DEPTH-RECURRENT RESIDUAL CONNECTIONS FOR SUPER-RESOLUTION OF REAL-TIME RENDERINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inferring temporally coherent data features is crucial for a large variety of learning tasks. We propose a network architecture that introduces temporal recurrent connections for the internal state of the widely used residual blocks. We demonstrate that, with these connections, convolutional neural networks can more robustly learn stable temporal states that persist between evaluations. We demonstrate their potential for inferring high-quality super-resolution images from low resolution images produced with real-time renderers. This data arises in a wide range of applications, and is particularly challenging as it contains a strongly aliased signal. Hence, the data differs substantially from the smooth inputs encountered in natural videos, and existing techniques do not succeed at producing acceptable image quality. We additionally propose a series of careful adjustments of typical generative adversarial architectures for video super-resolution to arrive at a first model that can produce detailed, yet temporally coherent images from an aliased stream of inputs from a real-time renderer.

1 INTRODUCTION

Learning expressive and stable representations is a goal that lies at the heart of a vast range of deep learning tasks (Dahl et al., 2011; Radford et al., 2015; Chen et al., 2016). While typical recurrent architectures focus on feedback loops to form persistent latent-spaces (Rumelhart et al., 1988; Chaitanya et al., 2017), we show that for inference tasks where the result is conditioned on a stream of inputs, these existing architectures unnecessarily complicate the learning task, and fail to reliably stabilize the inference. With our work, we propose a new type of connection for the very widely used building blocks of ResNet architectures (He et al., 2015) that lets the network easily compare internal states in-place. The learned representation can then, e.g., yield a detailed image sequence with natural changes. We demonstrate this with a particularly challenging learning objective: we aim for the synthesis of detailed images from a stream of strongly aliased inputs. Specifically, we show that adversarially trained convolutional neural networks (CNNs) can be leveraged to produce detailed images from unfiltered, low-resolution images generated via point-sampling with a rasterization-based real-time renderer. Real-time graphics are the basis for a wide range of applications: Generating images with a sufficient resolution from low resolution, yet computationally light-weight renderings is a task that is, e.g., important for generating content for the high resolution screens of mobile devices, and is especially interesting for streaming services of games in order to compute the final resolution only on the client.

Our work shares its goal with a variety of approaches that have been proposed for generating high-quality images for raytracing algorithms (Zhang et al., 2016; Chaitanya et al., 2017) and purely image-based super-resolution algorithms (Sajjadi et al., 2017; Sajjadi et al., 2018; Chu et al., 2018). Our architecture differs from previous works as the proposed recurrent connection allows the network to learn a temporally stable latent-space representation that does not negatively impact the residual flow of a ResNet architecture. Also, the temporal connections for deeper layers of the network are important for successful learning, as we will demonstrate below.

While the basic concept of depth-recurrent connections could potentially be applied to a variety of sequence-based learning tasks, we focus on demonstrating its potential for pushing forward the limits of real-time rendering. Hence, we additionally outline a series of modifications to existing architectures which are crucial for achieving high quality of the strongly aliased input images from

Figure 1: Given a strongly aliased low-resolution input rendering with one sample per pixel, recurrent non-adversarial training ((Chaitanya et al., 2017) with modifications for fair comparisons) produces blurry results, and existing adversarial methods ((Chu et al., 2018), re-trained) introduce strong flickering artifacts. Trained on the same data, due to the proposed DRR connections our network infers more consistent spatio-temporal features (see the supplemental footage for a clear assessment of the temporal differences).

typical real-time rendering pipelines. A typical input for our network is shown on the left of Fig. 1. This application scenario is especially challenging for CNNs, since it requires to work with images that need to be rendered at very high frame rates and, thus, exhibit severe aliasing due to point sampling and typically low resolutions. The aliasing not only distorts the spatial signal, but likewise affects the temporal changes. Therefore, a super-resolution (SR) network can't rely on receiving smoothly changing, filtered inputs that allow for localization of small features. Rather, it has to learn over the course of multiple frames to infer consistent output images(Fig. 1, right) from spatially and temporally aliased input content. As we will demonstrate in a number of studies below, this task is where our proposed depth-recurrent connections unfold their strength. They enable the network to match the data distribution of the targets, i.e., to synthesize images with a high visual quality in terms of detail as well as their temporal behavior. We show results and comparisons in the paper, and provide many additional evaluations in the supplemental material [1], where videos more clearly show spatial and temporal differences.

2 Related Work

Deep learning has been successfully applied to a large variety of image-based super-resolution tasks (Wang et al., 2015; Kim et al., 2016; Ledig et al., 2016). Here, convolutional architectures (CNNs) (Dong et al., 2014) with residual blocks (He et al., 2015) are a very popular generator architecture that offers training stability as well as high qualify inference. Targeting photo-realism, Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) were introduced to prevent the undesirable smoothing of direct loss formulations (Sajjadi et al., 2017; Ledig et al., 2016). For GAN architectures, a second discriminator network is trained to classify real and generated samples that is used to guide the generator network. The results of these GAN approaches were improved upon, e.g., by modifying the residual blocks and perceptual loss function (Wang et al., 2018), by employing the Earth Movers distance to stabilize the training (Arjovsky et al., 2017), and by accounting for the a-priori knowledge that fake samples exist (Jolicoeur-Martineau, 2018). Using the feature-space differences of image classification networks, e.g. a pre-trained VGG network (Johnson et al., 2016) or the discriminator in a GAN setting, as perceptual loss was shown to be highly effective as well. Natural temporal changes of the generated content are crucial for video SR tasks. Often multiple subsequent frames are used to generate a high-resolution (HR) frame (Liao et al., 2015). These multi-frame approaches benefit from aligning the frames via warping which requires an estimation of the image space motions. As this is usually not readily available for natural videos, optical flow estimation networks are a popular choice (Caballero et al., 2016). As for spatial content, employing an L^2 loss to enforce temporal coherence is not optimal. Instead, extending the adversarial loss to the temporal domain improves the temporal coherence of small-scale details (Pérez-Pellitero et al., 2018; Xie et al., 2018; Chu et al., 2018). We likewise employ a spatio-temporal GAN architecture in the following.

[1] Anonymous and time-stamped supplemental material for our submission can be downloaded at `https://www.dropbox.com/sh/4tco2sb9ja034jo/AAAQC-8IBqoC6qt8M-lJ4asFa?dl=0`. All source code will be published upon acceptance.

For deep learning methods, a variety of recurrent neural networks has been proposed (Rumelhart et al., 1988; Hochreiter & Schmidhuber, 1997; Cho et al., 2014), and were shown to be useful for image generation tasks. Chaitanya et al. (2017) uses recurrent connections to propagate a latent state over time inside the network, while Sajjadi et al. (2018) use the previously generated high-resolution (HR) output as input. The SR task also bears a certain similarity to other image enhancement techniques like inpainting (Pathak et al., 2016; Zhang et al., 2018a), where progressive methods provide state-of-the-art quality, or the removal of compression artifacts and text (Mao et al., 2016). Likewise, temporal coherence is an important aspect in style transfer (Huang et al., 2017). We use spatio-temporal self-supervision similar to previous work (Chu et al., 2018), but target a substantially more challenging data domain than natural videos in the following.

Images of real-time renderers that can be generated quickly often exhibit strong aliasing. In this context, the existing work focuses on enhancing path-traced images. A common application is image denoising, e.g. for Monte-Carlo ray-tracing (Zhang et al., 2016; Chaitanya et al., 2017), where a network learns to predict a noise pattern in order to infer a smooth image from a sparse set of importance samples. Computer games and other real-time graphics applications, on the other hand, use a rasterization-based rendering pipeline. Simple geometry is shaded with simplified but increasingly complex lighting and texturing computations, and often only 1 sample per pixel is taken to maintain the required frame rates. Compared to natural and path-traced synthetic images these images are strongly under-sampled and exhibit aliasing artifacts that are different from the noisy images of path-tracing algorithms. Super-Sampling (SSAA), i.e. rendering at a higher resolution and averaging, is a straight-forward, but expensive anti-aliasing (AA) solution. While Multi-Sampling (MSAA) (Akenine-Moller et al., 2018) is a possible optimization, image-space methods (Lottes, 2009; Jimenez et al., 2011; 2012) are popular and try to reduce aliasing by detecting and smoothing edges in image-space after rendering, temporal methods like Temporal Anti-Aliasing (TAA) (Korein & Badler, 1983) use the warped previous samples to smooth and stabilize edges. Here, the use of deep leaning models for image enhancement in real-time settings is sparse. To the best of our knowledge, only a closed-source solution exist, provided by Nvidia in the Turing architecture (NVIDIA Corporation, 2018), and little is known about its internal realization. Instead, our goal is to provide an open solution, and at the same time improve the state-of-the-art in GAN-based SR.

3 METHOD

Our main goal is to facilitate the learning of a stable internal network representation that persists between repeated SR inference steps. In this way, the network can pass information from one frame to the next such that a persistent and stable prediction over time is achieved. This is crucial for processing strongly aliased input data streams. The internal network representation can then be used to generate the desired output distribution and, as demonstrated below, this improves temporal coherence and generates stable small-scale details in SR rendering. Fig. 2 demonstrates the quality that is achieved by the proposed architecture. To enable the network to learn this internal representation, we augment the generator network with aligned, temporal connections inside the residual blocks (ResBlocks).

In order to demonstrate the advantages of our depth-recurrent architecture and the extensions for the real-time setting, we build on a GAN-based video super-resolution architecture from previous work (Chu et al., 2018). Here, the generator G is trained on image sequences and processes the current LR input as well as its warped previous output through a series of ResBlocks to produce a detailed and temporally coherent output. The spatio-temporal discriminator $D_{s,t}$ sees shorter 3-frame sequences of the output as well as the LR input as condition. We also employ a perceptual *feature loss* via the discriminator network. In addition to the adversarial and the feature loss, the generator is trained with a Ping-Pong term for long-term coherence and an L^2 loss in image space, the *content loss*, for stabilization.

3.1 DEPTH-RECURRENT RESIDUALS

The residual blocks of frame $t-1$ and frame t are connected to pass the activations of a convolutional layer from frame $t-1$ to t. As we work with video data, we align these activations by warping them with the screen-space motions before they are concatenated to the ones in frame t and passed as input to the next layer. We call this setup Depth-Recurrent Residuals, and it is visualized in Fig. 3, DRR. In contrast to frame-recurrent networks, we employ connections in latent space, and different

Figure 2: Our DRR model applied to 2 different test scenes. The comparisons show input LR color images and the high-resolution outputs inferred by our method. The inputs exhibit strong spatial aliasing, and a similar amount of temporal aliasing, as is visible in the videos of the supplemental material. Despite these challenges, our model infers a stable and detailed output sequence.

to commonly used "feed-back" loops of recurrent architectures, we use a "feed-forward" design that includes a warping step. Within a multi-layer CNN, a variety of possible recurrent connections are imaginable. The goal of the specific variant we propose is that of temporal stability: with our connections a receiving layer can compare an aligned set of activations from the previous frame with the current activations. Hence, operations such as temporal derivatives are trivial to perform with in-place differences.

The operations of a regular ResBlock at depth b can be summarized as follows: two convolutional layers $\mathbf{F}_{b,2} = C_{b,2}(\mathbf{F}_{b,1})$, and $\mathbf{F}_{b,1} = C_{b,1}(\mathbf{F}_{b-1,+})^+$ are followed by an addition, yielding the final output $\mathbf{F}_{b,+} = \mathbf{F}_{b,2} + \mathbf{F}_{b-1,+}$. Here, $C_{\text{<ResBlock>,<layer>}}$ denotes a convolution, and $^+$ ReLU activation. For our depth-recurrent residual blocks, we replace the second convolutional layer with the following operation: $\mathbf{F}^t_{b,2} = C_{b,2}(\mathbf{F}^t_{b,1} \oplus W(\mathbf{F}^{t-1}_{b,1}))$. This is followed by a summation as before to produce the output $\mathbf{F}_{b,+}$. Here, W applies warping for the current layer based on externally computed velocities (either with another CNN, or via rendering as for our inputs), and $\oplus$ denotes feature concatenation. Compared to regular ResBlock convolution, we use $18,75\%$ fewer features for DRR blocks in order to keep the overall number of weights per ResBlock constant.

During our experiments, we apply the following curriculum learning steps, which typically help to stabilize the training runs and lead to improved final results in our tests: The depth recurrent connections are linearly faded in during training in a staggered fashion, beginning with fading in the connection of first block, then the second block and so on. Similarly, the discriminator network is faded in after a pre-training step of the generator (details are given in App. B).

Discussion To analyze whether the intuition that the DRR connection support learning stable features is correct, we have evaluated alternative connectivities. One variant from previous work connects the outputs of a ResBlock as inputs into the same ResBlock for the next frame to produce a hidden state. This is inspired by the recurrent denoising autoencoders Chaitanya et al. (2017), and shown as *RDA* in Fig. 3. Note however, that this version in the following uses our improved architecture and warping operations, both of which are not employed in the original version. In terms of the notation above, this variant is obtained via $\mathbf{F}^t_{b,1} = C_{b,1}(\mathbf{F}^t_{b-1,+} \oplus W(\mathbf{F}^{t-1}_{b,+}))^+$. It yields spatially slightly worse but strongly flickering results, as the layer now effectively has to compare different feature sets: that of the previous ResBlock, and the features of its own from the previous frame.

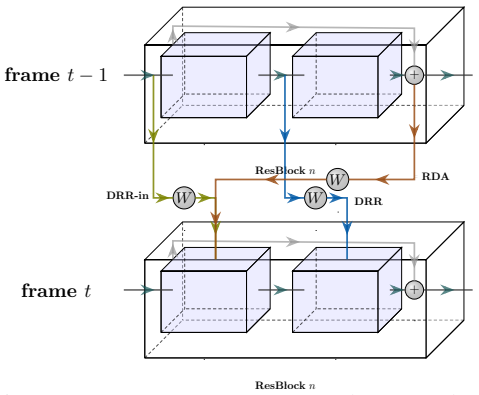

Figure 3: We propose to connect the center latent-space of ResBlock n at time $t-1$ to the same ResBlock n at time t, yielding our DRR connection. A variant of our approach is DRR-in, in addition to a recurrent connections similar to previous work (RDA).

Figure 4: A visual comparison of the different types of recurrent connections with otherwise identical models (all three using our discriminator supervision, generator and warping). Our DRRs yield an improved image quality and temporal stability, as shown in the supplemental material.

A variant of our recurrent ResNet architecture is to use an in-place connection instead. Thus, feeding the outputs of the previous ResBlock of both the current and previous frame as input to each ResBlock as shown in Fig. 3 DRR-in. I.e., $\mathbf{F}^t_{b,1} = C_{b,1}(\mathbf{F}^t_{b-1,+} \oplus W(\mathbf{F}^{t-1}_{b-1,+}))^+$. This variant of our DRR connections stabilizes the results, but does typically result in slightly lower quality inference. Qualitative results for each variant are shown in Fig. 4.

We believe that the primary reason for the improved stability of our variant is that the proposed connection is de-coupled from the regular flow of information along the residual blocks. Hence, the learned temporal features change the way the latent-space within the ResBlock is shaped, but it does influence and possibly impede the residuals added at the beginning and end of a ResBlock. Therefore, the network can focus on learning suitable temporal features, instead of additionally having to learn to work with a combination of spatial and temporal information (as with previous work and the variants discussed above). Thus, our approach preserves the structure which led to the success and widespread use of the original ResBlocks (He et al., 2015).

In all three variants, applying the W function, i.e. warping with the screen-space motion vectors, improves the results. Otherwise, the networks easily converge towards solutions with noticeable streak-like artifacts. It is also worth noting that the depth recurrent connections are substantially different from the recurrent input to the generator network, i.e., the previously generated HR frame, as they pass on learned latent-space features at the same location within the network. Thus, for new inference tasks the network does not need to analyze again the information encoded in an image. However, it is still challenging for the network to encode and transport coherent small-scale features across frames. Removing the recurrent input, using only our depth recurrent connections leads to temporally coherent results, but induces a significant loss of spatial detail. Hence, rather than replacing it, the depth recurrent connections work best in conjunction with a recurrent input.

4 DATA

We use images rendered in real-time with a rasterization-based rendering pipeline with 1 sample/pixel as our training data. The network is trained with 15-frame sequences of matching HR-LR image pairs, both rendered using *Unity's* high-definition rendering pipeline (HDRP). From there, we also capture the rendered screen-space motion that is used for warping. Details are given in App. A.

Typical super-resolution models for natural videos are trained with a down-sampled version of the target. This gives the network a smooth, reliable average of the reference content. A generator network can detect the changes and sharpen existing features to generate detail. As our synthetic data exhibit strong aliasing artifacts, there are larger changes between adjacent pixels as well as temporal aliasing. This makes the data much more difficult to analyze. In contrast to a spatio-temporally integrated signal, as for natural video SR, sub-pixel features only show up rarely, and most of the time do not have a signal at all in our setting. This makes it much harder to correlate image space content in subsequent frames and achieve temporal coherence. Aliasing also makes it more difficult to detect the actual, underlying edges over a larger area and correct the input accordingly. These combined challenges make it necessary to introduce more temporal context, which we enable with the DRR connections.

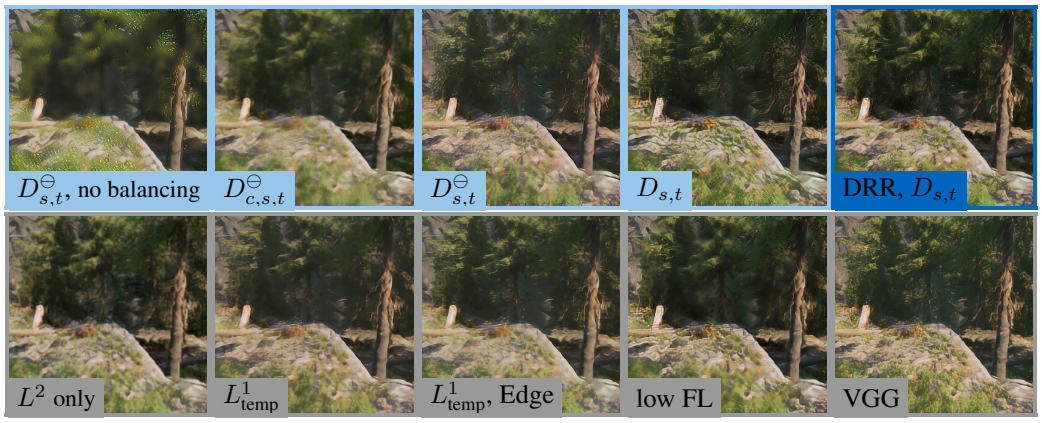

Figure 5: Ablation study for the discriminator network: The top row shows different discriminator setups: without adaptive balancing, with conditional LR color input, the standard size and our larger variant. The bottom row shows the influence of different auxiliary losses.

5 Super-Resolution Architecture

In addition to the depth-recurrent residuals described above, our model is also subject to multiple changes which we found crucial for dealing with the challenging input data from real-time renderers. First, we use the motion vectors generated by the renderer for all warping operations. These are faster to compute and more accurate than the motion estimated by an auxiliary network. We also found resize-convolutions (Odena et al., 2016) with bilinear interpolation instead of (strided) deconvolutions important to reduce checkerboard artifacts. The specific task the network needs to perform in our target application reveals the limits of the deconvolution approach, requiring to incorporate resize-convolutions as described below.

Providing depth as LR input to the generator also slightly enhances edges and reduces aliasing artifacts such as staircasing. This effect is stronger near the camera, which suggests that it is related to the non-linear dependency of depth to view-space distance. Introducing additional data fields from the rasterization pipeline, such as surface normals, did not yield improvements.

Previous work proposes to let the generator output be a residual that is added to the bicubically up-scaled LR input color, instead of directly generating the final output. Interestingly, for our strongly aliased data we found it beneficial to instead use bilinear interpolation for up-scaling the inputs, as bicubic interpolation often leads to overshooting near edges. This caused difficulties for the generator and did not yield any improvements in terms of image quality. However, also the bilinearly up-scaled LR color, to which the residual is applied, still contains strong aliasing. To perform the necessary anti-aliasing, the network has to detect structures based on single samples in the LR input, as it does not have direct access to the up-scaled version in this case, and then perform the necessary correction via the residual addition to the HR version on the other end of the network. This is sufficient for the smoother data of natural videos, where detail enhancement and sharpening operations can be less accurate and still yield good results. Replacing the final addition of the residual with 2 additional convolutional layers further improves image quality. For these convolutions, we concatenate the up-scaled color after the resize convolutions (details in App. B). With this modification, the network does not need to transport the LR color through the LR ResBlocks and can choose based on the up-sampled LR color what parts of it to use. This adds ca. 1% additional weights compared to the standard setup, but is computationally more expensive as high-res tensors are convolved. Nonetheless, the gain in image quality outweighed the slight increases in computations and memory.

The concatenation of the DRRs increases the number of input channels and, therefore, the number of weights of the receiving layer. Thus, to keep the overall amount of weights consistent with previous versions and allow for a fair comparison, the number of base channels of the ResBlocks is reduced when using these recurrent connections.

6 ABLATION STUDIES

Discriminator The discriminator facilitates the generation of sharp and detailed images. While a network trained with only L^2 supervision produces a more blurred output, "L^2 only" in Fig. 5, activating the spatio-temporal discriminator yields a significant improvement, see "$D^{\ominus}_{s,t}$". As visible in Fig. 5, we found it important to use a discriminator without conditional input in order to balance the networks. As a large part of the loss function employs the discriminator (both adversarial and feature losses), its quality is crucial. E.g., by increasing the size, i.e., adding layers or channels, and therefore complexity of the discriminator, further improvements are achieved. Increasing depth values allows D to detect more complex features, thus providing gradients towards more complex image content. This also avoids that the generator quality is limited by the lack of discriminator gradient information. The feature loss similarly profits from more complex features, and while a larger discriminator is slower to train, it does not affect performance during inference. However, a larger discriminator can make the training more unstable, and eventually become unbalanced. We found a discriminator with 14 layers and 3.7M weights ($D_{s,t}$), and a generator with 26 layers and 769k weights to be a stable combination that yields high quality results, as long as adaptive balancing is used as outlined in App. B.

Depth-Recurrent Residuals A generator with Depth-Recurrent Residuals (DRRs) gives good results, even when trained with the largest discriminator where a generator without DRRs tends to produce strong artifacts, Fig. 5 ($D_{s,t}$). The generator network can process the detailed gradient feedback it receives from the large discriminator network, and use it to generate a large amount of image-space detail. Letting the DRR connections fade into the network further improves the results. When the connections are activated right from the beginning of the training process, the image quality is typically reduced.

Loss Functions Despite the stabilization via DRR connections, the proposed learning setup requires careful balancing. The additional details provided by a larger discriminator, combined with the strong learning objective to produce detail, yields overall better results but more small-scale differences to the target. This in turn increases the L^2 content loss, which however is needed to stabilize the training. In practice, a parameter is introduced to choose between temporal stability and large amounts of image-space detail. An important source of detail seems to stem from the discriminator feature loss, i.e., by using only the regular adversarial loss, details are noticeably reduced. In practice, a strong feature loss effectively reduces the smoothing caused by the content loss and gives the best results.

Similar to a spatial L^2 loss, we found that a temporal L^1 loss, as employed by Chaitanya et al. (2017), leads to an undesirable smoothing in our GAN setting. Even worse, when using a temporal L^1 loss without warping, it produces strong streak-like artifacts. Modifications such as additional edge loss terms did not alleviate this behavior. While others have reported improvements by using perceptual losses with pre-trained networks (such as VGG), we likewise found that this yields different, but not necessarily better outputs. Presumably, the different nature of the images produced by the rasterization pipelines is less amenable to the VGG features. The loss variants used in previous works are shown in the bottom row of Fig. 5.

7 EVALUATION

We compare our model with and without DRR connections to state-of-the-art methods from previous works. Specifically, we compare to supervised approaches (DUF (Jo et al., 2018), Enet (Sajjadi et al., 2017), FRVSR (Sajjadi et al., 2018)), and TecoGAN (Chu et al., 2018) as a GAN-based alternative. As shown by the qualitative comparison in Fig. 6 and the supplemental videos, all existing approaches have difficulties to stabilize the input data and produce strong artifacts. We also re-trained a TecoGAN model with our data set, which also leads to undesirable results.

Standard metrics for image quality often fail to relate to the image quality perceived by humans. PSNR values fluctuate during training and across results and in practice have very limited significance. We thus use perceptual metrics, measuring LPIPS for perceptual similarity to single target images, i.e. LPIPS$(G(x_t), y_t)$ (Zhang et al., 2018b). As measure for temporal coherence we evaluate tLP $=$ |LPIPS$(G(x_{t-1}), G(x_t))$ $-$ LPIPS(y_{t-1}, y_t)| (Chu et al., 2018), which suffers from the

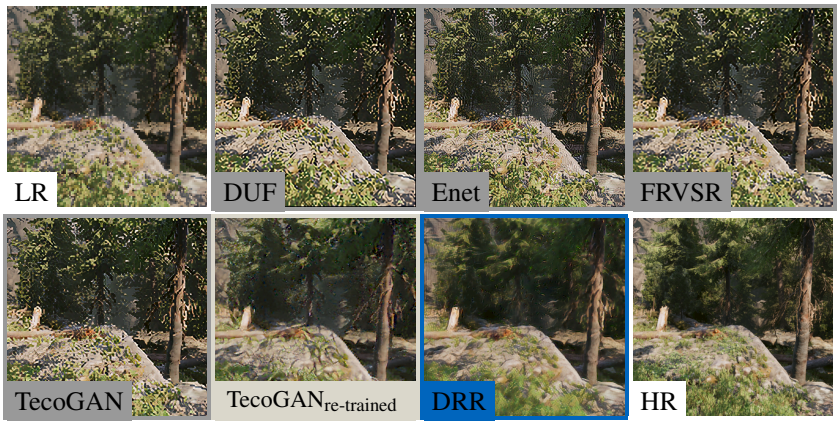

Figure 6: Models trained with natural video all enhance aliasing as the larger changes between adjacent pixels are detected as edges. Also, re-training a model such as TecoGAN with our data set yields undesirable results.

Table 1: Quantitative evaluation: LPIPS yields single-image (spatial) perceptual similarity to GT, tLP measures differences in perceptual temporal coherence to GT, while T-diff computes the absolute difference of warped subsequent frames (not accurate, only shown for completeness). $D^{\ominus}_{s,t}$ and $D^{*}_{s,t}$ are models using normal ResBlocks instead of DRRs, while $D^{\ominus}_{s,t}$ uses a smaller discriminator. The best two scores for LPIPS and tLP rows are highlighted in bold.

Models:	GT	TecoGAN	L^2	$D^{\ominus}_{s,t}$	$D^{*}_{s,t}$	RDA	DRR-in	DRR
LPIPS$\times$10 $\downarrow$	-	3.089	4.129	3.426	3.210	2.904	**2.888**	**2.686**
T-diff$\times$100 $\downarrow$	4.853	8.108	4.850	5.673	5.550	5.677	5.672	5.796
tLP$\times$100 $\downarrow$	-	4.258	**2.015**	4.329	3.521	4.353	3.586	**2.752**

common problem that the temporal coherence of blurry data is generally better, as can be seen from "L^2". Thus, it should be viewed in conjunction with the spatial LPIPS for assessing the overall video quality. For completeness, we also compute the in-place metric T-diff $= ||W(G(x_{t-1})) - G(x_t)||_2$) (Chen et al., 2017), using the captured HR motion for increased accuracy. The results are given in table 1, and confirm the qualitative comparisons so far: our method yields the best temporal coherence in terms of tLP score. In addition, our generators with warped connections all yield excellent LPIPS scores around 2.7 (last three columns), but the DRR connections are crucial to obtain good temporal changes in addition to the details. The DRR tLP score is only surpassed by L^2, which, however, yields clearly sub-optimal image details.

8 Discussion and Conclusions

As we focus on real-time rendering as our use case scenario, ideally the performance of the inference step needs to surpass the performance of the renderer. For a desired output resolution of 1920×1080, our pre-trained model takes 113ms per frame on average.[2] Although this is not yet fast enough for real-time applications, we expect that techniques such as network compression (Choi et al., 2018; Molchanov et al., 2016) and evaluation of the models with dedicated hardware (NVIDIA Corporation, 2017) will easily yield very significant performance improvements.

We have demonstrated how depth-recurrent residual connections can be leveraged to learn stable internal latent-space representations in conditional generator architectures. The DRR connections are particularly promising for iterative models with strongly aliased data, such as low-resolution inputs from a real-time renderer. We have additionally shown how to achieve high quality synthesis in the context of real-time rendering by carefully analyzing and adjusting the network architecture. We anticipate that DRRs could be beneficial for a variety of other tasks such as object tracking (Ning et al., 2017) and physics predictions (Li et al., 2019).

[2]Measured on an NVidia GeForce RTX 2070 GPU.

REFERENCES

Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering*. AK Peters/CRC Press, 2018.

Martin Arjovsky, Soumith Chintala, and Lon Bottou. Wasserstein GAN, 2017. URL http://arxiv.org/abs/1701.07875.

Jose Caballero, Christian Ledig, Andrew P. Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. *CoRR*, abs/1611.05250, 2016. URL http://arxiv.org/abs/1611.05250.

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.*, 36(4):98:1–98:12, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073601. URL http://doi.acm.org/10.1145/3072959.3073601.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proc. Intl. Conf. Computer Vision (ICCV)*, 2017.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Universal deep neural network compression. *arXiv preprint arXiv:1802.02271*, 2018.

Mengyu Chu, You Xie, Laura Leal-Taixé, and Nils Thuerey. Temporally coherent gans for video super-resolution (tecogan). *CoRR*, abs/1811.09393, 2018. URL http://arxiv.org/abs/1811.09393.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 184–199, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2672–2680, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969125.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu. Real-time neural style transfer for videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7044–7052, July 2017. doi: 10.1109/CVPR.2017.745.

Jorge Jimenez, Belen Masia, Jose I. Echevarria, Fernando Navarro, and Diego Gutierrez. *GPU Pro 2*, chapter Practical Morphological Anti-Aliasing. AK Peters Ltd., 2011.

Jorge Jimenez, Jose I. Echevarria, Tiago Sousa, and Diego Gutierrez. Smaa: Enhanced morphological antialiasing. *Computer Graphics Forum (Proc. EUROGRAPHICS 2012)*, 31(2), 2012.

Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3224–3232, 2018.

Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. URL http://arxiv.org/abs/1603.08155.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *CoRR*, abs/1807.00734, 2018. URL http://arxiv.org/abs/1807.00734.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1637–1645, 2016.

Jonathan Korein and Norman Badler. Temporal anti-aliasing in computer generated animation. In *ACM SIGGRAPH Computer Graphics*, volume 17(3), pp. 377–388. ACM, 1983.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. URL http://arxiv.org/abs/1609.04802.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *ICLR*, 2019.

R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 531–539, Dec 2015. doi: 10.1109/ICCV.2015.68.

Timothy Lottes. FXAA. Technical report, NVIDIA, 02 2009. URL http://developer.download.nvidia.com/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf.

Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *CoRR*, abs/1606.08921, 2016. URL http://arxiv.org/abs/1606.08921.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *ISCAS*. IEEE, 2017.

NVIDIA Corporation. NVIDIA Turing Architecture Whitepaper. https://www.nvidia.com/content/dam/enzz/Solutions/design-visualization/technologies/turingarchitecture/NVIDIA-Turing-Architecture-Whitepaper.pdf, 2017. June 2017.

NVIDIA Corporation. NVIDIA TURING GPU ARCHITECTURE. Technical report, NVIDIA Corporation, 2018. URL https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf.

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003. URL http://distill.pub/2016/deconv-checkerboard.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016. URL http://arxiv.org/abs/1604.07379.

Eduardo Pérez-Pellitero, Mehdi S. M. Sajjadi, Michael Hirsch, and Bernhard Schölkopf. Photorealistic video super resolution. *CoRR*, abs/1807.07930, 2018. URL http://arxiv.org/abs/1807.07930.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

M. S. M. Sajjadi, B. Schlkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4501–4510, Oct 2017. doi: 10.1109/ICCV.2017.481.

Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. *CoRR*, abs/1801.04590, 2018. URL http://arxiv.org/abs/1801.04590.

Unity Technologies. Book of the Dead: Environment. https://assetstore.unity.com/packages/essentials/tutorial-projects/book-of-the-dead-environment-121175, 2019a. [Online; accessed 19.03.2019].

Unity Technologies. FPS Sample. https://github.com/Unity-Technologies/FPSSample, 2019b. [Online; accessed 07.03.2019].

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018. URL http://arxiv.org/abs/1809.00219.

Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*, pp. 370–378, 2015.

You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric GAN for super-resolution fluid flow. *CoRR*, abs/1801.09710, 2018. URL http://arxiv.org/abs/1801.09710.

Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. Semantic image inpainting with progressive generative networks. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pp. 1939–1947, New York, NY, USA, 2018a. ACM. ISBN 978-1-4503-5665-7. doi: 10.1145/3240508.3240625. URL http://doi.acm.org/10.1145/3240508.3240625.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *CoRR*, abs/1608.03981, 2016. URL http://arxiv.org/abs/1608.03981.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018b. URL http://arxiv.org/abs/1801.03924.

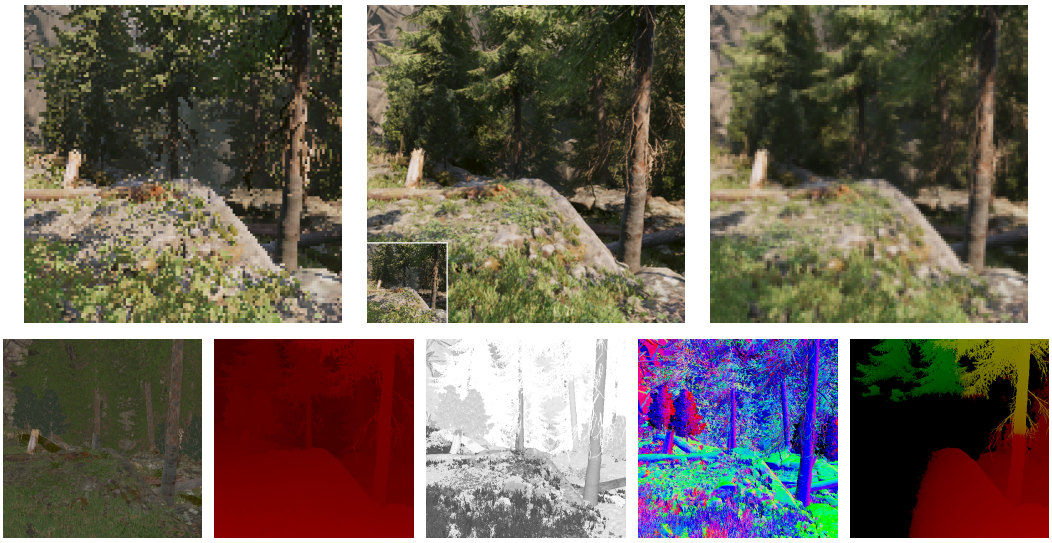

Figure 7: Top row: Examples of input (LR) and target (HR) data used for training. Natural VSR uses a smooth input, illustrated with down-sampling on the right. The bottom row shows additional data captured from the renderer: diffuse color, depth, roughness, surface normals and motion.

A DATA

As source of our data we use the projects "FPS Sample" and "Book of the Dead: Environment" (Unity Technologies, 2019b;a) for the *Unity* engine, both use the HDRP. We captured a total of 57 120-frame sequences, split 50-5-2 for training, validation and testing. For each frame we have lit color (the final image), unlit diffuse color, view-space surface normals, roughness, screen-space motion and depth for both HR and LR. This data is easy to acquire as it can be inferred from the scene, geometry and materials and is rendered by default in Unity's HDRP. However, the use of unlit color, normals or roughness had no tangible effects during our tests. Most post-processing effect have been turned off, but the HR color is augmented with TAA. HR is rendered and captured at a resolution of 512×512, LR at 128×128.

B NETWORK ARCHITECTURE AND TRAINING

The details of our network architecture are given in figures 8 and 9. Our generator network represents a modified TecoGAN$^{\ominus}$ generator, and a lager version of TecoGAN's discriminator is likewise used.

A typical training takes 400,000 iterations (ca. 287 epochs). All discriminator related losses, the loss for the discriminator itself as well as the adversarial and feature loss to the generator, are linearly faded in over the first 40k iterations starting from 0. When training with DRR the temporal connections are faded in later during training in a staggered fashion, beginning at 60k iterations with the first block and 8k iterations between each. Every block takes 10k iterations to fully fade in. We also use an exponential learning rate decay during the last 150k iterations, decaying to 65% every 30k iterations. The most dominant losses are the L^2, the feature loss and the ping-pong loss.

To balance the adversarial and feature loss during training, the discriminator is only trained if it is not too strong to keep it from overwhelming the generator. We train the discriminator only if the EMA of the sum of $b_{disc} = -\sum log(D(y)) - \sum logD(G(x))$, is below a threshold. For our setting, 0.4 yields stable GAN training runs.

For our final DRR version we use use following weights for the terms of our loss function. L2 content loss: 1.0, ping-pong loss: 0.5, feature loss: 3.6, and adversarial loss: 0.1. In addition, we use the following hyper-parameter settings. discriminator learning rate: 3.5e-5, discriminator balancing threshold: 0.4, learning rate: 5e-5, adam beta: 0.9.

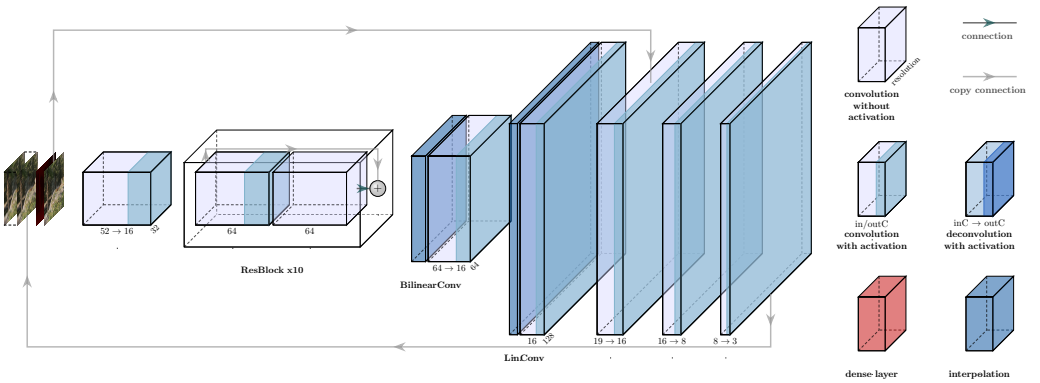

Figure 8: Our modified generator: in addition to the LR color and frame-recurrent input (generator output of the previous frame encoded in $3 * 16$ channel) we also add the LR depth. After an initial convolution most work is done in the 10 sequential ResBlocks (same as TecoGAN$^{\ominus}$, 9 omitted for visibility). The latent image is then scaled to output resolution by 2 resize convolutions and fine-tuned by another 3 convolutions after the bilinear interpolated LR color is appended. All convolutions have 3×3 kernel size and ReLU or no activation.

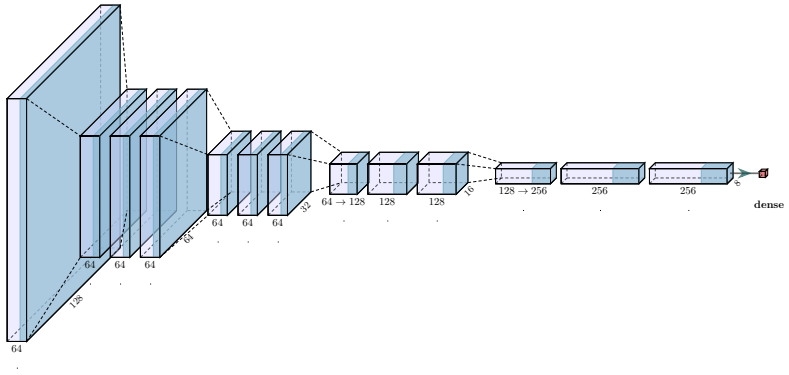

Figure 9: The extended discriminator: 3-frame sequences of HR color are provided as input from the left. The down-scaling is done by strided convolutions with a stride of 2. All convolutions have 4×4 kernel size and lReLU activation with leak 0.2. The dense layer for the final score has σ activation.

C Additional Results

To highlight the range of content our trained model can produce, we show additional sequences in terms of their low-resolution input and inferred output in figures 10 and 11. In order to fully assess the quality, please check the supplemental material document, which contains animated sequences.

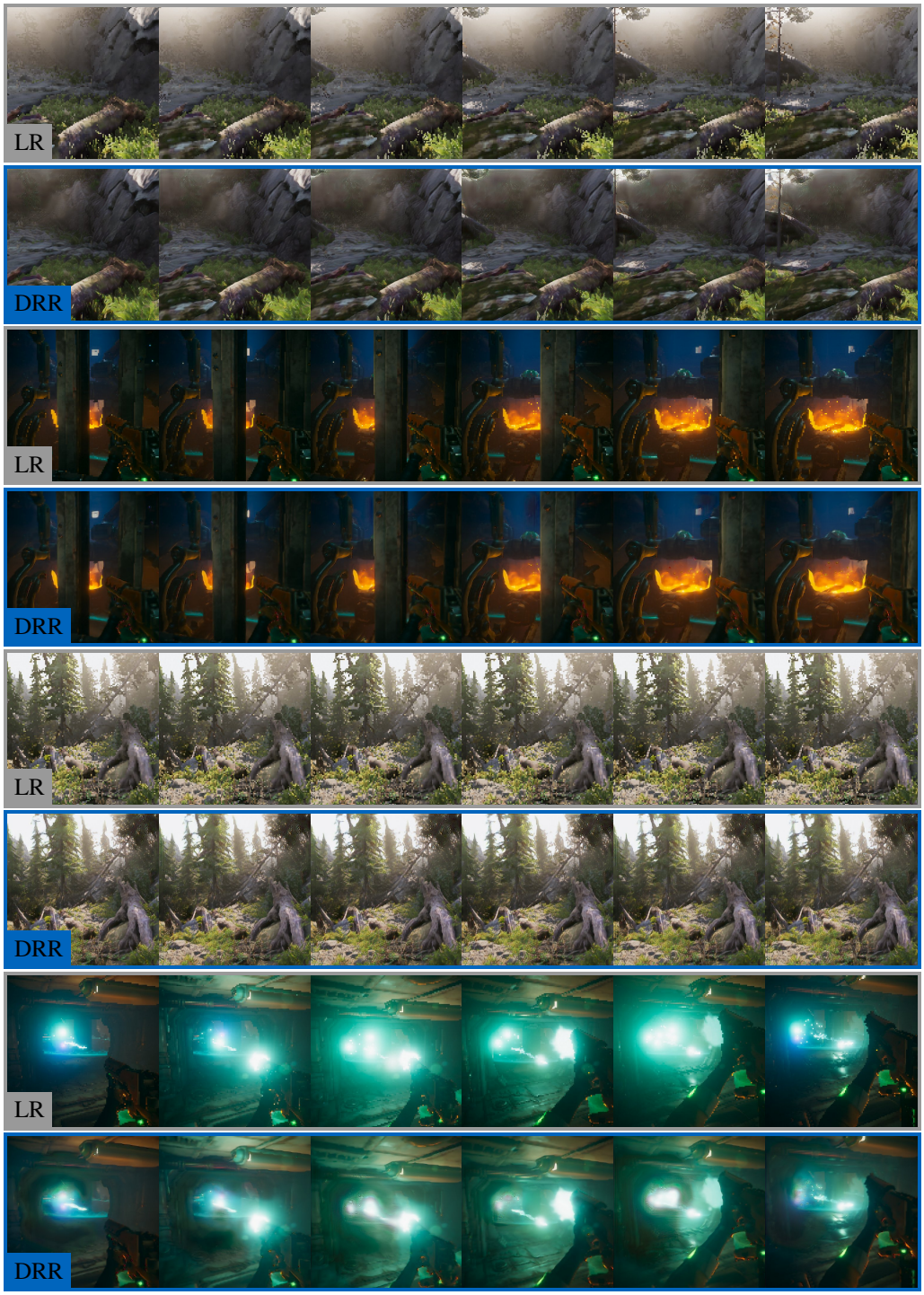

Figure 10: Several example sequences from our test data set (each top row shows the input, the bottom row the result inferred by our model).

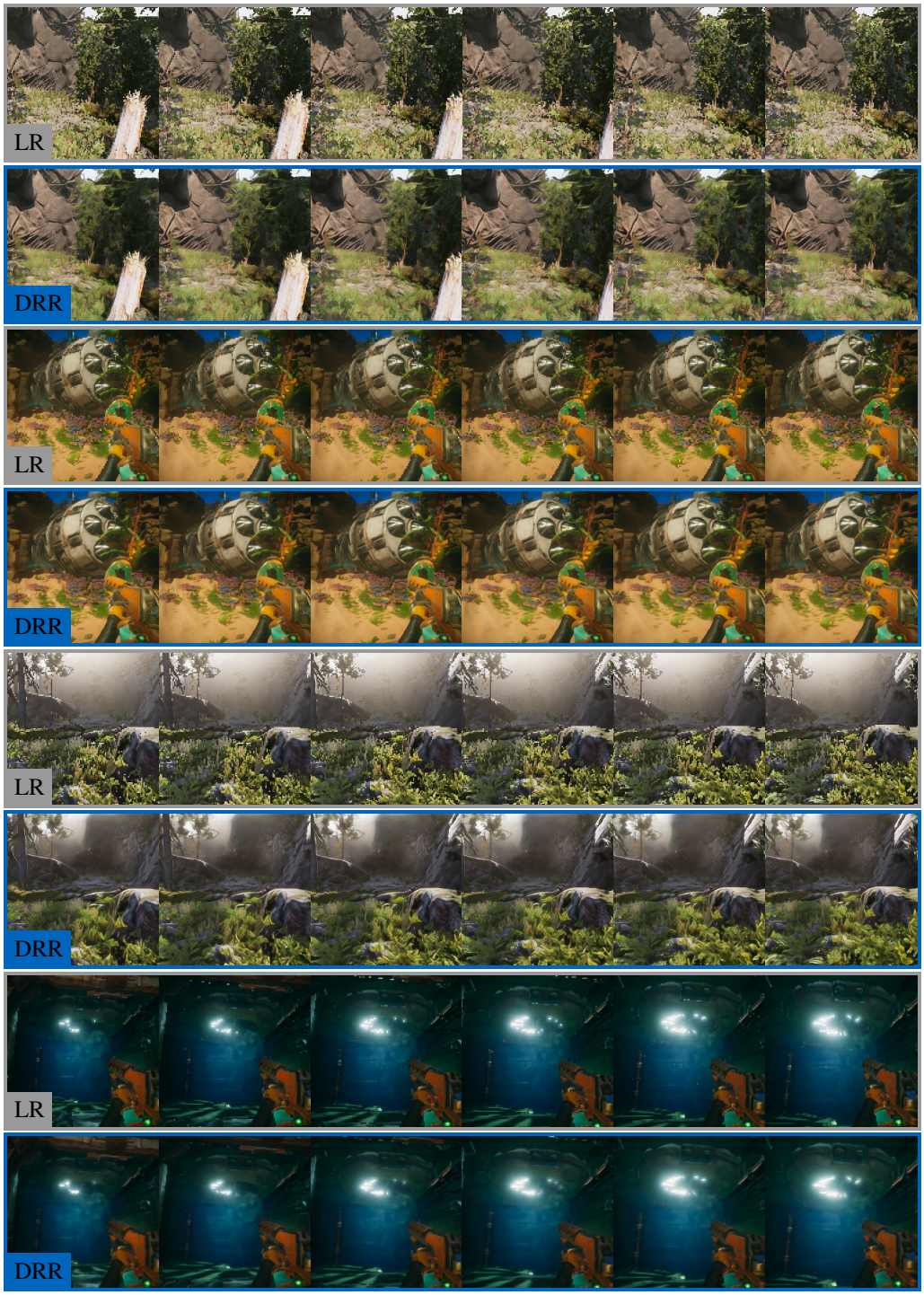

Figure 11: Additional example sequences from our test data set (each top row shows the input, the bottom row the result inferred by our model).