Layout-Guided Multi-Object Scene Novel View Synthesis Network: Postprint

Authors: Gao Xiaotian, Zhang Qian, Lü Fan, Hu Fuyuan, Hu Fuyuan

Date: 2022-04-07T16:20:37Z

Abstract

The novel view synthesis task refers to generating novel-view images of a scene from multiple reference images. However, in multi-object scenes, inter-object occlusion leads to incomplete acquisition of object information, resulting in artifacts and misalignment issues in the generated novel-view scene images. To address this problem, this paper proposes a novel view synthesis network guided by scene layout maps and introduces a newly annotated multi-object scene dataset (Multi-Objects Novel View Synthesis, MONVS). First, multiple layout map information of the scene and corresponding camera pose information are input into a layout map prediction module to compute the scene layout map information at the novel view; then, using the annotated object bounding box information in the scene to construct object sets for different objects, each object' s information in the novel-view scene is generated via a pixel prediction module; finally, the obtained novel-view layout map and individual object information are input into a scene generator to construct the novel-view scene image. Comparisons with several state-of-the-art methods were conducted on the MONVS and ShapeNet Cars datasets. Experimental data and visualization results demonstrate that the proposed method achieves favorable performance on both datasets in novel-view image synthesis for multi-object scenes, effectively resolving the problems of artifacts in generated images and inaccurate positional information of multiple objects within the scene.

Full Text

Multi-object scenes novel view synthesis via layout projection

Gao Xiaotian[1] ,[1] , Zhang Qian[2], Lyu Fan[2], Hu Fuyuan[1] ,[1] †

[1]Suzhou University of Science and Technology

a. College of Electronic & Information Engineering

b. Suzhou Key Laboratory for Virtual Reality Intelligent Interaction & Application Technology
c. Suzhou Key Laboratory for Big Data & Information Service
Suzhou, Jiangsu 215009, China
[2]College of Intelligence & Computing, Tianjin University
Tianjin 300354, China

Abstract

Novel view synthesis (NVS) aims to generate images of a scene from new viewpoints given multiple reference images. However, multi-object scenes suffer from inter-object occlusions and incomplete object information, leading to artifacts and misalignment in synthesized images. To address this problem, we propose a layout-guided novel view synthesis network and introduce a new multi-object scene dataset (Multi-Objects Novel View Synthesis, MONVS). Our approach first computes the layout map for the novel view by feeding multiple scene layout maps and corresponding camera poses into a layout prediction module. Next, we construct object sets using annotated bounding box information and employ a pixel prediction module to generate each object' s appearance from the new viewpoint. Finally, the predicted layout map and individual object information are fed into a scene generator to construct the complete scene image. Experimental comparisons with state-of-the-art methods on both MONVS and ShapeNet Cars datasets demonstrate that our method achieves superior performance in multi-object novel view synthesis, effectively eliminating artifacts and improving positional accuracy of objects in the generated scenes.

Keywords: multi-object scene; occlusion; image artifacts; layout; novel view synthesis

0 Introduction

Novel view synthesis (NVS) generates images of objects or scenes from arbitrary viewpoints given multiple input images and their camera poses. This task has widespread applications in virtual reality, robotics, and static image animation. By avoiding complex 3D model construction during arbitrary viewpoint generation, NVS improves efficiency and has attracted significant research attention.

Early NVS methods relied on camera imaging principles, using interpolation in pixel or ray space to synthesize new views [?]. With deep learning advances, convolutional networks were employed to generate novel views of rigid objects [?], though they struggled with fine details and produced blurry contours. Subsequent work incorporated geometric priors [?, ?], projecting input image pixels onto output views based on object geometry or 3D point cloud information [?, ?]. While effective for single objects, these approaches treat the scene as a monolithic entity [?], failing to extract features of occluded objects or learn their geometry in realistic multi-object scenes. This results in artifacts, blurring, and even object disappearance, as illustrated in Fig. 1(a).

To combat artifacts, depth maps were used as prior information to guide novel view synthesis [?]. However, depth acquisition requires specialized equipment and cannot resolve boundary blurring caused by inter-object occlusion. Layout maps [?], which contain object categories and bounding boxes, are more readily obtainable. Inspired by layout-to-image generation, we propose a layout-guided NVS network (Fig. 1(b)). Unlike previous deep learning-based NVS methods, our approach eliminates the need for complex depth maps or point clouds and generalizes to multi-object scenes.

Our method first computes object motion trajectories from multi-view layout information, deriving the novel view layout through camera pose relationships. This addresses positional inaccuracies caused by occlusion. Based on the layout map, we decompose the multi-object NVS task into multiple single-object generation tasks. To preserve object details, a pixel predictor progressively refines results as input images transform. Finally, the predicted layout guides the scene generator to synthesize the complete image.

1.1 Layout-to-Image Generation

Current image generation often leverages auxiliary information [?, ?] (e.g., category labels, textual descriptions [?], scene graphs) as prior knowledge. However, priors like depth maps are constrained by acquisition conditions. Layout maps offer an accessible alternative.

In [?], layout maps and object information guide text-to-image and scene-graph-to-image synthesis by matching object shapes to feature repositories. [?] proposes manipulating layout boundaries to reconstruct scenes with modified object styles and positions. Recent work [?, ?] optimizes layout-to-image networks by estimating scene layouts from source views with depth constraints to generate room layouts. Inspired by these methods, we exploit the relationship between viewpoint changes and object bounding box transformations to derive novel view layouts from known viewpoints, guiding NVS.

1.2 Novel View Synthesis

NVS synthesizes new images from arbitrary viewpoints given multiple inputs and camera poses. Early methods [?, ?] warped input pixels via geometric mapping or interpolation, but produced poor texture details and could not hallucinate missing pixels. Deep learning approaches [?] directly generated novel views using CNNs, achieving good results on single rigid objects (e.g., chairs, cars) but still failing on occluded regions. Sun et al. [?] introduced optical flow prediction and pixel generation modules with self-learned confidence aggregation, yet detail rendering remained suboptimal.

Depth-based methods [?] and 3D structure priors [?] mapped source pixels to target views [?]. Some approaches [?, ?] reconstructed 3D geometry before synthesizing images from novel poses, requiring extensive training time and resources. Mildenhall et al. [?] proposed Neural Radiance Fields (NeRF), which uses a 5D vector (spatial coordinates + viewing direction) to output color and volume density, achieving impressive results in complex scenes. However, NeRF requires numerous input views per scene and lacks generalization. Yu et al. [?] improved this with pixelNeRF, enabling few-shot reconstruction with better efficiency and generalization, yet still cannot resolve occlusion-induced artifacts.

2 Layout-Guided Novel View Synthesis

We propose a layout-guided multi-object NVS method. The layout prediction module provides novel view layout information, while the pixel predictor generates individual object images. These are fed into the scene generator to produce the final image. The overall architecture is shown in Fig. 2.

Given multi-view images with layout maps $\{L_i\}_{i=1}^n$, where each L_i contains bounding box information (x_i, y_i, h_i, w_i) for every object o_i in image i, we input the layout maps into the layout prediction module to compute the novel view layout L_t. The model samples each object instance o_i from input images and concatenates them with camera pose matrices along the channel dimension to construct input tensors. These tensors feed into the pixel predictor to generate per-object images $\{I_i^t\}_{i=1}^n$ for the novel view. Finally, L_t and $\{I_i^t\}$ enter the scene generator, where object images pass through an encoder and fusion module to produce a combined feature representation, which the decoder uses to generate the scene image.

2.1 Layout Prediction Module

Through camera calibration [?, ?], we map objects from multiple input images into a unified world coordinate system. During camera movement, each object follows an elliptical trajectory. For a single object, we model its initial trajectory ellipse as:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

where A, B, C, D, E, F are ellipse parameters. Using Faster R-CNN to obtain layout information from input images, we compute these coefficients from multi-view layout maps. The input layout maps are processed by the layout prediction module to derive object trajectories and compute the novel view layout (Fig. 3).

Layout information constructs bounding box sets per object category, yielding center coordinate sets $\{(x_i, y_i)\}_{i=1}^n$. We use least squares fitting to solve for ellipse parameters A, B, C, D, E, F.

Occlusion introduces errors in annotated bounding boxes. We propose an iterative refinement method to correct trajectories and bounding boxes. First, we compute the minimum distance $d_{\min}$ between each center coordinate and the trajectory curve, comparing it against a threshold to identify coordinates requiring correction. These coordinates approach the trajectory curve with step size $d_{\min}$. After each update, we compute distances to the previous bounding box' s four vertices and expand the box using the maximum distance as a constraint. This iterative process optimizes the solution. The objective function is:

$$f = \sum_{i=1}^{n} \left(\frac{(x_i - x)^2}{w_i^2} + \frac{(y_i - y)^2}{h_i^2} \right)$$

where (x, y) are points on the elliptical trajectory.

Generally, an object' s bounding box size varies linearly with camera distance. We split the trajectory into left and right halves. On both sides, the object' s y-coordinate correlates with bounding box dimensions: coordinates closer to the ellipse's lower half indicate nearer objects with larger boxes, while distant objects have smaller boxes. To compute the novel view layout, we transform corrected bounding boxes and camera pose information between coordinate systems:

$$\begin{cases} y = k_1 w + b_1 \\ y = k_2 h + b_2 \end{cases}$$

where k_1, b_1, k_2, b_2 are parameters solved from the trajectory. Using the novel view camera pose coordinates, we compute corresponding object bounding boxes.

2.2.1 Pixel Predictor

Existing layout-to-image methods use convolutional feature extraction, which often focuses on texture transfer while losing geometric details. We introduce a pixel predictor that directly regresses pixel values to predict missing pixels in target views, preserving object textures. The layout map' s category information constrains object geometry for structural consistency.

The predictor is an encoder-decoder network with Convolutional Long-Short-Term Memory (ConvLSTM) in the bottleneck, passing convolutional features to corresponding deconvolution layers for richer information. Multi-view input images each generate novel view predictions, which are aggregated via averaging to produce the final target image (Fig. 4).

First, we vectorize discrete camera poses using one-hot encoding into an N-dimensional vector based on total camera count N, computing pose difference $P_{\text{diff}} = P_{\text{target}} - P_{\text{input}}$. This difference is tiled spatially to $H \times W \times v$, where v is the pose vector dimension. Input images are cropped using bounding boxes to obtain N object image sets $\{I_i^s\}_{i=1}^{N}$, which are bilinearly interpolated and concatenated with pose tensor P_{diff} along channels before feeding into the pixel predictor:

$$I_i^t = \Phi_{\text{pixel}}(I_i^s \oplus P_{\text{diff}})$$

where Φ_{pixel} is the pixel predictor, I_i^t is the predicted image, and $\oplus$ denotes channel-wise concatenation. Predictions are aggregated as:

$$I_{\text{target}} = \frac{1}{n}\sum_{i=1}^{n} I_i^t$$

The pixel generator is trained to minimize:

$$\mathcal{L}_{\text{pixel}} = \sum_{i=1}^{N} \|I_i^t - I_{\text{target}}\|^2$$

As shown in Fig. 5, feature-based methods on ShapeNet only produce car outlines, while our pixel-based approach preserves fine textures.

2.2.2 Scene Generator

Predicted object images and bounding boxes $\{B_i\}_{i=1}^N$ construct object feature maps F_i, which feed into the scene generator. Object categories y_i are encoded via Word Embedding and concatenated with object features F_i within their bounding boxes:

$$Z_i = F_i \oplus \text{Embed}(y_i)$$

where $\oplus$ is the vector concatenation operator and $\text{Embed}(y_i)$ copies object information into the bounding box region.

To encode all object instances at desired positions, we add a multi-layer ConvLSTM after the decoder to fuse sampled object features, outputting a hidden layout map H containing all object positions, categories, and features. This H is decoded to generate the target image.

To guide realistic synthesis and prevent feature loss in H, we crop generated images I_{gen} using the same bounding boxes to obtain single object images I_i', which feed into a latent code estimator to produce mean and variance vectors. The mean vector serves as regressed latent code Z_i' and compares against the pixel predictor output Z_i:

$$\mathcal{L}_{\text{latent}} = \sum_{i=1}^{N} \|Z_i' - Z_i\|^2$$

Overlapping bounding boxes (occlusions) cause artifacts during fusion. We adopt VGG-19-based perceptual loss [**?**, **?**]:

$$\mathcal{L}_{\text{percept}} = \sum_j \frac{1}{C_j H_j W_j} \|\phi_j(I_{\text{gen}}) - \phi_j(I_{\text{target}})\|^2$$

where j indexes VGG-19 intermediate layers (we use layers 0, 2, 3), ϕ_j extracts features, and C_j, H_j, W_j are channel, height, and width dimensions.

3 Experimental Results

Experiments use PyTorch on Ubuntu 16.04 with four NVIDIA 1080Ti GPUs.

3.1 Datasets

We construct two datasets of varying difficulty: MONVS (Blender/Real) and ShapeNet Cars.

MONVS Blender/Real contains: (1) MONVS Blender with geometric primitives rendered in 10 random colors at random positions; (2) MONVS Real with 3 objects randomly selected from 10 real-world categories against a monochrome background. **ShapeNet Cars** synthesizes multi-object scenes by randomly placing 3 cars from 10 models on a two-tone background. Both datasets use 10 fixed camera positions with constant elevation, containing 100 scenes (1,000 images) each at 64×64 resolution. Sample images are shown in Fig. 6.

3.2 Results Analysis

We evaluate using SSIM, PSNR, and Learned Perceptual Image Patch Similarity (LPIPS). LPIPS learns a reverse mapping from generated to ground-truth images, prioritizing perceptual similarity over traditional metrics like L2/PSNR or SSIM. Lower LPIPS indicates higher similarity. Complexity is measured via Floating-point Operations (FLOPs).

Since prior work lacks identical settings, we train comparison methods using only multi-view images and poses. Testing shows 6 input images yield optimal trajectory fitting accuracy and speed. We train on 800 randomly selected images per dataset and test on 200 images at 64×64 resolution. We compare against TB-network [?], uORF-main [?], and SVNVS [?].

Figs. 7-9 visualize challenging large-viewpoint transformations. uORF-main combines single-object 3D representations with depth inference but struggles to infer object correspondences, producing unclear images (columns 3-4). TB-network generates high-quality 3D structures and voxels but loses background details, creating holes (columns 5-6). SVNVS uses self-supervised depth probability estimation but cannot accurately generate depth for large viewpoint changes, causing blurred object boundaries (columns 7-8).

Our layout-guided approach avoids 3D structure estimation and depth dependency, accurately recovering object-object and object-background relationships.

Qualitative analysis (Table 1) shows our method achieves the best SSIM, PSNR, and LPIPS across datasets. FLOPs are significantly lower as we only compute bounding boxes rather than estimating 3D information (depth/voxels), demonstrating layout priors outperform depth-based and implicit 3D methods.

3.3 Ablation Study

We validate each module on MONVS. Without layout error correction, object positions are inaccurate (Fig. 10, row 2, column 4). Without perceptual loss, severe artifacts appear (row 3, column 2). Without layout priors, objects are misplaced and incomplete (row 4, column 4). Perceptual loss constrains synthesis, eliminating artifacts from bounding box overlap during fusion.

Quantitative evaluation uses FID and LPIPS (Table 2). Layout correction improves realism by 7.9%, while perceptual loss boosts it by 58%, producing accurate colors and eliminating artifacts.

4 Conclusion

We propose a layout-guided NVS method that computes novel view layouts from multi-view layout information, preventing object loss during viewpoint transformation. Perceptual loss in the scene generator eliminates fusion artifacts. Experiments demonstrate superior performance and quality in simple multi-object scenes. Limitations include: (1) layout prediction only works for circular camera trajectories; (2) foreground-background boundary pixels remain blurry. Future work will incorporate neural radiance field methods to improve generalization on irregular capture patterns for complex outdoor scenes like gardens.

References

[1] Zhou T, Tulsiani S, Sun W, et al. View synthesis by appearance flow [C]// European conference on computer vision. Springer, Cham, 2016: 286-301.

[2] Tatarchenko M, Dosovitskiy A, Brox T. Multi-view 3d models from single images with a convolutional network [C]// European Conference on Computer Vision. Springer, Cham, 2016: 322-337.

[3] Hani N, Engin S, Chao J J, et al. Continuous object representation networks: novel view synthesis without target view supervision [J]. Advances in Neural Information Processing Systems, 2020, 33: 6086-6097.

[4] Shi Y, Li H, Yu X. Self-Supervised Visibility Learning for Novel View Synthesis [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 9675-9684.

[5] Song Z, Chen W, Campbell D, et al. Deep Novel View Synthesis from Colored 3D Point Clouds [C]// European Conference on Computer Vision. Springer, Cham, 2020: 1-17.

[6] Le H A, Mensink T, Das P, et al. Novel view synthesis from single images via point cloud transformation [J]. arXiv preprint arXiv: 2009.08321, 2020.

[7] Park E, Yang J, Yumer E, et al. Transformation-Grounded Image Generation Network for Novel 3D View Synthesis [C]// IEEE Conference on Computer Vision and Pattern Recognition. 2017: 702-711.

[8] Choi I, Gallo O, Troccoli A, et al. Extreme view synthesis [C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7781-7790.

[9] Huang P H, Matzen K, Kopf J, et al. Deepmvs: Learning multi-view stereopsis [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2821-2830.

[10] Zhao B, Meng L, Yin W, et al. Image generation from layout [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 8584-8593.

[11] Herzig R, Bar A, Xu H, et al. Learning canonical representations for scene graph to image generation [C]// European Conference on Computer Vision. Springer, Cham, 2020: 210-227.

[12] Lan H, Liu Q. A scene graph-to-image generation model for graph attention networks [J]. Chinese Journal of Image Graphics, 2020, 25(08): 1591-1603.

[13] Lan H, Chen Z, Liu Q. Text-Oriented Image Editing Based on Transformer [J/OL]. Application Research of Computers: 1-6 [2022-03-09]. http://www.arocmag.com/article/02-2022-05-032.html

[14] Sun W, Wu T. Image synthesis from reconfigurable layout and style [C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 10531-10540.

[15] Xu J, Zheng J, Xu Y, et al. Layout-Guided Novel View Synthesis from a Single Indoor Panorama [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 16438-16447.

[16] Zou C, Colburn A, Shan Q, et al. Layoutnet: Reconstructing the 3d room layout from a single rgb image [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2051-2059.

[17] Sun S H, Huh M, Liao Y H, et al. Multi-view to novel view: Synthesizing novel views with self-learned confidence [C]// Proceedings of the European Conference on Computer Vision. 2018: 155-171.

[18] Flynn J, Neulander I, Philbin J, et al. Deepstereo: Learning to predict new views from the world’ s imagery [C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 5515-5524.

[19] Azinović D, Martin-Brualla R, Goldman D B, et al. Neural RGB-D surface reconstruction [J]. arXiv preprint arXiv: 2104.04532, 2021.

[20] Guo P, Bautista M A, Colburn A, et al. Fast and Explicit Neural View Synthesis [C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022: 3791-3800.

[21] Wei X, Li J, Sun X, et al. Multi-view image generation algorithm based on hybrid generative adversarial network [J]. Chinese Journal of Automation, 2021, 47(11): 2623-2636.

[22] Mildenhall B, Srinivasan P P, Tancik M, et al. Nerf: Representing scenes as neural radiance fields for view synthesis [C]// European conference on computer vision. Springer, Cham, 2020: 405-421.

[23] Yu A, Ye V, Tancik M, et al. pixelnerf: Neural radiance fields from one or few images [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 4578-4587.

[24] Zhang J, Yu H, Deng H, et al. A robust and rapid camera calibration method by one captured image [J]. IEEE Trans on Instrumentation and Measurement, 2018, 68(10): 4112-4121.

[25] Zhao M, Liu Y, Wu G, et al. Surround camera calibration method under strong constraints [J]. Application Research of Computers, 2017, 34(11): 3463-3467.

[26] Wang L, Liu T, Dong Q, et al. Camera weighted calibration method for defocus blur estimation [J]. Journal of Computer Aided Design and Graphics, 2020, 32(3): 410-417.

[27] Yang Q, Yan P, Zhang Y, et al. Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss [J]. IEEE Trans on medical imaging, 2018, 37(6): 1348-1357.

[28] Wu C, Chen X, Ji D, et al. Image Denoising Combined with Deep Residual Learning and Perceptual Loss [J]. Chinese Journal of Image Graphics, 2018, 23(10): 1483-1491.

[29] Olszewski K, Tulyakov S, Woodford O, et al. Transformable bottleneck networks [C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7648-7657.

[30] Yu H X, Guibas L J, Wu J. Unsupervised discovery of object radiance fields [J]. arXiv preprint arXiv: 2107.07905, 2021.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.