Long-Term Photometric Consistent Novel View Synthesis with Diffusion Models (Supplementary Materials)

Jason J. Yu
York University
Toronto
jjyu@yorku.ca

Fereshteh Forghani
York University
Toronto
forghani@yorku.ca

Kosta G. Derpanis
York University
Toronto
kosta@yorku.ca

Marcus A. Brubaker
York University
Toronto
mab@eecs.yorku.ca

Abstract

This document provides additional material that is supplemental to the main submission. Section 1 outlines finer implementation details of our model, and provides a link to our released code. Section 3 provides qualitative sampling results using stochastic conditioning on RealEstate10K. Section 4 describes the additional qualitative results that can be found in the included supplemental webpage. Section 5 provides visualizations of quantitative TSED results, and describes an interactive demo of the metric that can be found on the supplemental webpage. Section 6 discusses the limitations of our method for novel view synthesis.

1. Architecture Details

In this section, we provide additional details of our model described in Section 3.2 of the main paper. Our model is based on *Noise Conditional Score Network*++ (NCSN++) [8]. An overview of the main backbone is provided in Tables 1 and 2. Two streams of the backbone are used to process the conditioning and generated image. We modify the original architecture by adding cross-attention layers throughout the backbone, which attend to features in the opposite stream. The residual blocks are based on the residual blocks used in BigGAN [1]. Upsampling and downsampling is also performed in the network using BigGAN residual blocks [1]. Inputs to the backbone encoder are provided at various layers using a multi-scale pyramid. Outputs of the network are accumulated from multiple layers of the decoder using a multi-scale residual pyramid. Specific implementation details can be found in the code release: https://yorkucvil.github.io/Photoconsistent-NVS/.

2. TSED Sensitivity Analysis.

A drawback to using epipolar geometry to measure consistency between correspondences and the camera poses is the potential for TSED to be insensitive to positional errors in the correspondences along epipolar lines. We empirically analyse the sensitivity of TSED on ground truth image pairs from RealEstate10K [10] under three classes of camera motion: dominant forward-backward motions, dominant left-right motions, and motion that contains more than ten degrees of azimuth rotation. Using $T_{\text{error}} = 2$, we compute TSED over 100 random image pairs in each class while adding perturbations to the 2D positions of the correspondences in each view by a constant magnitude along horizontal and vertical directions. In the ideal case when TSED is maximally sensitive, it should show a sharp reduction when the perturbations have a magnitude of T_{error} or greater. Results from our sensitivity analysis are shown in Fig. 1. As expected, TSED is least sensitive to horizontal perturbations for when there are left-right camera motions since most of the epipolar lines are horizontal. For image-pairs with greater than 10 degrees of azimuth rotation, there are fewer horizontal epipolar lines, and TSED is more sensitive to horizontal perturbations than with dominant left-right motion. The results also show that TSED is most sensitive for forward-backward motions since the epipolar lines have a variety of orientations.

Layer	Output size	Additional inputs	Additional outputs
Input image	$4 \times 32 \times 32$		Skip 0,In Pyramid
ResBlock	$256 \times 32 \times 32$	Time emb.	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Skip 1, Cross
ResBlock	$256 \times 32 \times 32$	Time emb.	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Skip 2, Cross
ResBlockDown	$256 \times 16 \times 16$	Time emb.	
Combiner	$256 \times 16 \times 16$	In Pyramid 1	Skip 3
ResBlock	$256 \times 16 \times 16$	Time emb.	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Skip 4, Cross
ResBlock	$256 \times 16 \times 16$	Time emb.	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Skip 5, Cross
ResBlockDown	$256 \times 8 \times 8$	Time emb.	
Combiner	$256 \times 8 \times 8$	In Pyramid 2	Skip 6
ResBlock	$256 \times 8 \times 8$	Time emb.	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Skip 7, Cross
ResBlock	$256 \times 8 \times 8$	Time emb.	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Skip 8, Cross
ResBlockDown	$256 \times 4 \times 4$	Time emb.	
Combiner	$256 \times 4 \times 4$	In Pyramid 3	Skip 9
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Skip 10, Cross
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Skip 11, Cross
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
ResBlock	$256 \times 4 \times 4$	Time emb.	

Table 1: NSCN++ U-Net backbone encoder. ResBlocks are BigGAN [1] style residual blocks, ResBlocksDown layers are the same, but configured with a downsampling option. Time emb. is the time information provided for the diffusion model. Skip inputs are skip connections that go to the decoder. Rays are the camera ray conditioning, and Cross is a cross-attention connection to the other stream.

Layer	Output size	Additional inputs	Additional outputs
Encoder input	$256 \times 4 \times 4$		
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 11	
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 10	
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 9	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Cross
Conv3 $\times$ 3	$256 \times 4 \times 4$	Out Pyramid 1	
ResBlockUp	$256 \times 8 \times 8$	Time emb.	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 8	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 7	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 6	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Cross
Conv3 $\times$ 3	$256 \times 8 \times 8$	Out Pyramid 2	
ResBlockUp	$256 \times 16 \times 16$	Time emb.	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 5	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 4	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 3	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Cross
Conv3 $\times$ 3	$256 \times 16 \times 16$	Out Pyramid 3	
ResBlockUp	$256 \times 32 \times 32$	Time emb.	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 2	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 1	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 0	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Cross
Conv3 $\times$ 3	$256 \times 32 \times 32$	Out Pyramid 4	

Table 2: NSCN++ U-Net backbone decoder. ResBlocks are BigGAN [1] style residual blocks, ResBlocksUp layers are the same, but configured with an upsampling option. Time emb. is the time information provided for the diffusion model. Skip inputs are skip connections coming from the encoder. Rays are the camera ray conditioning, and Cross is the cross-attention connection to the other stream.

3. Stochastic Conditioning on RealEstate10K

Previous work [9] proposed a heuristic for extending a novel view diffusion model to use an arbitrary number of source views, called *stochastic conditioning*. Given m possible source views, each iteration of the diffusion sampling process is modified to be randomly conditioned on one of the m views. Results using stochastic conditioning on CLEVR [4] are provided in the main paper in Section 4.2. Previous work [9] used stochastic conditioning to condition on all previous frames. We also apply this heuristic for generating sets of views on RealEstate10K [10], but we conditioned on up to two of the previous frames. Qualitative results shown in Figure 2 exhibit a significant reduction in quality, and contain noticeable artifacts. As a consequence, we did not include results based on stochastic conditioning with our method.

4. Additional Qualitative Results

Additional qualitative results are provided with an interactive viewer on our project page, https://yorkucvil.github.io/Photoconsistent-NVS/, under the **RealEstate10K Qualitative Results - Out-of-Distribution Trajectories** and **RealEstate10K Qualitative Results - In-Distribution Trajectories** sections. The viewer allows the images along a trajectory to be explored for multiple scenes, and sampling instances. Due to the stochastic nature of our model and the baselines, different plausible extrapolations of the scene are shown in the different instances of sampling. Additional qualitative results for Matterport3D [2] are also available on our project page, https://yorkucvil.github.io/Photoconsistent-NVS/, under the **Matterport3D Qualitative Results - Out-of-Distribution Trajectories** and **Matterport3D Qualitative Results - In-Distribution Trajectories** sections.

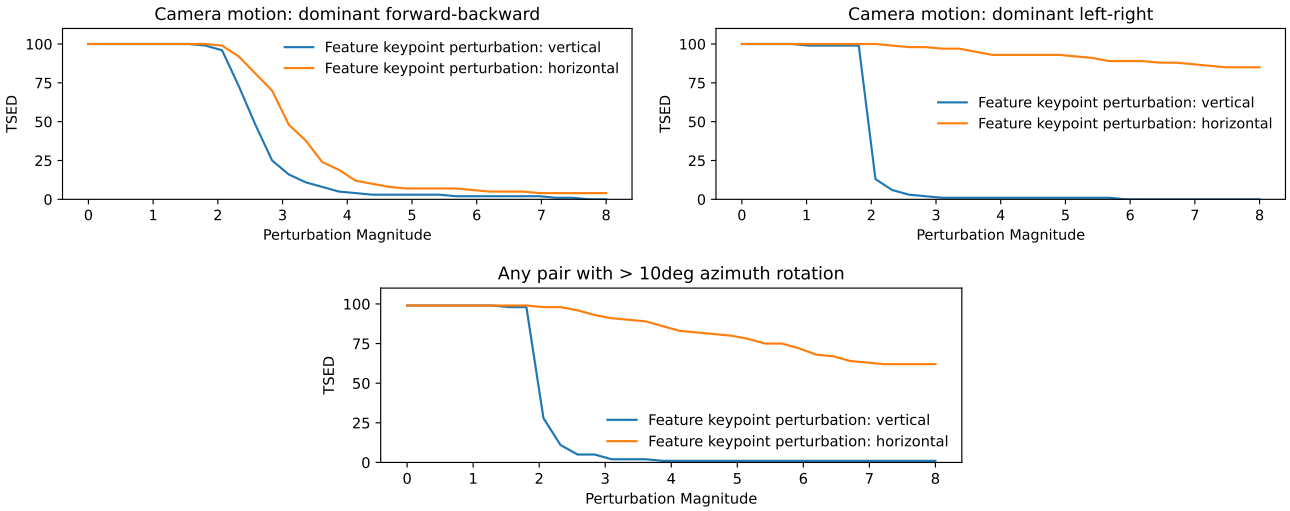


Figure 1: TSED sensitivity analysis for image pairs with different dominant camera motions using $T_{\text{error}} = 2$. TSED scores are plotted for perturbations to the 2D correspondence locations with constant magnitude along horizontal and vertical directions. Camera motion determines the orientations of the epipolar lines, which can make the metric insensitive in some cases when many epipolar lines share the same orientation.

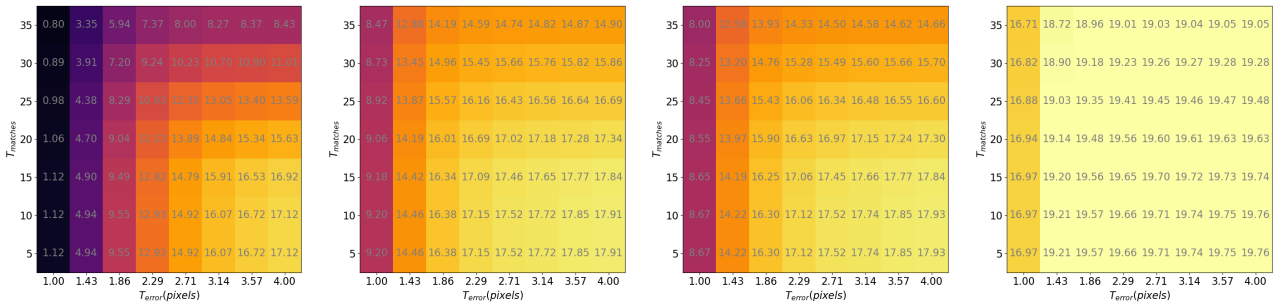
(a) Source image. (b) Frame 5 of Markov sampling. (c) Frame 7 of Markov sampling.

(d) Frame 5 with stochastic conditioning. (e) Frame 7 with stochastic conditioning.

Figure 2: Comparison of generation using a Markov dependency vs stochastic conditioning with the previous two frames as input. Both methods were generated using the same trajectory and source image. Notice the reduction of quality when stochastic conditioning is applied.

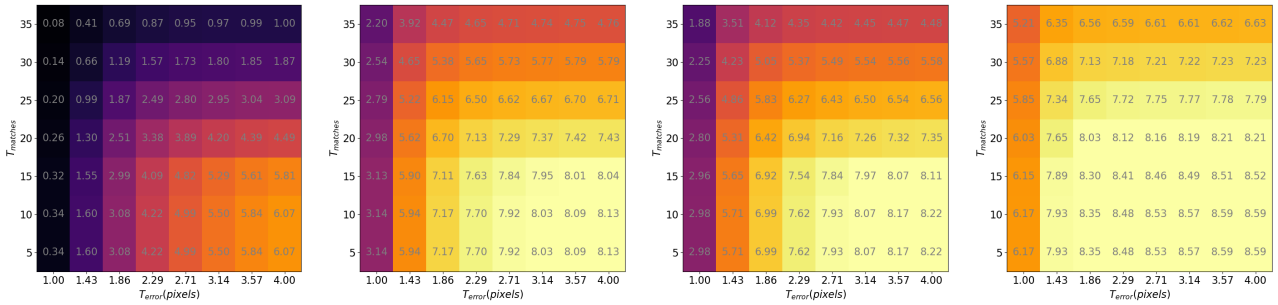
5. Additional Results with TSED

We provide additional quantitative results using TSED in Figures 3, 4, 5, and 6 for images generated using in-distribution trajectories, and the orbit, spin, hop out-of-distribution trajectories, respectively. We sweep across a range of values for both $T_{\textbf{error}}$ and $T_{\textbf{matches}}$. Pairs of images with less than $T_{\textbf{matches}}$ SIFT [5] matches, or a median SED [3] lower than $T_{\textbf{error}}$, are considered not consistent. In all trajectory types, GeoGPT [7] is the most affected by $T_{\textbf{matches}}$ due to a lack of photometric consistency, which leads to a low number of SIFT correspondences. The TSED for both variants of Lookout [6] do not vary as severely as GeoGPT with respect to $T_{\textbf{matches}}$. Image pairs generated with our method tend to yield more SIFT matches, and are mainly affected by $T_{\textbf{error}}$. The quantitative TSED results in the main paper were evaluated at $T_{\textbf{matches}} = 10$, but these extended results show that our method yields higher TSED scores, remains consistent over a range of $T_{\textbf{matches}}$ values, in all cases.



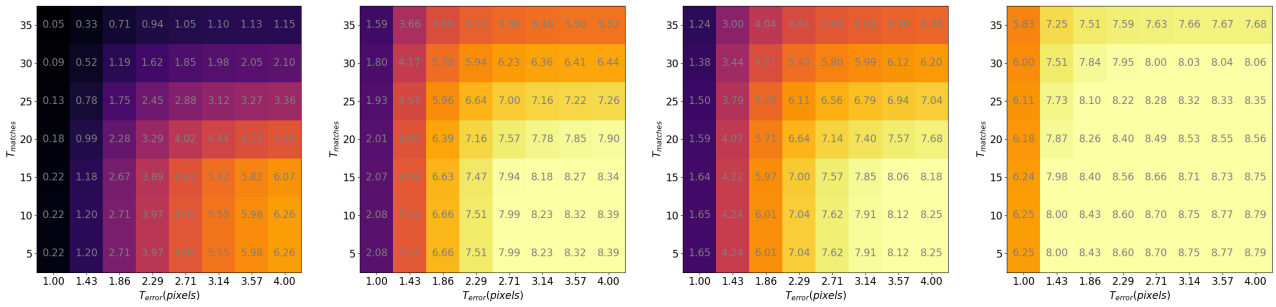
(a) GeoGPT (b) Lookout-noerror (c) Lookout (d) Ours

Figure 3: TSED computed using images generated over in-distribution trajectories. We sweep over a range of values for T_{matches} and T_{error}. The values are provided as the average number of consistent pairs per sequence out of 20.



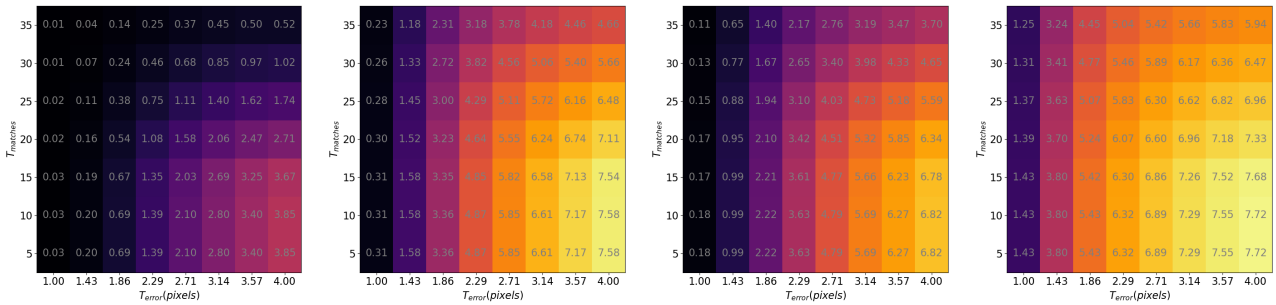
(a) GeoGPT (b) Lookout-noerror (c) Lookout (d) Ours

Figure 4: TSED computed using images generated over *orbit* trajectory. We sweep over a range of values for T_{matches} and T_{error}. The values are provided as the average number of consistent pairs per sequence out of 9.



(a) GeoGPT (b) Lookout-noerror (c) Lookout (d) Ours

Figure 5: TSED computed using images generated over *spin* trajectory. We sweep over a range of values for T_{matches} and T_{error}. The values are provided as the average number of consistent pairs per sequence out of 9.



(a) GeoGPT (b) Lookout-noerror (c) Lookout (d) Ours

Figure 6: TSED computed using images generated over our *hop* trajectory. We sweep over a range of values for T_{matches} and T_{error}. The values are provided as the average number of consistent pairs per sequence out of 9.

To provide a better intuition on how symmetric epipolar distance (SED) [3] provides a measure of consistency, we provide an interactive demo on our project page, https://yorkucvil.github.io/Photoconsistent-NVS/, under the **Visualization of SED** section. The demo visualizes how SED varies in response to the positions of two correspondences in a pair of views with known relative camera geometry. Each point creates an epipolar line on the opposite image, and the minimal distance line between a point and a line on the same image is shown.

6. Limitations of Autoregressive Sampling

Our method and the baselines are limited by the use of sequential generation with a fixed budget for conditioning images. Regions that become occluded and subsequently disoccluded in a sequence are very likely to change appearance. For example, conditioning on one image prevents information about previously disoccluded regions from informing the generation of those same regions beyond one frame. Qualitative examples of this phenomenon can be seen on our project page, https://yorkucvil.github.io/Photoconsistent-NVS/, under the **RealEstate10K Qualitative Results - Out-of-Distribution Trajectories** section, with the **Spin** motion. The described phenomenon can be observed at the edges of the images with **Spin** motion, where those regions of the scene often move beyond the image boundaries before returning in the future. A qualitative example of this is shown in Figure 7.

Conditioning on an arbitrary number of frames could theoretically solve this problem. However, the practicality of this solution is limited by the ability to design models that can process an arbitrary number of inputs, and the model's ability to generalize to out-of-distribution camera poses (*e.g.*, far away cameras in large scenes). Leveraging many images for generation is a potentially significant direction for future work.

(a) Initial image

(b) Image after returning close to the initial camera position.

Figure 7: The initial frame and the final frame from a generated sequence with the *spin* motion. Notice the final frame has returned to a location similar to the initial frame, but the bottom left region on the floor has changed appearance.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1, 2

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *Proceedings of the International Conference on 3D Vision (3DV)*, 2017. 2

[3] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 5, 6

[4] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999. 5

[6] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3D scene video from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[7] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3D priors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 5

[8] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1

[9] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2

[10] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (TOG)*, 2018. 1, 2