Painting Many Pasts: Synthesizing Time Lapse Videos of Paintings

Amy Zhao
MIT
xamyzhao@mit.edu

Guha Balakrishnan
MIT
balakg@mit.edu

Kathleen M. Lewis
MIT
kmlewis@mit.edu

Frédo Durand
MIT
fredo@mit.edu

John V. Guttag
MIT
guttag@mit.edu

Adrian V. Dalca
MIT, MGH
adalca@mit.edu

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.

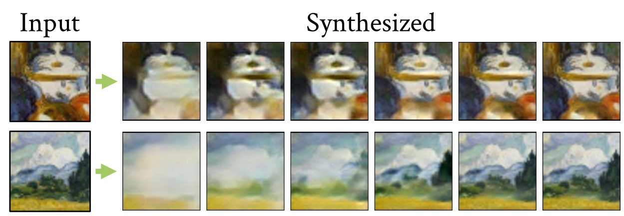


Figure 1: We present a probabilistic model for synthesizing time lapse videos of paintings. We demonstrate our model on *Still Life with a Watermelon and Pomegranates* by Paul Cezanne (top), and *Wheat Field with Cypresses* by Vincent van Gogh (bottom).

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 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 dramatic 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 progressions 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

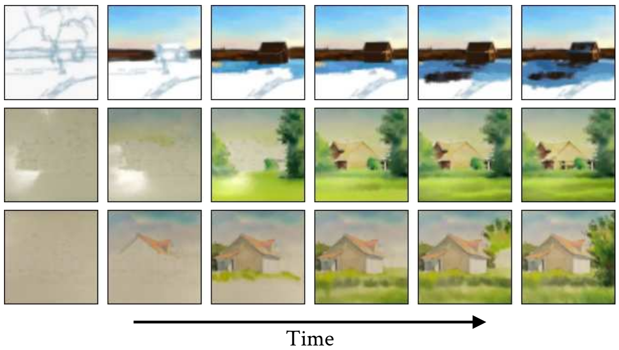


Figure 2: **Several real painting progressions of similar-looking scenes**. Each artist fills in the house, sky and field in a different order.

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.

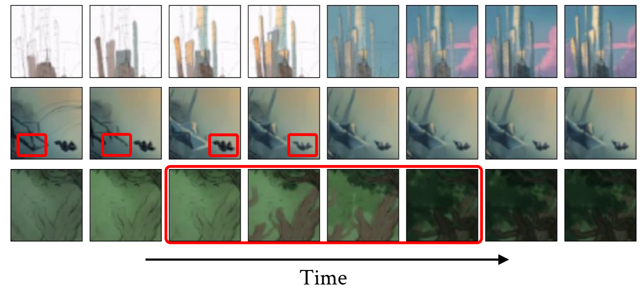


Figure 3: **Example digital painting sequences**. These sequences show a variety of ways to add paint, including fine strokes and filling (row 1), and broad strokes (row 3). We use red boxes to outline challenges, including erasing (row 2) and drastic changes in color and composition (row 3).

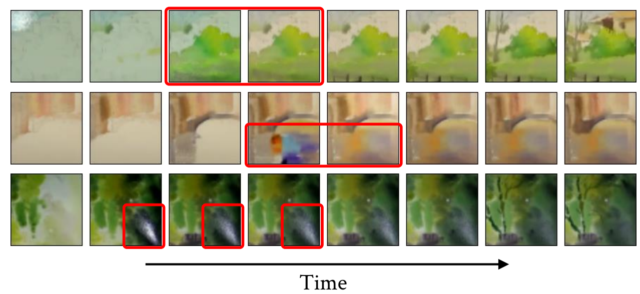


Figure 4: **Example watercolor painting sequences**. The outlined areas highlight some watercolor-specific challenges, including changes in lighting (row 1), diffusion and fading effects as paint dries (row 2), and specular effects on wet paint (row 3).

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 $x_1, \cdots, x_{T-1}$. Suppose we have a training set of real time lapse videos $\{\mathbf{x}^{(i)} = x_1^{(i)}, \cdots, x_{T^{(i)}}^{(i)}\}$. 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 δ_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 δ_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

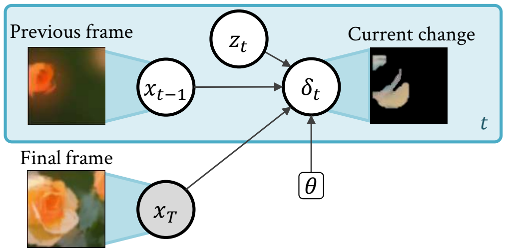


Figure 5: **The proposed probabilistic model**. Circles represent random variables; the shaded circle denotes a variable that is observed at inference time. The rounded rectangle represents model parameters.

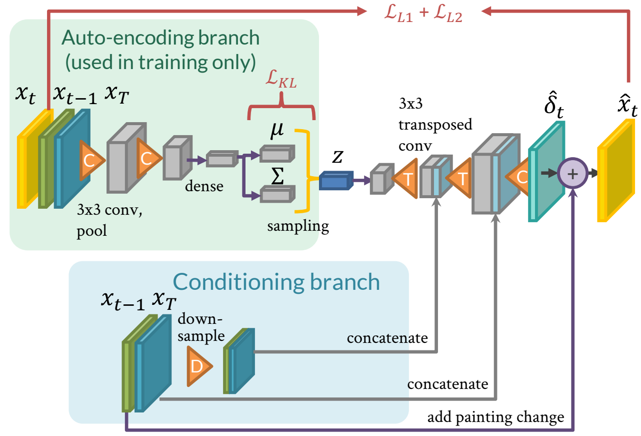


Figure 6: **Neural network architecture**. We implement our model using a conditional variational autoencoder framework. At training time, the network is encouraged to reconstruct the current frame x_t, while sampling the latent z_t from a distribution that is close to the standard normal. At test time, the auto-encoding branch is removed, and z_t is sampled from the standard normal. We use the shorthand $\hat{\delta}_t = g_\theta(z_t, x_{t-1}, x_T), \hat{x}_t = x_{t-1} + \hat{\delta}_t$.

$p_\theta(\delta_t|z_t, x_{t-1}; x_T)$ (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 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 the likelihood take the form:

$$\begin{aligned} &p_\theta(\delta_t|z_t, x_{t-1}; x_T) \\ &\propto e^{-\frac{1}{\sigma_1^2}|\delta_t - \hat{\delta}_t|} \mathcal{N}\big(V(x_{t-1}+\delta_t); V(x_{t-1}+\hat{\delta}_t), \sigma_2^2 \mathbb{I}\big), \end{aligned} \tag{1}$$

where $\hat{\delta}_t = g_\theta(z_t, x_{t-1}, x_T)$, $g_\theta(\cdot)$ represents a function parameterized by θ, $V(\cdot)$ is a function that extracts normalized VGG features, and σ_1, σ_2 are fixed noise parameters.

We assume the latent variable z_t is generated from the multivariate standard normal distribution:

$$p(z_t) = \mathcal{N}(z_t; 0, \mathbb{I}). \tag{2}$$

We aim to find model parameters θ that best explain all videos in our dataset:

$$\begin{aligned} &\arg\max_\theta \Pi_i \Pi_t p_\theta(\delta_t^{(i)}, x_{t-1}^{(i)}, x_{T^{(i)}}^{(i)}) \\ &= \arg\max_\theta \Pi_i \Pi_t \int_{z_t} p_\theta(\delta_t^{(i)}|z_t^{(i)}, x_{t-1}^{(i)}; x_{T^{(i)}}^{(i)}) dz_t. \end{aligned} \tag{3}$$

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

$$\begin{aligned} &q_\phi(z_t|\delta_t, x_{t-1}, x_T) \\ &= \mathcal{N}\big(z_t; \mu_\phi(\delta_t, x_{t-1}, x_T), \Sigma_\phi(\delta_t, x_{t-1}, x_T)\big), \end{aligned} \tag{4}$$

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

4.1.1 Neural network framework

We implement the functions g_θ, μ_ϕ and Σ_ϕ as a convolutional encoder-decoders parameterized by θ and ϕ, using a conditional variational autoencoder (CVAE) framework [54, 64]. We use an architecture (summarized in Figure 6) similar to [64]. We include full details in the appendix.

4.2. Learning

We learn model parameters using short sequences from the training video dataset, which we discuss in further detail in Section 5.1. We use two stages of optimization to facilitate convergence: pairwise optimization, and sequence optimization.

4.2.1 Pairwise optimization

From Equations (3) and (4), we obtain an expression for each *pair* of consecutive frames (a derivation is provided in the appendix):

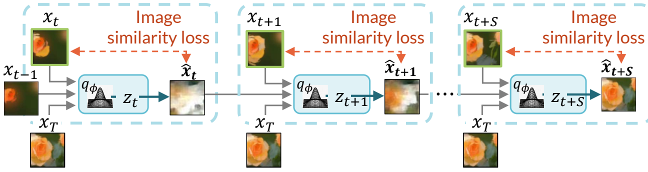


Figure 7: **Sequential CVAE training**. Our model is trained to reconstruct a real frame (outlined in green) while building upon its previous predictions for S time steps.

$$\log p_\theta(\delta_t, x_{t-1}, x_T) \geq \mathbb{E}_{z_t \sim q_\phi(z_t|x_{t-1},\delta_t;x_T)}\left[\log p_\theta(\delta_t|z_t, x_{t-1}; x_T)\right] - KL[q_\phi(z_t|\delta_t, x_{t-1}; x_T)||p(z_t)], \quad (5)$$

where $KL[\cdot||\cdot]$ denotes the Kullback-Liebler divergence. Combining Equations (1), (2), (4), and (5), we minimize:

$$\mathcal{L}_{KL} + \frac{1}{\sigma_1^2}\mathcal{L}_{L1}(\delta_t, \hat{\delta}_t) + \frac{1}{2\sigma_2^2}\mathcal{L}_{L2}(VGG(x_{t-1}+\delta_t), VGG(x_{t-1}+\hat{\delta}_t)), \quad (6)$$

where $\mathcal{L}_{KL} = \frac{1}{2}\big(-\log\Sigma_\phi + \Sigma_\phi + \mu_\phi^2\big)$, and the image similarity terms $\mathcal{L}_{L1}, \mathcal{L}_{L2}$ 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 *starter sequences* teach the model how to start a painting at inference time.

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 [52]. 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.

Sequential CVAE training encourages *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 $z_t \sim q_\phi(z_t|x_t - \hat{x}_{t-1}, \hat{x}_{t-1}, x_T)$. We compare each predicted frame to the corresponding real frame using the image similarity losses in Eq. (6). We illustrate this in Figure 7.

Sequential sampling training encourages random samples from our learned distribution to look like *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 [15], 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 [19]:

$$\mathcal{L}_{critic} = \mathbb{E}_{z_t \sim p(z_t)}\left[D_\psi(\hat{x}_t, \hat{x}_{t-1}, x_T)\right] - \mathbb{E}_{x_t}\left[D_\psi(x_t, x_{t-1}, x_T)\right], \quad (7)$$

where $D_\psi(\cdot)$ is a critic function with parameters ψ. The critic encourages the distribution of sampled painting changes $\hat{\delta}_t = g_\theta(z_t, \hat{x}_{t-1}, x_T), z_t \sim p(z_t)$ to match the distribution of training painting changes δ_t. We use a critic architecture based on [10] and optimize it using WGAN-GP [19].

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

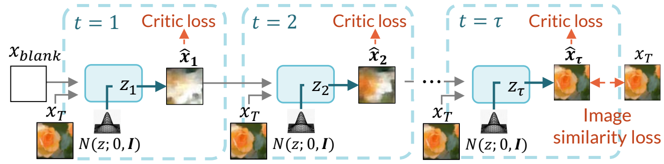


Figure 8: **Sequential sampling training**. We use a conditional frame critic to encourage all frames sampled from our model to look realistic. The image similarity loss on the final frame encourages the model to complete the painting in τ time steps.

4.3. Inference: video synthesis

Given a completed painting x_T and learned model parameters θ, ϕ, 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 $\hat{x}_0 = x_{blank}$, a blank frame.

4.4. Implementation

We implement our model using Keras [11] and Tensorflow [1]. We experimentally selected the hyperparameters controlling the reconstruction loss weights to be $\sigma_1 = \sigma_2 = 0.1$, using the validation set.

5. Experiments

5.1. Datasets

We collected time lapse recordings of paintings from YouTube and Vimeo. We selected digital and watercolor

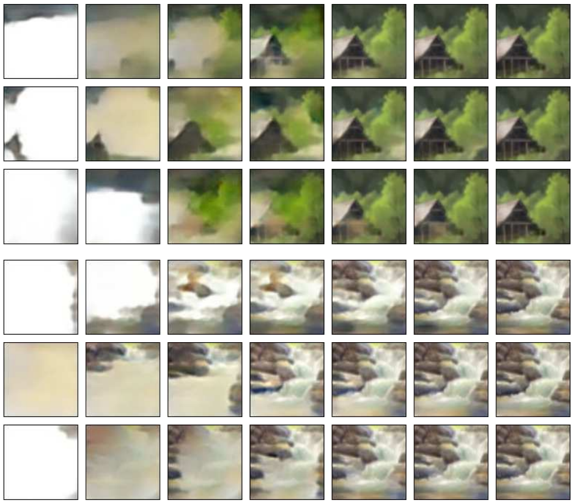


Figure 9: **Diversity of sampled videos**. We show examples of our method applied to a digital (top 3 rows) and a watercolor (bottom 3 rows) painting from the test set. Our method captures diverse and plausible painting trajectories.

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 360×640 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 γ frames in each synthesized time step), with a maximum variance of ϵ 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 γ and ϵ 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 126×168, so that most patches contain visually interesting content and spatial context, and then extract 50×50 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.

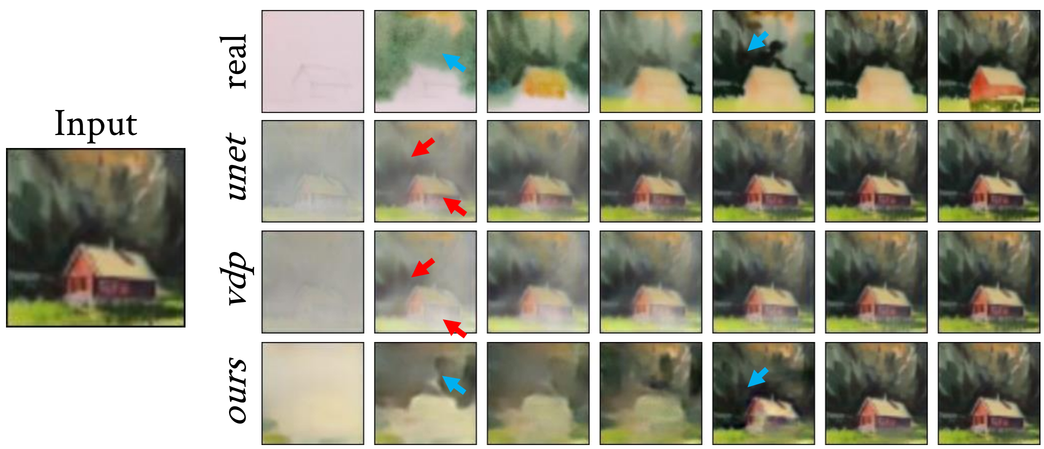
Stochastic video synthesis (*vdp*): Visual deprojection synthesizes a distribution of videos from a single temporally-projected input image [5].

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`.

(a) **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.



(b) **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 *vdp* and *ours*, we show the nearest sample to the real video out of 2000 samples. We show additional results in the appendices.

Comparison	All paintings	Watercolor paintings	Digital paintings
real > *vdp*	90%	90%	90%
real > *ours*	55%	60%	51%
ours > *vdp*	91%	90%	88%

Table 1: **User study results.** Users compared the realism of pairs of videos randomly sampled from *ours*, *vdp*, and real videos. The vast majority of participants preferred our videos over *vdp* videos ($p < 0.0001$). Similarly, most participants chose real videos over *vdp* videos ($p < 0.0001$). Users preferred real videos over 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 *ours*, *vdp*, or the real videos. In this study, we omit the weaker baseline *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.

Figure 10 shows videos synthesized by each method. To objectively compare the stochastic methods *vdp* and *ours*, we show the most similar predicted video by L1 distance to the ground truth video. The ground truth videos show that artists tend to paint in a coarse-to-fine manner, using broad strokes near the start of a painting, and finer strokes near the end. Artists also tend to focus on one or a few semantic regions in each time step. As we highlight with arrows, our method captures these tendencies better than baselines, having learned to make changes within separate semantic regions such as mountains, cabins and trees. Our predicted trajectories are similar to the ground truth, showing that our sequential modeling approach is effective at capturing realistic temporal progressions. In contrast, the baselines tend to make blurry changes without separating the scene into components.

5.3.3 Quantitative results

In a stochastic task, comparing synthesized results to "ground truth" is ill-defined, and developing quantitative measures of realism is difficult [25, 48]; these challenges motivated our user study above. In this section, we explore quantitative metrics designed to measure aspects of time lapse realism. For each video in the test set, we extract a 40-frame long sequence according to the criteria described in Section 5.1.1, and evaluate each method on 5 random crops using several video similarity metrics:

Best (across k samples) overall video distance (lower is better): For each crop, we draw k sample videos from each model and report the closest sample to the true video by L1 distance [5]. This captures whether a model produces some realistic samples, and whether the model is diverse enough to capture each artist's specific choices.

Method	Digital paintings		Watercolor paintings	
	L1	Change IOU	L1	Change IOU
interp	0.49 (0.13)	0.17 (0.06)	0.38 (0.09)	0.17 (0.09)
unet	0.18 (0.08)	0.24 (0.08)	0.15 (0.06)	0.27 (0.07)
vdp	**0.16** (0.06)	0.31 (0.10)	**0.14** (0.05)	0.32 (0.08)
ours	**0.16** (0.05)	**0.36** (0.09)	**0.14** (0.05)	**0.36** (0.07)

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

[1] Martín Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Inc. Amazon Mechanical Turk. Amazon mechanical turk: Overview, 2005.

[3] Ryoichi Ando and Reiji Tsuruno. Segmental brush synthesis with stroke images. 2010.

[4] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[5] Guha Balakrishnan, Adrian V. Dalca, Amy Zhao, John V. Guttag, Fredo Durand, and William T Freeman. Visual deprojection: Probabilistic recovery of collapsed dimensions. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[6] William Baxter, Yuanxin Liu, and Ming C Lin. A viscous paint model for interactive applications. *Computer Animation and Virtual Worlds*, 15(3-4):433–441, 2004.

[7] William V Baxter and Ming C Lin. A versatile interactive 3d brush model. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 319–328. IEEE, 2004.

[8] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 3601–3606. IEEE, 2002.

[9] Zhili Chen, Byungmoon Kim, Daichi Ito, and Huamin Wang. Wetbrush: Gpu-based 3d painting simulation at the bristle level. *ACM Transactions on Graphics (TOG)*, 34(6):200, 2015.

[10] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[11] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[12] Nelson S-H Chu and Chiew-Lan Tai. Moxi: real-time ink dispersion in absorbent paper. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 504–511. ACM, 2005.

[13] Adrian V Dalca, Ramesh Sridharan, Natalia Rost, and Polina Golland. tipix: Rapid visualization of large image collections. *MICCAI-IMIC Interactive Medical Image Computing Workshop*.

[14] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666, 2016.

[15] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018.

[16] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *KDD*, volume 99, pages 63–72, 1999.

[17] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, SM Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. *arXiv preprint arXiv:1804.01118*, 2018.

[18] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[20] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460. ACM, 1998.

[21] Fay Huang, Bo-Hui Wu, and Bo-Ru Huang. Synthesis of oil-style paintings. In *Pacific-Rim Symposium on Image and Video Technology*, pages 15–26. Springer, 2015.

[22] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[23] Savage Interactive. *Procreate Artists' Handbook*. Savage, 2016.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[26] Biao Jia, Jonathan Brandt, Radomír Mech, Byungmoon Kim, and Dinesh Manocha. Lpaintb: Learning to paint from self-supervision. *CoRR*, abs/1906.06841, 2019.

[27] Biao Jia, Chen Fang, Jonathan Brandt, Byungmoon Kim, and Dinesh Manocha. Paintbot: A reinforcement learning approach for natural media painting. *CoRR*, abs/1904.02201, 2019.

[28] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.

[29] Pierre-Marc Jodoin, Emric Epstein, Martin Granger-Piché, and Victor Ostromoukhov. Hatching by example: a statistical approach. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 29–36. ACM, 2002.

[30] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[31] Mikyung Kim and Hyun Joon Shin. An example-based approach to synthesize artistic strokes using graphs. In *Computer Graphics Forum*, volume 29, pages 2145–2152. Wiley Online Library, 2010.

[32] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[34] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010.

[35] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017.

[36] Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam Finkelstein. Realbrush: painting with examples of physical media. *ACM Transactions on Graphics (TOG)*, 32(4):117, 2013.

[37] Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Brushables: Example-based edge-aware directional texture painting. In *Computer Graphics Forum*, volume 34, pages 257–267. Wiley Online Library, 2015.

[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 interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1418, 2015.

[40] Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent grammar cells"". In *Advances in neural information processing systems*, pages 1925–1933, 2014.

[41] Roni Mittelman, Benjamin Kuipers, Silvio Savarese, and Honglak Lee. Structured recurrent temporal restricted boltzmann machines. In *International Conference on Machine Learning*, pages 1647–1655, 2014.

[42] Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Romain Vergne, Joëlle Thollot, Hans-Martin Rall, and Davide Benvenuti. Edge-and substrate-based effects for watercolor stylization. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 2. ACM, 2017.

[43] Seonghyeon Nam, Chongyang Ma, Menglei Chai, William Brendel, Ning Xu, and Seon Joo Kim. End-to-end time-lapse video synthesis from a single outdoor image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1409–1418, 2019.

[44] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018.

[45] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.

[46] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

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

[50] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608, 2009.

[51] Dizan Vasquez and Thierry Fraichard. Motion prediction for moving objects: a statistical approach. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3931–3936. IEEE, 2004.

[52] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017.

[53] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[54] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[55] Jacob Walker, Abhinav Gupta, and Martial Hebert. Patch to the future: Unsupervised visual prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3302–3309, 2014.

[56] Miaoyi Wang, Bin Wang, Yun Fei, Kanglai Qian, Wenping Wang, Jiating Chen, and Jun-Hai Yong. Towards photo watercolorization with artistic verisimilitude. *IEEE transactions on visualization and computer graphics*, 20(10):1451–1460, 2014.

[57] Der-Lor Way and Zen-Chung Shih. The Synthesis of Rock Textures in Chinese Landscape Painting. *Computer Graphics Forum*, 2001.

[58] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. Optical flow guided tv-l 1 video interpolation and

restoration. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 273–286. Springer, 2011.

[59] Ning Xie, Hirotaka Hachiya, and Masashi Sugiyama. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting. *CoRR*, abs/1206.4634, 2012.

[60] Ning Xie, Tingting Zhao, Feng Tian, Xiaohua Zhang, and Masashi Sugiyama. Stroke-based stylization learning and rendering with inverse reinforcement learning. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2531–2537. AAAI Press, 2015.

[61] Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG)*, 33(6):172, 2014.

[62] Songhua Xu, Min Tang, Francis Lau, and Yunhe Pan. A solid model based virtual hairy brush. In *Computer Graphics Forum*, volume 21, pages 299–308. Wiley Online Library, 2002.

[63] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems*, pages 91–99, 2016.

[64] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.

[65] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1235–1248, 2013.

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[67] Yong Zhang, Weiming Dong, Chongyang Ma, Xing Mei, Ke Li, Feiyue Huang, Bao-Gang Hu, and Oliver Deussen. Data-driven synthesis of cartoon faces using different styles. *IEEE Transactions on image processing*, 26(1):464–478, 2017.

[68] Ming Zheng, Antoine Milliez, Markus Gross, and Robert W Sumner. Example-based brushes for coherent stylized renderings. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 3. ACM, 2017.

[69] Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2019.

[70] Yipin Zhou and Tamara L. Berg. Learning temporal transformations from time-lapse videos. volume 9912, pages 262–277, 10 2016.

[71] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.