Two-Stream Convolutional Networks for Dynamic Texture Synthesis

BY

MATTHEW TESFALDET

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

August, 2018

Abstract

This thesis introduces a two-stream model for dynamic texture synthesis. The 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 encapsulate the per-frame appearance of the input texture, while statistics of filter responses from the optical flow ConvNet model its dynamics. To generate a novel texture, a randomly initialized input sequence is optimized to match the feature statistics from each stream of an example texture. Inspired by recent work on image style transfer and enabled by the two-stream model, the synthesis approach is applied to combine the texture appearance from one texture with the dynamics of another to generate entirely novel dynamic textures. The proposed approach generates novel, high quality samples that match both the framewise appearance and temporal evolution of input texture. Finally, a quantitative evaluation of the proposed dynamic texture synthesis approach is performed via a thorough user study.

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The natural world is rich in visual texture. While a precise definition of texture remains to be found, most research consider it as visual patterns that exhibit local spatial variations while maintaining global homogeneity, as shown in Fig. 1.1.

Textures can be static or dynamic: static textures exist in two-dimensional (2D) image space (*e.g.*, grass and water) while dynamic textures extend the notion across time (*e.g.*, fluttering grass and wavy water). As a result, local spatial variations and global homogeneity extend across space *and* time. These temporal patterns have previously been studied under a variety of names, including turbulent flow [23] (for extracting optical flow from fluids undergoing irregular fluctuations), temporal textures [34] (for motion recognition of moving patterns such as windblown trees or rippling water), time-varying textures [3] (for synthesizing stochastically-moving patterns), dynamic textures [8] (for modelling and synthesizing stochastically-moving patterns), textured motion [51] (for modelling and

Figure 1.1: A static texture is a visual pattern that exhibits local spatial variations while maintaining global homogeneity. Observing small apertures across the texture should reveal visual content that looks roughly the same.

synthesizing patterns undergoing stochastic or consistent motion), and spacetime textures [7] (for classifying moving patterns). In this thesis, the term "dynamic texture" is adopted.

Both static and dynamic texture cues play important roles in our perception of surfaces. Understanding and characterizing these patterns has long been a problem of interest in human perception, computer vision, and computer graphics. In computer vision, studying the underlying statistics of textures allows us to gain insight as to how these complex structures can be interpreted and how we may be able to leverage this knowledge to inform certain vision-related tasks. Examples of such tasks include shape from texture [18], texture synthesis [22], and more recently, image style transfer [16].

Shape from texture involves recovering the three-dimensional (3D) shape of an object from a 2D image by using texture as a cue. Gibson [18] proposed the *texture gradient* as the primary basis of surface perception by humans. He revealed that neighbouring areas on a textured surface are perceived differently only due to differences in surface orientation and distance from the observer.

Texture synthesis (Fig. 1.2) is the process of algorithmically constructing a

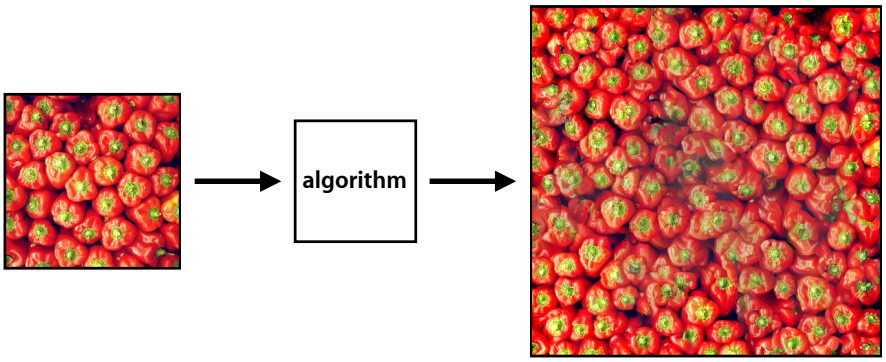

Figure 1.2: (Static) texture synthesis is the process of algorithmically constructing a texture (right) that matches or extends a given source texture (left) by taking advantage of its structural content.

texture that matches or extends a given source texture by taking advantage of its structural content. Heeger and Bergen [22] took advantage of the fact that two textures are often difficult to discriminate when they produce a similar distribution of responses from a bank of linear filters. They used a combination of Laplacian and steerable pyramids to deconstruct a given texture and synthesized a new texture by matching the distributions of responses from each pyramid level. More recently, Gatys *et al.* [15] demonstrated impressive results for texture synthesis by using a convolutional network (ConvNet) instead of a linear bank of filters to model the non-linear spatial statistics of a given texture.

Image style transfer (Fig. 1.3) is a recent extension of texture synthesis where the goal is to synthesize a texture from a source image while constraining the process in order to preserve the semantic content of another image. This can be considered as a texture transfer problem, as previously demonstrated by Efros *et al.* [10] where they transferred a given texture to another image by stitching together small patches of the given texture while conforming to the luminance of the other image. Although the simplicity of their approach was attractive, it failed

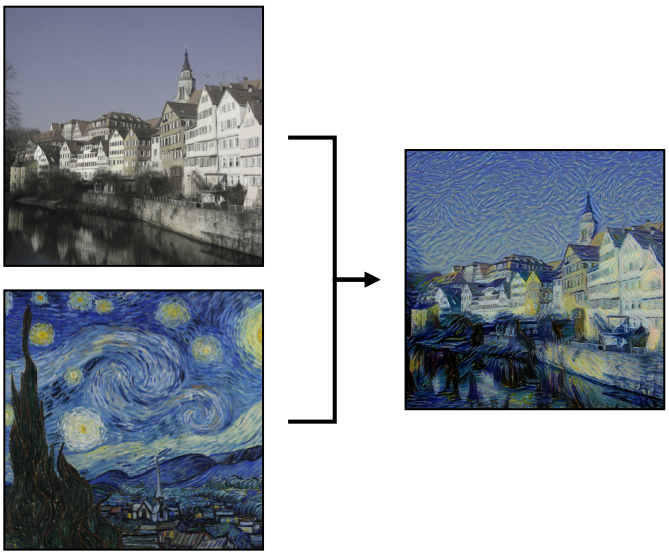

Figure 1.3: Image style transfer. The goal is to synthesize a texture (right) from a source image (bottom-left) while constraining the process in order to preserve the semantic content of another image (top-left).

for highly structured textures due to patch boundary inconsistencies, limiting the selection of acceptable textures. As a follow-up to [15], Gatys *et al.* modified their previous ConvNet to support texture transfer by including an additional objective that enforced the synthesized texture to match the semantic content of a given image, naming the process *image style transfer* [16]. Unlike the patch-based method of Efros *et al.* [10], Gatys *et al.*'s [16] approach of using a ConvNet was more robust to textures with long-range consistencies.

This work extends the ConvNet model of Gatys *et al.* [16] for texture synthesis. Specifically, the focus of this work is on the synthesis of dynamic texture samples based on a single exemplar through the use of ConvNets.

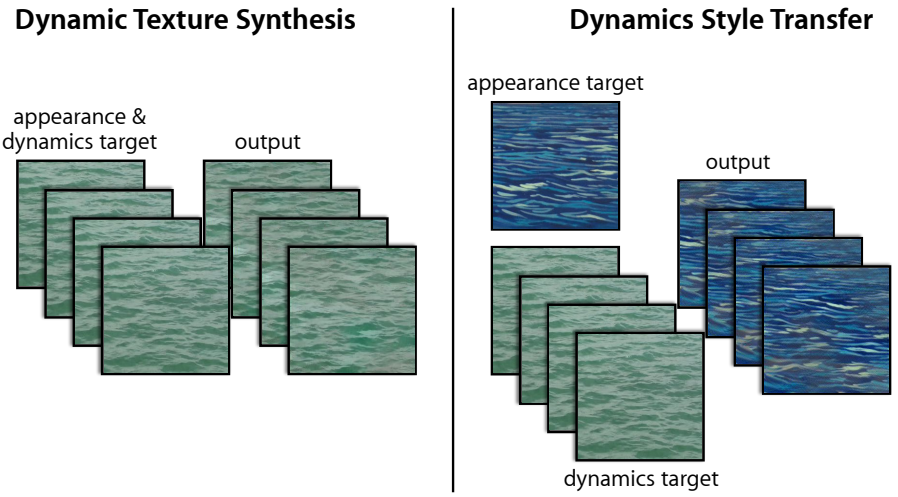

Figure 1.4: Dynamic texture synthesis. (left) Given an input dynamic texture as the target, the two-stream model synthesizes a novel dynamic texture that preserves the target's appearance and dynamics characteristics. (right) The two-stream approach enables synthesis that combines the texture appearance from one target with the dynamics from another, resulting in a composition of the two.

1.2 Outline of thesis

Many common dynamic textures are naturally described by the ensemble of appearance and dynamics (*i.e.*, spatial and temporal pattern variation) of their constituent elements. In this work, a factored analysis of dynamic textures in terms of their appearance and dynamics is proposed. This factorization is then used to enable dynamic texture synthesis which, based on example dynamic texture inputs, will generate a novel dynamic texture instance. It shall also enable a novel form of style transfer where the target appearance and dynamics can be taken from different sources—termed *dynamics style transfer*. An overview of dynamic texture synthesis and dynamics style transfer is shown in Fig. 1.4.

The proposed model is constructed from two 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 [15, 22, 37], an input dynamic texture is summarized in terms of a set of spatiotemporal statistics of filter outputs from each stream. The appearance stream models the per-frame appearance of the input texture, while the dynamics stream models its temporal dynamics. The synthesis process consists of optimizing a randomly initialized 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 [6] 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 [19] models the human visual cortex in terms of two pathways, the ventral stream (involved with object recognition) and the dorsal stream (involved with motion processing). Two-stream networks have also been used for video understanding tasks in computer vision, with particular attention to action recognition [44, 13].

In this thesis, the two-stream analysis of dynamic textures is applied to texture synthesis. A range of dynamic textures are considered and it is demonstrated that the proposed approach generates novel, high quality samples that match both the frame-wise appearance and temporal evolution of an input example. Further, as stated previously, the factorization of appearance and dynamics enables a novel form of style transfer, where the dynamics of one texture are combined with the appearance of a different one, *cf*. [16]. This can even be done using a single image as an appearance target, which allows static images to be animated. Finally, the perceived realism of the generated textures is validated through an extensive user study.

1.3 Contributions

Matthew split contributions into a numbered list (5)

The contributions of this work span both theory and application. First, theoretical insight into the characterization of dynamic textures is provided by building a novel factored representation of both appearance and dynamics. Second, for the representation of dynamics, a novel ConvNet is constructed and trained for optical flow prediction. Third, through a qualitative evaluation, it is shown that the two-stream representation is effective in generating visually compelling, novel instances of a wide range of dynamic textures. Fourth, a novel form of style transfer is demonstrated, where the dynamics of a dynamic texture can be mixed with the spatial appearance of a different (static or dynamic) texture. This is enabled by the proposed factored representation. Finally, a quantitative evaluation on the limitations of the method is performed through the inclusion of a broad range of textures and an extensive user study. This analysis will show which sequences work better than others and why. This may point to future work to address limitations of the proposed model. In terms of specific applications, there are many in the creative-industry including, but not limited to, computer- generated imagery, digital painting, and image editing. More broadly, the ability to animate static imagery via dynamics style transfer can meaningfully contribute to the emerging artistic medium of computer-generated art.

Chapter 2

Related work Matthew unfinished (6)

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 [11, 30, 41, 53], and statistical parametric models. As the proposed approach is an instance of a parametric model, this thesis will focus on these parametric approaches.

The statistical characterization of visual textures was introduced in the seminal work of Julesz [27]. 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 [22, 37]. In particular, inspired by models of the early stages of visual processing, statistics of (handcrafted) multi-scale oriented filter responses were used to optimize an initial noise pattern to match the filter response statistics of an input texture. More recently, Gatys *et al.* [15] 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 modelled 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 [16], where the style of one image was combined with the image content of another to produce a new image. Ruder *et al*. [40] extended this model to video by using optical flow to enforce temporal consistency of the resulting imagery.

Variants of linear autoregressive models have been studied [47, 8] that jointly model appearance and dynamics of the spatiotemporal pattern. More recent work has considered ConvNets as a basis for modelling dynamic textures. Xie *et al*. [54] proposed a spatiotemporal generative model where each dynamic texture is modelled as a random field defined by multiscale, spatiotemporal ConvNet filter responses and dynamic textures are realized by sampling the model. Unlike my current work, which assumes pretrained fixed networks, this approach requires the ConvNet weights to be trained using the input texture prior to synthesis.

A recent preprint [14] described preliminary results extending the framework of Gatys *et al*. [15] to model and synthesize dynamic textures by computing a Gram matrix of filter activations over a small temporal window. In contrast, the proposed two-stream filtering architecture is more expressive as the dynamics stream is specifically tuned to spatiotemporal dynamics. Moreover, the factorization in terms of appearance and dynamics enables a novel form of style transfer, where the dynamics of one pattern are transferred to the appearance of another to generate an entirely new dynamic texture. This work is the first to demonstrate this form of style transfer.

The recovery of optical flow from temporal imagery has long been studied in computer vision. Traditionally, it has been addressed by handcrafted approaches *e.g*., [24, 33, 39]. Recently, ConvNet approaches [9, 38, 25, 55] have been demon-

strated as viable alternatives. Most closely related to my approach are energy models of visual motion [2, 21, 43, 35, 7, 29] 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. Here, an energy model motivates the representation of observed dynamics which will then be encoded as a ConvNet. Significantly, a completely analytically-defined oriented energy ConvNet model provides the current state-of-the-art for the related task of dynamic texture recognition [20].

Chapter 3

Technical approach

Matthew unfinished (7)

The proposed two-stream approach consists of an appearance stream, representing the static (texture) appearance of each frame, and a dynamics stream, representing temporal variations between frames. Each stream consists of a ConvNet whose activation statistics are used to characterize the dynamic texture. Synthesizing a dynamic texture is formulated as an optimization problem with the objective of matching these activation statistics. The dynamic texture synthesis approach is summarized in Fig. 3.1 and the individual pieces are described in turn in the following sections.

3.1 Texture model: Appearance stream

The appearance stream follows the spatial texture model introduced by Gatys *et al.* [15] which was reviewed in the previous chapter **Matthew** talk about Gatys style transfer in related work (8).

Matthew talk about texture synthesis in related work (9) To briefly review, the key idea is that feature auto-correlations (*i.e.*, normalized *Gram* matrices) in a ConvNet trained for object recognition (*e.g.*, VGG-19 [45]) capture texture appearance. The same publicly available normalized VGG-19 ConvNet [45] used by Gatys et al. [15] is used here.

3.1.1 Target texture appearance

To capture the appearance of an input dynamic texture, an initial forward pass through VGG-19 is performed with each frame of the image sequence to compute the feature activations (filter responses), $\mathbf{A}^{lt} \in \mathbb{R}^{N_l \times M_l}$, for various levels in the network, where N_l and M_l denote the number of feature activations and the number of spatial locations of layer l at time t, respectively. The auto-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:

$$G^l_{ij} = \frac{1}{TN_lM_l} \sum_{t=1}^{T} \sum_{k=1}^{M_l} A^{lt}_{ik} A^{lt}_{jk} \ , \tag{3.1}$$

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.

3.1.2 Synthesized texture appearance

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:

$$\hat{G}^{lt}_{ij} = \frac{1}{N_l M_l} \sum_{k=1}^{M_l} \hat{A}^{lt}_{ik} \hat{A}^{lt}_{jk} \ , \tag{3.2}$$

where $\hat{A}^{lt}_{ik}$ denotes the activation of feature i at location k in layer l on the synthesized frame t.

3.1.3 Appearance loss

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 synthesized texture computed at each frame:

$$\mathcal{L}_{\text{appearance}} = \frac{1}{L_{\text{app}} T_{\text{out}}} \sum_{t=1}^{T_{\text{out}}} \sum_{l} \| \mathbf{G}^{l} - \hat{\mathbf{G}}^{lt} \|_F^2 \ , \tag{3.3}$$

where L_{app} is the number of VGG-19 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 previous work [15], Gram matrices are computed on the following layers: *conv1_1*, *pool1*, *pool2*, *pool3*, and *pool4*.

3.2 Texture model: Dynamics stream

There are three primary goals in designing the dynamics stream ConvNet.

1. The activations of the ConvNet must represent the temporal variation of the

input pattern.

2. The activations should be largely invariant to the appearance (*i.e.*, spatial content) of the images (which should be characterized by the appearance stream described above).

3. The dynamics representation must be differentiable to enable synthesis via a ConvNet.

By analogy to the appearance stream, an obvious choice is a ConvNet architecture suited for computing optical flow **Matthew** talk about optical flow in related work (10) (*e.g.*, [9, 25]) which is naturally differentiable. However, with most such models it is unclear how invariant their layers are to appearance. Instead, a novel network architecture is proposed which is motivated by the spacetime-oriented energy model [7, 43]. **Matthew** talk about SOE for texture recognition in related work? (11)

3.2.1 Review: Marginalized spacetime oriented energies

In motion energy models, the velocity of image content (*i.e.*, motion) is interpreted as a 3D orientation in the x-y-t spatiotemporal domain [2, 12, 21, 43, 52]. **Matthew** include figure of motion in xyt (12) 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. **Matthew** include figure of motion in frequency space (13) 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, the constituent spacetime orientations for a spectrum of common visual patterns can serve as a basis for describing the temporal variation of an image sequence

[7]. This observation suggests that motion energy models may form an ideal basis for the dynamics stream. Specifically, the spacetime-oriented energy models proposed by Derpanis *et al.* [7] and Simoncelli *et al.* [43] are used to motivate the network architecture, which is briefly reviewed here; see Chapter 2 **Matthew** talk about Derpanis' MSOE in related work (14) for a more in-depth description.

Given an input video, a bank of oriented 3D filters, which are sensitive to a range of spatiotemporal orientations, are applied. These filter activations are then 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. At this point, each oriented energy measurement is confounded with spatial orientation. Consequently, in cases where the spatial image structure varies wildly about an otherwise coherent dynamic region, the responses of the bank of oriented filters will reflect this behaviour and thereby become dependent on spatial appearance; whereas, a description consisting purely of pattern dynamics is sought. To remove this difficulty, the spatial orientation component of each filter is discounted via "marginalization". Specifically, filter activations consistent with the same temporal orientation (not necessarily the same spatial orientation) are summed. **Matthew** clarify with Kosta (15) 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 that makes it 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 orientation responses which results in a representation that is robust to local appearance variations but highly selective to spacetime orientation.

3.2.2 ConvNet architecture

Using this model as the basis, the following fully convolutional network [42] **Matthew** talk about this in relate is proposed. The ConvNet input is a pair of temporally consecutive greyscale images, $\mathbf{I} \in \mathbb{R}^{H \times W \times C \times T}$ (height $\times$ width $\times$ channels $\times$ time), where $C = 1$ and $T = 2$. From here forth, the channel dimension (C) will be omitted for simplicity. Each input pair is first normalized to have zero-mean and unit variance (*i.e.*, contrast normalization or "instance normalization" [50]), as follows:

$$\mathbf{I}_N = \frac{\mathbf{I} - \mu}{\sigma + \epsilon} \ , \tag{3.4}$$

where μ is the average pixel value of the input pair, σ is the standard deviation of the input pair, and ϵ is a small value to prevent dividing by zero. This step provides a level of invariance **Matthew** Rick: Given that you're using normalized, bandpass filters, is this preprocessing necessary? (17) to overall brightness and contrast, *i.e.*, global additive and multiplicative signal variations, easing the training process of the ConvNet. **Matthew** maybe talk about how it relates to batchnorm? (18 The first layer consists of a 3D convolution over the normalized input pair with a bank of 32 3D filters of size $11 \times 11 \times 2$ (height $\times$ width $\times$ time), resulting in an output of spacetime oriented energy measurements, **Matthew** Rick: Really only 2 taps in time? (19)

$$E_F(\mathbf{x}) = F * \mathbf{I}_N(\mathbf{x}) \ , \tag{3.5}$$

where E_F denotes the response of filter F (of size $11 \times 11 \times 2$) after a convolution, $*$, centered about $\mathbf{x} \equiv (x, y, t)$. 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:

$$\bar{E}_F(\mathbf{x}) = \max_{i \in \Omega} \{E_F(i)^2\} \ , \tag{3.6}$$

where Ω is a 5×5 spatial neighbourhood centered about $\mathbf{x}$. A 2D convolution follows with 64 filters of size 1×1 that combines energy measurements that are consistent with the same orientation:

$$E_G(\mathbf{x}) = G * \bar{E}_F(\mathbf{x}) \ , \tag{3.7}$$

where $\tilde{E}_G$ denotes the response of filter G (of size 1×1) after a convolution. Finally, to remove local contrast dependence, an L_1 divisive normalization is applied to each spatial location: **Matthew** explain why this is not redundant due to contrast norm on input (20)

$$\bar{E}_G(\mathbf{x}) = \frac{E_G(\mathbf{x})}{\|E_G(\mathbf{x})\|_1 + \epsilon} \ , \tag{3.8}$$

where $\| \cdot \|_1$ is the L_1 norm computed over the filter responses of all filters.

To capture spacetime orientations beyond those capable with the limited receptive fields used in the initial layer, a five-level spatial Gaussian pyramid is computed. Each pyramid level is processed independently with the same spacetime-oriented energy model and then bilinearly upsampled to the original resolution and concatenated:

$$E(\mathbf{x}) = \left(\bar{E}_G(\mathbf{x}) \| \bar{E}_G(\mathbf{x}) \| \bar{E}_G(\mathbf{x})\right) \tag{3.9}$$

Training

Prior energy model instantiations (*e.g.*, [2, 7, 43]) used handcrafted filter weights. While a similar approach could be followed here, we 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 apply a 3×3 convolution (with 64 filters), ReLU activation, and a 1×1 convolution (with 2 filters) that yields a two channel output encoding the optical flow directly. The proposed architecture is illustrated in Fig. 3.2.

For training, 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 [39] to estimate optical flow for training, *cf*. [48]. For training data, we used videos from the UCF101 dataset [46] with geometric and photometric data augmentations similar to those used by FlowNet [9], and optimized the aEPE loss using Adam [28]. Inspection of the learned filters in the initial layer showed evidence of spacetime-oriented filters, consistent with the handcrafted filters used in previous work [7].

Similar to the appearance stream, filter response correlations 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 filter response 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 $\hat{D}^{lt}_{ik}$ 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:

$$\mathcal{L}_{\text{dynamics}} = \frac{1}{L_{\text{dyn}}(T_{\text{out}}-1)} \sum_{t=1}^{T_{\text{out}}-1} \sum_{l} \|\mathbf{G}^l - \hat{\mathbf{G}}^{lt}\|_F^2, \tag{3.10}$$

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 as shown in our evaluation. 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 a result, the flow estimates will tend to be an average of the underlying orientation measurements and consequently not descriptive. A comparison between the texture synthesis results using the concatenation layer and the predicted flow output is provided in Sec. 4.

3.3 Texture generation

The overall dynamic texture loss consists of the combination of the appearance loss, Eq. (3.3), and the dynamics loss, Eq. (3.10):

$$\mathcal{L}_{\text{dynamic texture}} = \alpha \mathcal{L}_{\text{appearance}} + \beta \mathcal{L}_{\text{dynamics}}, \tag{3.11}$$

where α and β are the weighting factors for the appearance and dynamics content, respectively. Dynamic textures are implicitly defined as the (local) minima of this loss. Textures are generated by optimizing Eq. (3.11) 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 [15], we use L-BFGS [32] optimization.

Naive application of the outlined approach will consume increasing amounts of memory as the temporal extent of the dynamic texture grows; this makes it impractical to generate longer sequences. Instead, long sequences can be incrementally generated by separating the sequence into subsequences and optimizing them sequentially. This is realized by initializing the first frame of a subsequence as the last frame from the previous subsequence and keeping it fixed throughout the optimization. The remaining frames of the subsequence are initialized randomly and optimized as above. This ensures temporal consistency across synthesized subsequences and can be viewed as a form of coordinate descent 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 frame regions are updated.

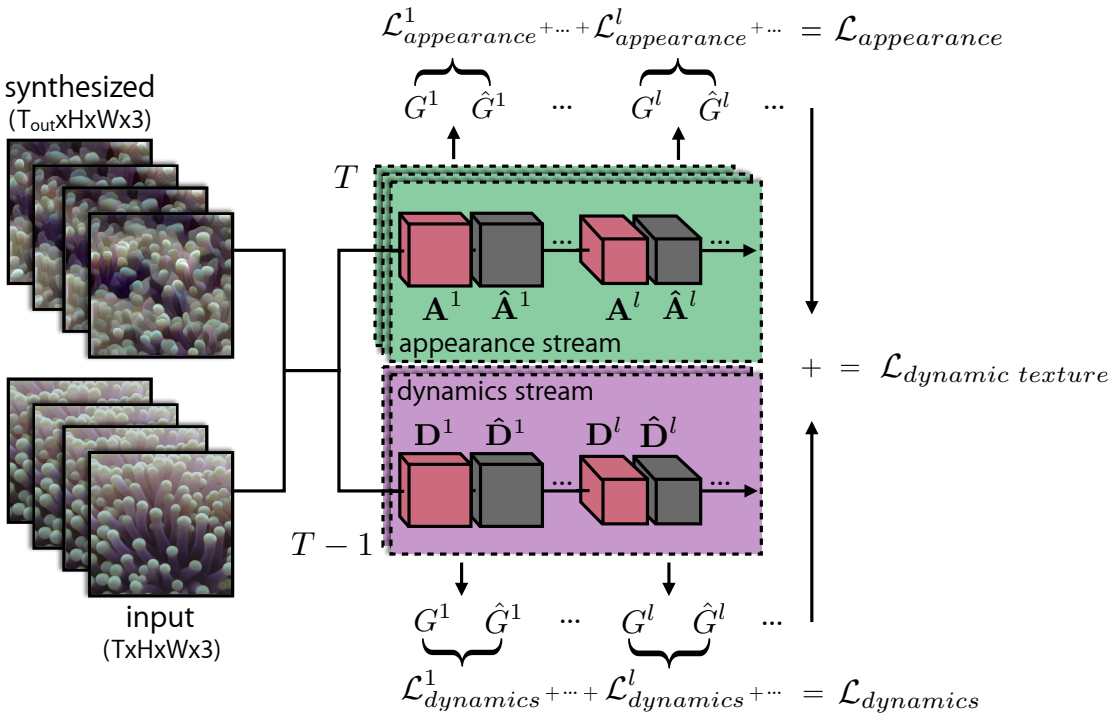

Figure 3.1: Two-stream dynamic texture generation. Two sets of Gram matrices represent a texture's appearance and dynamics. Matching these statistics will allow for the generation of novel textures as well as style transfer between textures. Here, G^l and $\hat{G}^l$ are the Gram matrices of activations A^l and $\hat{A}^l$ (or D^l and $\hat{D}^l$) corresponding to the target and synthesized sequence, respectively, computed at layer l of the appearance stream (or dynamics stream) and averaged over time T (or $T-1$). $\mathcal{L}^l_{\text{appearance}}$ is the appearance loss at layer l, computed as the squared Frobenius norm between G^l and $\hat{G}^l$ from the appearance stream. Similarly, $\mathcal{L}^l_{\text{dynamics}}$ is the dynamics loss at layer l for the dynamics stream. By summing each loss computed at various layers, we arrive at $\mathcal{L}_{\text{appearance}}$ and $\mathcal{L}_{\text{dynamics}}$, which, when summed, form the combined dynamic texture loss, $\mathcal{L}_{\text{dynamic texture}}$, that is to be minimized.

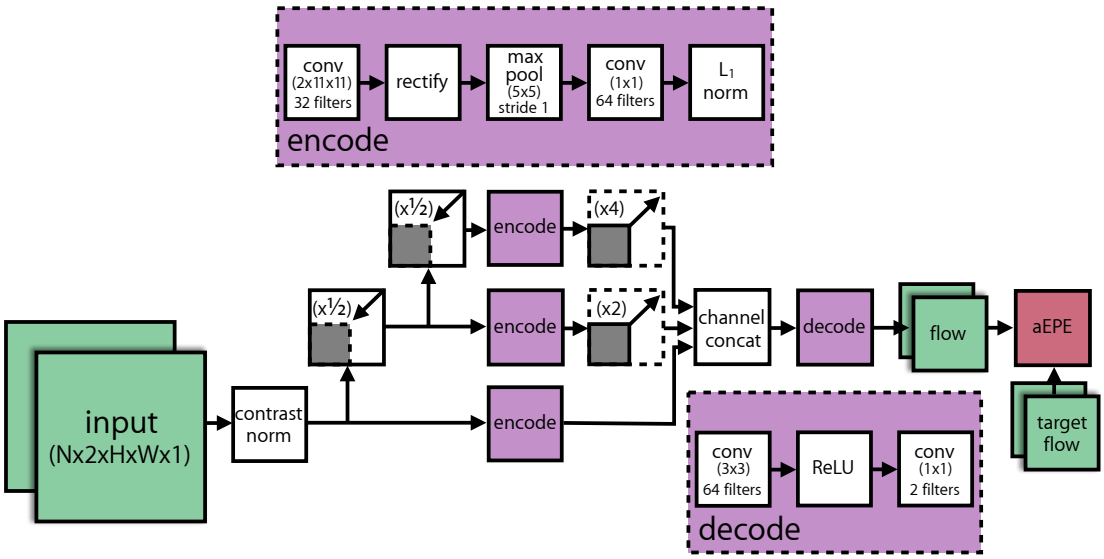

Figure 3.2: Dynamics stream ConvNet. The ConvNet is based on a spacetime-oriented energy model [7, 43] and will be trained for optical flow prediction. Three scales are shown for illustration; in practice five scales will be used.

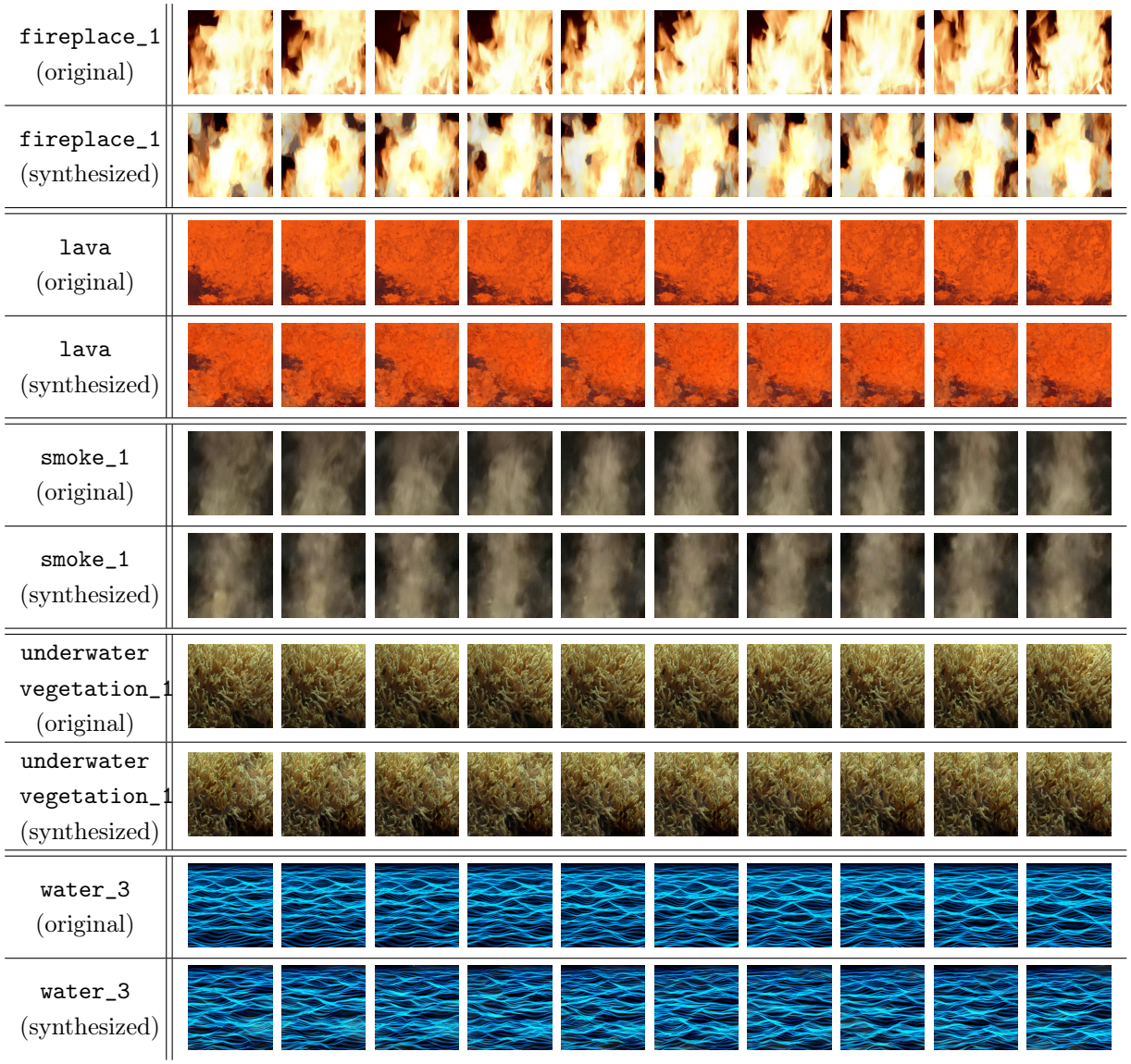

Figure 3.3: Dynamic texture synthesis success examples. Names correspond to files in the supplemental material.

Chapter 4

Evaluation

Matthew haven't touched yet (21)

The goal of (dynamic) texture synthesis is to generate samples that are indistinguishable from the real input target texture by a human observer. In this section, we present a variety of synthesis results including a user study to quantitatively evaluate the realism of our results. Given their temporal nature, our results are best viewed as videos. Our two-stream architecture was implemented using TensorFlow [1]. 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. For our full synthesis results and source code, please refer to the supplemental material on the project website: ryersonvisionlab.github.io/two-stream-projpage.

4.1 Dynamic texture synthesis

We applied our dynamic texture synthesis process to a wide range of textures which were selected from the DynTex [36] database and others we collected in the wild. Included in our supplemental material are synthesized results of nearly 60 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. 3.3 but we encourage readers to view the videos to fully appreciate the results. In addition, we performed a comparison with [14] and [54]. Generally, we found our results to be qualitatively comparable or better than these methods. See the supplemental for more details on the comparisons with these methods.

We also generated dynamic textures incrementally, as described in Sec. 3.3. The resulting textures were perceptually indistinguishable from those generated with the batch process. Another extension that we explored were textures with no discernible temporal seam between the last and first frames. Played as a loop, these textures appear to be temporally endless. This was achieved by assuming that the first frame follows the final frame and adding an additional loss for the dynamics stream evaluated on that pair of frames.

Example failure modes of our method are presented in Fig. 4.2. In general, we find that most failures result from inputs that violate the underlying assumption of a dynamic texture, *i.e.*, the appearance and/or dynamics are not spatiotemporally homogeneous. In the case of the `escalator` example, the long edge structures in the appearance are not spatially homogeneous, and the dynamics vary due to perspective effects that change the motion from downward to outward. The

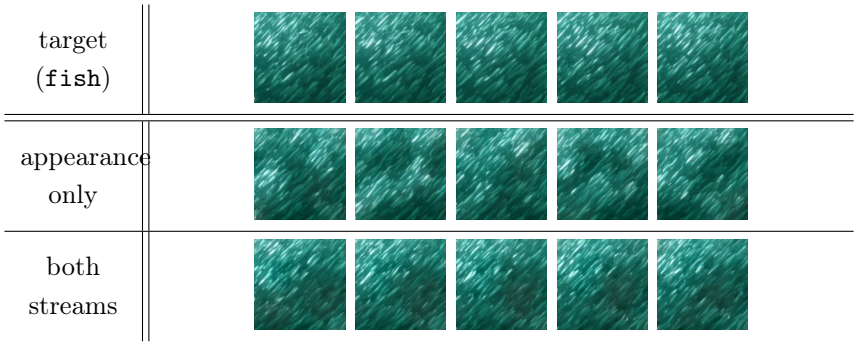

Figure 4.1: Dynamic texture synthesis versus texture synthesis. (top row) Target texture. (middle) Texture synthesis without dynamics constraints shows consistent per-frame appearance but no temporal coherence. (bottom) Including both streams induces consistent appearance and dynamics.

resulting synthesized texture captures an overall downward motion but lacks the perspective effects and is unable to consistently reproduce the long edge structures. This is consistent with previous observations on static texture synthesis [15] and suggests it is a limitation of the appearance stream.

Another example is the `flag` sequence where the rippling dynamics are relatively homogeneous across the pattern but the appearance varies spatially. 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`, which consists of a swirling pattern. Our model faithfully reproduces the appearance but is unable to capture the spatially varying dynamics. Interestingly, it still produces a result which is statistically indistinguishable from real in our user study discussed below.

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 consis-

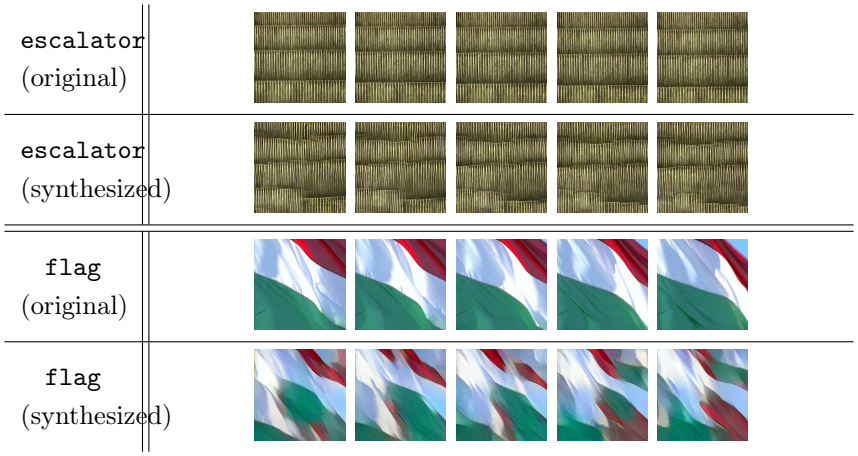

Figure 4.2: Dynamic texture synthesis failure examples. In these cases, the failures are attributed to either the appearance or the dynamics not being homogeneous.

tent with the input, we generated frames starting from randomly generated noise but only using the appearance statistics and corresponding loss, *i.e.*, Eq. 3.3. As expected, this produced frames that were valid textures but with no coherent dynamics present. Results for a sequence containing a school of fish are shown in Fig. 4.1; to examine the dynamics, see `fish` in the supplemental material.

Similarly, to validate that the dynamics stream did not inadvertently include appearance information, we generated videos using the dynamics loss only, *i.e.*, Eq. 3.10. The resulting frames had no visible appearance and had an extremely low dynamic range, *i.e.*, the standard deviation of pixel intensities was 10 for values in $[0, 255]$. This indicates a general invariance to appearance and suggests that our two-stream dynamic texture representation has factored appearance and dynamics, as desired.

4.2 User study

Quantitative evaluation for texture synthesis is a particularly challenging task as there is no single correct output when synthesizing new samples of a texture. Like in other image generation tasks (*e.g.*, rendering), human perception is ultimately the most important measure. Thus, we performed a user study to evaluate the perceived realism of our synthesized textures.

Similar to previous image synthesis work (*e.g.*, [5]), we conducted a perceptual experiment with human observers to quantitatively evaluate our synthesis results. We employed a forced-choice evaluation on Amazon Mechanical Turk (AMT) with 200 different users. Each user performed 59 pairwise comparisons between a synthesized dynamic texture and its target. Users were asked to choose which appeared more realistic after viewing the textures for an exposure time sampled randomly from discrete intervals between 0.3 and 4.8 seconds. Measures were taken to control the experimental conditions and minimize the possibility of low quality data. See the supplemental material for further experimental details of our user study.

For comparison, we constructed a baseline by using the flow decode layer in the dynamics loss of Eq. 3.10. This corresponds with attempting to mimic the optical flow statistics of the texture directly. Textures were synthesized with this model and the user study was repeated with an additional 200 users. To differentiate between the models, we label "Flow decode layer" and "Concat layer" in the figures to describe our baseline and final model, respectively.

The results of this study are summarized in Fig. 4.3 which shows user accuracy in differentiating real versus generated textures as a function of time for both

methods. Overall, users are able to correctly identify the real texture 66.1%±2.5% of the time for brief exposures of 0.3 seconds. This rises to 79.6% ± 1.1% with exposures of 1.2 seconds and higher. Note that "perfect" synthesis results would have an accuracy of 50%, indicating that users were unable to differentiate between the real and generated textures and higher accuracy indicating less convincing textures.

The results clearly show that the use of the concatenation layer activations is far more effective than the flow decode layer. This is not surprising as optical flow alone is known to be unreliable on many textures, particularly those with transparency or chaotic motion (*e.g.*, water, smoke, flames, etc.). Also evident in these results is the time-dependant nature of perception for textures from both models. Users' ability to identify the generated texture improved as exposure times increased to 1.2 seconds and remained relatively flat for longer exposures.

To better understand the performance of our approach, we grouped and analyzed the results in terms of appearance and dynamics characteristics. For appearance we used the taxonomy presented in [31] and grouped textures as either regular/near-regular (*e.g.*, periodic tiling and brick wall), irregular (*e.g.*, a field of flowers), or stochastic/near-stochastic (*e.g.*, tv static or water). For dynamics we grouped textures as either spatially-consistent (*e.g.*, closeup of rippling sea water) or spatially-inconsistent (*e.g.*, rippling sea water juxtaposed with translating clouds in the sky). Results based on these groupings can be seen in Fig. 4.4.

A full breakdown of the user study results by texture and grouping can be found in the supplemental material. Here we discuss some of the overall trends. Based on appearance it is clear that textures with large-scale spatial consistencies (regular, near-regular, and irregular textures) tend to perform poorly. Examples

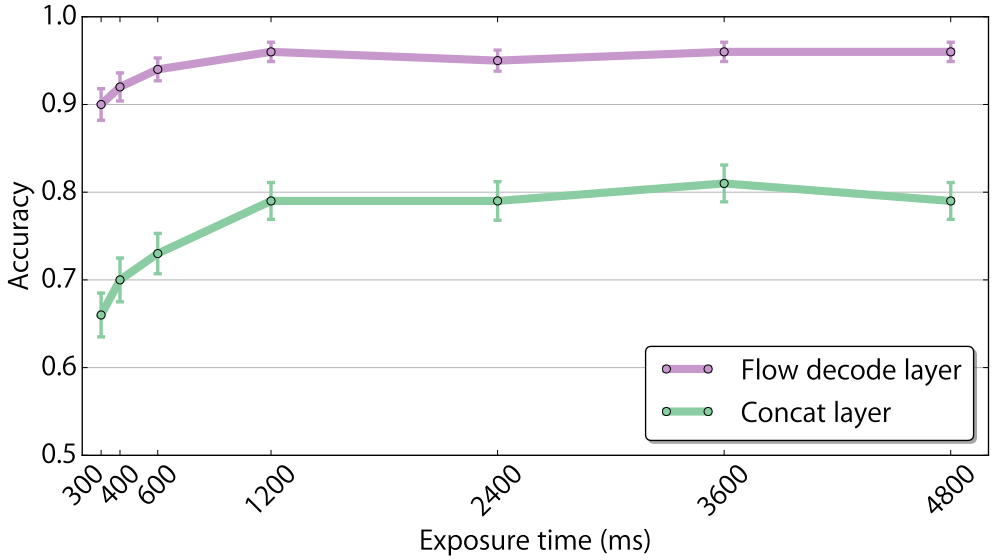

Figure 4.3: Time-limited pairwise comparisons across all textures with 95% statistical confidence intervals.

being `flag` and `fountain_2` with user accuracies of $98.9\% \pm 1.6\%$ and $90.8\% \pm 4.3\%$ averaged across all exposures, respectively. This is not unexpected and is a fundamental limitation of the local nature of the Gram matrix representation used in the appearance stream which was observed in static texture synthesis [15]. In contrast, stochastic and near-stochastic textures performed significantly better as their smaller-scale local variations are well captured by the appearance stream, for instance `water_1` and `lava` which had average accuracies of $53.8\% \pm 7.4\%$ and $55.6\% \pm 7.4\%$, respectively, making them both statistically indistinguishable from real.

In terms of dynamics, we find that textures with spatially-consistent dynamics (*e.g.*, `tv_static`, `water_*`, and `calm_water_*`) perform significantly better than those with spatially-inconsistent dynamics (*e.g.*, `candle_flame`, `fountain_2`, and `snake_*`), where the dynamics drastically differ across spatial locations. For ex-

ample, `tv_static` and `calm_water_6` have average accuracies of $48.6\% \pm 7.4\%$ and $63.2\% \pm 7.2\%$, respectively, while `candle_flame` and `snake_5` have average accuracies of $92.4\% \pm 4\%$ and $92.1\% \pm 4\%$, respectively. Overall, our model is capable of reproducing a full spectrum of spatially-consistent dynamics. However, as the appearance shifts from containing small-scale spatial consistencies to containing large-scale consistencies, performance degrades. This was evident in the user study where the best-performing textures typically consisted of a stochastic or near-stochastic appearance with spatially-consistent dynamics. In contrast the worst-performing textures consisted of regular, near-regular, or irregular appearance with spatially-inconsistent dynamics.

4.3 Dynamics 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, 17] 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. 4.5 (top), using two real videos. Additional examples are available in the supplemental material. 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 produce implausible results.

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. 4.5 (bottom), where the appearance is taken from a painting and the dynamics from a real world video.

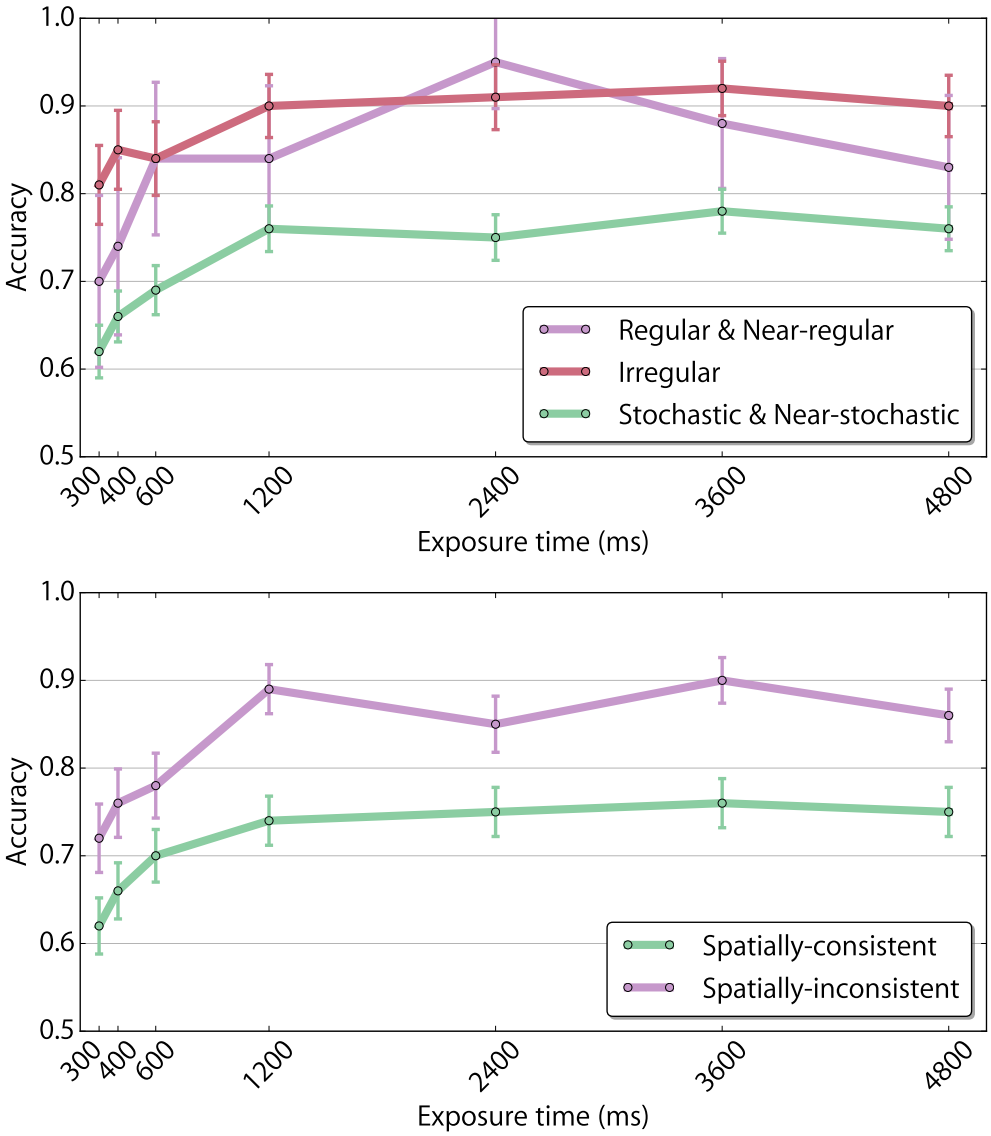

Figure 4.4: Time-limited pairwise comparisons across all textures, grouped by appearance (top) and dynamics (bottom). Shown with 95% statistical confidence intervals.

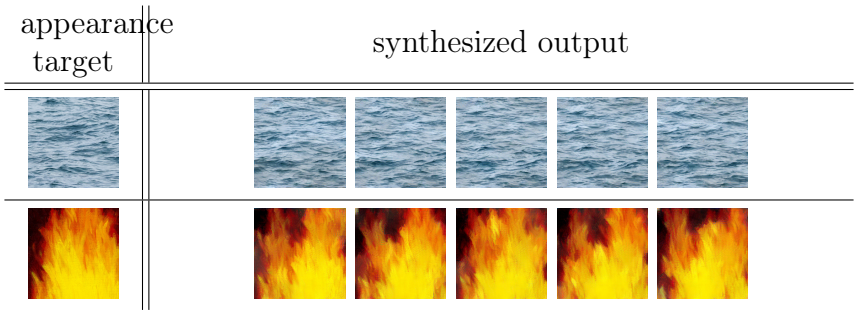

Figure 4.5: Dynamics style transfer. (top row) Appearance of still water was used with the dynamics of a different water dynamic texture (`water_4`). (bottom row) The appearance of a painting of fire was used with the dynamics of a real fire (`fireplace_1`). Animated results and additional examples are available in the supplemental material.

Chapter 5

Conclusion

Matthew haven't touched yet (22)

In this paper, we 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. This was validated both qualitatively and quantitatively through a large user study. 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 which we leave as directions for future work. First, much like has been reported in recent image style transfer work [16], we have found that high frequency noise and chromatic aberrations are a problem in generation. Another issue that arises is

the model fails to capture textures with spatially-variant appearance, (*e.g.*, `flag` in Fig. 4.2) and spatially-inconsistent dynamics (*e.g.*, `escalator` in Fig. 4.2). By collapsing the local statistics into a Gram matrix, the spatial and temporal organization is lost. Simple post-processing methods may alleviate some of these issues but we believe that they also point to a need for a better representation. Beyond addressing these limitations, a natural next step would be to extend the idea of a factorized representation into feed-forward generative networks that have found success in static image synthesis, *e.g.*, [26, 49].

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 24

[2] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A (JOSA-A)*, 2(2):284–299, 1985. 10, 14, 17

[3] Ziv Bar-Joseph, Ran El-Yaniv, Dani Lischinski, and Michael Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics (T-VCG)*, 7(2):120–135, 2001. 1

[4] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv:1603.01768*, 2016. 31

[5] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 28

[6] James E. Cutting. Blowing in the wind: Perceiving structure in trees and bushes. *Cognition*, 12(1):25 – 44, 1982. 6

[7] Konstantinos G. Derpanis and Richard P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(6):1193–1205, 2012. 2, 10, 14, 15, 17, 18, 22

[8] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision (IJCV)*, 51(2):91–109, 2003. 1, 9

[9] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 9, 14, 18

[10] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346, 2001. 3, 4

[11] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1033–1038, 1999. 8

[12] 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. 14

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[14] Christina M. Funke, Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Synthesising dynamic textures using convolutional neural networks. *arXiv:1702.07006*, 2017. 9, 25

[15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Conference on Neural Information Processing Systems (NIPS)*, pages 262–270, 2015. 3, 4, 6, 8, 9, 11, 12, 13, 20, 26, 30

[16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 2, 4, 6, 9, 35

[17] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 31

[18] James J. Gibson. The perception of the visual world. 1950. 2

[19] Melvyn A. Goodale and A. David. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992. 6

[20] Isma Hadji and Richard P. Wildes. A spatiotemporal oriented energy network for dynamic texture recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 10

[21] David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision (IJCV)*, 1(4):279–302, 1988. 10, 14

[22] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238, 1995. 2, 3, 6, 8

[23] David J. Heeger and Alex P. Pentland. Seeing structure through chaos. In *IEEE Motion Workshop: Representation and Analysis*, pages 131–136, 1986. 1

[24] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence (A.I.)*, 17:185–203, 1981. 9

[25] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 9, 14

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711, 2016. 36

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

[28] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014. 18

[29] Kishore Konda, Roland Memisevic, and Vincent Michalski. Learning to encode motion using spatio-temporal synchrony international conference on learning representation. In *International Conference on Learning Representations (ICLR)*, 2014. 10

[30] 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. 8

[31] Wen-Chieh Lin, James Hays, Chenyu Wu, Yanxi Liu, and Vivek Kwatra. Quantitative evaluation of near regular texture synthesis algorithms. In *CVPR*, volume 1, pages 427–434, 2006. 29

[32] Dong C. Liu and Jorge Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989. 20

[33] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981. 9

[34] R. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. *International Conference on Visualization, Graphics and Image Processing (CVGIP)*, 56(1), 1992. 1

[35] 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. 10

[36] Renaud Péteri, Sándor Fazekas, and Mark J. Huiskes. DynTex: A Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters (PRL)*, 31(12), 2010. 25

[37] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision (IJCV)*, 40(1):49–70, 2000. 6, 8

[38] A. Ranjan and M. J. Black. Optical Flow Estimation using a Spatial Pyramid Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 9

[39] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1172, 2015. 9, 18

[40] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition (GCPR)*, pages 26–36, 2016. 9

[41] Arno Schödl, Richard Szeliski, David Salesin, and Irfan A. Essa. Video textures. In *SIGGRAPH*, pages 489–498, 2000. 8

[42] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(4):640–651, 2017. 16

[43] Eero P. Simoncelli and David J. Heeger. A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743 – 761, 1998. 10, 14, 15, 17, 22

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Conference on Neural Information Processing Systems (NIPS)*, pages 568–576, 2014. 6

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 12

[46] 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. 18

[47] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *IEEE International Conference on Image Processing (ICIP)*, pages 823–826, 1996. 9

[48] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Deep end2end voxel2voxel prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 402–409, 2016. 18

[49] 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. 36

[50] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4105–4113, 2017. 16

[51] Yizhou Wang and Song Chun Zhu. Modeling textured motion: Particle, wave and sketch. In *IEEE International Conference on Computer Vision (ICCV)*, pages 213–220, 2003. 1

[52] Andrew B. Watson and Albert J. Ahumada. A look at motion in the frequency domain. In *Motion workshop: Perception and representation*, pages 1–10, 1983. 14

[53] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488, 2000. 8

[54] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 9, 25

[55] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In *European Conference on Computer Vision (ECCV) Workshops*, 2016. 9

To do...

- ☐ 1 (p. iii): **Matthew** unfinished
- ☐ 2 (p. iii): **Matthew** unfinished
- ☐ 3 (p. iv): **Matthew** haven't touched yet
- ☐ 4 (p. iv): **Matthew** haven't touched yet
- ☐ 5 (p. 7): **Matthew** split contributions into a numbered list
- ☐ 6 (p. 8): **Matthew** unfinished
- ☐ 7 (p. 11): **Matthew** unfinished
- ☐ 8 (p. 11): **Matthew** talk about Gatys style transfer in related work
- ☐ 9 (p. 12): **Matthew** talk about texture synthesis in related work
- ☐ 10 (p. 14): **Matthew** talk about optical flow in related work
- ☐ 11 (p. 14): **Matthew** talk about SOE for texture recognition in related work?
- ☐ 12 (p. 14): **Matthew** include figure of motion in xyt
- ☐ 13 (p. 14): **Matthew** include figure of motion in frequency space

- ☐ 14 (p. 15): **Matthew** talk about Derpanis' MSOE in related work
- ☐ 15 (p. 15): **Matthew** clarify with Kosta
- ☐ 16 (p. 16): **Matthew** talk about this in related work?
- ☐ 17 (p. 16): **Matthew** Rick: Given that you're using normalized, bandpass filters, is this preprocessing necessary?
- ☐ 18 (p. 16): **Matthew** maybe talk about how it relates to batch-norm?
- ☐ 19 (p. 16): **Matthew** Rick: Really only 2 taps in time?
- ☐ 20 (p. 17): **Matthew** explain why this is not redundant due to contrast norm on input
- ☐ 21 (p. 24): **Matthew** haven't touched yet
- ☐ 22 (p. 35): **Matthew** haven't touched yet