Two-Stream Convolutional Networks for Dynamic Texture Synthesis

Matthew Tesfaldet
Department of Electrical Engineering and Computer Science
York University
Toronto, Canada
mtesfald@eecs.yorku.ca

Konstantinos G. Derpanis
Department of Computer Science
Ryerson University
Toronto, Canada
kosta@scs.ryerson.ca

Marcus A. Brubaker
Department of Electrical Engineering and Computer Science
York University
Toronto, Canada
mab@eecs.yorku.ca

Abstract

We introduce a two-stream model for dynamic texture synthesis. Our model is based on pre-trained convolutional networks (ConvNets) that target two independent tasks: (i) object recognition, and (ii) optical flow prediction. Given an input dynamic texture, statistics of filter responses from the object recognition ConvNet encapsulates the per frame appearance of the input texture, while statistics of filter responses from the optical flow ConvNet models its dynamics. To generate a novel texture, a noise input sequence is optimized to simultaneously match the feature statistics from each stream of the example texture. Inspired by recent work on image style transfer and enabled by the two-stream model, we also apply the synthesis approach to combine the texture appearance from one texture with the dynamics of another to generate entirely novel dynamic textures. We show that our approach generates novel, high quality samples that match both the framewise appearance and temporal evolution of an input example.

1 Introduction

Many common temporal visual patterns are naturally described by the ensemble of appearance and dynamics (*i.e.*, temporal pattern variation) of their constituent elements. Examples of such patterns include fire, fluttering vegetation, wavy water among others. Understanding and characterizing these temporal patterns has long been a problem of interest in human perception, computer vision, and computer graphics. These patterns have been studied under a variety of names, including turbulent-flow motion [18], temporal textures [28], time-varying textures [3], dynamic textures [7], textured motion [43] and spacetime textures [6]. Here, we adopt the term "dynamic texture". In this work, we propose a factored analysis of dynamic textures in terms of appearance and temporal dynamics.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

This factorization is then used to enable dynamic texture synthesis which, based on example texture inputs, generates a novel dynamic texture instance that is perceptually indistinguishable.

Our model is constructed from two convolutional networks (ConvNets), an appearance stream and a dynamics stream, which have been pre-trained for object recognition and optical flow prediction, respectively. Similar to previous work on spatial textures [17, 31, 12], we summarize an input dynamic texture in terms of a set of spatiotemporal statistics of filter outputs from each stream. The appearance stream ConvNet models the per frame appearance of the input texture, while the dynamics stream ConvNet models its temporal dynamics. The synthesis process consists of iteratively coercing an initial white noise pattern such that its spatiotemporal statistics from each stream match those of the input texture. The architecture is inspired by insights from human perception and neuroscience. In particular, psychophysical studies [5] show that humans are able to perceive the structure of a dynamic texture even in the absence of appearance cues, suggesting that the two streams are effectively independent. Similarly, the two-stream hypothesis [15] models the human visual cortex in terms of two pathways, the ventral stream (involved with form representation and object recognition) and the dorsal stream (involved with motion processing).

In this paper, our two-stream analysis of dynamic textures is applied to texture synthesis. We consider a range of dynamic textures and show that our approach generates novel, high quality samples that match both the framewise appearance and temporal evolution of an input example. Further, the factorization of appearance and dynamics enables a novel form of style-transfer, where dynamics of one texture are combined with the appearance of a different one, *cf.* [13]. This can even be done using a single image as an appearance target, which allows portions of static images to be animated.

2 Related work

There are two general approaches that have dominated the texture synthesis literature: non-parametric sampling approaches that synthesize a texture by sampling pixels of a given source texture [9, 45, 35, 25], and statistical parametric models (*e.g.*, [12]). As our approach is an instance of a parametric model, here we focus on these approaches.

The statistical characterization of visual textures was introduced by the seminal work of Julesz [22]. He conjectured that particular statistics of pixel intensities were sufficient to partition spatial textures into metameric (*i.e.*, perceptually indistinguishable) classes. Later work leveraged this notion for texture synthesis [17, 31]. In particular, inspired by the early stages of visual processing, statistics of (handcrafted) multi-scale oriented filter responses were used to iteratively coerce an initial noise pattern to match the filter response statistics of an input texture. More recently, Gatys et al. [12] demonstrated impressive results by replacing the linear filter bank with a ConvNet that, in effect, served as a proxy for the ventral visual processing stream. Textures are modeled in terms of the correlations between filter responses within several layers of the network. In subsequent work, this texture model was used in image style transfer [13], where the style of one image was combined with the image content of another to produce a new image. Ruder et al. [34] extended this model to video and used optical flow to enforce the temporal consistency of the resulting stylized imagery.

Variants of linear autoregressive (AR) models have been studied [40, 7] that jointly model appearance and dynamics of the spatiotemporal pattern. More recent work has considered ConvNets as a basis for modeling dynamic textures. Xie et al. [46] proposed a spatiotemporal generative model where each dynamic texture is modeled as a random field defined by multiscale, spatiotemporal ConvNet filter responses and dynamic textures are realized by sampling the model. Unlike our current work, which assumes pretrained fixed networks, this approach requires the ConvNet weights to be trained using the input texture prior to synthesis. Most closely related to our approach is the recent spatiotemporal extension of Gatys et al. [12] to model and synthesize dynamic textures [11]. In addition to modeling the texture appearance for each frame via summary statistics of the correlation of ConvNet filter responses within a layer, the (purely) temporal correlation of filter responses were considered. In contrast, our temporal filtering architecture is more expressive as it is tuned to spatiotemporal oriented structures. Moreover, as will be demonstrated, this factorization of a pattern in terms of its appearance and dynamics enables a novel form of style transfer, where the dynamics of one pattern are transfered to the appearance of another to generate an entirely new dynamics texture. To the best of our knowledge, we are the first to demonstrate this form of style transfer.

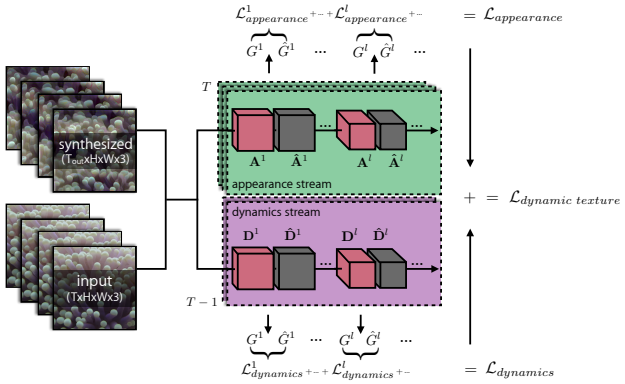

Figure 1: Two-stream dynamic texture generation. Separate sets of Gram matrices represent the appearance and dynamics of the texture. Matching these statistics allows for the generation of novel textures as well as the transfer of style between textures.

The recovery of optical flow from temporal imagery has been a long studied problem in computer vision. Traditionally, it has been addressed by handcrafted approaches *e.g.*, [19, 27, 33]. Recently, ConvNet approaches [8, 32, 20] have been demonstrated to be viable alternatives. Most closely related to our approach are energy models of visual motion [2, 16, 37, 29, 6, 24] that have been motivated and studied in a variety of contexts, including computer vision, visual neuroscience, and visual psychology. Given an input image sequence, these models consist of an alternating sequence of linear and non-linear operations that yield a distributed representation (*i.e.*, implicitly coded) of pixelwise optical flow. In our current work, an energy model motivates the representation of observed dynamics which is then encoded in the architecture of a ConvNet.

3 Technical approach

Our proposed two-stream approach consists of the appearance stream, representing the static (texture) appearance of each frame, and the dynamics stream, representing temporal variations between frames. Each stream consists of a ConvNet and the activation statistics of these networks are used to characterize the dynamic texture. Synthesizing either the appearance or the dynamics of a dynamic texture is then formulated as an optimization problem with the objective of matching the activation statistics. This is summarized in Fig. 1 and the individual pieces are described in turn in the following section.

3.1 Texture model: Appearance stream

The appearance stream follows the spatial texture model introduced by Gatys et al. [12] which we briefly review here. The key idea is that the feature correlations at various levels in a ConvNet trained on an object recognition task captures texture appearance. We use the same publicly available normalized VGG-19 network [38] used by Gatys et al. [12].

To capture the appearance of an input dynamic texture, we first perform a forward pass with each frame of the image sequence through the ConvNet and compute the feature activations, $\mathbf{A}^{lt} \in \mathbb{R}^{N_l \times M_l}$, for various levels in the network, where N_l and M_l denote the number of filters and the number of spatial locations of layer l at time t, respectively. The correlations of the filter responses in a particular layer are averaged over the frames and encapsulated by a Gram matrix $\mathbf{G}^l \in \mathbb{R}^{N_l \times N_l}$ whose entries are given by $G^l_{ij} = \frac{1}{TN_lM_l} \sum_{t=1}^{T} \sum_{k=1}^{M_l} A^{lt}_{ik} A^{lt}_{jk}$, where T denotes the number of input frames and A^{lt}_{ik} denotes the activation of feature i at location k in layer l on the target frame t. The synthesized texture appearance is similarly represented by a Gram matrix $\hat{\mathbf{G}}^{lt} \in \mathbb{R}^{N_l \times N_l}$ whose activations are given by $\hat{G}^{lt}_{ij} = \frac{1}{N_lM_l} \sum_{k=1}^{M_l} \hat{A}^{lt}_{ik} \hat{A}^{lt}_{jk}$, where $\hat{A}^{lt}_{ik}$ denotes the activation of feature i at location k in layer l on the synthesized frame t. The appearance loss, $\mathcal{L}_{\text{appearance}}$, is then defined as the temporal average of the mean squared error between the Gram matrix of the input texture and that of the generated

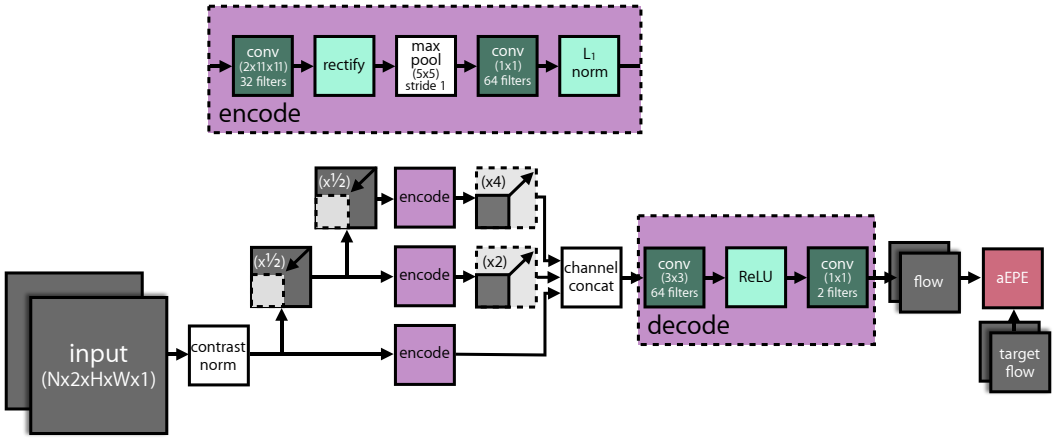

Figure 2: Dynamics stream convolutional network, based on spacetime oriented energy models [37, 6], is trained for optical flow prediction. In this case only three different scales are shown for illustration while in practice we used 5 different scales.

texture computed at each frame:

$$\mathcal{L}_{\text{appearance}} = \frac{1}{L_{\text{app}} T_{\text{out}}} \sum_{t=1}^{T_{\text{out}}} \sum_{l} \|\mathbf{G}^l - \hat{\mathbf{G}}^{lt}\|_F^2 \,, \tag{1}$$

where L_{app} is the number of layers used to compute Gram matrices, T_{out} is the number of frames being generated in the output and $\|\cdot\|_F$ is the Frobenius norm. Consistent with [12], we compute Gram matrices on the following layers: *conv1_1*, *pool1*, *pool2*, *pool3*, and *pool4*.

3.2 Texture model: Dynamics stream

In designing the dynamics stream of our model there are three primary goals. First, the activations of the network must represent the input pattern's temporal variation. Second, the activations should be largely invariant to the appearance of the images which should be characterized by the appearance stream described above. Finally, the representation must be differentiable to enable synthesis. By analogy to the appearance stream, an obvious choice is a ConvNet architecture suited for computing optical flow (*e.g.*, [8, 20]) which is naturally differentiable. However, with most such models it is unclear how invariant their layers are to appearance. Instead, we propose a novel network architecture which is motivated by the spacetime oriented energy model [37, 6].

In motion energy models, the velocity of image content (*i.e.*, motion) is interpreted as a three-dimensional orientation in the x-y-t spatiotemporal domain [10, 2, 44, 16, 37]. In the frequency domain, the signal energy of a translating pattern can be shown to lie on a plane through the origin where the slant of the plane is defined by the velocity of the pattern. Thus, motion energy models attempt to identify this orientation-plane (and hence the pattern's velocity) via a set of image filtering operations. More generally, as discussed in Derpanis et al. [6], the constituent spacetime orientations for a spectrum of common visual patterns (including translation and dynamic textures) can serve as a basis for describing the temporal variation of an image sequence. This suggests that such motion energy models may form an ideal basis for our dynamics stream.

Specifically, we use the spacetime oriented energy model [37, 6] to motivate our network architecture which we briefly review here; see [6] for a more in-depth description. Given an input spacetime volume, a bank of oriented 3D filters are applied which are sensitive to a range of spatiotemporal orientations. These filter activations are rectified (squared) and pooled over local regions to make the responses robust to the phase of the input signal, *i.e.*, robust to the alignment of the filter with the underlying image structure. Next, filter activations consistent with similar spacetime orientations are summed. These responses provide a pixelwise distributed measure of which orientations (frequency domain planes) are present in the input. However, these responses are confounded by local image contrast and, as a result, it is difficult to determine whether a high response is indicative of the presence of a spacetime orientation or simply due to high image contrast. To address this ambiguity,

an L_1 normalization is applied across orientations which results in a representation that is robust to local appearance variations but highly selective to spacetime orientation.

Using this model as our basis, we propose the following fully convolutional network architecture [36]. The input to our ConvNet is a pair of greyscale images. These are first normalized to have mean zero and unit variance. This step provides a level of invariance to overall brightness and contrast, *i.e.*, global additive and multiplicative signal variations. The first layer consists of 32 3D spacetime convolution filters of size $11 \times 11 \times 2$ (height $\times$ width $\times$ time). Next, a squaring activation function and 5×5 spatial max-pooling (with a stride of one) is applied to make the responses robust to local signal phase. Following this, a 1×1 convolution layer with 64 filters allows for the combination of energy measurements which are consistent with the same orientation. Finally, to remove local contrast dependence, an L_1 divisive normalization is applied.

To capture spacetime orientations beyond those capable with the limited receptive fields used in the initial layer, we compute a five-level spatial pyramid consisting of downsampling by a factor of two between each level. The multi-resolution results are processed independently with the same spacetime oriented energy model and then bilinearly upsampled to the original resolution and concatenated.

Prior energy model instantiations (*e.g.*, [2, 37, 6]) use handcrafted filter weights. While a similar approach could be followed here, we instead opt to learn the weights so that they are better tuned to natural imagery. To train the network weights, we add additional decoding layers that take the concatenated distributed representation and applies: 3×3 convolution (with 64 filters), ReLU activation and a 1×1 (with 64 filters) convolution and finally a two channel output that encodes optical flow directly. The proposed architecture is illustrated in Fig. 2.

To train the network, we use the standard average endpoint error (aEPE) flow metric (*i.e.*, L_2 norm) between the predicted flow and the ground truth flow as the loss. Since no large-scale flow dataset exists that captures natural imagery with groundtruth flow, we take an unlabeled video dataset and apply an existing flow estimator [33] to estimate optical flow for training, as was done in, *e.g.*, [41]. For training data we used videos from the UCF-101 dataset [39] augmented with random 90 degree rotations and optimized the aEPE loss using Adam [23]. Inspection of the filters learned in the initial layer showed evidence of spacetime oriented filters, consistent with those found in [6].

As with the appearance stream, correlations of the filter responses in a particular layer of the dynamics stream are averaged over the number of image frame pairs and encapsulated by a Gram matrix $\mathbf{G}^l \in \mathbb{R}^{N_l \times N_l}$ whose entries are given by $G^l_{ij} = \frac{1}{(T-1)N_l M_l} \sum_{t=1}^{T-1} \sum_{k=1}^{M_l} D^{lt}_{ik} D^{lt}_{jk}$, where D^{lt}_{ik} denotes the activation of feature i at location k in layer l on the target frames t and $t+1$. The dynamics of the synthesized texture is represented by a Gram matrix of feature activation correlations computed separately for each pair of frames $\hat{\mathbf{G}}^{lt} \in \mathbb{R}^{N_l \times N_l}$ with entries $\hat{G}^{lt}_{ij} = \frac{1}{N_l M_l} \sum_{k=1}^{M_l} \hat{D}^{lt}_{ik} \hat{D}^{lt}_{jk}$, where D^{lt}_{ik} denotes the activation of feature i at location k in layer l on the synthesized frames t and $t+1$. The dynamics loss, $\mathcal{L}_{\text{dynamics}}$, is defined as the average of the mean squared error between the Gram matrices of the input texture and those of the generated texture:

$$\mathcal{L}_{\text{dynamics}} = \frac{1}{L_{\text{dyn}}(T_{\text{out}} - 1)} \sum_{t=1}^{T_{\text{out}}-1} \sum_{l} \|\mathbf{G}^l - \hat{\mathbf{G}}^{lt}\|_F^2, \tag{2}$$

where L_{dyn} is the number of ConvNet layers being used in the dynamics stream.

Here we propose to use the output of the concatenation layer, where the multiscale distributed representation of orientations is stored, as the layer to compute the Gram matrix. While it is tempting to use the predicted flow output from the network, this generally yields poor results. Due to the complex, temporal variation present in dynamic textures, they contain a variety of local spacetime orientations rather than a single dominant orientation. As result, the flow estimates will tend to be an average of the underlying orientation measurements and consequently not descriptive. We also explored using the outputs of the L_1 normalized layers. This worked reasonably for simple motions, but we generally found that the concatenation layer provided more pleasing results. Examples of the results with these other layers can be found in the supplemental material.

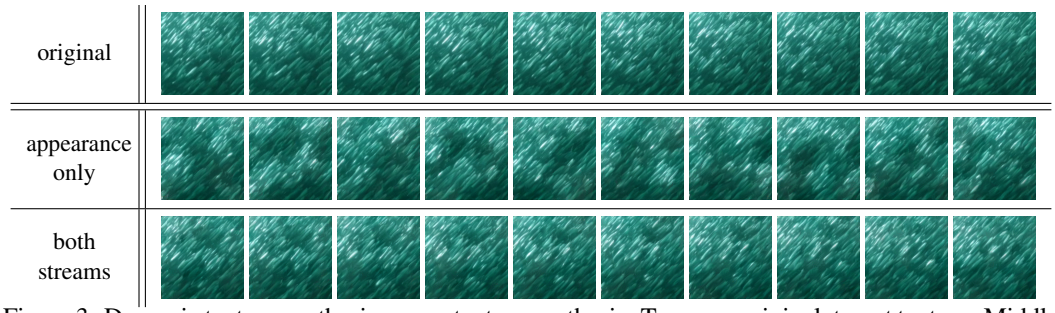

Figure 3: Dynamic texture synthesis versus texture synthesis. Top row: original, target texture. Middle row: texture synthesis without dynamics constraints shows no temporal coherence but consistent, per-frame appearance. Bottom row: including both streams induces a consistent motion.

3.3 Texture generation

The overall dynamic texture loss consists of the combination of the appearance loss, (1), and the dynamics loss, (2):

$$\mathcal{L}_{\text{dynamic texture}} = \alpha \mathcal{L}_{\text{appearance}} + \beta \mathcal{L}_{\text{dynamics}}, \tag{3}$$

where α and β are the weighting factors for the appearance and dynamics content, respectively. Dynamics textures are implicitly defined as the local minima of this loss. Generation of textures is done by optimizing Eq. (3) with respect to the spacetime volume, *i.e.*, the pixels of the video. Variations in the resulting texture are found by initializing the optimization process using IID Gaussian noise. Consistent with previous work [12], we use L-BFGS [26] to perform the optimization.

Naive application of the outlined approach will consume increasing amounts of memory as the temporal extent of the dynamic texture grows, making it impractical to generate longer sequences. Instead, long sequences can be incrementally generated by increasing the length of the sequence and only optimizing the most recent frames of the sequence. In particular, the first frames of the sequence would be produced directly as described. However, subsequent frames are generated in small batches by fixing their initial frame to be the last synthesized frame of the previous batch. This ensures temporal consistency across synthesized batches and can be viewed as a form of coordinate descent optimization for the full sequence objective. The flexibility of this framework allows other texture generation problems to be handled simply by altering the initialization of frames and controlling which frames or regions of frames can be updated. This is described further below.

4 Empirical evaluation

Ultimately, the goal of (dynamic) texture synthesis is to generate samples that cannot be distinguished from the input texture by a human observer. In this section, we present a variety of synthesis results. Given their temporal nature, our results are best viewed as videos. Please refer to the supplemental materials for our full texture synthesis results. Our two-stream architecture was implemented using TensorFlow [1]. Source code will be made available upon the paper's publication. Results were generated using an NVIDIA Titan X (Pascal) GPU and synthesis times ranged between one to three hours to generate 12 frames with an image resolution of 256×256.

4.1 Dynamic texture synthesis

We applied our dynamic texture synthesis process to a wide range of textures which were selected from the DynTex [30] as well as others collected in-the-wild. Included in our supplemental material are synthesized results of over 50 different textures that encapsulate a range of phenomena, such as flowing water, waves, clouds, fire, rippling flags, waving plants, and schools of fish. Some sample frames are shown in Fig. 4 but we encourage readers to view the videos to fully appreciate the results which demonstrate that our two-stream synthesis approach produces compelling dynamic textures. The supplemental material also includes sequences generated incrementally, as described in Sec. 3.3. No discernible temporal discontinuity is observed in these sequences.

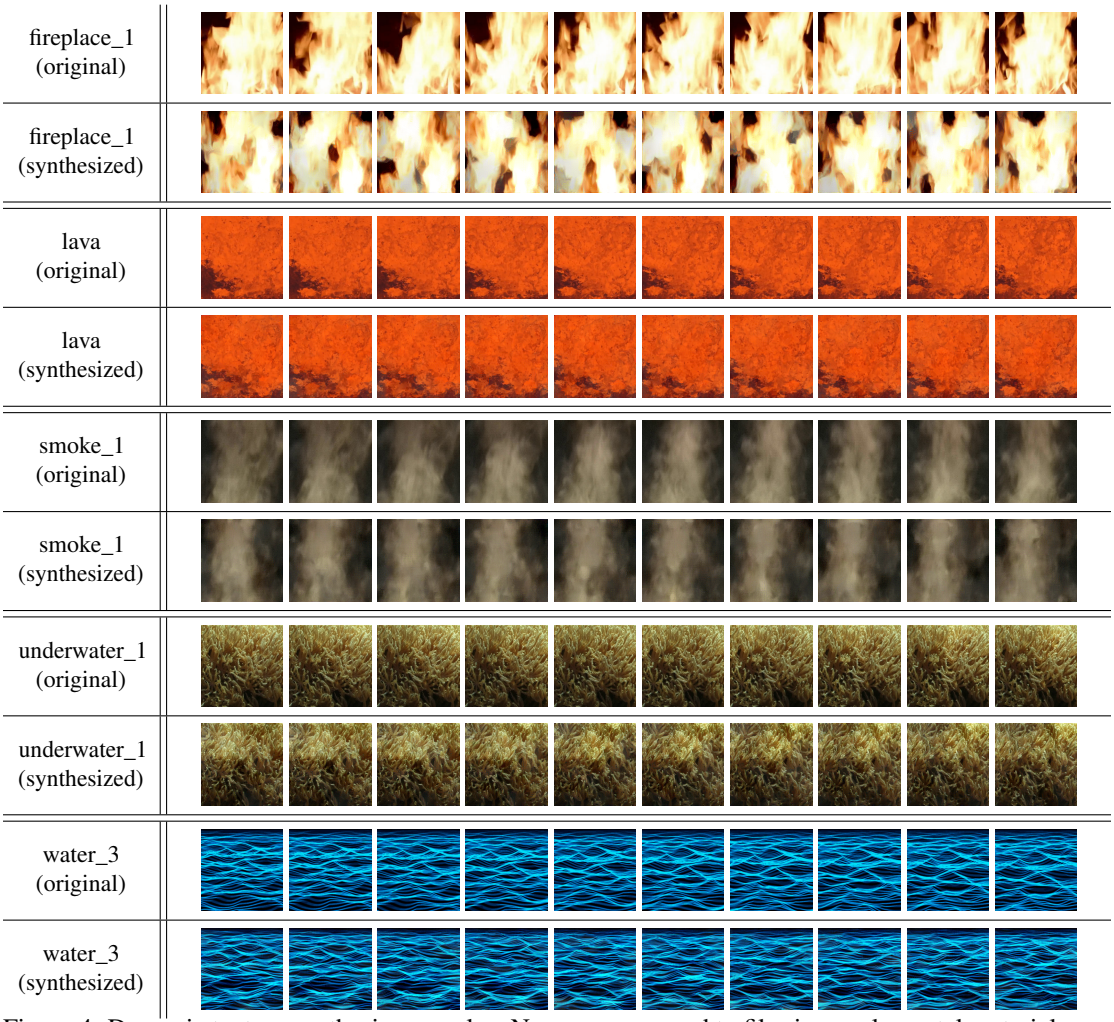

Figure 4: Dynamic texture synthesis examples. Names correspond to files in supplemental material.

An interesting extension that we briefly explored and also provide in the supplemental material, are textures where there is no discernible temporal seam between the last and first frames. Played as a loop, these textures appear to be temporally endless. This is trivially achieved by adding an additional loss to the dynamics stream that ties the last frame to the first.

Some of the failure modes of our method are presented in Fig. 5. In general, we find that most failures result from inputs which violate the underlying assumption of a dynamic texture, *i.e.*, the appearance and/or dynamics are not spatially homogeneous. In the case of the escalator example, the long edge structures in the appearance are not spatially homogeneous, further the dynamics are somewhat variable as perspective effects change the motion from downward to outward. The resulting synthesized texture captures an overall downward motion but lacks the perspective effects and is unable to reproduce the long edge structures. This is consistent to what was seen by [12] with appearance alone. Another example is the flag sequence. In this case the rippling dynamics are relatively homogeneous but the appearance spatially varies. As expected, the generated texture does not faithfully reproduce the appearance; however, it does exhibit plausible rippling dynamics. In the supplemental material, we include an additional failure case (`cranberries` video) consisting of a swirling pattern. Our model faithfully reproduces the appearance but is not able to capture the spatially varying dynamics. Interestingly, it does still produce a plausible dynamic texture.

Appearance vs. dynamics streams We sought to verify that the appearance and dynamics streams were capturing complementary information. To validate that the texture generation of multiple frames would not induce dynamics consistent with the input, we generated frames starting from

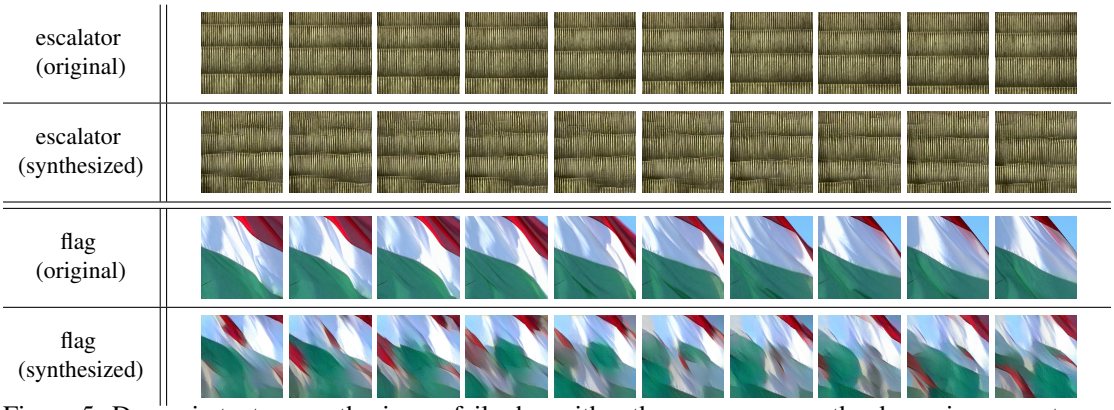

Figure 5: Dynamic texture synthesis can fail when either the appearance or the dynamics are not homogeneous as in the case of these sequences.

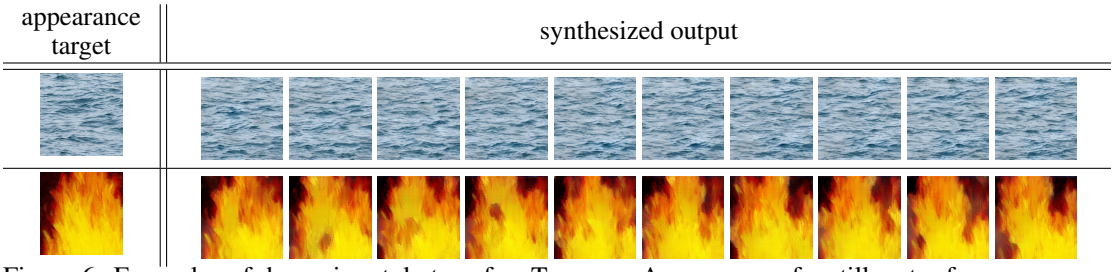

Figure 6: Examples of dynamics style transfer. Top row: Appearance of a still water frame was used with the dynamics of a different water texture (water_4 in supplemental). Bottom row: The appearance of a painting of fire was used with the dynamics of a real fire (fireplace_1 in Fig. 4 and supplemental). Animated results and additional examples are available in the supplemental material.

randomly generated noise but only using the appearance statistics and corresponding loss, *i.e.*, Eq. 1. As expected, this produced frames that were valid textures but with no coherent dynamics present. Results for a sequence containing a school of fish is shown in Fig. 3; to examine the dynamics, see `fish` in the supplemental materials.

Similarly, to validate that the dynamics stream did not inadvertently include appearance information, we generated volumes using the dynamics statistics and corresponding loss only, *i.e.*, Eq. 2. The resulting frames had extremely low dynamic range, indicating a general invariance to appearance. Due to the low dynamic range of the generated results, we do not present them. This suggests that our two-stream dynamic texture representation factors appearance and dynamics, as desired.

4.2 Dynamic style transfer

The underlying assumption of our model is that appearance and dynamics of texture can be factorized. As such, it should allow for the transfer of the dynamics of one texture onto the appearance of another. This has been explored previously for artistic style transfer [4, 14] with static imagery. We accomplish this with our model by performing the same optimization as above, but with the target Gram matrices for appearance and dynamics computed from different textures.

A dynamics style transfer result is shown in Fig. 6 (top), using two real videos. Additional, examples of dynamics style transfer are available in the supplemental materials. We note that when performing dynamics style transfer it is important that the appearance structure be similar in scale and semantics; otherwise, the generated dynamic textures will look unnatural. For instance, transferring the dynamics of a flame onto a water scene will generally be ineffective.

We can also apply the dynamics of a texture to a static input image, as the target Gram matrices for the appearance loss can be computed on just a single frame. This allows us to effectively animate regions of a static image. The result of this process can be striking and is visualized in Fig. 6 (bottom), where the appearance is taken from a painting and the dynamics from a real world video.

5 Discussion and summary

In this paper, we have presented a novel, two-stream model of dynamic textures using ConvNets to represent the appearance and dynamics. We applied this model to a variety of dynamic texture synthesis tasks and showed that, so long as the input textures are generally true dynamic textures, *i.e.*, have spatially invariant statistics and spatiotemporally invariant dynamics, the resulting synthesized textures are compelling. Further, we showed that the two-stream model enabled dynamics style transfer, where the appearance and dynamics information from different sources can be combined to generate a novel texture.

We have explored this model thoroughly and found a few limitations. First, much like has been reported in recent image style transfer work [13], we have found that high frequency noise and chromatic aberrations are a problem in generation. Another issue that arises is the model fails to faithfully capture spatially inhomogeneous patterns, *e.g.*, the escalator in Fig. 5 (both appearance and dynamics spatially vary) and swirling patterns (see `cranberries` video in supplemental where dynamics spatially vary). By collapsing the local statistics into a Gram matrix, the spatial pattern organization is lost. Simple post-processing methods may alleviate some of these issues but we believe that they may also point to a need for a better targeted representation. Beyond removing limitations, a natural next step would be to extend the idea of a factorized representation into the feedforward networks that have found success in static image synthesis, *e.g.*, [21, 42].

References

[1] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *JOSA–A*, 2(2):284–299, 1985.

[3] Ziv Bar-Joseph, Ran El-Yaniv, Dani Lischinski, and Michael Werman. Texture mixing and texture movie synthesis using statistical learning. *T-VCG*, 7(2):120–135, 2001.

[4] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks.

[5] James E. Cutting. Blowing in the wind: Perceiving structure in trees and bushes. *Cognition*, 12(1):25 – 44, 1982.

[6] Konstantinos G. Derpanis and Richard P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. *PAMI*, 34(6):1193–1205, 2012.

[7] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *IJCV*, 51(2):91–109, 2003.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[9] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.

[10] M. Fahle and T. Poggio. Visual hyperacuity: Spatiotemporal interpolation in human vision. *Proceedings of the Royal Society of London B: Biological Sciences*, 213(1193):451–477, 1981.

[11] Christina M. Funke, Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Synthesising dynamic textures using convolutional neural networks. *CoRR*, abs/1702.07006, 2017.

[12] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, pages 262–270, 2015.

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[14] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. *arXiv:1611.07865*, 2016.

[15] Melvyn A. Goodale and A. David. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15, 1992.

[16] David J. Heeger. Optical flow using spatiotemporal filters. *IJCV*, 1(4):279–302, 1988.

[17] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238, 1995.

[18] David J. Heeger and Alex P. Pentland. Seeing structure through chaos. In *IEEE Motion Workshop: Representation and Analysis*, pages 131–136, 1986.

[19] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *A.I.*, 17:185–203, 1981.

[20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[22] Béla Julesz. Visual pattern discrimination. *IRE Trans. Information Theory*, 8(2):84–92, 1962.

[23] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

[24] Kishore Konda, Roland Memisevic, and Vincent Michalski. Learning to encode motion using spatio-temporal synchrony international conference on learning representation. In *ICLR*, 2014.

[25] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *SIGGRAPH*, pages 277–286, 2003.

[26] Dong C. Liu and Jorge Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989.

[27] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[28] R. Nelson and R. Polana. Qualitative recognition of motion using temporal textures. *CVGIP*, 56(1), 1992.

[29] Shinji Nishimoto and Jack L. Gallant. A three-dimensional spatiotemporal receptive field model explains responses of area mt neurons to naturalistic movies. *Journal of Neuroscience*, 31(41):14551–14564, 2011.

[30] Renaud Péteri, Sándor Fazekas, and Mark J. Huiskes. DynTex : a Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters*, doi: 10.1016/j.patrec.2010.05.009. http://projects.cwi.nl/dyntex/.

[31] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, 2000.

[32] A. Ranjan and M. J. Black. Optical Flow Estimation using a Spatial Pyramid Network. *ArXiv e-prints*, November 2016.

[33] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, pages 1164–1172, 2015.

[34] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *GCPR*, pages 26–36, 2016.

[35] Arno Schödl, Richard Szeliski, David Salesin, and Irfan A. Essa. Video textures. In *SIGGRAPH*, pages 489–498, 2000.

[36] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *PAMI*, 39(4):640–651, 2017.

[37] Eero P. Simoncelli and David J. Heeger. A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743 – 761, 1998.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012.

[40] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *ICIP*, pages 823–826, 1996.

[41] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Deep end2end voxel2voxel prediction. In *CVPR Workshops*, pages 402–409.

[42] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[43] Yizhou Wang and Song Chun Zhu. Modeling textured motion: Particle, wave and sketch. In *ICCV*, pages 213–220, 2003.

[44] Andrew B. Watson and Albert J. Ahumada Jr. A look at motion in the frequency domain. In *Motion workshop: Perception and representation*, pages 1–10, 1983.

[45] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488, 2000.

[46] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic textures and sounds by spatial-temporal generative convnet. *arXiv:1606.00972*, 2016.