Abstract

We introduce a new video synthesis task: synthesizing time lapse videos depicting how a given painting might have been created. Artists paint using unique combinations of brushes, strokes, and colors. There are often many possible ways to create a given painting. Our goal is to learn to capture this rich range of possibilities.

Creating distributions of long-term videos is a challenge for learning-based video synthesis methods. We present a probabilistic model that, given a single image of a completed painting, recurrently synthesizes steps of the painting process. We implement this model as a convolutional neural network, and introduce a novel training scheme to enable learning from a limited dataset of painting time lapses. We demonstrate that this model can be used to sample many time steps, enabling long-term stochastic video synthesis.

We evaluate our method on digital and watercolor paintings collected from video websites, and show that human raters find our synthetic videos to be similar to time lapse videos produced by real artists. Our code is available at https://xamyzhao.github.io/timecraft.

1. Introduction

Skilled artists can often look at a piece of artwork and determine how to recreate it. In this work, we explore whether we can use machine learning and computer vision to mimic this ability. We define a new video synthesis problem: given a painting, can we synthesize a time lapse video depicting how an artist might have painted it?

Artistic time lapses present many challenges for video synthesis methods. There is a great deal of variation in how people create art. Suppose two artists are asked to paint the same landscape. One artist might start with the sky, while the other might start with the mountains in the distance. One might finish each object before moving onto the next, while the other might work a little at a time on each object. During the painting process, there are often few visual cues indicating where the artist will apply the next stroke. The painting process is also long, often spanning hundreds of paint strokes and dozens of minutes.

In this work, we present a solution to the painting time lapse synthesis problem. We begin by defining the problem and describing its unique challenges. We then derive a principled, learning-based model to capture a distribution of steps that a human might use to create a given painting. We introduce a training scheme that encourages the method to produce realistic changes over many time steps. We demonstrate that our model can learn to solve this task, even when trained using a small, noisy dataset of painting time lapses collected from the web. We show that human evaluators almost always prefer our method to an existing video synthesis baseline, and often find our results indistinguishable from time lapses produced by real artists.

This work presents several technical contributions:

1. We use a probabilistic model to capture stochastic decisions made by artists, thereby capturing a distribution of plausible ways to create a painting.
2. Unlike work in future frame prediction or frame interpolation, we synthesize long-term videos spanning dozens of minutes.
of time steps and many real-time minutes.

3. We demonstrate a model that successfully learns from painting time lapses “from the wild.” This data is small and noisy, having been collected from uncontrolled environments with variable lighting, spatial resolution and video capture rates.

2. Related work

To the best of our knowledge, this is the first work that models and synthesizes distributions of videos of the past, given a single final frame. The most similar work to ours is a recent method called visual deprojection [5]. Given a single input image depicting a temporal aggregation of frames, their model captures a distribution of videos that could have produced that image. We compare our method to theirs in our experiments. Here, we review additional related research in three main areas: video prediction, video interpolation, and art synthesis.

2.1. Video prediction

Video prediction, or future frame prediction, is the problem of predicting the next frame or few frames of a video given a sequence of past frames. Early work in this area focused on predicting motion trajectories [8, 16, 34, 51, 55] or synthesizing motions in small frames [40, 41, 50]. Recent methods train convolutional neural networks on large video datasets to synthesize videos of natural scenes and human actions [35, 38, 46, 52, 53]. A recent work on time lapse synthesis focuses on outdoor scenes [43], simulating illumination changes over time while keeping the content of the scene constant. In contrast, creating painting time lapses requires adding content while keeping illumination constant. Another recent time lapse method outputs only a few frames depicting specific physical processes: melting, rotting, or flowers blooming [70].

Our problem differs from video prediction in several key ways. First, most video prediction methods focus on short time scales, synthesizing frames on the order of seconds into the future, and encompassing relatively small changes. In contrast, painting time lapses span minutes or even hours, and depict dynamic content changes over time. Second, most video predictors output a single most likely sequence, making them ill-suited for capturing a variety of different plausible painting trajectories. One study [63] uses a conditional variational autoencoder to model a distribution of plausible future frames of moving humans. We build upon these ideas to model painting changes across multiple time steps. Finally, video prediction methods focus on natural videos, which depict of motions of people and objects [52, 53, 63] or physical processes [70]. The input frames often contain visual cues about how the motion, action or physical process will progress, limiting the space of possibilities that must be captured. In contrast, snapshots of paintings provide few visual cues, leading to many plausible future trajectories.

2.2. Video frame interpolation

Our problem can be thought of as a long-term frame interpolation task between a blank canvas and a completed work of art, with many possible painting trajectories between them. In video frame interpolation, the goal is to temporally interpolate between two frames in time. Classical approaches focus on natural videos, and estimate dense flow fields [4, 58, 65] or phase [39] to guide interpolation. More recent methods use convolutional neural networks to directly synthesize the interpolated frame [45], or combine flow fields with estimates of scene information [28, 44]. Most frame interpolation methods predict a single or a few intermediate frames, and are not easily extended to predicting long sequences, or predicting distributions of sequences.

2.3. Art synthesis

The graphics community has long been interested in simulating physically realistic paint strokes in digital media. Many existing methods focus on physics-based models of fluids or brush bristles [6, 7, 9, 12, 57, 62]. More recent learning-based methods leverage datasets of real paint strokes [31, 36, 68], often posing the artistic stroke synthesis problem as a texture transfer or style transfer problem [3, 37]. Several works focus on simulating watercolor-specific effects such as edge darkening [42, 56]. We focus on capturing large-scale, long-term painting processes, rather than fine-scale details of individual paint strokes.

In style transfer, images are transformed to simulate a specific style, such as a painting-like style [20, 21] or a cartoon-like style [67]. More recently, neural networks have been used for generalized artistic style transfer [18, 71]. We leverage insights from these methods to synthesize a realistic progression of paintings.

Several recent papers apply reinforcement learning or similar techniques to the process of painting. These approaches involve designing parameterized brush strokes, and then training an agent to apply strokes to produce a given painting [17, 22, 26, 27, 59, 60, 69]. Some works focus on specific artistic tasks such as hatching or other repetitive strokes [29, 61]. These approaches require careful hand-engineering, and are not optimized to produce varied or realistic painting progressions. In contrast, we learn a broad set of effects from real painting time lapse data.

3. Problem overview

Given a completed painting, our goal is to synthesize different ways that an artist might have created it. We work with recordings of digital and watercolor painting time lapses collected from video websites. Compared to
natural videos of scenes and human actions, videos of paintings present unique challenges.

**High Variability**

_**Painting trajectories:**_ Even for the same scene, different artists will likely paint objects in different temporal orders (Figure 2).

_**Painting rates:**_ Artists work at different speeds, and apply paint in different amounts.

_**Scales and shapes:**_ Over the course of a painting, artists use strokes that vary in size and shape. Artists often use broad strokes early on, and add fine details later.

_**Data availability:**_ Due to the limited number of available videos in the wild, it is challenging to gather a dataset that captures the aforementioned types of variability.

**Medium-specific challenges**

_**Non-paint effects:**_ In digital art applications (e.g., [23]), there are many tools that apply local blurring, smudging, or specialized paint brush shapes. Artists can also apply global effects simulating varied lighting or tones.

_**Erasing effects:**_ In digital art applications, artists can erase or undo past actions, as shown in Figure 3.

_**Physical effects in watercolor paintings:**_ Watercolor painting videos exhibit distinctive effects resulting from the physical interaction of paint, water, and paper. These effects include specular lighting on wet paint, pigments fading as they dry, and water spreading from the point of contact with the brush (Figure 4).

In this work, we design a learning-based model to handle the challenges of high variability and painting medium-specific effects.

---

**Figure 2: Several real painting progressions of similar-looking scenes.** Each artist fills in the house, sky and field in a different order.

**Figure 3: Example digital painting sequences.** These sequences show a variety of ways to add paint, including fine strokes and filling (row 1), and broad strokes (row 3). We use red boxes to outline challenges, including erasing (row 2) and drastic changes in color and composition (row 3).

**Figure 4: Example watercolor painting sequences.** The outlined areas highlight some watercolor-specific challenges, including changes in lighting (row 1), diffusion and fading effects as paint dries (row 2), and specular effects on wet paint (row 3).

---

**4. Method**

We begin by formalizing the time lapse video synthesis problem. Given a painting $x_T$, our task is to synthesize the past frames $x_1, \ldots, x_{T-1}$. Suppose we have a training set of real time lapse videos $\{x^{(i)} = x_1^{(i)}, \ldots, x_T^{(i)}\}$. We first define a principled probabilistic model, and then learn its parameters using these videos. At test time, given a completed painting, we sample from the model to create new videos of realistic-looking painting processes.

**4.1. Model**

We propose a probabilistic, temporally recurrent model for changes made during the painting process. At each time instance $t$, the model predicts a pixel-wise intensity change $\delta_t$ that should be added to the previous frame to produce the current frame; that is, $x_t = x_{t-1} + \delta_t$. This change could represent one or multiple physical or digital paint strokes, or other effects such as erasing or fading.

We model $\delta_t$ as being generated from a random latent variable $z_t$, the completed piece $x_T$, and the image content at the previous time step $x_{t-1}$; the likelihood is
**Figure 5:** The proposed probabilistic model. Circles represent random variables; the shaded circle denotes a variable that is observed at inference time. The rounded rectangle represents model parameters.

\[ p(\delta_t | z_t, x_{t-1}; x_T) \] (Figure 5). Using a random variable \( z_t \) helps to capture the stochastic nature of painting. Using both \( x_T \) and \( x_{t-1} \) enables the model to capture time-varying effects such as the progression of coarse to fine brush sizes, while the Markovian assumption facilitates learning from a small number of video examples.

It is common to define such image likelihoods as a per-pixel normal distribution, which results in an L2 image similarity loss term in maximum likelihood formulations [33]. In synthesis tasks, using L2 loss often produces blurry results [24]. We instead design our image similarity loss as the L1 distance in pixel space and the L2 distance in a perceptual feature space. Perceptual losses are commonly used in image synthesis and style transfer [14, 24, 30, 45, 66]. We use the L2 distance between tasks to produce sharper and more visually pleasing results [14, 24, 30, 45, 66]. We use the L2 distance between normalized VGG features [49] as described in [66]. We let \( \theta \) represent model parameters.

We assume the latent variable \( z_t \) is generated from a distribution that is close to the standard normal. We use the shorthand \( \delta_t = g_\theta(z_t, x_{t-1}, x_T) \), \( \tilde{x}_t = x_t - \delta_t \).

\[ p(z_t | \delta_t, x_{t-1}; x_T) = q_\phi(z_t | \delta_t, x_{t-1}; x_T) \] [32, 63, 64]. We let this approximate distribution take the form of a multivariate normal:

\[ q_\phi(z_t | \delta_t, x_{t-1}, x_T) = \mathcal{N}(z_t; \mu_\phi(\delta_t, x_{t-1}, x_T), \Sigma_\phi(\delta_t, x_{t-1}, x_T)), \] (4)

where \( \mu_\phi(\cdot), \Sigma_\phi(\cdot) \) are functions parameterized by \( \phi \), and \( \Sigma_\phi(\cdot) \) is diagonal.

### 4.1.1 Neural network framework

We implement our model using a conditional variational autoencoder framework. At training time, the network is encouraged to reconstruct the current frame \( x_t \), while sampling the latent \( z_t \) from a distribution that is close to the standard normal. At test time, the auto-encoding branch is removed, and \( z_t \) is sampled from the standard normal. We use the shorthand \( \delta_t = g_\theta(z_t, x_{t-1}, x_T) \), \( \tilde{x}_t = x_t - \delta_t \).

We aim to find model parameters \( \theta \) that best explain all videos in our dataset:

\[
\arg\max_{\theta} \prod_{t} p(\delta_t^{(t)} | z_t^{(t)}, x_{t-1}^{(t)}, x_T^{(t)})
= \arg\max_{\theta} \prod_{t} \int_{z_t} p(\delta_t^{(t)} | z_t^{(t)}, x_{t-1}^{(t)}, x_T^{(t)}) dz_t.
\] (3)

This integral is intractable, and the posterior \( p(z_t | \delta_t, x_{t-1}; x_T) \) is also intractable, preventing the use of the EM algorithm. We instead use variational inference and introduce an approximate posterior distribution.
\[
\log p_\theta(\delta_t, x_{t-1}, x_T) \\
\geq \mathbb{E}_{z_t \sim q_\phi(z_t|x_{t-1}, \delta_t;x_{t};x_{t+1})} \left[ \log p_\theta(\delta_t|z_t, x_{t-1};x_{t};x_{t+1}) \right] \\
- KL[q_\phi(z_t|\delta_t, x_{t-1};x_{t};x_{t+1}) || p(z_t)], \tag{5}
\]
where \( KL[\cdot||\cdot] \) denotes the Kullback-Liebler divergence.

Combining Equations (1), (2), (4), and (5), we minimize:

\[
L_{KL} + \frac{1}{\sigma_1^2} L_{L1}(\delta_t, \hat{\delta}_t) + \frac{1}{2\sigma_2^2} L_{L2}(VGG(x_{t-1} + \delta_t), VGG(x_{t-1} + \hat{\delta}_t)), \tag{6}
\]
where \( L_{KL} = \frac{1}{2}( -\log \Sigma_\phi + \Sigma_\phi + \mu_\phi^2) \), and the image similarity terms \( L_{L1}, L_{L2} \) represent L1 and L2 distance respectively.

We optimize Equation (6) on single time steps, which we obtain by sampling all pairs of consecutive frames from the dataset. We also train the model to produce the first frame \( x_1 \) from videos that begin with a blank canvas, given a white input frame \( x_{blank} \) and \( x_T \). These \textit{starter sequences} teach the model how to start a painting at inference time.

\subsection{4.2.2 Sequence optimization}

To synthesize an entire video, we run our model recurrently for multiple time steps, building upon its own predicted frames. It is common when making sequential predictions to observe compounding errors or artifacts over time \cite{bengio2015estimating}. We use a novel sequential training scheme to encourage outputs of the model to be accurate and realistic over multiple time steps. We alternate between two training modes.

\textbf{Sequential CVAE training} encourages \textit{sequences} of frames to be well-captured by the learned distribution, by reducing the compounding of errors. We train the model sequentially for a few time steps, predicting each intermediate frame \( \hat{x}_t \) using the model’s prediction from the previous time step: \( \hat{x}_t = \hat{x}_{t-1} + g_\theta(z_t, \hat{x}_{t-1}, x_T) \) for \( z_t \sim q_\phi(z_t|x_{t-1}, \hat{x}_{t-1}, x_T) \). We compare each predicted frame to the corresponding real frame using the image similarity losses in Eq. (6). We illustrate this in Figure 7.

\textbf{Sequential sampling training} encourages random samples from our learned distribution to look like \textit{realistic} partially-completed paintings. During inference (described below), we rely on sampling from the prior \( p(z_t) \) at each time step to synthesize new videos. A limitation of the variational strategy is the limited coverage of the latent space \( z_t \) during training \cite{kingma2013auto}, sometimes leading to unrealistic predictions \( \hat{x}_t = \hat{x}_{t-1} + g_\theta(z_t, \hat{x}_{t-1}, x_T) \) for \( z_t \sim p(z_t) \). To compensate for this, we introduce supervision on such samples by amending the image similarity term in Equation (5) with a conditional critic loss term \cite{arjovsky2017wasserstein}:

\[
L_{\text{critic}} = \mathbb{E}_{z_t \sim p(z_t)} \left[ \psi(\hat{x}_t, \hat{x}_{t-1}, x_T) \right] - \mathbb{E}_{x_t} \left[ \psi(x_t, x_{t-1}, x_T) \right], \tag{7}
\]

where \( \psi(\cdot) \) is a critic function with parameters \( \psi \).

The critic encourages the distribution of sampled painting changes \( \delta_t = g_\theta(z_t, \hat{x}_{t-1}, x_T) \) to match the distribution of training painting changes \( \delta_t \). We use a critic architecture based on \cite{gulrajani2017improved} and optimize it using WGAN-GP \cite{arjovsky2017wasserstein}.

In addition to the critic loss, we apply the image similarity losses discussed above after \( \tau \) time steps, to encourage the model to eventually produce the completed painting. This training scheme is summarized in Figure 8.

\subsection{4.3. Inference: video synthesis}

Given a completed painting \( x_T \) and learned model parameters \( \theta, \phi \), we synthesize videos by sampling from the model at each time step. Specifically, we synthesize each frame \( \hat{x}_t = \hat{x}_{t-1} + g_\theta(z_t, \hat{x}_{t-1}, x_T) \) using the synthesized previous frame \( \hat{x}_{t-1} \) and a randomly sampled \( z_t \sim p(z_t) \). We start each video using \( \hat{x}_0 = x_{\text{blank}} \), a blank frame.

\subsection{4.4. Implementation}

We implement our model using Keras \cite{chollet2015keras} and TensorFlow \cite{abadi2016tensorflow}. We experimentally selected the hyperparameters controlling the reconstruction loss weights to be \( \sigma_1 = \sigma_2 = 0.1 \), using the validation set.

\section{5. Experiments}

\subsection{5.1. Datasets}

We collected time lapse recordings of paintings from YouTube and Vimeo. We selected digital and watercolor
paintings (which are common painting methods on these websites), and focused on landscapes or still lifes (which are common subjects for both mediums). We downloaded each video at $360 \times 640$ resolution and cropped it temporally and spatially to include only the painting process (excluding other content such as introductions or sketching). We split each dataset in a 70:15:15 ratio into training, validation, and held-out test video sets.

**Digital paintings**: We collected 117 digital painting time lapses. The average duration is 4 minutes, with many videos having already been sped up by artists using the Procreate application [23]. We selected videos with minimal zooming and panning. We manually removed segments that contained movements such as translations, flipping and zooming. Figure 3 shows example video sequences.

**Watercolor paintings**: We collected 116 watercolor time lapses, with an average duration of 20 minutes. We only kept videos that contained minimal movement of the paper, and manually corrected any small translations of the painting. We show examples in Figure 4.

A challenge with videos of physical paintings is the presence of the hand, paintbrush and shadows in many frames. We trained a simple convolutional neural network to identify and remove frames that contained these artifacts.

### 5.1.1 Sequence extraction

We synthesize time lapses at a lower temporal resolution than real-time for computational feasibility. We extract training sequences from raw videos at a period of $\gamma > 0$ frames (i.e., skipping $\gamma$ frames in each synthesized time step), with a maximum variance of $\epsilon$ frames. Allowing some variance in the sampling rate is useful for (1) improving robustness to varied painting rates, and (2) extracting sequences from watercolor painting videos where many frames containing hands or paintbrushes have been removed. We select $\gamma$ and $\epsilon$ independently for each dataset. We avoid capturing static segments of each video (e.g., when the artist is speaking) by requiring that adjacent frames in each sequence have at least 1% of the pixels changing by a fixed intensity threshold. We use a dynamic programming method to find all training and validation sequences that satisfy these criteria. We train on sequences of length 3 or 5 for sequential CVAE training, and length $\tau = 40$ for sequential sampling training, which we determined using experiments on the validation set. For evaluations on the test set, we extract a single sequence from each test video that satisfies the filtering criteria.

### 5.1.2 Crop extraction

To facilitate learning from small numbers of videos, we use multiple crops from each video. We first downsample each video spatially to $126 \times 168$, so that most patches contain visually interesting content and spatial context, and then extract $50 \times 50$ crops with minimal overlap.

### 5.2. Baselines

**Deterministic video synthesis** (*unet*): In image synthesis tasks, it is common to use an encoder-decoder architecture with skip connections, similar to U-Net [24, 47]. We adapt this technique to synthesize an entire video at once.

**Stochastic video synthesis** (*vdp*): Visual deprojection synthesizes a distribution of videos from a single temporally-projected input image [5].

We design each baseline model architecture to have a comparable number of parameters to our model. Both baselines output videos of a fixed length, which we choose to be 40 to be comparable to our choice of $\tau = 40$ in Section 5.1.

### 5.3. Results

We conducted both quantitative and qualitative evaluations. We first present a user study quantifying human perception of the realism of our synthesized videos. Next, we qualitatively examine our synthetic videos, and discuss characteristics that contribute to their realism. Finally, we discuss quantitative metrics for comparing sets of sampled videos to real videos. We show additional results, including videos and visualizations using the tipiX tool [13] on our project page at https://xamyzhao.github.io/timecraft.
Similarly to the artist, our method paints in a coarse-to-fine manner. Blue arrows show where our method first applies a flat color, and then adds fine details. Red arrows indicate where the baselines add fine details even in the first time step.

Our method works on similar regions to the artist, although it does not use the same color layers to achieve the completed painting. Blue arrows show where our method paints similar parts of the scene to the artist (filling in the background first, and then the house, and then adding details to the background). Red arrows indicate where the baselines do not paint according to semantic boundaries, gradually fading in the background and the house in the same time step.

Figure 10: Videos predicted from the digital (top) and watercolor (bottom) test sets. For the stochastic methods \textit{vdp} and \textit{ours}, we show the nearest sample to the real video out of 2000 samples. We show additional results in the appendices.

<table>
<thead>
<tr>
<th>Comparison</th>
<th>All paintings</th>
<th>Watercolor paintings</th>
<th>Digital paintings</th>
</tr>
</thead>
<tbody>
<tr>
<td>real &gt; vdp</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>real &gt; ours</td>
<td>55%</td>
<td>60%</td>
<td>51%</td>
</tr>
<tr>
<td>ours &gt; vdp</td>
<td>91%</td>
<td>90%</td>
<td>88%</td>
</tr>
</tbody>
</table>

Table 1: User study results. Users compared the realism of pairs of videos randomly sampled from \textit{ours}, \textit{vdp}, and real videos. The vast majority of participants preferred our videos over \textit{vdp} videos ($p < 0.0001$). Similarly, most participants chose real videos over \textit{vdp} videos ($p < 0.0001$). Users preferred real videos over \textit{ours} ($p = 0.0004$), but many participants confused our videos with the real videos, especially for digital paintings.

We experimented with training each method on digital or watercolor paintings only, as well as on the combined paintings dataset. For all methods, we found that training on the combined dataset produced the best qualitative and quantitative results (likely due to our limited dataset size), and we only present results for those models.

5.3.1 Human evaluations

We surveyed 158 people using Amazon Mechanical Turk [2]. Participants compared the realism of pairs of videos randomly sampled from \textit{ours}, \textit{vdp}, or the real videos. In this study, we omit the weaker baseline \textit{unet}, which performed consistently worse on all metrics (discussed below).

We first trained the participants by showing them several examples of real painting time lapses. We then presented a pair of time lapse videos generated by different methods for the center crop of the same painting, and asked “Which
video in each pair shows a more realistic painting process?” We repeated this process for 14 randomly sampled paintings from the test set. Full study details are in the appendix.

Table 1 indicates that almost every participant thought videos synthesized by our model looked more realistic than those synthesized by vdp (p < 0.0001). Furthermore, participants confused our synthetic videos with real videos nearly half of the time. In the next sections, we show example synthetic videos and discuss aspects that make our model’s results appear more realistic, offering an explanation for these promising user study results.

5.3.2 Qualitative results

Figure 9 shows sample sequences produced by our model for two input paintings. Our model chooses different orderings of semantic regions during the painting process, leading to different paths that still converge to the same completed painting.

For each crop, we draw 8442 samples and report the closest one to the ground truth. We quantify how similar the set of painting change shapes are between the ground truth and each predicted video, disregarding the order in which they were performed. We define the painting change shape as a binary map of the changes made in each time step. For each time step in each test video, we compare the artist’s change shape to the most similarly shaped change synthesized by each method, as measured by intersection-over-union (IOU). This captures whether a method paints in similar semantic regions to the artist.

Table 2: We compare videos synthesized from the digital and watercolor painting test sets to the artists’ videos. For the stochastic methods vdp and ours, we draw 2000 video samples and report the closest one to the ground truth.

Best (across k samples) painting change shape similarity (higher is better): We quantify how similar the set of painting change shapes are between the ground truth and each predicted video, disregarding the order in which they were performed. We define the painting change shape as a binary map of the changes made in each time step. For each time step in each test video, we compare the artist’s change shape to the most similarly shaped change synthesized by each method, as measured by intersection-over-union (IOU). This captures whether a method paints in similar semantic regions to the artist.

We summarize these results in Table 2. We introduce the interp baseline, which linearly interpolates in time, as a quantitative lower bound. The deterministic interp and unet approaches perform poorly for both metrics. For k = 2000, vdp and our method produce samples that lead to comparable “best video similarity” by L1 distance, highlighting the strength of methods designed to capture distributions of videos. The painting change IOU metric shows that our method synthesizes changes that are significantly more realistic than the other methods.

6. Conclusion

In this work, we introduce a new video synthesis problem: making time lapse videos that depict the creation of paintings. We proposed a recurrent probabilistic model that captures the stochastic decisions of human artists. We introduced an alternating sequential training scheme that encourages the model to make realistic predictions over many time steps. We demonstrated our model on digital and watercolor paintings, and used it to synthesize realistic and varied painting videos. Our results, including human evaluations, indicate that the proposed model is a powerful first tool for capturing stochastic changes from small video datasets.

7. Acknowledgments

We thank Zoya Bylinskii of Adobe Inc. for her insights around designing effective and accurate user studies. This work was funded by Wistron Corporation.
References

Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Michael Mathieu, Camille Couprie, and Yann Lecun. Deep
Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Roni Mittelman, Benjamin Kuipers, Silvio Savarese, and
Seonghyeon Nam, Chongyang Ma, Menglei Chai, William Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam
Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense
Mikyung Kim and Hyun Joon Shin. An example-based ap-
Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual
Diederik P Kingma, Shakir Mohamed, Danilo Jimenez
Diederik P Kingma and Max Wellin. Auto-encoding variational
Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense Diederik P Kingma, Shakir Mohamed, Danilo Jimenez
edge-aware directional texture painting. In Computer Graph-
Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam
Jingwen Lu, Connelly Barnes, Stephen DiVerdi, and Adam
Finkelstein. Realbrush: painting with examples of physical
Michael Mathieu, Camille Couprie, and Yann Lecun. Deep
multiscale video prediction beyond mean square error. 11
Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame inter-
Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent gram-
Roni Mittelman, Benjamin Kuiipers, Silvio Savarese, and Honglak Lee. Structured recurrent temporal restricted boltz-
Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Romain Vergne, Joëlle Thollot, Hans-Martin Rall, and Da-

[38] Michael Mathieu, Camille Couprie, and Yann Lecun. Deep multi-scale video prediction beyond mean square error. 11 2016.
[39] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame inter-
[40] Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent gram-
[41] Roni Mittelman, Benjamin Kuiipers, Silvio Savarese, and Honglak Lee. Structured recurrent temporal restricted boltz-
[42] Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Romain Vergne, Joëlle Thollot, Hans-Martin Rall, and Da-


