# Learning Residual Color for Novel View Synthesis

Lei Han*, Dawei Zhong*, Lin Li, Kai Zheng, Lu FANG§

***Abstract*—Scene Representation Networks (SRN) have been proven as a powerful tool for novel view synthesis in recent works. They learn a mapping function from the world coordinates of spatial points to radiance color and the scene's density using a fully connected network. However, scene texture contains complex high-frequency details in practice that is hard to be memorized by a network with limited parameters, leading to disturbing blurry effects when rendering novel views. In this paper, we propose to learn 'residual color' instead of 'radiance color' for novel view synthesis, i.e., the residuals between surface color and reference color. Here the reference color is calculated based on spatial color priors, which are extracted from input view observations. The beauty of such a strategy lies in that the residuals between radiance color and reference are close to zero for most spatial points thus are easier to learn. A novel view synthesis system that learns the residual color using SRN is presented in this paper. Experiments on public datasets demonstrate that the proposed method achieves competitive performance in preserving high-resolution details, leading to visually more pleasant results than the state of the arts.**

## I. Introduction

Novel view synthesis, serving as a fundamental technique for virtual reality applications, aims to create new views from given observation samples of the scenes. Recent works such as GoogleJump [2], DeepView [6] etc. have shown significant progress by employing a synchronized structured camera array as capture devices. However, it remains a challenging task for high-quality novel view synthesis from a sparse view input. Existing methods try to solve the problem by either reconstructing an explicit geometric model of the scene [7], [10] or employing probabilistic depth representation [27], [37]. Typically, model-based methods enjoy higher freedom with few input views, yet require high-resolution and precise 3D models. Moreover, it cannot reflect the change of light from different views. On the other hand, probabilistic depth-based methods model the scene geometry as a probabilistic distribution instead of an explicit depth surface. For instance, StereoMagnify [37] employs multi-plane images for scene representation and renders novel views based on alpha composition, NeRF [24] parametrizes the scene as a radiance field using an implicit scene representation network and applies volumetric rendering for novel view synthesis.

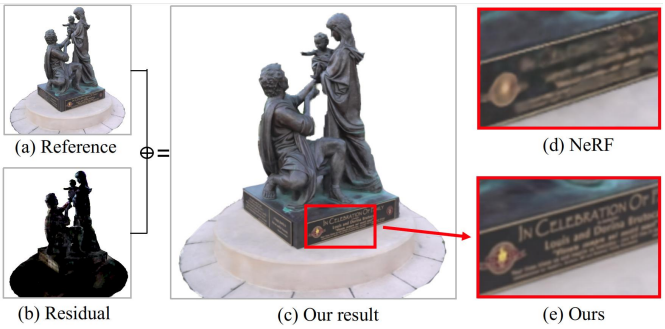



Fig. 1: For each point in 3D space, we calculate its reference color based on the multiview observations. The residual color as well as density that are memorized by Scene Representation Networks. Reference color image (a) and residual color image (b) are composed based on volumetric rendering of sampling points and combined for novel view synthesis. Note that the reference color image contains most high-frequency texture information, while the SRN only needs to represent the residual color and geometry (density field) that are composed of low-frequency information. As shown in (d,e), the proposed method generates more clear details than state-of-the-art method NeRF [24] for novel view synthesis.

Examining the latest work, NeRF [24] achieves superior performance by employing a fully connected network to represent the underlying continuous volumetric radiance field of the complex scenes. The network can be trained on a sparse set of input 2D pictures directly, without additional 3D supervision. Benefiting from the volumetric scene representation, NeRF generates continuous novel view synthesis for free moving cameras. Unfortunately, due to the inherent nature that neural networks overfit low-frequency information [28], the synthesized images lost high-frequency texture details even with the positional encoding scheme, which leads to disturbing blurry effects.

We argue that current implicit scene representation networks simply encode the spatial coordinates as the representations of each point while neglecting that the points may own different characteristics when being back-projected onto the input views. Specifically, the back-projected observations (denoted as spatial color priors) at different view angles are consistent for points on Lambertian surfaces while varying significantly for non-surface points. As a result, there exists a strong connection between spatial color priors and the actual radiance color for each point.

Based on this observation, we propose a residual-color

∗ : Equal Contribution

Dawei Zhong and Lu FANG are with Dept. of Electronic Engineering and Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, and also with Beijing National Research Center for Information Science and Technology (BNRist), Beijing 100084. Lei Han, Lin Li and Kai Zheng are with Hisilicon.

This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 62125106, 61860206003 and 62088102, in part by Ministry of Science and Technology of China under contract No. 2021ZD0109901, in part by Shenzhen Science and Technology Research and Development Funds (JCYJ20180507183706645), in part by Beijing National Research Center for Information Science and Technology (BNRist) under Grant No. BNR2020RC01002.

§ Correspondence Author: E-mail: fanglu@tsinghua.edu.cn

learning framework for novel view synthesis. Specifically, for each point, we take its spatial color priors as the reference color and employ a scene representation network (e.g., NeRF [24]) to regress residuals between surface color and reference color. Fig. 1 shows the decomposition of our rendering result. Note that the residuals are small values or close to zero for most spatial points. Thus, they are easier to learn than previous methods that directly enforce the network to memorize the intricate texture details. We demonstrate that our scheme preserves more clear details for the novel view synthesis, leading to more pleasing visual results than the state-of-the-arts. Notably, for complex scenes, previous methods such as NeRF [24] suffer blurry artifacts, while our method achieves significant improvement thanks to the residual learning scheme. The technical contributions are summarized as follows.

- Spatial color priors: given the insight that multiview observation conveys prior information for radiance color, our learning framework is equipped with the spatial color priors based on the input view observations, which serves as complementary information for implicit scene representation networks that simply map the world coordinate of points to local scene properties.
- Residual color learning: by taking the proposed spatial color priors as the reference, we propose a residual color learning framework to regress the residuals between surface colors and the reference. The residuals are close to zero for most spatial points thus are easier to learn than previous works that directly regress surface colors. The proposed residual learning framework in neural rendering is simple yet effective, which can be easily incorporated with other implicit scene representation approaches.

## II. Related Work

Realistic rendering aims to generate arbitrary novel views based on limited observations, which is mainly divided into two different pipelines: texture-based rendering and image-based rendering. The texture-based rendering follows a classical rendering pipeline, which constructs explicit 3D models and gets rendering images based on ray tracing. While image-based rendering uses soft 3D representation, such as probabilistic depth or neural network, for implicit scene representation without explicit 3D models. In the following, we will introduce the current progress of texture-based rendering and image-based rendering, respectively.

**Texture-based Rendering.** Texture-based rendering aims to reconstruct an accurate colored 3D model of the environment for novel view rendering. [7], [10] use dense matching of multiview observations and epipolar geometry to reconstruct the 3D model. ElasticFusion [33] uses frame-to-model registration and windowed surfel-based fusion. [36] uses volumetric fusion based on spatial hashing [19] and TSDF fusion [17] to achieve real-time dense reconstruction. With the development of machine learning, the neural network is also used in predicting an explicit 3D model. [16], [32] projects 2D feature to 3D voxel grid and uses 3D convolution to get the voxel model. [20] uses a differentiable point-based renderer to get the 3D model. The coordinate and color of points are the learning parameters. [8] uses a multilayer perceptron to complete the point cloud into a mesh model. [13] trains a patch-based conditional discriminator to guide the texture optimization to be tolerant to misalignment. Its performance is limited by the quality of the existing 3D model.

With the help of an explicit 3D model, texture-based rendering has good efficiency and editability. However, it is difficult to avoid distortions, holes, and blurred parts in the reconstructed models, especially for a messy scene. The deficiencies in the generated models will bring artifacts and blurred details in rendering images.

**Image-based Rendering.** Different from texture-based rendering, image-based rendering generates novel views without an explicit 3D model. [4], [12] generate novel views by transforming sampling images. The sampling images are warped into a novel view based on the estimation of camera poses estimation. [5], [14] use Bayesian estimation to get the color value at each pixel in novel views. The neural network is widely used in implicit scene representation. It shows great potential in memorizing a scene, including both geometry and texture. Geometry can be easily represented by the neural network thanks to its low-frequency nature while high-frequency texture details are harder to be memorized by the neural network. [6], [37] generate the multiplane images with different transparency in different layers and the novel view can be obtained by integration of the multiplane images. [23] generates novel views by a weighted combination of transformed neighboring multiplane images, which are modulated by the corresponding transparency. [25] deduces differentiable volumetric rendering for TSDF value prediction. [31] maps world coordinates to a feature representation of local scene properties and uses a scene representation network to predict a special network for different kinds of scenes. [22] uses an encoder to produce a latent code $z$ based on the multiview images, then decodes it into a volume that gives color and transparency values for each voxel. NPBG [1] takes a set of RGB views and a point cloud as input. A neural descriptor is fitted to each point, after which new views of a scene can be rendered. FVS [30] computes 3D proxy geometry with the input images via multi-view stereo. Given a target view,nearby source images are mapped into the target view based on the projection depth and then the mapped images are blended using a recurrent convolutional network. Both methods need high-quality 3D geometry as input. The rendering performance is highly influenced by the quality of the point cloud or reconstructed 3D geometry. If the 3D model used for mapping misses large parts of the scene or has gross outliers, the pipeline will produce visible artifacts. NeRF [24] represents a scene by a multilayer perceptron and trains it by volumetric rendering. Positional encoding and hierarchical sampling are used to improve rendering performance.

The implicit scene representation has shown great potential in realistic rendering but remains a challenging task. NeRF [24] uses multilayer perceptron and volumetric rendering for the implicit representation. It achieves significant rendering performance improvement, and there are many methods to improve NeRF. Both NSVF and our method aim to improve

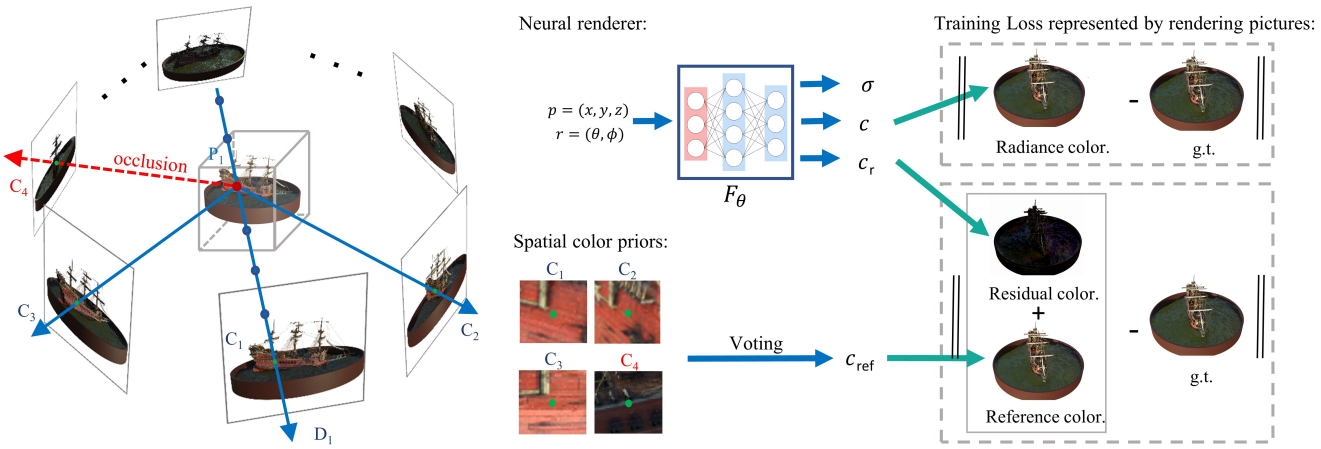



Fig. 2: An overview of the proposed residual color learning scheme. For each spatial point, we calculate its reference color $c_{ref}$ from the input multiview observations and predict density $\sigma$, radiance color $c$ and residual color $c_r$ using a scene representation network. The spatial color priors are the projected pixels (e.g., $C_1$, $C_2$, $C_3$ and $C_4$ of $P_1$, which are the center pixels of the presented images patches). The image patches are used to filter out the occluded pixels. The reference color $c_{ref}$ of the point $P_1$ is estimated by voting from back-projected pixels of the point. For novel view synthesis at a given viewpoint, volumetric rendering is applied by integrating $c_r + c_{ref}$ of spatial points along all pixel rays based on the predicted density $\sigma$. Radiance color $c$ is integrated to predict a coarse image for occlusion detection (removing $C4$ in this case) for better reference color prediction. During the training stage, input views are sampled as ground truth, and $F(\theta)$ is trained using the rendering loss of both radiance color and residual color.

the rendering quality of novel view synthesis from different angles. NSVF employs the prior that surfaces are sparse in 3D space thus only voxels passing through the surface need to be processed and uses local parameters to improve the capability of scene representation network. Differently, our method proposes spatial color prior to reduce the learning difficulty of high frequency texture details by calculating the reference color from projection pixels. Nerfies [26] introduces deformation code to handle dynamic scenes and uses appearance code to handle light changes. KiloNeRF [29] utilizes thousands of tiny MLPs to replace the original single large MLP for acceleration. Our method is supplementary for such methods. To improve the ability to keep high-frequency textures, we propose novel residual-based multiview priors based on multiview observations. With the proposed spatial color priors, a residual learning scheme is introduced for high-quality implicit scene representation.

## III. Method

Our approach takes a sparse set of views as input and aims to render novel views at a given viewpoint. The overall framework is illustrated in Fig. 2. We propose 'spatial color priors' based on the input multiview observations, while the occluded pixel is removed by a proposed patch feature filter. The reference color is obtained from the spatial color priors by a voting strategy. With that, a residual color learning scheme is introduced in the implicit scene representation network to reduce the network capacity requirements for high-frequency information.

In the following contents, we first introduce the implicit scene representation in Sec. III-A, then elaborate spatial color priors in Sec. III-B and residual color learning scheme in Sec. III-C, followed by the implementation details in Sec. III-D.

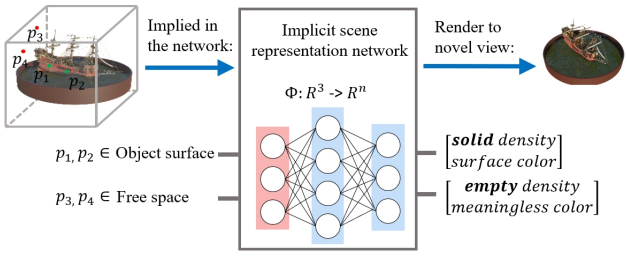



Fig. 3: Unlike traditional 3D reconstruction, which has an explicit model, the implicit scene representation uses a function to fit scene information. The function takes point position as input and outputs its spatial characteristics which can be rendered as images. Through this, the sampling images can be used for training the function, which can then be used to generate novel views.

### A. Implicit Scene Representation

[31] employ a fully connected network to describe the scene implicitly. It learns a function that maps the continuous 3D coordinates to a feature representation of the scene at those feature coordinates. The feature representation may be translated to properties such as density [24] or signed distance function [25] (Fig. 3) for different targets.

The representative SRN method NeRF [24] models the scene as a neural radiance field and applies volumetric rendering [15] for novel view synthesis. Each spatial point is represented by its 3D coordinates $p = (x, y, z)$ and view

direction $d_r = (\theta, \phi)$, which are mapped to the density (opacity) $\sigma$ and radiance color $c$ using a fully connected network. The expected color $C(r)$ of a camera ray $r$ can be rendered from the classical volumetric rendering techniques as shown in Eq. 1.

$$\bar{C}(r) = \int_{t_n}^{t_f} \exp\big(-\int_{t_n}^{t} \sigma(s)ds\big)\sigma(t)c(t, d_r)dt, \quad (1)$$

where $t_n$ and $t_f$ are the near and far bounds of $r$ respectively, $dt$ is the distance among the camera ray. $d_r$ indicates the view direction of $r$ and "exp" is the exponential function. Based on the volumetric rendering [15], the continuous integration of Eq. 1 can be replaced by numerical quadrature:

$$\begin{aligned} T_i &= \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j), \\ w_i &= T_i\big(1 - \exp(-\sigma_i \delta_i)\big), \\ \bar{C}(r) &= \sum_{i=1}^{N} w_i c_i. \end{aligned} \quad (2)$$

$\sigma_i, c_i$ are represented by a fully connected network $F_\theta(p_i, d_r)$, which means the color and density of the i-th sampled point respectively. $\delta_i$ is the distance between two sampling points. $\bar{C}(r)$ is calculated by summing all sampling points among a ray based on weight $w_i$. Here $F_\theta(p_i, d_r)$ can be learned from the given sparse input views by minimizing the difference between the rendered views $\bar{C}(r)$ and the observed views $C(r)$:

$$L = \sum_{r \in R} \|\bar{C}(r) - C(r)\|, \quad (3)$$

where $R$ is the set of all camera rays. Its number is equal to the number of all image pixels.

### *B. Spatial Color Priors*

Recall that the scene geometry and texture information are implied in the color consistency of multiview observations, based on which, we propose 'spatial color priors' and a residual color learning scheme to reduce the network capacity requirements for high-frequency information.

The spatial points are firstly projected onto the observation images to obtain its projection histogram. The training images are denoted as $\boldsymbol{I} = \{I_i, i \in N\}$ and the corresponding camera poses are denoted as $\boldsymbol{H} = \{H_i, i \in N\}$. We calculate the distance between current camera pose $H_c$ and $\boldsymbol{H}$, and select $M$ closest images from training images $\boldsymbol{I}$. The local images are $\boldsymbol{I}_{local} = \{I^i_{local}, i \in M\}$. Then the back-projection pixels are calculated based on the multiview geometry [9]:

$$u_i = K H_i H_c^{-1} p, i \in M, \quad (4)$$

where $K$ is the intrinsic of camera. $\boldsymbol{u} = \{u_i, i \in M\}$ are the projection pixels in local images $\boldsymbol{I_{local}}$ of the point $p$. The projection histogram of point $p$ is defined as the statistical histogram of $\boldsymbol{u}$.

Whether the sampling point is near or far away from the object surface leads to different characteristics of the projection histogram. Fig. 4 illustrates the projection histograms for the cases of non-surface and on-surface points. For the non-surface point, the observations from different views are irrelevant, as indicated by its scattered projection histogram. For the point on the object surface, the observations from different views are consistent and its projection histogram is centralized. As the color consistency of the projection histogram implies scene geometry and texture information, for each spatial point, we propose 'spatial color priors' based on the information in its projection histogram.

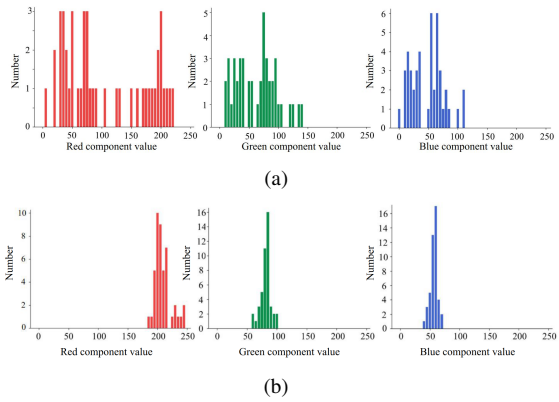



Fig. 4: Spatial color priors (the histogram of projected pixels, which comes from 45 projection views) for (a) a non-surface point and (b) an on-surface point of the scene of Fig. 2. From left to right: observed color histogram in red, green, and blue of the point when back projected to input views. Note that the histogram for the non-surface point (a) is distributed, while the histogram for the on-surface point (b) is centralized thus reference color can be robustly estimated based on our proposed spatial color priors. The other non-surface points and on-surface points have the similar situation.

If the point is on a Lambertian surface, the projection pixels are similar except occluded pixels. As the occluded pixels are irrelevant to other projection pixels, they are meaningless noise for spatial color priors. To handle it, we adopt a patch feature filter to remove occluded pixels from the projection histogram. The local image patches of the same 3D point in different perspectives are expected to be similar except for occlusion, which is suitable for occlusion removal. The $3 \times 3$ patches of the half size picture is used as pixel features, since the downsampled image has bigger receptive field with the same local patch size. The patch feature of projection pixels is compared with the current view. We calculate the $l_2$ norm and remove pixels whose differences with the current view are bigger than the threshold. The proposed patch filter is a simple but effective method to handle multiple occlusion in surrounding scenes. It is not needed to be very accurate since the residual color prediction will compensate for the small bias.

For training, the patch feature of the current view is extracted from the training images. For inference, the patch feature of the current view is extracted from the predicted radiance color $C$ (see Sec. III-C). With the patch feature filter, the occluded pixels can be removed. Afterward, we

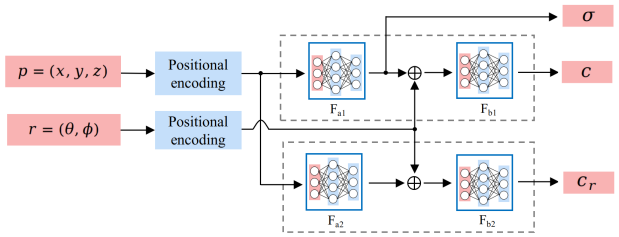
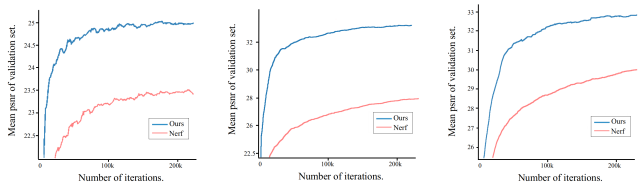



Fig. 5: The network architecture. The input is position ($x$,$y$,$z$) and viewing direction ($\theta$,$\phi$). The positional encoding of the input location $p$ is passed through 8 fully-connected ReLU layers, each with 256 channels ($F_{a1}$ and $F_{a2}$). Then the output 256 feature is combined with positional encoding of the input viewing direction $r$ and is passed through 4 fully-connected ReLU layers, each with 128 channels ($F_{b1}$ and $F_{b2}$). The output is density $\sigma$, radiance color $c$ and residual color $c_r$.




(a) Attic (b) Library (c) Family

Fig. 6: The convergence curve of NeRF and our proposed method in 'Attic' (a) and 'Library' (b) from casual 3D dataset [11] and 'Family' (c) from tank and temples dataset [18]. Both methods use the same training parameters. The batchsize is 1024 and the learning rate is $5^{-4}$. Our method achieves faster convergence under the same number of iterations.

calculate the reference color $c_{ref}$ through a voting strategy based on the remaining projected pixels $\mathbf{u}'$. Although we remove the occluded pixels by feature filter, some projection pixels with strong reflectance may still influence the reference color calculation. Thus we calculate the mean value of the $\mathbf{u}'$, then remove the values that are bigger than the threshold from the mean. The pixels with strong reflectance are removed by the voting strategy. Then we calculate the reference color by the mean value of the remaining pixels. Note that without feature filter, the residual learning for spatial color priors already improves the performance obviously in most areas but introduces small artifacts in occlusion. We introduce the feature filter to handle the occlusion. However, directly using the feature filter will bring worse results (Fig. 14), which is because although the feature filter provides more accurate reference color. It also makes the projection histogram of some non-surface points more centralized, which leads to less accurate density prediction. The feature filter must be combined with the joint-training of Eq. 8 to enhance the robustness of density prediction. Then the artifacts in occlusion will be dropped successfully.

### C. Residual Color Learning

With the reference color calculation of spatial points, we propose a residual color learning scheme to apply the spatial color priors for novel view synthesis. For each spatial point, we calculate its reference color $c_{ref}$ based on spatial color priors as shown in Sec. III-B and predict its residual color $c_r$ by the SRN $F_\theta$. The reference color and residual color are combined as the predicted color $c$ for volumetric rendering of color $\bar{C}_R$ at ray $r$ as shown here:

$$\begin{aligned} c^i_{com} &= c^i_{ref} + c^i_r, \\ \bar{C}_R(r) &= \sum_{i=1}^{N} T_i\big(1-\exp(-\sigma_i\delta_i)\big)c^i_{com}. \end{aligned} \tag{5}$$

The pixel colors of different views are similar for the points on Lambertian surface. With robust reference color calculation, the residual color predictions of different views are much smaller than original radiance color prediction. The learning task of complex high-frequency texture details is simplified to learning the residual color that is close to 0 for most spatial points, reducing the burden of the network significantly.

$$L = \sum_{r\in R} \|\bar{C}_R(r) - C(r)\| \tag{6}$$

However, we also observe that learning the network based on residual color merely as shown in Eq. 6 may lead to overfitting as non-surface points may be assigned to non-zero density if its reference color is similar to the target color. The radiance color and residual color can have respective density prediction. However, to enhance the robustness of density prediction after introducing feature filter, we propose a joint-training scheme, which is leveraging radiance color loss for density prediction by learning both residual color and radiance color with the same density:

$$\begin{aligned} \bar{C}_W(r) &= \sum_{i=1}^{N} T_i\big(1-\exp(-\sigma_i\delta_i)\big)c_i, \\ \bar{C}_R(r) &= \sum_{i=1}^{N} T_i\big(1-\exp(-\sigma_i\delta_i)\big)(c^i_{ref} + c^i_r). \end{aligned} \tag{7}$$

$\delta_i$, $c_i$ and $c^i_r$ are the outputs of fully connected network $F_\theta(p_i, d_r)$. $\delta_i$ is the density prediction. $c_i$ and $c^i_r$ are the radiance color and residual color output respectively. The network is trained jointly by the rendering loss of both radiance image $\bar{C}_W(r)$ and residual image $\bar{C}_R(r)$:

$$L = \sum_{r\in R} \|\bar{C}_W(r) - C(r)\| + \sum_{r\in R} \|\bar{C}_R(r) - C(r)\|. \tag{8}$$

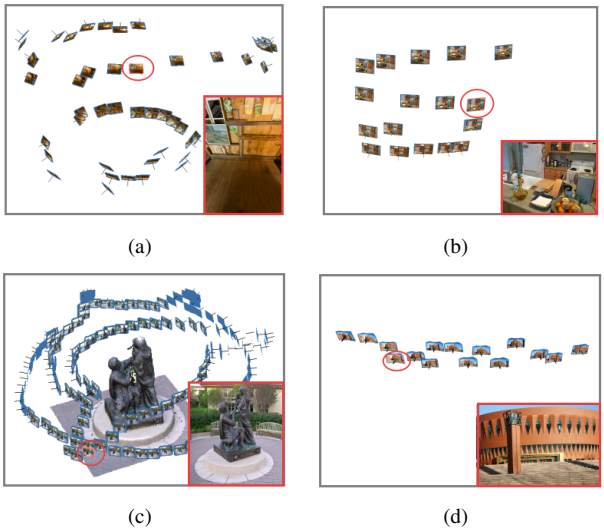
The proposed residual color learning scheme greatly reduces the burden of network. As a result, our proposed method achieves better performance and converges with fewer iterations than NeRF (Fig. 6).

(a) Reference image, PSNR: 30.08, SSIM: 0.950. (b) Residual image. (c) Our result, PSNR: 33.80, SSIM: 0.965.

Fig. 7: Illustration of rendering result decomposition. The residual image corrects the distortion in reference image caused by wrong projection pixels (red block) and adds view-dependent light shadows (green block). After the remedy of residual image, the PSNR of rendering result is raised from 30.08 to 33.80.

### D. Implementation Details

Following NeRF [24], we train a SRN for every single scene supervised by the input views. The network architecture is illustrated in Fig. 5. In the training step, pixel rays are randomly sampled from the training views. A hierarchical sampling strategy proposed in NeRF[24] is applied to sample the volumetric space more efficiently. It optimizes two networks: one coarse and one fine. The coarse network uses stratified sampling and the fine network uses a more informed sampling based on the output of coarse network. This procedure allocates more samples to regions we expect to contain visible content. Spatial color priors are computed for all sampling points in the training stage while only points with a weight ($w_i$ in Eq. 2) larger than $10^{-3}$ in inference for efficiency.



(a) (b) (c) (d)

Fig. 8: The input scenes and inference viewpoints of 4 kinds of representative input scenes. The input images are displayed according to their camera pose. (a) is indoor surrounding data from casual 3D [11] dataset (Fig. 8(a)). (b) is forward-facing data from LLFF [23] (Fig. 8(c)). (c) is encircling data from tanks and temples [18] dataset (Fig. 8(d)). (d) is self-collected outdoor large-scale data (Fig. 9(a)).

## IV. Experiment

For a fair comparison with previous methods, we evaluate our method on various datasets: forward-facing data from LLFF [23], synthetic data from NeRF [24], indoor surrounding data from casual 3D [11] dataset, self-collected outdoor large-scale data ('Auditorium' and 'Theater' in Table I), and encircling data from tanks and temples [18] dataset. Fig. 8 shows the different shooting trajectories for different kinds of data. In the following, both quantitative and qualitative evaluations are implemented to verify the proposed method's performance.

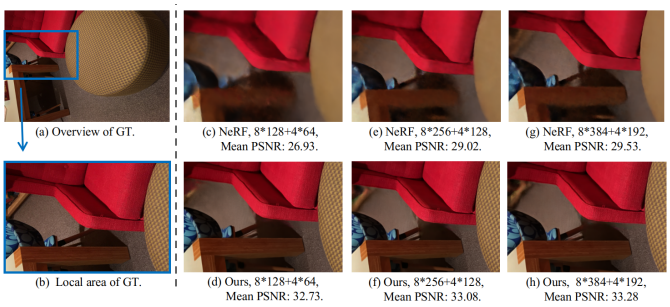



Fig. 9: Comparison between NeRF and our method in 'Library' from casual 3D dataset [11]. Both methods use the same codebase with 8+4 fully-connected ReLU layers (introduced in Fig. 5). We test their performance with the different hidden channels. (a,b) is the overview and local area of the groundtruth. (c,e,g) is the NeRF results while (d,f,h) is our results. The PSNR value is the mean result of the validation set.

### A. Quantitative Evaluations

The quantitative evaluation is evaluated using the PSNR, SSIM, and LPIPS [35]. The smaller value of PSNR and SSIM implies higher accuracy while the higher value of LPIPS implies better visual quality. We compared our method with previous state of the arts including SRN [31], NV [22], LLFF [23] and NeRF [24] as shown in Table I.

For the simple scenes with a small scope of view, e.g., 'Room' and 'Fortress' from LLFF dataset, NeRF achieves good performance with enough memory capacity. Spatial color priors helps to reveal high frequency details and the improvement is relatively small. For complex scenes with large-scale surrounding views, e.g., 'Library' and 'Attic' from casual 3D dataset [11], NeRF performs badly due to the limitation of network size. Our proposed method achieves much better performance since the proposed spatial color priors help to reduce the network capacity requirements for large scale scenes. As shown in Fig.9, NeRF achieves better performance with the growth of network size, which supports that the rendering quality of NeRF is restricted by its network capacity. However, larger memory size requires more complexity, which restricts the size from increasing too much. Also the improvement brought by network growth is small. On the other hand, with the help of the proposed spatial color priors, the requirement for network capacity is reduced a lot and our proposed residual

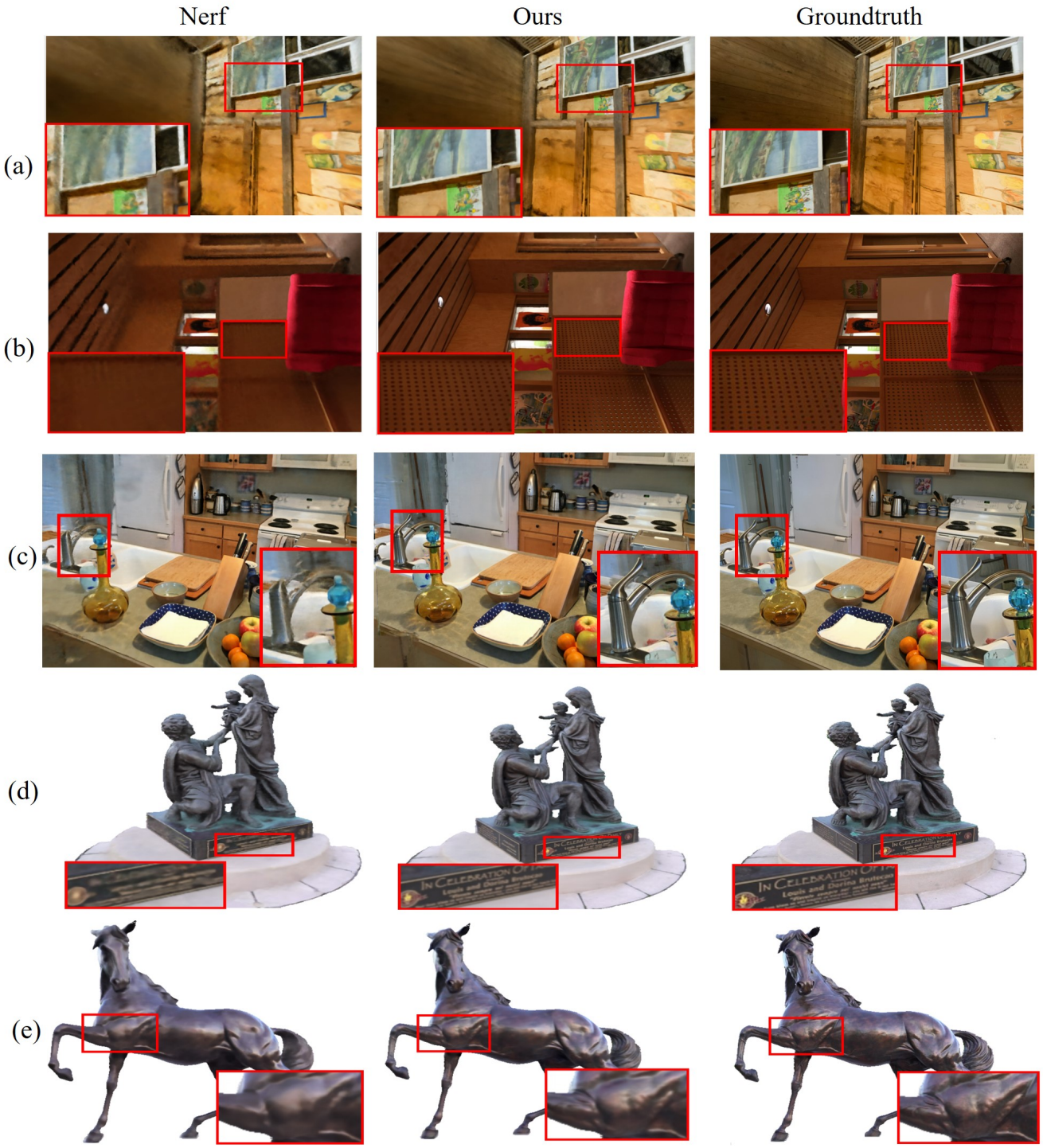



Fig. 10: Qualitative evaluations compared with previous method NeRF [24] on public datasets: (a,b,c) are from casual 3D [11], (d,e) are from tanks and temples [18]. Experiments show that our residual learning scheme based on the proposed spatial color priors produces clearer details compared with previous state-of-the-art method.

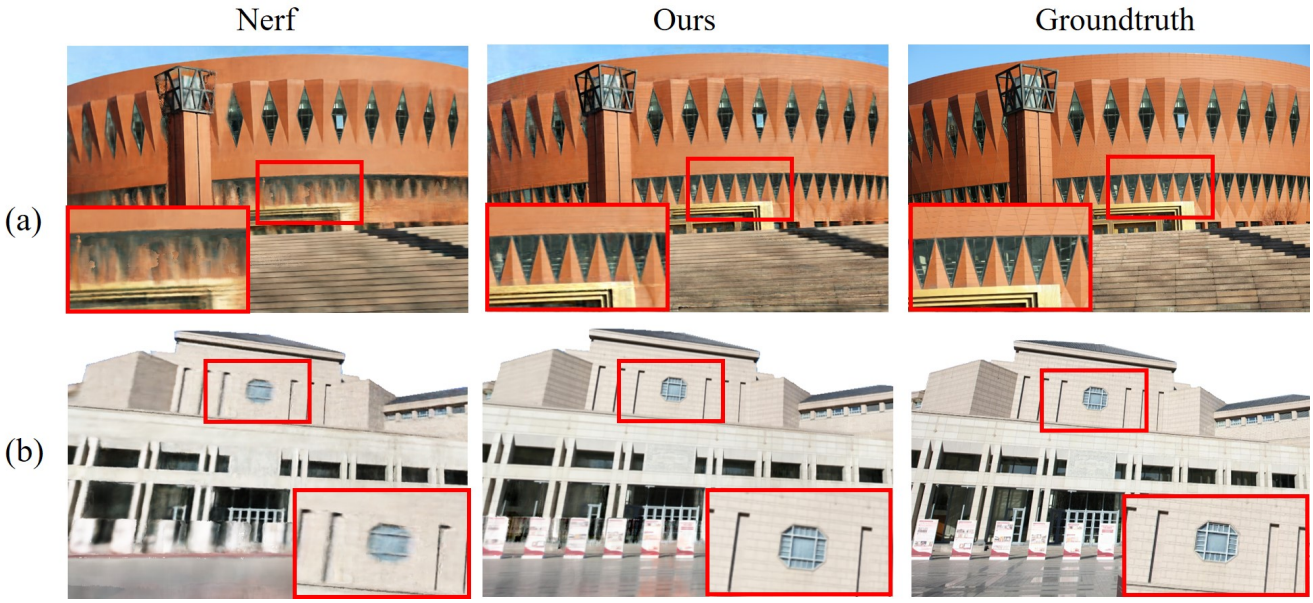



Fig. 11: Qualitative evaluations compared with previous method NeRF [24] on self-collected large-scale outdoor scenes 'Theater' (a) and 'Auditorium' (b). Experiments show that our method achieves competitive performance in large-scale scene than NeRF.

TABLE I: Quantitative evaluations on public datasets in terms of three metrics (PSNR (↑), SSIM (↑) and LPIPS (↓)). The scores are the mean value of all testing images.

| | Room [23] | | | Fortress [23] | | | Drums [24] | | | Ship [24] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| SRN [31] | 27.29 | 0.883 | 0.240 | 26.63 | 0.641 | 0.453 | 17.18 | 0.766 | 0.267 | 20.60 | 0.757 | 0.299 |
| NV [22] | | - | | | - | | 22.58 | 0.873 | 0.214 | 23.93 | 0.784 | 0.276 |
| LLFF [23] | 28.42 | 0.932 | 0.155 | 29.40 | 0.872 | 0.173 | 21.13 | 0.890 | 0.126 | 23.22 | 0.823 | 0.218 |
| NeRF [24] | 32.70 | 0.948 | 0.178 | **31.16** | 0.881 | 0.171 | 25.01 | 0.925 | **0.091** | 28.65 | 0.856 | 0.206 |
| Ours | **32.89** | **0.955** | **0.151** | 31.15 | **0.905** | **0.144** | **26.06** | **0.934** | 0.099 | **30.09** | **0.863** | **0.199** |

| | Library [11] | | | Attic [11] | | | Kitchen [11] | | | Troll [11] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| NeRF [24] | 29.02 | 0.784 | 0.481 | 23.64 | 0.744 | 0.535 | 26.13 | 0.826 | 0.334 | 26.04 | 0.643 | 0.515 |
| Ours | **33.08** | **0.926** | **0.183** | **25.25** | **0.780** | **0.424** | **27.70** | **0.878** | **0.229** | **26.74** | **0.696** | **0.364** |
| | Auditorium | | | Theater | | | Family [18] | | | Horse [18] | | |
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| NeRF [24] | 21.81 | 0.766 | 0.334 | 21.50 | 0.666 | 0.425 | 31.07 | 0.924 | 0.126 | 30.41 | 0.932 | 0.144 |
| Ours | **23.58** | **0.834** | **0.210** | **23.38** | **0.691** | **0.323** | **32.71** | **0.953** | **0.069** | **31.08** | **0.948** | **0.104** |

TABLE II: PSNR comparison between NeRF and our method at different resolutions. 'library' and 'attic' are surrounding indoor data from casual 3D [11] dataset.

| | Library 960×720 | Library 1200×900 | Attic 592×880 | Attic 886×1330 |
|---|---|---|---|---|
| NeRF | 29.02 | 28.09 | 23.44 | 23.64 |
| Ours | 33.08 | 33.21 | 24.93 | 25.25 |

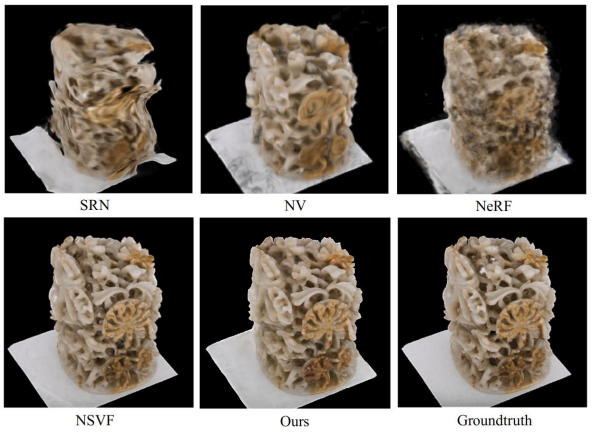
learning scheme achieves much better quality even with a smaller network.

We also compared the performance of NeRF and our proposed method for rendering novel views at different resolutions as shown in Table II. For higher resolution, the gap between our method and NeRF is larger, demonstrating the ability of our proposed method for generating realistic rendering results at high resolution.

### *B. Qualitative Evaluations*

The reference color is calculated based on the spatial color priors. It is close to the real rendering result in most area and may suffer from distortion in the corner due to the wrong projection of pixels. The residual color prediction has the potential to partially correct these issues. Also, it can add different light shadows from different perspectives (as we can see from Fig. 7). The following qualitative evaluations show that our proposed method achieves robust reference color calculation and high-quality rendering performance.

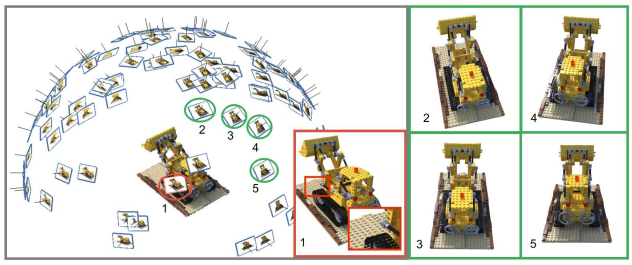
- **Overall Performance.** Our method puts forward a residual-based framework to utilize spatial color priors, and such an idea was put into practice on NeRF. Fig. 10 and Fig. 11 show the qualitative comparison with NeRF [24] in different kinds of scenes. For NeRF, the high-frequency information of texture is difficult to learn. It loses detailed information. Our method turns the high-frequency learning task into a low-frequency one. The residual color only needs to memorize the



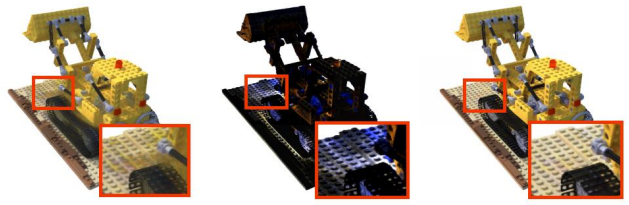

(a)


(b)

Fig. 12: Qualitative evaluations compared with methods SRN [31], NV [22], NeRF [24] and newly published NSVF [21] on 'Jade' from BlendedMVS dataset [34] (a) and 'Family' from tank and temple dataset [18] (b). Experiments show that our method achieves much better performance than previous method (SRN, NV and NeRF) and comparable performance with NSVF.



(a) Input scene. The image in red box is the inference viewpoint. The images in green box are part of its adjacent views. The zoomed-in area is occlusion region, which is visible in the inference viewpoint and is invisible in the presented adjacent views.



(b) Without feature filter and joint training

(c) With feature filter and joint training

Fig. 13: Comparison of rendering images with or without feature filter and joint-training, respectively. From left to right of (b,c): reference images, residual images, and result images. The reference color image of (b) has obvious artifacts caused by occluded projection pixels, whereas the reference color image of (c) is more accurate. As a result, with feature filter and joint-training, the residual color image does not need to make up for mistakes and the result image has improvement in occlusion.

low-frequency information, since the calculated reference color already captures the high-frequency texture of the scene. As a result, the high-frequency information is better preserved and our method has clearer details. In particular, NeRF is only able to recover a limited scene otherwise with low quality. For the large-scale scenes, NeRF tends to perform badly due to the network size limitation. While our method can effectively deal with large-scale scenes, because the proposed spatial color priors help to reduce the network capacity requirements. E.g., for the complex indoor scenes of Fig. 10 (b,c) and large-scale outdoor scene Fig. 11 (a,b), our result shows significant improvement in the high-quality rendering. Also Fig. 12 shows the qualitative comparison with SRN [31], NV [22], NeRF [24] and newly published NSVF [21]. Experiments show that our method achieves much better performance than previous method (SRN, NV and NeRF) and comparable performance with NSVF.

- **Occlusion Handling.** The reference color is calculated by projected pixels. If there is occlusion, the wrong projection pixel may influence the quality of the reference image. We apply a feature filter and joint-training to handle this limitation. Fig. 13 (b) shows that without occlusion detection, the reference color suffers from obvious artifacts due to the wrong projection of pixels. The reference color suffers from obvious distortion due to the wrong projection of pixels. The residual color prediction owns potential to partially correct these issues, yet the remedy can not be perfect, where the resultant image still suffers certain artifacts. With our patch feature filter and joint-training, the calculation of reference color is not affected by occlusion, as shown in Fig. 13 (c). Meanwhile, as demonstrated in Fig. 14, the performance gain mainly originates from the residual learning scheme. The feature filter works for handling the occlusions. Thus it needs to be incorporated with the joint-training.

(a) NeRF PSNR: 33.64 (b) No radiance color loss PSNR: 36.06

(c) No joint-training PSNR: 27.50 (d) Complete model PSNR: 36.51

Fig. 14: (a) is the result of original NeRF. (b) uses residual learning to take advantage of spatial color priors without additional radiance color loss. It already achieves better performance than NeRF while the red boxes shows that the occlusion area is blurry (top box) or has ghosting. (c,d) introduce extra radiance color loss in addition to residual learning scheme. (c) uses feature filter without joint-training, which means the radiance color and residual color uses own density prediction. (d) uses feature filter and joint-training together. The comparison shows that the performance improvement is mainly from residual learning scheme. While feature filter must be combined with joint-training for remove artifacts caused by occlusion.

(a) NeRF (b) Ours (c) Groundtruth

Fig. 15: Comparison of Lambertian surface (red block) and reflective surface (green block). This figure shows that our residual-based method outperforms the NeRF in the Lambertian surface, whereas for the reflective surface our method works similarly with NeRF.

### C. Limitations

Although our proposed method can generate high-quality novel views and outperform the existing state-of-the-art method, the performance of reflection area is still poor. The residual color learning scheme turns the high-frequency learning task into low-frequency in Lambertian surfaces. However, in the reflective area, observations from different view angles are very different, so the proposed residual-based method does not have an advantage over NeRF. From Fig. 15 we can see, our proposed method works best for Lambertian surfaces while it works similarly with NeRF for a reflective surface. It is still challenging to keep precise details in the specular area.

The proposed residual learning for spatial color priors is both efficient and effective. It is inspiring for neural rendering and can be easily introduced into other frameworks. At the same time, the proposed additional training loss of radiance color increases the computational complexity mainly for removing small artifacts caused by occlusion. The extra computation burden is inefficient for performance improving. Without the additional MLP for radiance color, our proposed residual learning scheme still outperforms NeRF in most areas while just introducing small artifacts in occlusion. The existing methods of neural rendering still suffer from poor rendering quality which must be addressed for practical usage, so this paper mainly focuses on synthesizing high-quality novel views without caring about the complexity. In future, we will explore more efficient method to handle occlusion in our residual learning framework.

## V. Conclusion

In this paper, aiming to improve the immersive experience of novel view synthesis from free-moving cameras, we argue that conventional scene representation networks that try to memorize the texture details and geometry of the environment using a fully connected network fails to preserve high-frequency details in practice, and propose a novel framework that learns residual colors instead by employing the proposed spatial color priors as a reference for radiance color prediction. Experiments demonstrate that the proposed approach achieves more visually pleasant results that previous state of the arts, especially for environments that contain complex textures and large surface area. The proposed approach works best for Lambertian surfaces and only achieves comparable performance with the previous approach for non-Lambertian surfaces. Detecting such areas using segmentation approaches [3] might be helpful for this challenge, which will be left for future investigations.

## VI. Acknowledgement

This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 62125106, 61860206003 and 62088102, in part by Shenzhen Science and Technology Research and Development Funds (JCYJ20180507183706645), in part by Beijing National Research Center for Information Science and Technology (BNRist) under Grant No. BNR2020RC01002.

## References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 696–712. Springer, 2020.

[2] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)*, 35(6):1–13, 2016.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[4] Paul Debevec, Yizhou Yu, and George Boshokov. Efficient view-dependent ibr with projective texture-mapping. In *EG Rendering Workshop*, volume 4, 1998.

[5] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005.

[6] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019.

[7] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3d photography. *ACM Transactions on Graphics (TOG)*, 36(6):1–15, 2017.

[11] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3D Photography. 36(6):234:1–234:15, 2017.

[12] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[13] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, and Thomas Funkhouser. Adversarial texture optimization from rgb-d scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020.

[14] Michal Irani, Tal Hassner, and P Anandan. What does the scene look like from a scene point? In *European Conference on Computer Vision*, pages 883–897. Springer, 2002.

[15] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

[16] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*, pages 365–376, 2017.

[17] Matthew Klingensmith, Ivan Dryanovski, Siddhartha Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. 07 2015.

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[19] O. Kähler, V. Adrian Prisacariu, C. Yuheng Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1241–1250, Nov 2015.

[20] Christoph Lassner. Fast differentiable raycasting for neural rendering using sphere-based representations. *arXiv preprint arXiv:2004.07484*, 2020.

[21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33, 2020.

[22] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

[23] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

[25] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[26] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[27] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017.

[28] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[29] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021.

[30] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Computer Vision – ECCV 2020*, pages 623–640, 2020.

[31] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019.

[32] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2634, 2017.

[33] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[34] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[36] Dawei Zhong, Lei Han, and Lu Fang. idfusion: Globally consistent dense 3d reconstruction from rgb-d and inertial measurements. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 962–970, 2019.

[37] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.


**Lei Han** studied Electrical Engineering at the Hong Kong University of Science and Technology (China) and Tsinghua University (China). He received the B.S. degree in July 2013 and joined the Department of Electrical Computing Engineering at the Hong Kong University of Science and Technology in September 2016, where he is pursuing the PhD degree. His current research focuses on multi-view geometry and 3D computer vision.

**Dawei Zhong** studied Data Science and Information Technology at Tsinghua-Berkeley Shenzhen Insititute(TBSI) of Tsinghua University. He received the Bachelor degree from Tongji University in July 2019. His current research is about 3D computer vision.

**Lin Li** received the B.S. and the integrated M.S. and Ph. D. degrees from the Department of Electronic Information Technology and Instrument Institute of ZheJiang University in 2010 and 2015, respectively. She is engineer in Huawei.

**Kai Zheng** received the B.S from Harbin Institute of Technology,Weihai,China in 2016 and Master degree from The computer major at the Harbin Institute of Technology (HIT),Harbin,China,in 2018.He is currently a Research and Development Enginneer in Hisilicon.His research interest include Smart Industry, 3D Reconstruction and Auto Vehicle.

**Lu Fang** is currently an Associate Professor in the Department of Electronic Engineering, Tsinghua University. She received Ph.D from the Hong Kong Univ. of Science and Technology in 2011, and B.E. from Univ. of Science and Technology of China in 2007. Her research interests include computational imaging and visual intelligence. Dr. Fang is currently IEEE Senior Member, Associate Editor of IEEE TIP and TMM.