorithm in this paper removes false edges near the edge

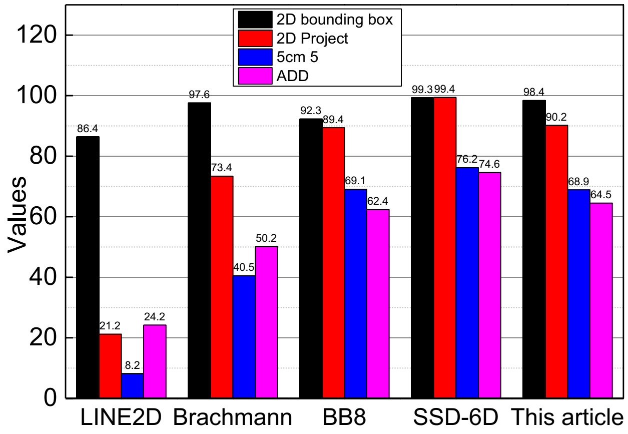


Fig. 4 Comparison table of average errors of different methods

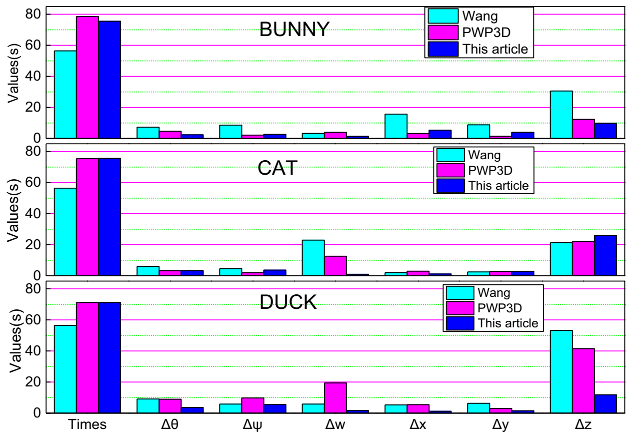


Fig. 5 Video frame running time and pose parameter error analysis

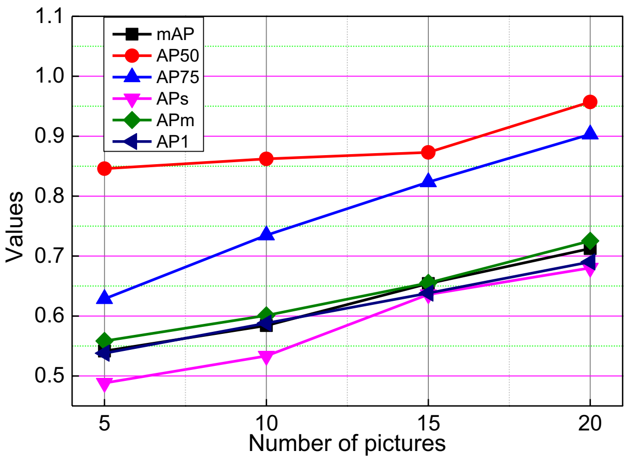


Fig. 6 Improved results with composite lighting

of the contour, avoids mismatch problems, and can achieve accurate tracking in most test scenarios. Due to the interference of the same color area in the front background, PWP3D is prone to drift.

We found that the impact of this sample collection method on improving experimental results is huge. The experiment will select some pictures from the data set according to a fixed index order to ensure that the light source conditions are consistent, and then divide it into a training set and a test set. And then test the test set to get the results in Fig. 6, you can see that each index will make great progress when increasing the number of pictures, it is clear that the data collection method can greatly improve my method The effect, the resulting improvement is even greater than the model improvement. Good data is heavier in deep learning methods than well-designed algorithms.

Table 2 Scene complexity

Testing scenarios	Number of vertices to test	Test the number of quads
Scenes	Number of triangles (K)	Number of vertices (K)
A1	102	312
A2	240	652
A3	324	546
A4	425	737

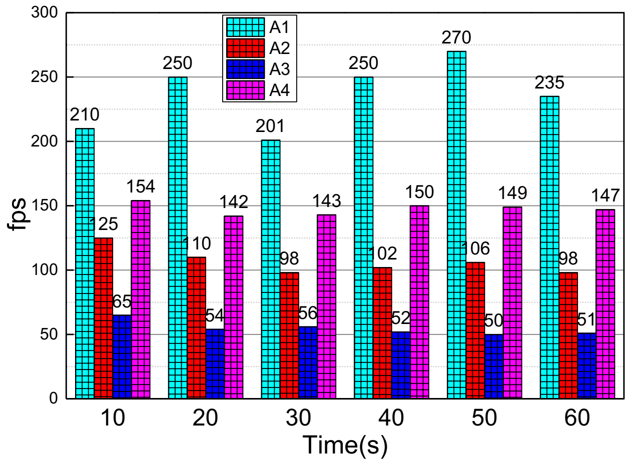


Fig. 7 Frame rate chart of different test scenarios

3.2 Analysis of rendering algorithms

Test the performance and efficiency of the algorithm in this paper. The number of vertices and the number of quadrangles represent the order of magnitude of the objects drawn in the scene, as shown in Table 2. Because the performance requirements of the calculation in this article are not too high, the test graphics card is Intel HD630. A vertex represents the high symmetry point of each figure. The following table records the frame rate performance within 120 s of continuously rendering the scene under different loads. When the model is more detailed, the frame rate can still be maintained above 50 fps. It fully meets the requirements of real-time Yunnan dyeing, and can also achieve very good lighting effects, which fully proves the efficiency of the algorithm in this paper.

The experiment showed all the lighting phenomena, as shown in Fig. 7, including the specular reflection of metal dragons, the diffuse reflection of red plastic dragons, the refraction effect of glass dragons, and the specular reflection effect of balls. Objects of different materials can be well integrated with the scene in the same scene, confirming once again the rationality of the algorithm in this paper, which can indeed bring better global illumination.

The effects of different refractive indices were tested. As shown in Fig. 8, higher refractive indices will cause greater reconstruction errors. The main reason is that the higher refractive index results in a wider range of radiation after refracted light, which is more difficult to capture, resulting in a reduction in reconstruction accuracy.

An intrusive method was used to reconstruct the real model. As shown in Fig. 9, the object was sprayed with DPT-5 developer and scanned and reconstructed using a high-end industrial scanner. Then iterative nearest point algorithm is used to align the initial model and the scan reconstruction model, and the distance between the reconstruction model and the scan reconstruction model is also used as a quantitative index for evaluation. The convergence speed of the real model during reconstruction is the simulation model. Although there are still some errors in our final reconstruction results, compared to the initial model (visual convex hull), it has been improved by 26%. The test curve shows that our method can significantly reduce the reconstruction error in about 20 iterations.

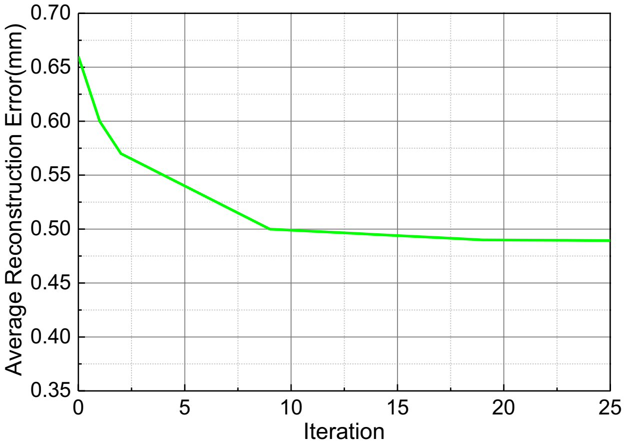


Fig. 9 Average reconstruction error at each step during the iteration

3.3 Overall framework analysis

The algorithm designed in this paper can achieve very good lighting effects in real-time rendering. The implementation process of the ambient lighting part was further improved to make it have better performance on low-end devices. The first part is the diffuse reflection irradiance map. The size of the original algorithm map is 512 × 512. Since the irradiance map stores the average value of the emissivity of the surrounding hemisphere, it is a low-frequency signal in the illumination equation. The pre-calculated cost is changed to a low-resolution (32 × 32) map here, and linear filtering is enabled to make the result smoother. The second part is to reduce the number of samples of pre-filtered environment maps, which can greatly reduce the time required for pre-calculation. As the roughness increases from left to right, reducing the number of samples will bring a lot of imaging to objects with higher roughness. Noise. After the optimization of the above algorithm, the time for processing pre-filtered environment maps is greatly reduced, which speeds up the pre-calculation process (Fig. 10).

Considering the performance and implementation complexity of the scheme, two simulation scenarios are set up. Bias voltage, signal amplitude, and number of active LED chips can all be used to adjust lighting and communication performance. It should be noted here that the spatial-domain dimming control with low complexity fixes the DC offset to alleviate the problem of color shift. In this work, the dynamic range of the LED is set to [0.00, 1.00]. Therefore, the optimal DC offset is set to the midpoint of the dynamic

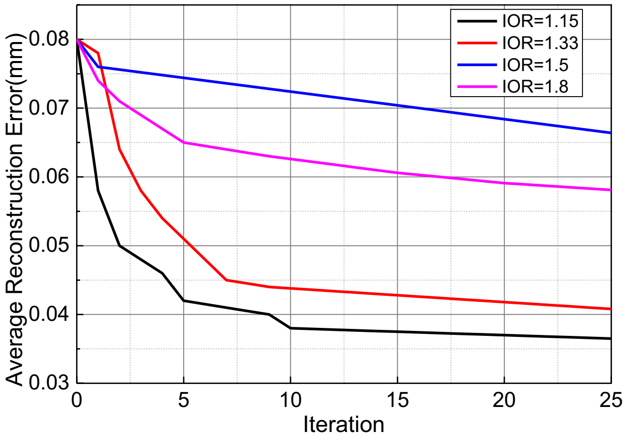


Fig. 8 Experimental results at different refractive indices

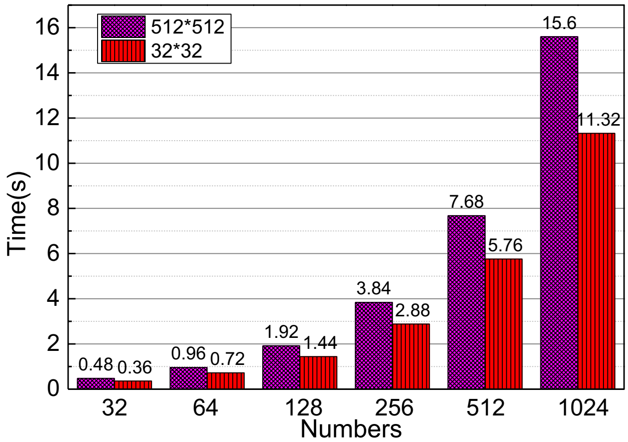


Fig. 10 Algorithm optimization comparison

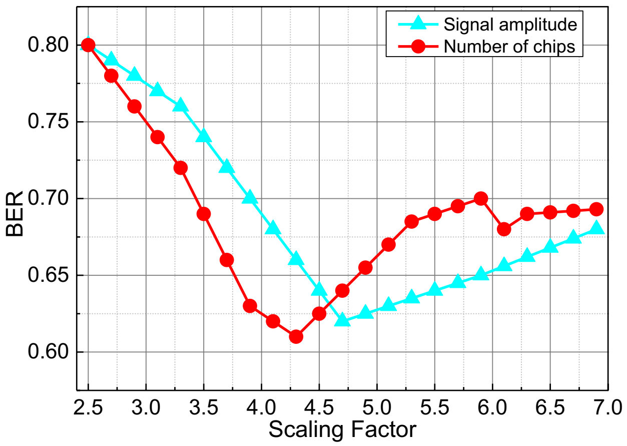


Fig. 11 Comparison of system BER performance under different normalized lighting levels

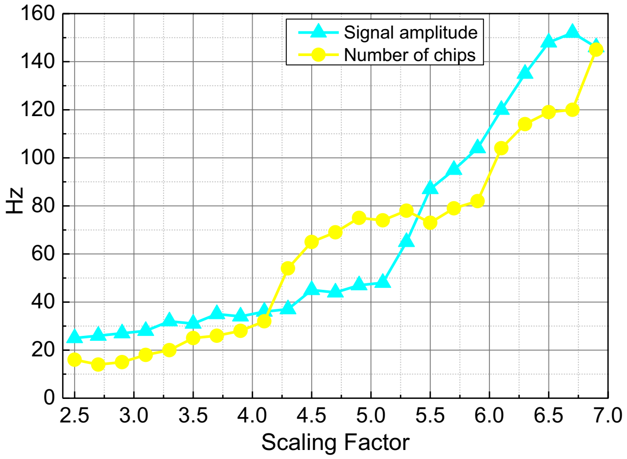


Fig. 12 Comparison of available spectral efficiency at different normalized brightness levels

range to ensure stable lighting and maximize modulation depth. Therefore, there are two parameters in the system (signal amplitude and number of active LED chips) that can be used to adjust different lighting levels. Figure 11 shows a comparison of the system's BER performance at different normalized lighting levels in this framework, and Fig. 12 shows a comparison of the available spectral efficiency at different normalized brightness levels.

4 Conclusion

Aiming at multi-scale target problems in the data set, this article takes the Faster R-CNN network as the basis and adopts three improvement measures on multi-scale problems. First of all, we used the image pyramid to reduce the defect to an appropriate ratio before inputting the image into the network, and regardless of its size, the detection network was relatively easy to handle. This part involves regional candidate network methods. In the step of selecting a suitable ROI, training is performed to make the selection more intelligent; second, the feature pyramid method is used in the feature extraction method, and the idea of the feature pyramid is included in the detection main network and the candidate frame extraction network, so that each feature map. One layer can contain multi-scale information, it is found that the recall rate for small targets is greatly increased through comparative experiments; finally, a correlation constraint method is used in the last layer feature map of the main network to enhance the non-correlation of the feature map, which can be in the first layer feature map Characterizing more information can effectively reduce the occurrence of overfitting. Use image-based lighting technology to calculate the ambient light of the entire scene, including the creation of environmental maps, the convolution generation of the irradiance maps of diffuse reflections, and the pre-filtered environmental maps of highlights and the generation of BRD 2D lookup textures; then calculate the phenomenon of reflected light and refracted light finally constitutes global illumination. These lights are based on physically rendering the object, making the object's appearance look more realistic in the virtual environment. Image enhancement technology is used in the final imaging. The model algorithm in this paper has practical application value in various performance indicators. The work of this paper makes a meaningful exploration for the detection of real-time augmented reality scene light sources and the construction of photorealistic rendering framework.

Acknowledgements This work is supported by the Key Research Base of Humanities and Social Sciences of Ministry of Education, Major Project of Bashu Culture Research Center of Sichuan Normal University (No. : bszd19-03).

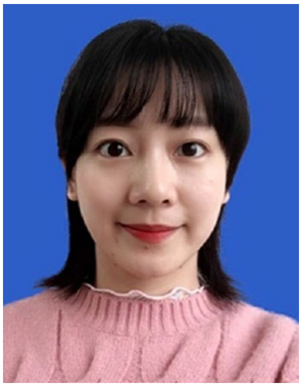
References

1. Rhee, T., Petikam, L., Allen, B., et al.: Mr360: mixed reality rendering for 360 panoramic videos[J]. IEEE Trans. Vis. Comput. Graph. **23**(4), 1379–1388 (2017)
2. Wang, K., Gou, C., Zheng, N., et al.: Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives[J]. Artif. Intell. Rev. **48**(3), 299–329 (2017)
3. Rohmer, K., Jendersie, J., Grosch, T.: Natural environment illumination: Coherent interactive augmented reality for mobile and non-mobile devices[J]. IEEE Trans. Vis. Comput. Graph. **23**(11), 2474–2484 (2017)
4. Meka, A., Fox, G., Zollhöfer, M., et al.: Live user-guided intrinsic video for static scenes[J]. IEEE Trans. Vis. Comput. Graph. **23**(11), 2447–2454 (2017)
5. Hettig, J., Engelhardt, S., Hansen, C., et al.: AR in VR: Assessing surgical augmented reality visualizations in a steerable virtual reality environment[J]. Int. J. Comput. Assist. Radiol. Surg. **13**(11), 1717–1725 (2018)

6. Morgand, A., Tamaazousti, M., Bartoli, A.: A geometric model for specularity prediction on planar surfaces with multiple light sources[J]. IEEE Trans. Vis. Comput. Graph. **24**(5), 1691–1704 (2017)
7. Kim, K., Billinghurst, M., Bruder, G., et al.: Revisiting trends in augmented reality research: a review of the 2nd decade of ISMAR (2008–2017)[J]. IEEE Trans. Vis. Comput. Graph. **24**(11), 2947–2962 (2018)
8. Guo, Y., Cai, J., Jiang, B., et al.: Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images[J]. IEEE Trans. Pattern Anal. Mach. Intell. **41**(6), 1294–1307 (2018)
9. Liu, B., Xu, K., Martin, R.R.: Static scene illumination estimation from videos with applications[J]. J. Comput. Sci. Technol. **32**(3), 430–442 (2017)
10. Chen, A., Wu, M., Zhang, Y., et al.: Deep surface light fields[J]. Proc. ACM Comput. Graph. Interact. Tech. **1**(1), 1–17 (2018)
11. Chu, Y., Li, X., Yang, X., et al.: Perception enhancement using importance-driven hybrid rendering for augmented reality based endoscopic surgical navigation[J]. Biomed. Opt. Express **9**(11), 5205–5226 (2018)
12. Wang, L., Liang, X., Meng, C., et al.: Fast ray-scene intersection for interactive shadow rendering with thousands of dynamic lights[J]. IEEE Trans. Visual Comput. Graph. **25**(6), 2242–2254 (2018)
13. Macedo, M.C.D.F., Apolinário, A.L.: Euclidean distance transform soft shadow mapping[C]. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 238–245. IEEE (2017)
14. Balcı, H., Güdükbay, U.: Sun position estimation and tracking for virtual object placement in time-lapse videos[J]. SIViP **11**(5), 817–824 (2017)
15. Barrile, V., Fotia, A., Bilotta, G.: Geomatics and augmented reality experiments for the cultural heritage[J]. Appl. Geomat. **10**(4), 569–578 (2018)
16. Chen, M., Lu, S., Liu, Q.: Uniform regularity for a Keller–Segel–Navier–Stokes system[J]. Appl. Math. Lett. **107**, 106476 (2020)
17. Kumara, W., Yen, S.H., Hsu, H.H., et al.: Real-time 3D human objects rendering based on multiple camera details[J]. Multimed. Tools Appl. **76**(9), 11687–11713 (2017)
18. De Paolis, L.T., De Luca, V.: Augmented visualization with depth perception cues to improve the surgeon's performance in minimally invasive surgery[J]. Med. Biol. Eng. Comput. **57**(5), 995–1013 (2019)
19. Nóbrega, R., Correia, N.: Interactive 3D content insertion in images for multimedia applications[J]. Multimedia Tools and Applications **76**(1), 163–197 (2017)
20. Fukuda, T., Yokoi, K., Yabuki, N., et al.: An indoor thermal environment design system for renovation using augmented reality[J]. J. Comput. Des. Eng. **6**(2), 179–188 (2019)
21. Huang, H., Fang, X., Ye, Y., et al.: Practical automatic background substitution for live video[J]. Comput. Vis. Media **3**(3), 273–284 (2017)
22. Milosavljević, A., Rančić, D., Dimitrijević, A., et al.: Integration of GIS and video surveillance[J]. Int. J. Geogr. Inf. Sci. **30**(10), 2089–2107 (2016)
23. Toisoul, A., Ghosh, A.: Practical acquisition and rendering of diffraction effects in surface reflectance[J]. ACM Trans. Graph. (TOG) **36**(5), 1–16 (2017)
24. Bui, G., Le, T., Morago, B., et al.: Point-based rendering enhancement via deep learning[J]. Vis. Comput. **34**(6–8), 829–841 (2018)
25. Carlson, A., Skinner, K.A., Vasudevan, R., et al.: Sensor transfer: Learning optimal sensor effect image augmentation for Sim-to-Real domain adaptation[J]. IEEE Robot. Autom. Lett. **4**(3), 2431–2438 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Taile Ni male, native, Wenzhou, Zhejiang, born in March 1984, graduated from Beijing University of Technology in June 2014. He is currently an associate professor at the School of Literature and Journalism at Xihua University and a national vocational skill level. He has long served as a teaching task for the "Dynamic Graphics" course and guided the practical activities of college students. Long-term project practice courses for college students, accumulating a large amount of commercial practical experience. Email: nitaile@sina.com.



Yingshuang Chen female, native, Aanyue, Sichuan, born in December 1992, graduated from Chengdu University Technology. She is currently an engineer at the School of Literature and Journalism at Xihua University and a national vocational skill level. She has been responsible for the development of virtual simulation projects and computer graphics production for a long time, and completed a large number of medical virtual simulation projects. Email: chendashu88@sina.com.

Shoupeng Liu male, born in Shangqiu, Henan Province, October 1982, graduated from Beijing University of Technology in June 2013. As a lecturer of Packaging School, Zhongshan Torch Polytechnic College, national professional skill appraisal assessor, and digital animation design engineer, teaching Three-dimensional Art and UI Design & instructing social practice for college students with rich experiences of teaching, scientific research and practical operation. Email: nremington@sina.com.

Jinglong Wu male, native, Heilongjiang Shangzhi, born in February 1983, graduated from Beijing University of Technology in July 2014. He is currently a lecturer at the Academy of Fine Arts, Taizhou University, Nanjing Normal University, director of the Animation Teaching and Research Office, and Director of the Laboratory. He has long served as an animation major, a major course in digital media art and related curriculum teaching tasks and guiding college students' practical activities. E-mail: wujinlong6868@sina.com.