Exploring Temporal Consistency in Image-Based Rendering for Immersive Video Transmission

Smitha Lingadahalli Ravi[*], Félix Henry[+], Luce Morin[*], Matthieu Gendrin[+]
[+]Orange Labs, 35510 Cesson Sévigné, France
[*]INSA Rennes – IETR, 35000 Rennes, France
slingada@insa-rennes.fr

Abstract

Image-based rendering (IBR) technique generates novel views by utilizing input images captured from various viewpoints to create an immersive video experience. However, current learning-based IBR methods have limitations as they only work at the still image level, and they do not maintain consistency between consecutive frames, leading to temporal noise. To address this, we propose an intra-only framework that identifies parts of input images causing temporal artifacts in synthesized views. Our method produces better and more stable novel views for immersive video transmission. We conclude that our framework is capable of detecting and correcting spatial features in still image level that produce artifacts in the temporal dimension.

Keywords : image-based rendering, temporal consistency, immersive video transmission

1 Introduction

Image-Based Rendering is a technique that involves synthesizing novel views of a scene from a set of input images. It is widely used in computer graphics and computer vision applications, such as virtual reality, video compression, and autonomous driving. In IBR methods, the source views are usually warped, resampled, and/or blended to obtain target viewpoints, which allows for high-resolution rendering. However, these methods typically require dense input views or explicit proxy geometry, making it challenging to estimate high-quality views without resulting in rendering artifacts. Earlier methods used dense sampling [2] or proxy geometry [3] to generate novel views. More recent techniques have introduced better modeling of scene structure [4], and learning-based methods have shown promising results.

Techniques that combine novel representations with differentiable rendering have produced high-quality novel views, with neural radiance fields (NeRF) [5] being the prominent one. However, NeRF requires per-scene optimization and overfitting, making it impractical for immersive video transmission. Additionally, the network parameters must be transmitted for each time instant of the video, which is not feasible. A new learning-based method called IBRNet [6] has been introduced, which combines ideas from IBR and NeRF. Unlike NeRF, IBRNet is a pure processor and does not require per-scene optimization. IBRNet is a general synthesizer that can work with any new content. The method involves selecting neighboring views of the target view, extracting dense features from each neighboring view, predicting volume densities and colors at continuous 5D locations, and compositing the colors and densities along each camera ray to produce the target image. During training, the color is rendered along each camera ray, and the mean squared error is minimized between the ground-truth pixel color and the rendered pixel color. As IBRNet is a pixel-by-pixel rendering and produces state-of-the-art results, we are using it to synthesize novel views for immersive video transmission.

Despite being a cutting-edge rendering technique, IBRNet has three main drawbacks. Firstly, it struggles to generate high-quality new views based on a limited number of input views. Secondly, it mainly deals with static images that do not have any moving objects in the scene. Thirdly, it suffers from intra-frame processing, meaning that views at a given time instance are only synthesized from other views at the same time, resulting in visual artifacts when displayed as a video. To solve this issue, we propose an intra-framework for IBRNet, which generates temporally consistent new views while still processing the views at an intra-frame level. The paper is structured as follows: Section 2 outlines our proposed intra-framework, Section 3 presents the experimental results, and Section 4 concludes the paper.

2 Proposed Method

In order to enhance the temporal consistency between successive frames, we limit the fine-tuning process of IBRNet to incorporate image pixels exclusively from the temporal artifacts area of the image. To recognize and isolate these pixels, we propose a temporal artifacts extraction technique, which can be performed in four stages. Figure depicting the production of a temporal guidance map can be seen in [1]. The first step of our method involves obtaining a motion mask by taking a pixel-by-pixel difference between the *t, t+1* consecutive frames of an original view. We use Algorithm 1 to set pixel values to white (255, 255, 255) or black (0, 0, 0) depending on whether their absolute difference value is

above a certain threshold (Th1). In the motion mask, the black pixels represent pixels with negligible motion, which are usually synthesized with more temporal stability. In the second step, we synthesize the original view at time *t* from neighbouring views using IBRNet, resulting in a synthesized view with both static and temporal artifacts. In the third step, we add the mask obtained in step 1 to both the original and synthesized view, which allows us to remove pixels affected by the original motion. Finally, in the fourth step, we extract only the region of temporal artifacts. We use Algorithm 2 to set pixel values to white if the distance between the original and synthesized value of a pixel is smaller than a certain threshold (Th2). Table 1 shows the values of thresholds used for each sequence. They were chosen experimentally to ensure that the number of active motion pixels is similar to the number of pixels affected by temporal noise (typically between 15 to 30 percent of the entire image, but this may vary between sequences).

ALGORITHM 1

Require: *T1* = Original view at *t*
T2 = Original view at *t*+1
M = Motion mask
Th1 = Predetermined threshold value
Let *(r1, g1, b1)*, *(r2, g2, b2)*, and *(rm, gm, bm)* be the RGB pixels of *T1*, *T2*, and *M* respectively at a given location.
For each pixel location:
```
d = sqrt((r1 - r2)**2 + (g1 - g2)**2 + (b1 - b2)**2)
if d > Th1 then
    (rm, gm, bm) = (255, 255, 255)
else:
    (rm, gm, bm) = (0, 0, 0)
end if
```

ALGORITHM 2

Require: *O* = Masked original view at *t*
S = Masked synthesized view at *t*
T = Temporal guidance map at *t*
Th2 = Predetermined threshold value
Let *(ro, go, bo)*, *(rs, gs, bs)*, and *(rt, gt, bt)* be the RGB pixels of *O*, *S* and *T* respectively at a given location.
For each pixel location:
```
d = sqrt((ro - rs)**2 + (go - gs)**2 + (bo - bs)**2)
if d > Th2 then
    (rt, gt, bt) = (ro, go, bo)
else:
    (rt, gt, bt) = (255, 255, 255)
end if
```

The result of the proposed method generates a temporal guidance map containing only temporally incoherent pixels. Each original image is associated with one map, which, along with camera parameters, is fed to IBRNet. During fine-tuning, each pixel of the temporal guidance map is read, and if a white pixel is found, its corresponding pixel in the original view is simply skipped.

3 Experimental Results

As shown in Table 1, our experiments were conducted using the MPEG-I test sequences, which consist of both real-world and computer-generated scenes captured using a sparse camera setup. To evaluate the proposed method's effectiveness in immersive video transmission, we conducted experiments with three use cases. In Use Case 1 (UC1), fine-tuning was done at the server-side, assuming lossless transmission of parameters to the client. Use Case 2 (UC2) involved per-scene fine-tuning at the client-side without the need to transmit any parameters, but with added complexity. Use Case 3 (UC3) was a universal solution that fine-tuned the network on five different sequences and then tested it with a new sequence. This scenario was the most realistic, as the resulting synthesizer was data-independent. The parameters were retrained once and for all using the temporal guidance maps, making it a classical IBRNet that could be used without the need for further parameter transmission.

Sequence	Type	Resolution	Frames	Camera Setup	Th1	Th2
Fan	Real-World	1920x1080	97	5x3	25	10
Mirror	Synthetic	1920x1080	97	5x3	25	10
Frog	Real-World	1920x1080	300	13x1	35	20
Shaman	Synthetic	1920x1080	300	5x5	25	10
Carpark	Real-World	1920x1088	150	9x1	25	10
Street	Real-World	1920x1088	150	9x1	25	10

Table 1. The MPEG-I sequences used for evaluation along with their type, resolution, number of frames, and the threshold utilized for tests.

Table 2 and Table 3 compare the quality of synthesized views for different use cases using various measures, with the anchor column indicating views synthesized using a model fine-tuned on other MPEG-I test sequences. Figure 1 shows that our novel views are consistently better than the anchor views, with more details from the original views. Table 2 evaluates MSE only on active pixels of the temporal guidance map to check for local improvement, where UC1 has better quality views than all use cases in every sequence since fine-tuning was on the same frames as inference. UC2 also showed improvement, while UC3 performed better even when fine-tuning was done on all sequences except the evaluated one. IBRNet identified specific spatial features producing temporal artifacts and improved performance, allowing it to be deployed once without parameter transmission.

Table 3 shows that VMAF, MS-SSIM, and PSNR were calculated on full images to evaluate if the fine-tuning impacts the rest of the synthesized view. UC1 and UC2 showed an increase in quality, but the most significant improvement was in UC3, where fine-tuning using temporal guidance maps improved synthesis, even in the most general case of offline fine-tuning. This suggests that the fine-tuned IBRNet can be deployed similarly to the anchor version.

4 Conclusion

This paper presents an intra-framework approach to enhance the temporal consistency in IBRNet for immersive video transmission. The proposed method requires no changes to the network architecture and is easy to implement. The experiments show that the technique significantly improves temporal stability in all use cases, even in the general case of offline fine-tuning. In future work, the authors plan to improve temporal stability by incorporating motion information as an input to the synthesis network.

Fig. 1. Qualitative comparison of temporal artifacts in synthesized views of Fan, Frog and Shaman (top to bottom)

MPEG-I Sequences	MSE ↓				PSNR ↑			
	Anchor	*UC1*	*UC2*	*UC3*	*Anchor*	*UC1*	*UC2*	*UC3*
Fan	507.61	**343.54**	361.82	486.19	22.67	**24.05**	23.64	22.93
Mirror	722.53	**601.39**	645.24	702.41	23.15	**24.68**	23.91	23.46
Frog	4517.12	**3341.92**	3402.54	3845.92	13.55	**17.21**	16.38	14.92
Shaman	321.89	**186.33**	214.82	299.43	23.41	**25.97**	24.37	23.82
Carpark	1419.26	**1128.47**	1206.31	1397.21	16.22	**17.85**	17.02	16.68
Street	1628.44	**1541.85**	1595.26	1633.75	17.48	**18.52**	17.98	17.65

Table 2. The comparison of average quality of synthesized views using MSE (lower means better), and PSNR (higher means better) metrics with respect to various use cases.

MPEG-I Sequences	VMAF ↑				MS-SSIM ↑				PSNR ↑			
	Anchor	*UC1*	*UC2*	*UC3*	*Anchor*	*UC1*	*UC2*	*UC3*	*Anchor*	*UC1*	*UC2*	*UC3*
Fan	52.62	53.05	**55.28**	53.86	0.9177	0.9203	**0.9365**	0.9248	27.18	27.72	**27.96**	27.53
Mirror	64.26	65.28	**66.85**	65.92	0.9311	0.9365	**0.9483**	0.9426	28.17	**29.36**	29.16	28.48
Frog	49.38	61.52	**63.74**	56.48	0.8928	0.9223	**0.9509**	0.9342	22.97	25.11	**25.35**	23.65
Shaman	57.72	**64.85**	64.32	58.54	0.9816	**0.9894**	0.9852	0.9821	32.84	**33.12**	32.91	32.86
Carpark	65.35	68.70	**70.18**	67.11	0.9426	0.9458	**0.9528**	0.9491	28.34	29.25	**29.97**	29.12
Street	73.63	75.28	**76.81**	74.92	0.9501	0.9546	**0.9612**	0.9558	29.01	29.67	**29.82**	29.38

Table 3. The comparison of average quality of synthesized views using VMAF (higher means better), MS-SSIM (higher means better), and PSNR (higher means better) metrics with respect to various use cases.

Appendix

This paper has been accepted and presented at the 10th European Workshop on Visual Information Processing held on 11th-14th September 2022, in Lisbon, Portugal.

References

[1] S. L. Ravi, F. Henry, L. Morin and M. Gendrin, "Exploring Temporal Consistency in Image-Based Rendering for Immersive Video Transmission," 2022 10th European Workshop on Visual Information Processing (EUVIP), Lisbon, Portugal, 2022, pp. 1-6, doi: 10.1109/EUVIP53989.2022.9922680.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F Cohen. The lumigraph. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 43–54, 1996.

[3] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 425–432, 2001.

[4] J. Shade, S. Gortler, L. He, and R. Szeliski, Layered depth images. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 231–242, 1998.

[5] B. Mildenhall, P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. ECCV, 2020.

[6] Q. Wang et al. "IBRNet: Learning multi-view image-based rendering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.