ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.32, No-1-2, 2024

Reflection probe interpolation for fast and accurate rendering of reflective materials

Katarina Gojković
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113
1000 Ljubljana, Slovenia
kg4775@student.uni-lj.si

Žiga Lesar
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113
1000 Ljubljana, Slovenia
ziga.lesar@fri.uni-lj.si

Matija Marolt
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113
1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

ABSTRACT

In this paper, we aim to improve rendering reflections using environment maps on moving reflective objects. Such scenarios require multiple reflection probes to be positioned at various locations in a scene. During rendering, the closest reflection probe is typically chosen as the environment map of a specific object, resulting in sharp transitions between the rendered reflections when the object moves around the scene. To solve this problem, we developed two convolutional neural networks that dynamically synthesize the best possible environment map at a given point in the scene. The first network generates an environment map from the coordinates of a given point through a decoder architecture. In the second approach, we triangulated the scene and captured environment maps at the triangle vertices – these represent reflection probes. The second network receives at the input three environment maps captured at the vertices of the triangle containing the query point, along with the distances between the query point and the vertices. Through an encoder-decoder architecture, the second network performs smart interpolation of the three environment maps. Both approaches are based on the phenomenon of overfitting, which made it necessary to train each network individually for specific scenes. Both networks are successful at predicting environment maps at arbitrary locations in the scene, even if these locations were not part of the training set. The accuracy of the predictions strongly depends on the complexity of the scene itself.

Keywords

reflection probes, interpolation, convolutional neural network

1 INTRODUCTION

Rendering reflective materials presents a formidable challenge in real-time computer graphics. Our goal is to render reflections that faithfully mirror the surroundings of a moving reflective object while ensuring fast rendering. Various techniques address this issue, yet the balance between speed and accuracy remains ever-present.

At one end of the spectrum, we encounter techniques yielding highly precise results but demanding considerable time and computational resources, often unsuitable for real-time applications. Path tracing, notably improved by the recent advancements in denoising techniques, serves as a prominent example.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conversely, there exist techniques capable of fast but less precise outcomes. By far the most common approach in practice are environment maps. However, environment maps capture only the surroundings of a single point in space (i.e. the reflection probe), which may result in inaccurate reflections if the location of the reflection point differs from that of the probe. To some extent, this issue can be addressed by placing multiple reflection probes in a scene and then selecting the closest one at runtime based on the location of the reflective object. This leads to sharp transitions between reflections when the selected reflection probe changes, which can be mitigated by linearly interpolating neighboring reflection probes. Linear interpolation, however, leads to noticeable inaccuracies and distortions in interpolated reflections.

To address these issues, we present two techniques based on convolutional neural networks (CNNs). In the first approach, a CNN with a decoder architecture takes the location of a reflective object as input and outputs an interpolated environment map at that location. The second technique uses a CNN with an encoder-decoder architecture. First, the set of reflection probes is triangu-

lated, yielding a triangle mesh covering the scene. During runtime, three reflection probes that form a triangle encompassing the reflective object are selected. Their corresponding environment maps, along with interpolation weights determined by the distance of the reflective object from the reflection probes, are input into the CNN. The CNN then outputs the interpolated environment map for the given location within the scene. Training both CNNs involves overfitting to specific scenes, necessitating separate training sessions for each scene. We evaluate the techniques on several scenes to demonstrate their performance and accuracy. Results show that both techniques produce more accurate reflections than simple linear interpolation, at the cost of computational complexity.

2 RELATED WORK

Rendering reflections in computer graphics is a richly explored domain crucial for achieving realistic scenes. While sophisticated algorithms like path tracing offer physically accurate illumination, there is a parallel focus on less computationally intensive methods leveraging ambient images and reflection probes.

Foundational work by Blinn and Newell [3] introduced techniques for applying textures and reflections to curved surfaces, elevating rendering quality using digital signal processing theory and curved surface mathematics. By incorporating a reflection term inspired by Phong's work [9], the authors aimed to simulate more realistic lighting effects, particularly on highly polished surfaces. This involves accurately modeling surface properties and employing precise normal vectors, along with a subdivision algorithm to facilitate the simulation of mirror reflections from curved surfaces.

Structured importance sampling, detailed by Agarwal et al. [1], offers a technique for illumination rendering based on surrounding images. It includes algorithms for sampling lighting textures with visibility consideration and hierarchical layering for surrounding image sampling, optimizing reflection rendering while reducing the required samples.

Ramamoorthi and Hanrahan's contributions [10] introduced efficient reflection representation with spherical harmonic reflection maps, allowing for accurate sampling frequency determination based on error analysis and fast prefiltering methods using spherical harmonic transformations.Further advancements in reflection appearance were made through the prefiltering of ambient images. Kautz et al. [4] presented three algorithms that unify prefiltering methods, including fast hierarchical filtering, machine-accelerated prefiltering, and anisotropic BRDF model prefiltering. The first algorithm provides approximately 10 times faster prefiltering than the brute force method, while the second, optimized for real-time usage, provides accelerated prefiltering. Lastly, the third enables the application of environment map techniques to anisotropic BRDF models for the first time.

Ashikhmin and Ghosh [2] simplified reflection creation with simple blurred reflections using OpenGL capabilities, while Křivánek and Colbert [6] developed filter importance sampling to reduce aliasing errors in real-time rendering.

Manson and Solan's fast filtering method [7] utilized hardware-accelerated trilinear sampling for efficient cube map prefiltering, optimizing coefficients to maintain high-quality results while reducing complexity.

McGuire et al. [8] introduced a novel data structure called light field probes, and two algorithms for real-time global illumination computation in static environments. The ideas from screen-space and voxel cone tracing techniques were applied to this data structure to efficiently sample radiance on world space rays, with correct visibility information, directly within a pixel and compute shaders. The approach improves traditional techniques by eliminating artifacts like light leaking and enabling complex illumination effects in real-time rendering scenarios.

Xia and Kuang [12] presented a novel method for non-uniform probe placement in probe-based global illumination algorithms, aiming to reduce memory waste and improve shading accuracy. The algorithm dynamically adjusts probe positions based on illumination information, by calculating irradiance errors between probes and shading points and employing gradient descent. The method achieves similar rendering quality to existing techniques like DDGI but with fewer probes. However, the algorithm is currently limited to static scenes and light sources.

Rodriguez et al. [11] presented a method for global illumination rendering using illumination textures and reflection probes, optimizing memory consumption with adaptive parametrization and introducing reflection probes to store light paths in the scene for specular reflections.

Finally, Kopanas et al. [5] introduced a novel approach for rendering scenes with curved reflectors, termed Neural Point Catacaustics, using a point-based representation and a neural warp field to model reflection trajectories. By leveraging neural rendering techniques and efficient point splatting, complex specular effects can be synthesized from casually captured input photos. Key contributions include the explicit representation of reflections with reflection and primary point clouds, enabling interactive high-quality renderings of novel views with accurate reflection flow.

3 INTERPOLATION NETWORKS

Our goal was to devise methods for generating accurate environment maps at arbitrary locations in a scene, used for rendering reflective objects. In this section, we present two approaches based on CNNs. Both approaches intentionally leverage overfitting. Our strategy involved training each model with a rich set of scene-specific data, which facilitated the learning of intricate details and subtleties within test scenes, thus enabling the models to adapt effectively to the test scenes.

3.1 First approach: point-based prediction

In the first approach, we provided the CNN with global coordinates (x,y,z) of a reflective object at its input, and tasked it with predicting a corresponding environment map for the given location. The CNN adopts a decoder architecture, transforming global coordinates into an RGB image. First, the input passes through 4 fully connected layers, which expand to 384 output neurons. Subsequently, the CNN features 16 convolutional layers, with batch normalization incorporated after each layer to ensure stable learning. To achieve the desired output image resolution of 512×256 pixels, 5 upsampling layers are inserted after every third convolutional layer, doubling the spatial resolution. ReLU activation functions are applied after each fully connected or convolutional layer. A schematic of the architecture is shown in Figure 1.

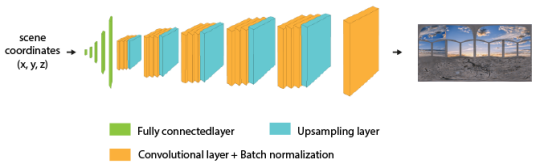


Figure 1: Schematic of the CNN used in the first approach.

3.2 Second approach: triangulation-based prediction

In the second approach, we aimed to enrich the network's input with additional scene context, providing three environment maps along with interpolation weights based on the distance from the reflection probes.

Interpolation weights w_1, w_2 and w_3 for a point $\mathbf{x}$ in a triangle defined by reflection probe locations $\mathbf{p_1}$, $\mathbf{p_2}$ and $\mathbf{p_3}$ are computed as follows:

$$
\begin{aligned}
w_1' &= (\|\mathbf{p_1} - \mathbf{x}\| + \varepsilon)^{-1} \\
w_2' &= (\|\mathbf{p_2} - \mathbf{x}\| + \varepsilon)^{-1} \\
w_3' &= (\|\mathbf{p_3} - \mathbf{x}\| + \varepsilon)^{-1} \\
w_1 &= w_1'/(w_1' + w_2' + w_3') \\
w_2 &= w_2'/(w_1' + w_2' + w_3') \\
w_3 &= w_3'/(w_1' + w_2' + w_3')
\end{aligned}
$$

We used $\varepsilon = 10^{-9}$ to avoid numerical errors.

3.2.1 *Architecture*

The CNN in the second approach adopts an autoencoder architecture. The encoder comprises 5 convolutional layers paired with max pool layers for each of the three input images. The interpolation weights are passed through 4 fully connected layers followed by 5 convolutional layers. Subsequently, the encoded data is passed through the decoder, consisting of 7 convolutional layers. In the decoder, upsampling layers are inserted after each convolution, except the last two, as the desired resolution, which was the same as in the first approach, was achieved at that point. Similar to the first approach, ReLU activation functions are applied after each fully connected or convolutional layer. A schematic of the architecture is shown in Figure 2.

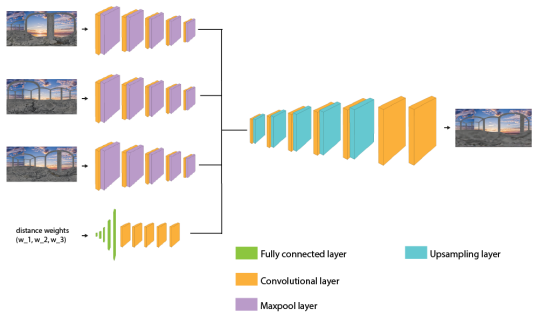


Figure 2: Schematic of the CNN used in the second approach.

3.2.2 *Scene triangulation*

Each scene has been triangulated based on a manually selected set of reflection probe locations. Delaunay triangulation has been used due to its tendency to avoid sliver triangles. The selection of reflection probe locations varied based on the size and complexity of each scene. In the largest and simplest scene, spanning from $(-400m, -400m)$ to $(400m, 400m)$, 25 reflection probes were determined, while the second and third scenes, although smaller in scale but more intricate, necessitated a denser placement of reflection probes. Specifically, the second scene, measuring $(-5m, 6m)$ by $(7m, 23m)$, had 21 reflection probes, while the third scene, sized $(2m, 25m)$ by $(32m, 50m)$, featured 33 reflection probes. Similarly, in the fourth scene, which spanned from $(5.7, -1.9)$ to $(11.7, -28.9)$, 25 reflection probes were uniformly placed. The fifth scene, although bigger, spanning from $(-45, 70)$ to $(100, -140)$, had its main focus between coordinates $(-45, 8)$ to $(100, -11)$ and $(23, -11)$ to $(40, -140)$. Consequently, there was a denser placement of 25 reflection probes within this area to capture the scene's intricacies. Similarly, In the first scene, reflection probes are densely concentrated in the central region, as it is the most relevant area of the scene, whereas, in

the subsequent three scenes, we aim to distribute the reflection probes as uniformly as possible across the entirety of the scene. Figure 3 shows how the scenes have been triangulated.

At each selected reflection probe, we captured the environment map required as input data for the CNN.

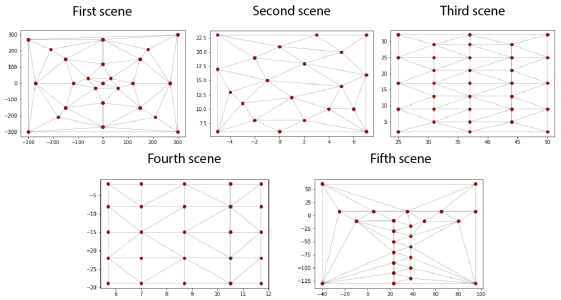


Figure 3: Triangulation of used scenes.

3.3 Learning data

Training the CNNs requires a diverse dataset encompassing input-output pairs for every scene.

For both approaches, panoramic environment maps serve as the reference output data, captured at uniform grid intervals adjusted for scene complexity and size. Specifically, 600 locations were determined for the first scene, 250 for the second scene, 805 for the third scene and 196 for the fourth scene where environment maps were captured. The fifth scene, however, was an exception, with 1210 locations not determined uniformly across the whole scene but only in the most relevant part, which was from $(-45, 8)$ to $(100, -11)$ and $(23, -11)$ to $(40, -140)$.

In the second approach, reflection probe locations were also determined, and panoramic images captured at these points served as input alongside interpolation weights for the locations of reflective objects.

Careful curation of the training set ensures an accurate representation of the depicted scene, facilitating effective network learning.

3.4 Learning procedure

We begin by outlining the training process for both CNNs before delving into a detailed presentation of their performance in Section 4.

During each iteration of the network, input data were first fed through the model to generate predictions in the form of image data. Then the Mean Square Error, which was used as a loss function, was computed from predicted images provided on the CNN's output and true images for the given input coordinates provided in the training set. The Adam optimizer was then employed to minimize the model's error.

For both developed CNNs, three key parameters were determined: the learning rate, batch size, and the number of epochs. Throughout the training of both networks, the learning rate was set to 0.0005. For the first network, the batch size was set to 32, and the number of epochs was set to 16,384, while these hyperparameters were halved for the second network due to the increased complexity of the input data, which included three additional images.

Both models for all three scenes were trained using an AMD Ryzen Threadripper 1950X 16-Core CPU in conjunction with an NVIDIA TITAN V GPU.

The learning procedures for each scene are shown in Figure 4. The loss function values of the last steps are presented in Table 1 and the training duration for both CNNs and each scene is presented in Table 2.

Las Value of Loss Function	First CNN	Second CNN
First Scene	129.688	81.607
Second Scene	192.072	119.955
Third Scene	391.773	333.901
Fourth Scene	109.982	70.636
Fifth Scene	194.705	*

Table 1: Loss function values for the last step by scenes.

	Training time of first CNN	Training time of second CNN
First scene	20 h 50 min	26 h 59 min
Second scene	9 h 10 min	10 h 40 min
Third Scene	28 h 2 min	36 h 21 min
Fourth Scene	6 h 53 min	8 h 58 min
Fifth Scene	42 h 9 min	54 h 30 min

Table 2: Training duration for each scene.

4 RESULTS

Our evaluation relies on both quantitative metrics, such as Mean Squared Error (MSE), and subjective assessments, leveraging the Learned Perceptual Image Patch Similarity (LPIPS) metric for a comprehensive understanding. Both CNNs were trained and tested across five distinct scenes: a simplistic building environment, a detailed room [3], a forest [4], a space ship corridor [5], and a city [6] (see Figure 5).

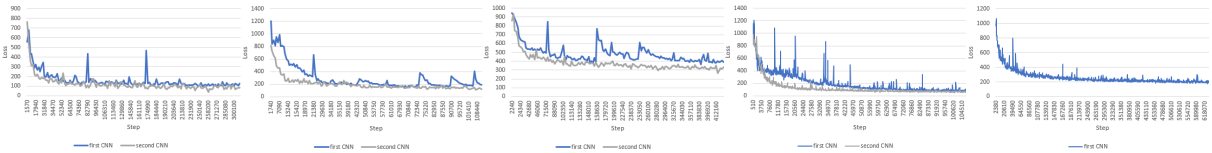
[2] Some initial training steps with significantly different values are excluded from the graph for better visualization.

[3] https://sketchfab.com/3d-models\/the-king-s-hall-d18155613363445b9b68c0c67196d98d

[4] https://www.cgtrader.com/3d-models/exterior/landscape/forest-house-scene

[5] https://www.cgtrader.com/free-3d-models/science/other/sci-fi-corridor-6b2e3194-8a54-4846-a290-5bf473a88a74

[6] https://www.cgtrader.com/free-3d-models/architectural/architectural-street/city-scene-719d674f-97b6-49a1-8399-de7722a60f5e



First scene Second scene Third scene Fourth scene Fifth scene

Figure 4: Change in the loss function during the training of the first and second CNNs for each scene.[2] The training data for the second CNN for the fifth scene is not presented due to corruption.

Figure 5: Five scenes on which we trained and tested CNNs.

4.1 Evaluation of predicted environment maps

To measure the accuracy of the predicted environment maps, we compared them against actual scenes at specific points in the room. We assessed both approaches and also generated interpolated images of reflection probes using the same spatial triangulation as described in Section 3. These interpolated images were created by weighting the pixels of three reflection probes.

For each scene, we selected four distinct points within the scene for evaluation. Two points were from the training set (Point A and Point B), and two were outside it (Point C and Point D). This evaluation process provided insights into the models' predictive accuracy across various scenarios.

Models were tested on Intel(R) Core(TM) i7-10510U 4-Core CPU and NVIDIA GeForce MX250 GPU.

4.1.1 First Scene: Simple Building

The first scene depicts a simple environment with minimal details—a building comprising a floor and eleven columns against a sky backdrop.

Mean Squared Error

The Mean Squared Error (MSE) values for the first scene are shown in Table 3.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	100.38	61.59	418.28
Point B	235.40	247.20	1085.67
Point C	718.93	744.58	1152.46
Point D	428.81	443.50	1451.75

Table 3: MSE values according to the real image of the surroundings for the first scene.

The first CNN shows slightly higher MSE values for points inside the training data set, while the second CNN exhibits marginally larger errors for points outside of it. Both CNNs significantly outperform interpolated reflection probes.

To visually represent the disparities between predicted and actual images, Figure 6 presents error maps, illustrating discrepancies for one point within the learning set and one outside. These maps delineate areas of inconsistency between predictions and real surroundings. Correct values are depicted in gray, while errors are indicated by varying colors, with greater intensity denoting larger errors. Error maps reveal fewer errors for both CNN approaches compared to the interpolated reflection probes. The first CNN demonstrates less error, particularly noticeable on the ground and distant towers. However, slight variations in sky color are noticeable in the predictions of the first CNN.

Learned Perceptual Image Patch Similarity

Table 4 shows the Learned Perceptual Image Patch Similarity (LPIPS) values for the first scene.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	0.0584	0.0418	0.0868
Point B	0.1172	0.1187	0.1870
Point C	0.1694	0.1795	0.1893
Point D	0.1302	0.1275	0.1993

Table 4: LPIPS values against the real image for the first scene.

Both CNNs exhibit comparable performance in terms of perceptual properties such as color, texture, and shape, outperforming interpolated reflection probes.

4.1.2 Second Scene: Room

The second scene depicts an intricately detailed room adorned with various decorations.

Mean Squared Error

The MSE values for the second scene are presented in Table 5.

The second CNN shows lower errors in most cases, especially within the training set. Both networks outperform interpolated reflection probes.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	134.81	114.39	952.968
Point B	178.43	134.51	1410.76
Point C	1075.77	995.61	1273.71
Point D	613.60	633.00	1197.21

Table 5: MSE values for the second scene.

Visualizations of error maps confirm the superiority of our approaches over interpolated reflection probes, particularly within the training set. Although the advantage diminishes slightly for scenarios outside the training set, our approaches consistently outperform the interpolated probes. The correlation between the error maps and mean squared error values further validates the robustness of our approaches.

Learned Perceptual Image Patch Similarity

Table 6 displays the LPIPS values for the second scene.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	0.1750	0.1538	0.2494
Point B	0.1673	0.1448	0.2401
Point C	0.2535	0.2418	0.2074
Point D	0.2238	0.2030	0.2565

Table 6: LPIPS values for the second scene.

Our approaches show a significant perceptual advantage over interpolated reflection probes on training set data. However, this advantage decreases with out-of-sample data, especially in the third case, where the LPIPS metric for interpolated reflection probes is lower than for our methods.

4.1.3 Third Scene: Forest

The third scene depicts a forest teeming with trees of various sizes, low shrubbery, and scattered rocks. Dominated by green hues, the scene boasts intricate details.

Mean Squared Error

The MSE values for the third scene are listed in Table 7.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	517.92	498.21	1876.48
Point B	367.71	338.59	1269.22
Point C	686.12	799.88	1203.10
Point D	805.69	832.67	1961.62

Table 7: MSE values for the third scene.

The second CNN performs better within the training set, while the first CNN is better for points outside the training set, contrary to visual impressions favoring the second approach. Both outperform interpolated reflection probes.

Analyzing the error maps reaffirms the superior performance of our approaches over the interpolated reflectance probes, although this distinction is less apparent in the actual images, especially within the forested regions.

Learned Perceptual Image Patch Similarity

Table 8 presents the LPIPS values for the third scene.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	0.3055	0.2653	0.2802
Point B	0.2720	0.2491	0.2043
Point C	0.3491	0.3284	0.2769
Point D	0.3316	0.2998	0.3088

Table 8: LPIPS values for the third scene.

The LPIPS values show less pronounced deviations, with interpolated reflection probes performing comparably to the CNNs.

4.1.4 Fourth Scene: Spaceship Corridor

The fourth scene depicts a small spaceship corridor characterized by predominant grayscale hues.

Mean Squared Error

The MSE values for the fourth scene are shown in Table 9.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	129.08	57.90	1031.44
Point B	111.29	67.39	1413.32
Point C	1303.07	1519.17	1394.56
Point D	856.83	1001.12	1321.07

Table 9: MSE values for the fourth scene.

The second CNN shows lower MSE for points within the training set. Surprisingly, the first CNN presents a marginally lower MSE for points outside the training set. Moreover, in these cases, the MSE of interpolated reflection probes is not substantially greater.

Error maps reveal more errors in the first CNN's predictions than the second's within the training set, and even more in interpolated reflection probes. Outside the training set, error maps are similar for all three approaches.

Learned Perceptual Image Patch Similarity

Table 10 shows the LPIPS values for the fourth scene.

The second CNN demonstrates superior perceptual performance, with both CNNs exhibiting better perceptual characteristics than interpolated reflection probes.

Point	First CNN	Second CNN	Interpolated Reflection Probes
Point A	0.1309	0.0878	0.2248
Point B	0.0991	0.0663	0.2193
Point C	0.2361	0.2334	0.1886
Point D	0.2317	0.1908	0.2405

Table 10: LPIPS values for the fourth scene.

4.1.5 Fifth Scene: City

The fifth scene depicts a cityscape with diverse buildings and two main thoroughfares, which were the focal points for both network training endeavors.

Mean Squared Error

The MSE values for the fifth scene are presented in Table 11.

point	first CNN	second CNN	interpolated reflection probes
Point A	168.45	214.49	2470.10
Point B	236.60	265.06	2303.81
Point C	578.39	476.49	3016.37
Point D	468.92	337.91	1169.55

Table 11: MSE values for the fifth scene.

The first CNN outperforms the second CNN for points within the training set, while the second CNN shows slightly better performance for points outside the training set. Both CNNs significantly outperform the interpolated reflection probes across all points.

For this scene as well, error maps were calculated, revealing fairly similar results between our first and second approaches for points within and outside the training set. Conversely, error regions for interpolated reflection probes are considerably larger and more intense for all points compared to our approaches.

Learned Perceptual Image Patch Similarity

Table 12 shows the LPIPS values for the fifth scene.

point	first CNN	second CNN	interpolated reflection probes
Point A	0.1261	0.1395	0.3036
Point B	0.1733	0.1607	0.3338
Point C	0.2074	0.2120	0.2697
Point D	0.1936	0.1933	0.2564

Table 12: LPIPS values for the fifth scene.

From a perceptual standpoint, both of our approaches are fairly equivalent across all points, exhibiting significantly superior performance compared to interpolated reflection probes.

4.2 Performance Evaluation Results

In addition to assessing prediction performance, we gathered runtime data for both CNNs to evaluate their suitability for rendering reflections in real-time. Total execution time (including pre-processing and post-processing) and prediction time were measured over 1000 consecutive predictions, from which the average time per prediction was calculated.

Regarding speed, the first network exhibited superior performance, with an average total execution time of approximately 173 milliseconds with a prediction time of about 95 milliseconds. In contrast, the second network had a total execution time of around 333 milliseconds, with a prediction time of approximately 295 milliseconds.

We also compared their performance in real-time execution in the Unity game engine. Specifically, we evaluated rendering speed, measured in frames per second (FPS) Table 13.

Speed	First CNN	Second CNN	Reflection Probe
Renderings	6-10 FPS	0.7-2 FPS	5-150 FPS

Table 13: Rendering speed comparison for different reflection rendering approaches.

The second approach's CNN proved to be unsuitable for real-time applications due to its slow performance. While the real-time reflection probe demonstrated excellent efficiency in the simplest scene, achieving a consistent rendering speed of approximately 150 FPS, it struggled in more complex scenes. In these scenarios, the first CNN performed comparably to the real-time reflection probe, but with smoother transitions between reflections, making it the preferred option for real-time applications.

5 DISCUSSION

In this section, we critically analyze the outcomes and metrics discussed in Section 4. We explore the advantages and drawbacks of each approach, consider their respective applications, and propose avenues for future enhancement.

Both approaches were trained and tested on five different scenes. We initiated the development of both approaches with the first and simplest scene. This scene provided insight into the promising direction of our approach, as we successfully trained the networks for a straightforward and repetitive environment. For the second scene, we selected a room filled with intricate details to assess the ability of the networks to learn such features. The third scene depicted a diverse environment rich in detail but predominantly characterized by a single color. Here, we aimed to evaluate the approaches' effectiveness in reproducing details in such environments. With the same objective, we trained the fourth scene, albeit significantly smaller than the third and featuring a different dominant color. This enabled us to assess the influence of dominant color and scene

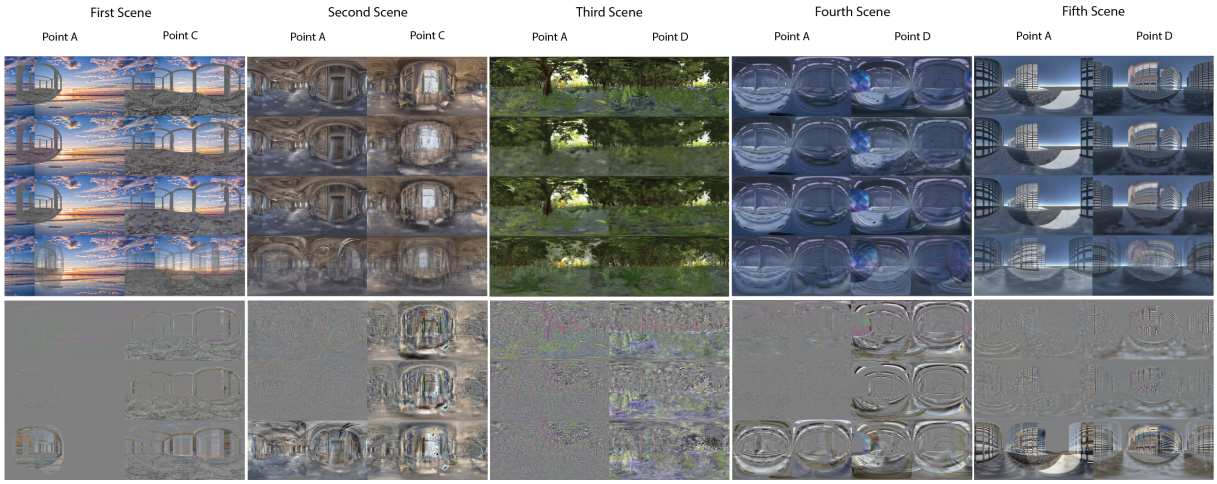


Figure 6: The upper half illustrates the prediction results of the two networks alongside the actual environment map and the interpolated reflection probes for all five scenes. In each scene, the first column showcases the outcomes for the point included in the training set, while the second column displays the outcomes for the point not included in it. The lower half displays error maps corresponding to the rendered results.

size on the results. Finally, the fifth scene contains numerous details, which are uniformly distributed and exhibit regular shapes throughout the scene (unlike the second scene, where details are unevenly distributed and irregular). Here, we aimed to investigate the impact of detail distribution and shape on the results.

5.1 Evaluation of results

We analyzed the effectiveness of the networks in various aspects, including prediction performance, color accuracy, and real-time processing.

Initially, we examined the fidelity of the surroundings' depiction, assessing whether objects were accurately represented and the level of detail in each depiction. Notably, the images generated by the second approach rendered buildings, especially those outside the training set, less accurately compared to the first network (Figure 7). Despite this, both networks performed similarly in rendering objects in other scenes. However, the second network's predictions exhibited more detail, likely due to the additional input data.

The first approach demonstrated superior proficiency in predicting details of rectilinear shapes, particularly when such shapes were in stark contrast with their surroundings, which was most pronounced in the first and fifth scenes.

Furthermore, the second network demonstrated superior color prediction, particularly in scenes with similar color tones between objects and the background (Figure 8). For instance, in the first scene, the sky and building's colors closely resemble each other, with the second network providing more accurate predictions. Conversely, the first network struggled with color nuances, especially apparent in the forest scene. The second network's enhanced color perception also contributed to its ability to capture finer details.

Figure 7: The figures illustrate the rendering of the first scene at the point, which was not part of the training set. The first image is the real image at that point, the second is the prediction of the first CNN, and the third is the image predicted by the second CNN.

Figure 8: The images in the first column depict actual surroundings, while those in the second column represent predictions made by the first CNN, and those in the third column depict predictions made by the second CNN. The images reveal an error in recognizing the correct lower hues by the first network.

In general, both approaches exhibited fairly comparable success in predicting environment maps at points outside the training set. However, the small amount of training data was evident in the second and fourth scenes, where the first approach performed relatively poorly. Consequently, it can be inferred that an undesirable degree of overfitting occurred. This assertion is further supported by the MSE values, which are very small for points within the training set but significantly larger for points outside the training set.

Effective utilization of our approaches requires careful scene design and consideration of lighting placement before data capture and network training. Retraining the CNN after any scene changes is crucial to maintaining accurate and realistic reflections. Additionally, careful placement of static objects and lighting effects ensures their influences are accurately captured in the training data and reflected in predictions.

Despite these considerations, a significant drawback of our approaches remains the absence of reflective object shadows in rendered reflections. Since the training data is captured without it being present in the scene, its shadows are not included, preventing the rendering of self-reflections and reflections between multiple reflective objects. This limitation persists even with meticulous scene design and data capture planning. Moreover, reflections of other non-static objects in the scene and their shadows are not present in the captured training data, as we cannot predict their spatial positions in advance at the time of data acquisition.

From a speed perspective, the first network outperformed the second due to processing less data. While the second network's additional input led to more detailed predictions, it came at the cost of slower processing.

Overall, both approaches have demonstrated the capability to predict and depict the surroundings of a given point in the scene fairly accurately, albeit sometimes with a lack of detail. Compared to traditional approaches such as using the nearest reflection probe or interpolating between reflection probes in the scene to depict reflections, our approaches predict and depict the environment of the object more accurately. However, it is notable that these predictions sometimes portray fewer details, resulting in slightly blurred surroundings. Regarding speed, both approaches are slower than traditional methods, although the first approach has a sufficiently fast rendering speed. When compared to real-time reflection probes, which also accurately depict the environment of the object and produce cleaner renderings with more details, our approaches excel particularly in complex environments where real-time reflection probes may falter, especially during the movement of reflective objects, where transitions between different rendered reflections are highly noticeable and sharp, a limitation not present in our approaches.

5.2 Usability of models

The CNN developed in the second approach proves unsuitable for real-time applications. Conversely, the first approach's network is well-suited for real-time applications like gaming, where heavy computational tasks are minimal, allowing for swift network predictions. On the other hand, both CNNs are valuable for rendering reflections in animations or design tools, offering quality results within reasonable processing times.

From the perspective of the quality of predicted environment maps and rendered reflections, our approaches are most useful for reflective objects that lack pronounced reflective properties (such as wrinkled or rough surfaces), where finer details in the rendered reflections are not crucial. However, their reflections contribute to creating a realistic appearance of the scene, where our approaches excel in providing generally accurately depicted surroundings in reflections and smooth, natural transitions between different areas of the environment as the reflective object moves through the scene.

Both approaches excel in environments with static scenes, making them ideal for rendering moving objects against a relatively stable backdrop. However, they are less effective for static objects with pronounced reflective properties, such as mirrors, where traditional reflection probes are more appropriate. Our approaches shine when rendering scenes where moving objects significantly enhance visual realism, such as characters or vehicles traversing a static environment, contributing to a lifelike visual perception of the surroundings.

5.3 Possible improvements

5.3.1 Enhancing robustness through incorporating varied object heights

Currently, both approaches are trained solely on scenes where the main activity occurs within a single plane, and the reflective objects move only at a single height (varying only in x and z coordinates). Consequently, both networks are trained with constant y coordinates. It would be worthwhile to further generalize the approaches and make them more robust by incorporating different y coordinates in the training data, thus teaching them to predict reflections for objects at varying heights. In such a scenario, the first CNN would not require additional upgrades, while the second approach would need some adjustments, primarily considering a different spatial decomposition.

5.3.2 Enhanced prediction quality

To elevate prediction quality, augmenting network complexity by integrating additional fully connected or convolutional layers could enhance adaptability to training data. However, caution is warranted as increased complexity may elongate prediction times. Advanced CNN layers like capsule and dynamic layers offer potential replacements for traditional convolutional layers, enabling better adaptability to training data without significant model slowdown. Moreover, employing sophisticated loss functions such as perceptual or contrastive loss could further refine predictions by preserving content and style fidelity. Also, post-processing

techniques like noise reduction, sharpening, and contrast enhancement could refine predicted images, augmenting overall quality.

5.3.3 Model speed optimization

Implementing model compression techniques like pruning and quantization reduces model size and prediction times, albeit with potential prediction quality trade-offs. Model caching, wherein predicted images for known inputs are stored to circumvent redundant predictions, offers an efficient strategy for recurrent input scenarios, albeit with increased memory overhead.

5.3.4 Realistic reflections

Exploring methods to prefilter environment maps for more realistic reflections akin to mirror reflections could enhance scene realism.

The absence of self-shadows in object reflections could be addressed via prerendering shadows in training data, possibly using shadow mapping techniques.

6 CONCLUSION

In this paper, we endeavor to enhance existing methods for rendering reflections, particularly those reliant on reflection probes. One of the foremost challenges with these methods lies in rendering reflections as reflective objects move within a scene, and in mitigating the jarring transitions between disparate reflection probes.

To address this issue, we introduce two novel methods leveraging convolutional neural networks (CNNs) to generate environment maps at specific points within a scene. Both methods demonstrate success in predicting scene surroundings, seamlessly blending reflections as objects move through the scene. Particularly in complex environments, the first method outperforms real-time reflection probes provided by the Unity game engine, offering smoother and more natural transitions between reflections. However, due to its slower processing speed, the second method's comparative evaluation remains unfeasible for real-time applications. Notably, the resolution of predicted environment maps varies with scene complexity, rendering the current methods more suitable for rendering blurred reflections.

Future research avenues may explore enhancing image resolution in complex scenes and optimizing method speed. Furthermore, exploring pre-filtering techniques for environment maps could enhance the realism of rendered reflections.

7 REFERENCES

[1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. Association for Computing Machinery, 2003.

[2] Michael Ashikhmin and Abhijeet Ghosh. Simple blurry reflections with environment maps. *Journal of Graphics Tools*, 2002.

[3] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. 1976.

[4] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. A unified approach to prefiltered environment maps. In Bernard Péroche and Holly Rushmeier, editors, *Rendering Techniques 2000*, pages 185–196, Vienna, 2000. Springer Vienna.

[5] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022.

[6] Jaroslav Křivánek and Mark Colbert. Real-time shading with filtered importance sampling. *Computer Graphics Forum*, pages 1147–1154, 2008.

[7] Josiah Manson and Peter-Pike Sloan. Fast filtering of reflection probes. *Computer Graphics Forum*, pages 119–127, 2016.

[8] Morgan McGuire, Michael Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, page 11, February 2017. I3D 2017. URL: https://casual-effects.com/research/McGuire2017LightField/index.html.

[9] Bui Tuong Phong. Illumination for computer generated pictures. In *Seminal graphics: pioneering efforts that shaped the field*, pages 95–101. 1998.

[10] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, page 517–526. Association for Computing Machinery, 2002.

[11] Simon Rodriguez, Thomas Leimkühler, Siddhant Prakash, Chris Wyman, Peter Shirley, and George Shirley. Glossy probe reprojection for interactive global illumination. *ACM Transactions on Graphics (TOG)*, pages 1–16, 2020.

[12] Bo Xia and Fen Kuang. Non-uniform illumination-guided probe placement. In *ICETIS 2022; 7th International Conference on Electronic Technology and Information Science*, pages 1–4, 2022.