End-to-End Time-Lapse Video Synthesis from a Single Outdoor Image – Supplementary Materials

Seonghyeon Nam[1] Chongyang Ma[2] Menglei Chai[2]
William Brendel[2] Ning Xu[3] Seon Joo Kim[1]
[1]Yonsei University [2]Snap Inc. [3]Amazon Go

1. Implementation Details

In Tables 1, 2, 3, and 4, we list the network parameters of the generator $\mathcal{G}_{\mathcal{B}}$ and the discriminator $\mathcal{D}_{\mathcal{B}}$ for images from the TLVDB dataset [3], as well as the generator $\mathcal{G}_{\mathcal{A}}$ and the discriminator $\mathcal{D}_{\mathcal{A}}$ for images from the AMOS dataset [2], based on the naming conventions of network components below:

- Conv2d(K, P): 2D convolution with the kernel size of K and the padding of P;
- BN: Batch normalization;
- LeakyReLU(S): Leaky ReLU with the negative slope of S;
- NN Upsampling: Nearest neighbor upsampling.

Module	Layers	Input size	Output size
(a) Image Encoder	Conv2d(3, 1), ReLU	$128 \times 128 \times 3$	$128 \times 128 \times 64$
	Conv2d(4, 2), BN, ReLU	$128 \times 128 \times 64$	$64 \times 64 \times 128$
	Conv2d(4, 2), BN, ReLU	$64 \times 64 \times 128$	$32 \times 32 \times 256$
	Conv2d(4, 2), BN, ReLU	$32 \times 32 \times 256$	$16 \times 16 \times 512$
(b) Timestamp Encoder	Linear, ReLU	1	64
	Linear, BN, ReLU	64	128
(c) Latent variable	Sample from $\mathcal{N}(0, 1)$		128
Concat (a), (b), and (c)	Conv2d(3, 1), BN, ReLU	$16 \times 16 \times 768$	$16 \times 16 \times 512$
Residual Blocks	$4\times$ Residual Block (below)	$16 \times 16 \times 512$	$16 \times 16 \times 512$
(d) Residual Block	Conv2d(3, 1), BN, ReLU	$16 \times 16 \times 512$	$16 \times 16 \times 512$
	Conv2d(3, 1), BN	$16 \times 16 \times 512$	$16 \times 16 \times 512$
	Input + (d)	$16 \times 16 \times 512$	$16 \times 16 \times 512$
Decoder	NN Upsampling ($2\times$)	$16 \times 16 \times 512$	$32 \times 32 \times 512$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$32 \times 32 \times 512$	$32 \times 32 \times 256$
	NN Upsampling ($2\times$)	$32 \times 32 \times 256$	$64 \times 64 \times 256$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$64 \times 64 \times 256$	$64 \times 64 \times 128$
	NN Upsampling ($2\times$)	$64 \times 64 \times 128$	$128 \times 128 \times 128$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$128 \times 128 \times 128$	$128 \times 128 \times 64$
	Conv2d(3, 1), Tanh	$128 \times 128 \times 64$	$128 \times 128 \times 3$

Table 1: The parameters of $\mathcal{G}_{\mathcal{B}}$. We add symmetric skip connection between layers in the image encoder and the decoder.

Module	Layers	Input size	Output size
Encoder	Conv2d(4, 2), LeakyReLU(0.2)	$128 \times 128 \times 3$	$64 \times 64 \times 64$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$64 \times 64 \times 64$	$32 \times 32 \times 128$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$32 \times 32 \times 128$	$16 \times 16 \times 256$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$16 \times 16 \times 256$	$8 \times 8 \times 512$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$8 \times 8 \times 512$	$4 \times 4 \times 512$
Classifier	Conv2d(4, 1)	$4 \times 4 \times 512$	$1 \times 1 \times 1$

Table 2: The parameters of $\mathcal{D}_{\mathcal{B}}$.

Module	Layers	Input size	Output size
Encoder	Conv2d(3, 1), ReLU	$128 \times 128 \times 3$	$128 \times 128 \times 64$
	Conv2d(4, 2), BN, ReLU	$128 \times 128 \times 64$	$64 \times 64 \times 128$
	Conv2d(4, 2), BN, ReLU	$64 \times 64 \times 128$	$32 \times 32 \times 256$
	Conv2d(4, 2), BN, ReLU	$32 \times 32 \times 256$	$16 \times 16 \times 512$
	Conv2d(4, 2), BN, ReLU	$16 \times 16 \times 512$	$8 \times 8 \times 512$
	Conv2d(4, 2), BN, ReLU	$8 \times 8 \times 512$	$4 \times 4 \times 512$
	Conv2d(4, 2), BN, ReLU	$4 \times 4 \times 512$	$2 \times 2 \times 512$
	NN Upsampling (2×)	$2 \times 2 \times 512$	$4 \times 4 \times 512$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$4 \times 4 \times 512$	$4 \times 4 \times 512$
Decoder	NN Upsampling (2×)	$4 \times 4 \times 512$	$8 \times 8 \times 512$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$8 \times 8 \times 512$	$8 \times 8 \times 512$
	NN Upsampling (2×)	$8 \times 8 \times 512$	$16 \times 16 \times 512$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$16 \times 16 \times 512$	$16 \times 16 \times 512$
	NN Upsampling (2×)	$16 \times 16 \times 512$	$32 \times 32 \times 512$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$32 \times 32 \times 512$	$32 \times 32 \times 256$
	NN Upsampling (2×)	$32 \times 32 \times 256$	$64 \times 64 \times 256$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$64 \times 64 \times 256$	$64 \times 64 \times 128$
	NN Upsampling (2×)	$64 \times 64 \times 128$	$128 \times 128 \times 128$
	Conv2d(3, 1), BN, LeakyReLU(0.2)	$128 \times 128 \times 128$	$128 \times 128 \times 64$
	Conv2d(3, 1), Tanh	$128 \times 128 \times 64$	$128 \times 128 \times 3$

Table 3: The parameters of $\mathcal{G}_{\mathcal{A}}$. We add symmetric skip connection between layers in the encoder and the decoder.

Module	Layers	Input size	Output size
(a) Image Encoder	Conv2d(4, 2), LeakyReLU(0.2)	$128 \times 128 \times 3$	$64 \times 64 \times 64$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$64 \times 64 \times 64$	$32 \times 32 \times 128$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$32 \times 32 \times 128$	$16 \times 16 \times 256$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$16 \times 16 \times 256$	$8 \times 8 \times 512$
	Conv2d(4, 2), BN, LeakyReLU(0.2)	$8 \times 8 \times 512$	$4 \times 4 \times 512$
	Conv2d(4, 1), BN, LeakyReLU(0.2)	$4 \times 4 \times 512$	$1 \times 1 \times 512$
(b) Timestamp Encoder	Linear, LeakyReLU(0.2)	1	64
	Linear, BN, LeakyReLU(0.2)	64	128
Concat (a) and (b)	Linear, BN, LeakyReLU(0.2)	640	512
	Temporal Max-Pooling	$T \times 512$	512
Conditional Classifier	Linear	512	1
Unconditional Classifier	Conv2d(1, 1)	$1 \times 1 \times 512$	$1 \times 1 \times 1$

Table 4: The parameters of $\mathcal{D}_{\mathcal{A}}$.

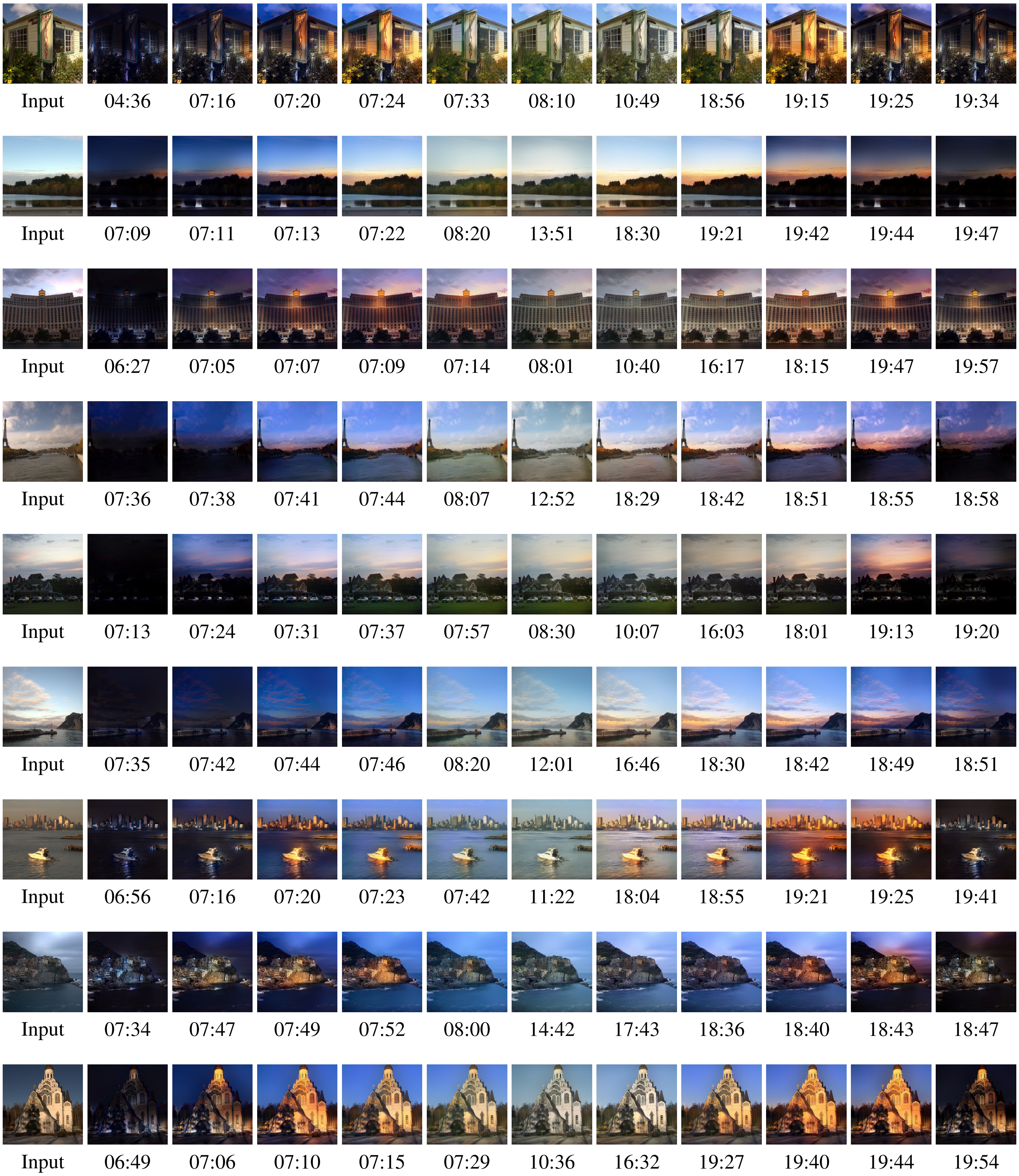

Figure 1: Additional results: our prediction results of the input images at different times of a day. The timestamp used for each output is shown below the corresponding image.

2. Additional Results

By using the timestamp as the conditional variable, we can synthesize continuous illumination changes over time from a single input image. In Figure 1, we show additional time-lapse video synthesis results using our method based on test images from the MIT-Adobe FiveK Dataset [1]. The input images are shown on the left of each row and the timestamps are shown below the corresponding output images. For each sequence, we randomly sample 11 output frames in the temporal domain focusing more on the transition time of the sunrise and the sunset. Note that for different sequences the night-to-day and day-to-night transitions may happen at different times of a day. This fact is evident in our training data and has been captured by our multi-frame joint conditional generation framework.

References

[1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2011.

[2] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

[3] Yichang Shih, Sylvain Paris, Frédo Durand, and William T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.*, 32(6):200:1–200:11, 2013.