Blowing in the Wind: CycleNet for Human Cinemagraphs from Still Images

Hugo Bertiche[1,2] Niloy J. Mitra[3,4] Kuldeep Kulkarni[4] Chun-Hao Paul Huang[4]
Tuanfeng Y. Wang[4] Meysam Madadi[1,2] Sergio Escalera[1,2] Duygu Ceylan[4]

[1]Universitat de Barcelona [2]Computer Vision Center [3]University College London [4]Adobe Research

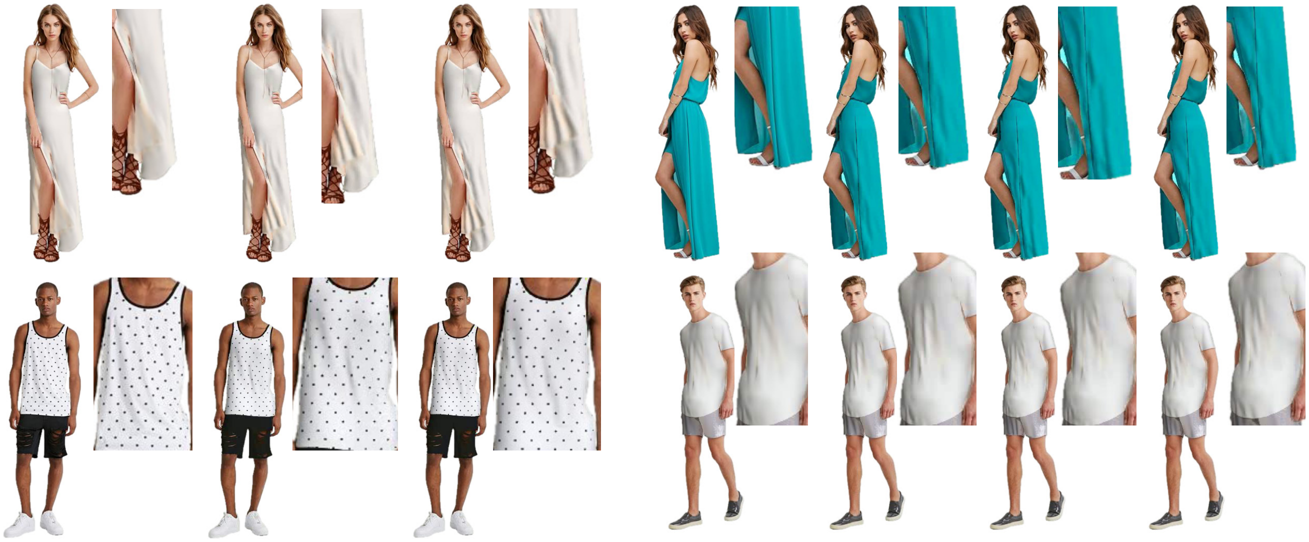


Figure 1. We introduce a method for automatic human cinemegraph generation from single RGB images focusing on generating plausible garment animation as if they are blown in the wind.

Abstract

Cinemagraphs are short looping videos created by adding subtle motions to a static image. This kind of media is popular and engaging. However, automatic generation of cinemagraphs is an underexplored area and current solutions require tedious low-level manual authoring by artists. In this paper, we present an automatic method that allows generating human cinemagraphs from single RGB images. We investigate the problem in the context of dressed humans under the wind. At the core of our method is a novel cyclic neural network that produces looping cinemagraphs for the target loop duration. To circumvent the problem of collecting real data, we demonstrate that it is possible, by working in the image normal space, to learn garment motion dynamics on synthetic data and generalize to real data. We evaluate our method on both synthetic and real data and demonstrate that it is possible to create compelling and plausible cinemagraphs from single RGB images.

1. Introduction

Cinemagraph, a term originally coined by Jamie Beck and Kevin Burg, refers to adding dynamism to still images by adding *minor and repeated movements*, forming a motion loop, to a still image. Such media format is both engaging and intriguing, as adding a simple and subtle motion can bring images to life. Creating such content, however, is challenging as it would require an artist to first set up and capture a suitable video, typically using a tripod, and then carefully mask out most of the movements in a post-processing stage.

We explore the problem of creating human cinemagraphs directly from a single RGB image of a person. Given a dataset of images and corresponding animated video pairs, a straightforward solution would be to train a fully supervised network to learn to map an input image to a plausible animated sequence. However, collecting such a dataset is extremely challenging and costly, as it would require capturing hundreds or thousands of videos of people holding a perfectly still pose under the influence of the wind from different known directions. While it is possible to simulate different wind force directions using oscillating fans in a lab setup [10], capturing the variability of garment geometry and appearance types in such a controlled setting is far from trivial. Hence, we explore the alternative approach of using synthetic data where different wind effects can easily be replicated using physically-based simulation. The challenge, then, is to close the synthetic-to-real gap, both in terms of garment dynamics and appearance variations.

We address this generalization concern by operating in the gradient domain, i.e., using surface normal maps. Being robust to lighting or appearance variations, surface normals are arguably easier to generalize from synthetic to real, compared to RGB images. Moreover, surface normals are indicative of the underlying garment geometry (i.e., folds and wrinkles) and hence provide a suitable representation to synthesize geometric and resultant appearance variations [21, 44] as the garment interacts with the wind.

Further, we make the following technical contributions. First, we propose a novel *cyclic* neural network formulation that directly outputs looped videos, with target time periods, without suffering from any temporal jumps. Second, we demonstrate how to condition the model architecture using wind parameters (e.g., direction) to enable control at test time. Finally, we propose a normal-based shading approach that takes the intermediate normals under the target wind attributes to produce RGB image frames. In Figure 1, we show that our method is applicable to a variety of real test images of different clothing types.

We evaluate our method on both synthetic and real images and discuss ablation results to evaluate the various design choices. We compare our approach against alternative approaches [27, 38] using various metrics as well as a user study to evaluate the plausibility of the generated methods. Our method achieves superior performance both in terms of quantitative metrics as well as the perceptual user study.

2. Related Work

2.1. Looping video generation

In this work, we are interested in synthesizing *cinemagraph* style looping animations where only certain parts of a frame are in motion. A typical method for creating such looping clips is to leverage video as input. Many approaches exist that solve an optimization problem to identify segments and transition points in the input video that can be looped seamlessly [1,4,7,14,23,24,32,35,42]. While we focus on generating such a looping clip from a static single image, we use a video based method [23] to ensure our training data is looped properly.

In the context of animating a single image in a looping manner, one approach is to warp regions of the image using Fourier methods in a stochastic manner which amounts to displacing the original texture [12]. Another approach is to transfer the phase patterns from an example video to the given input image [30]. Okabe et al. [28] also transfer the motion patterns from an example video to an input image of a fluid. Specifically, they map the example video to a constant flow and residual layers, which represent the high frequency motion patterns that are not explained by warping a reference frame using constant flow. Such residual patterns are transferred to the input image. These methods work best for natural phenomena such as water and fire where flow-based texture displacement and warping result in plausible animation. Halperin et al. [18] present another approach to animating a single image by focusing on repeating patterns. While demonstrating impressive results, such a method is not suitable for our problem since the motion a garment undergoes blowing in the wind is fundamentally different than displacing repeating patterns.

With the recent success of deep learning methods, several learning based approaches have been proposed to create looping animations from single images. While Endo et al. [16] predict a flow map to warp the input images directly, Holynski et al. [19] first generate a constant flow map directly from a single image and then warp image features using the generated flow map to synthesize the RGB frames. In a follow-up work, Mahapatra et al. [27] extend this framework to provide additional control of the motion direction and region of the image to be animated. We compare our method to this state-of-the-art approach and show that the assumption of constant flow is not suitable for garment motion and leads to unsatisfactory results. Recently, Fan et al. [17] present a method to animate fluids in a still image. Their method uses an additional depth map estimation to generate a surface mesh for the fluid region and thus utilizes physically based simulation priors to predict a motion field. While our approach of incorporating a surface normal map representation is similar, we focus on very different types of motions in our work.

2.2. Animating single images

With the success of deep learning, several methods have been recently proposed to animate a given image. One approach is based on using a driving video and focus on synthesizing specific type of content and motion such as time-lapse videos [11,26], facial and body animation [33,38]. We compare our method to the most recent method of Wang et al. [38] and show that it is not suitable to capture the subtle motions observed in a human cinemagraph.

Another line of work directly predicts video or future frames from a given single image [22, 40, 41, 43] or a semantic map [29]. Dorkenwald et al. [15] learn a generative model that encodes a latent residual representation and sample such latent code to synthesize a video from a given

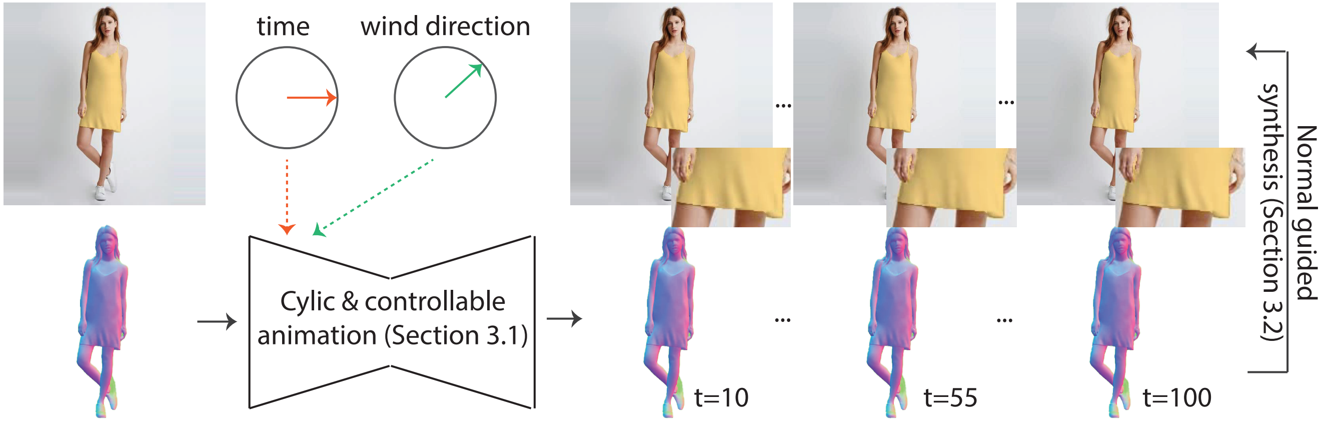

time
wind direction
Cylic & controllable animation (Section 3.1)
t=10
t=55
t=100
Normal guided synthesis (Section 3.2)

Figure 2. Given an input image (top, left) and its predicted surface normal map (bottom, left), we present a network that synthesizes a set of surface normals that resemble the effect of the garment blowing in the wind with a given direction. We ensure a looped animation by encoding the time t with a cyclic positional encoding with respect to a predefined loop duration (150 frames in our experiments). We then synthesize the corresponding RGB images demonstrating plausible garment deformation using an intrinsic image decomposition technique.

image. Many of these methods, however, synthesize multiple frames at the same time and hence operate only at low resolution without providing control. To address the latter challenge, Blatmann et al. [8, 9] enable the user to provide a poke that determines the final location of a sparse point in the input image. The resulting videos, however, are not looped in contrast to cinemagraphs.

Another interesting direction is to train a single image based generator [31], which can then be utilized to generate animations by providing random walks of the appearance of the object of interest in the latent space. Arora et al. [2] extend this approach to work with an input GIF. While impressive, such approaches do not provide the controllability we aim to achieve with our approach, however.

3. Methodology

Given a single input RGB image of a person, $\boldsymbol{I} \in \mathbb{R}^{W\times H\times 3}$, our goal is to generate a looped video sequence, $\boldsymbol{V} := \{\boldsymbol{I}_0, \boldsymbol{I}_1, ..., \boldsymbol{I}_t | \boldsymbol{I}_0 = \boldsymbol{I}_t\}$, where the loose garments worn by the person exhibit a plausible motion as if blown in the wind. We assume the direction of the wind can be provided by a unit vector $\boldsymbol{w}$ in the image plane to control the output animation. Hence, our goal is to learn the mapping $\mathcal{F}(\boldsymbol{I}, \boldsymbol{w}) \longrightarrow \boldsymbol{V}^{\boldsymbol{w}}$.

To more effectively represent the underlying garment geometry and the changes it undergoes due to the wind force, our method operates on the surface normal map $\boldsymbol{N}$ that corresponds to the input image $\boldsymbol{I}$. Specifically, given an input image $\boldsymbol{I}$, we first predict the surface normal map using an off-the-shelf normal estimator [3]. We then propose a novel cyclic network architecture that maps $\boldsymbol{N}$ to a sequence of normal maps $\boldsymbol{V}_{\boldsymbol{N}}^{\boldsymbol{w}} := \{\boldsymbol{N}_0, \boldsymbol{N}_1, ..., \boldsymbol{N}_t | \boldsymbol{N}_0 = \boldsymbol{N}_t\}$ that demonstrate plausible motion of the underlying garment under the influence of a wind force with a direction given by $\boldsymbol{w}$. Finally, we synthesize back the corresponding RGB images given the original input image and the sequence of animated normal maps using a constrained reshading approach. We provide the overall pipeline in Figure 2 and next discuss the details of our approach.

3.1. Cyclic and Controllable Animation

Given an input normal map $\boldsymbol{N}$ and a wind direction $\boldsymbol{w}$, our goal is to learn the mapping $\mathcal{F}_{\mathcal{N}}(\boldsymbol{N}, \boldsymbol{w}) \longrightarrow \boldsymbol{V}_{\boldsymbol{N}}^{\boldsymbol{w}} = \{\boldsymbol{N}_0, \boldsymbol{N}_1, ..., \boldsymbol{N}_t | \boldsymbol{N}_0 = \boldsymbol{N}_t\}$ where $\boldsymbol{V}_{\boldsymbol{N}}^{\boldsymbol{w}}$ demonstrates plausible garment animation. Our goal is to synthesize a cyclic animation sequence with a predefined period of T, so that $t = T - 1$. This amounts to synthesizing normal maps that satisfy constraints $\boldsymbol{N}_t = \boldsymbol{N}_{t+kT}\ \forall k \in \mathbb{Z}$.

We tackle this problem as an image-to-image translation task where our goal is to learn $f(\boldsymbol{N}_t, \Delta t, \boldsymbol{w}) \longrightarrow \boldsymbol{N}_{t+\Delta t}$ where $\Delta t \in [-T/2, T/2]$. Note that, since we are interested in looped animations, negative values for Δt correspond to valid animation samples. We realize the function f as a UNet architecture that is conditioned on both the residual time Δt and the wind direction $\boldsymbol{w}$, as shown in Figure 3. To enforce a cyclic behaviour, we first encode Δt using sinusoidal functions as:

$$\begin{aligned} \varphi_{\Delta t} &= \frac{2\pi n}{T}\Delta t, \quad n = 1, 2, 3, 4, 5. \\ \boldsymbol{x}_{\Delta t} &= \{\cos(\varphi_{\Delta t}), \sin(\varphi_{\Delta t})\}. \end{aligned} \tag{1}$$

This formulation ensures that $f(\boldsymbol{N}_t, \Delta t + kT, \boldsymbol{w})$ with $k \in \mathbb{Z}$ gives the same output resulting in a looping animation sequence. Similar to common practice in positional encoding [37], we observe that using multiples of the data frequency ($\omega = 2\pi n/T$) helps to learn higher frequency

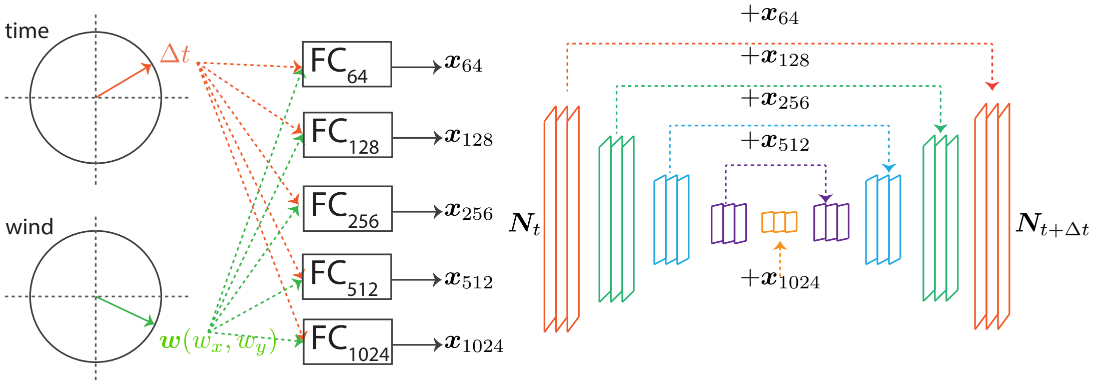


Figure 3. Cyclic wind-conditioned UNet. Given an input normal map N_t, a delta time increment Δt, and a wind direction $\boldsymbol{w}$; we extend the standard UNet architecture to give it a cyclic behaviour. We encode the time using a cylic positional encoding and concatenate with the wind direction. We pass the concatenated features through different fully convolutional layers to extract features of varying dimensions. The resulting features are provided as skip connections to the UNet architecture which synthesizes the final normal map $N_{t+\Delta t}$.

motions while still enforcing a global cyclic behaviour with period T. Note then how the time encoding $\boldsymbol{x}_{\Delta t}$ consists of multiple circumferences parameterized by Δt.

We represent the wind direction as a unit vector as $\boldsymbol{w}$ in the image plane. We concatenate $\boldsymbol{w}$ with $\boldsymbol{x}_{\Delta t}$ resulting in the final conditioning code $\boldsymbol{x} := \boldsymbol{x}_{\Delta t} \| \boldsymbol{w} = (x_{\Delta t,0}, x_{\Delta t,1}, ..., x_{\Delta t,2n}, w_x, w_y)$. We condition the UNet by introducing $\boldsymbol{x}$ at each feature map extracted by the encoder at different scales. To do so, we first linearly transform $\boldsymbol{x}$ to the corresponding feature map dimensionality with learnable weights $\{\boldsymbol{W}_i \in \mathbb{R}^{F_i \times D}\}$, where F_i is the number of channels of the i-th feature map and D is the dimensionality of $\boldsymbol{x}$. We apply 1×1 conovolutions to the feature maps before and after combining them with $\boldsymbol{x}$.

3.2. Normal Guided Synthesis

The final stage of our approach focuses on computing the final cinemagraph $\boldsymbol{V}$ given the original input RGB image $\boldsymbol{I}$ and the predicted normal map sequence $\boldsymbol{V}_{\boldsymbol{N}}^{\boldsymbol{w}}$. To this end, we rely on the concept of intrinsic image decomposition, which decomposes images into two layers $\boldsymbol{I} = \boldsymbol{S}\boldsymbol{R}$: (i) the reflectance $\boldsymbol{R} \in \mathbb{R}^{W \times H \times 3}$, which denotes the albedo invariant color of the materials, and (ii) the shading $\boldsymbol{S} \in \mathbb{R}^{W \times H}$ which is the result of the interaction of the light with the underlying geometry of the garment. In particular, the shading layer is crucial in how we perceive the changes in the fold and wrinkle patterns of the garment as it is animated. Given this observation, we synthesize a new shading layer that is consistent with the animated surface normal maps. Then, when composed with the original reflectance map reflects it generates the intended animation.

Given the input image $\boldsymbol{I}$, we first run an off-the-shelf intrinsic image decomposition method [5] to obtain the reflectance map $\boldsymbol{R}$ and the shading map $\boldsymbol{S}$. Assuming a simple lighting model composed of a directional and ambient light, we optimize for the light parameters using the predicted surface normal map from the input image:

$$\boldsymbol{S} = \max(0, -\boldsymbol{N}\boldsymbol{l}) + \delta, \tag{2}$$

where $\boldsymbol{l} \in \mathbb{R}^3$ is the light direction and $\delta \in \mathbb{R}^+$ is the ambient light. Given the predicted animated surface normal map sequence $\hat{\boldsymbol{V}}_{\boldsymbol{N}}$, we generate a new shading map sequence and composite it with the original reflectance map $\boldsymbol{R}$ to obtain the final RGB sequence $\hat{\boldsymbol{V}}$. At inference time, the user is required to provide a mask to denote the region of interest where motion is desired to be synthesized. Hence, we composite the original image and the synthesized RGB images based on this mask to provide the final output. While this approach changes only the shading without actually warping the texture of the garment, it is sufficient to provide the perception of a plausible animation.

Local vs Global. We design this methodology so it leans towards a local solution. The reasons for this are as follows. On one hand, cinemegraphs are characterized by subtle motions (local). On the other hand, *local* solutions generalize better, which is specially important for our approach to handle real test samples from a synthetic training set.

4. Experiments

In the following section, we describe the experimental setup and the qualitative and quantitative results. We detail the data used for training and evaluation, define the metrics, and briefly introduce the state-of-the-art baselines used for comparison. Finally, we provide a discussion of the results.

Datasets. In order to train our network, we generate a synthetic dataset that consists of different type of garments draped on human bodies with varying shape and pose. Specifically, we sample human body and garment pairs from the Cloth3D dataset [6], which is a large-scale dataset of clothed 3D humans. We select 1500 samples with

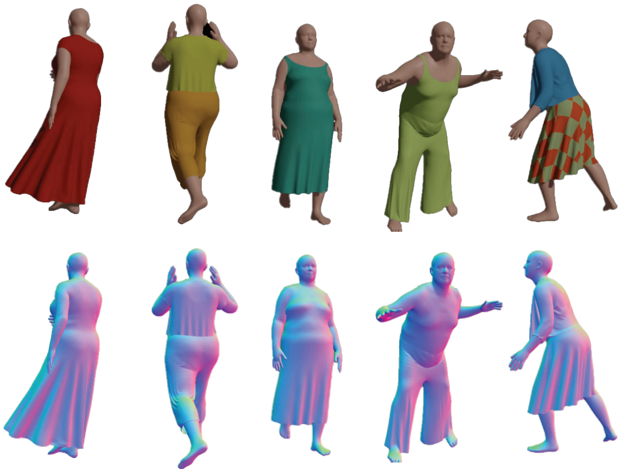


Figure 4. We train ours on a synthetic dataset that consists of different garment types draped on bodies with varying shape and poses acquired from the Cloth3D dataset [6]. We simulate the effect of wind and render the corresponding RGB and surface normal images.

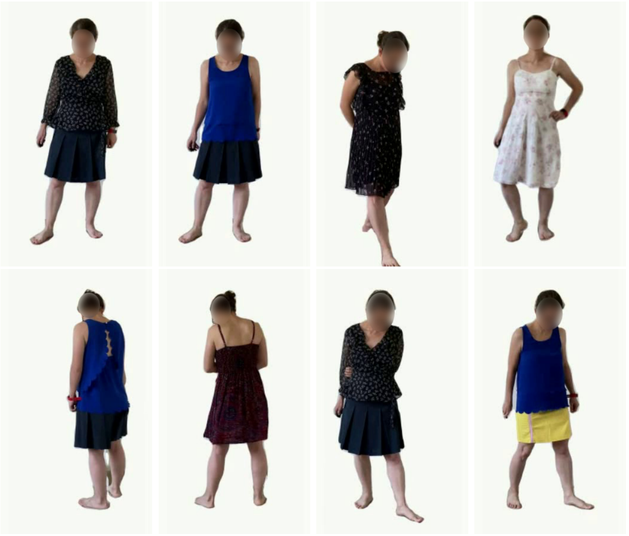


Figure 5. We capture a small real dataset where the subject keeps a still pose during the sequence while a fan generates wind. Different garment types show different dynamics.

skirt and dresses and 500 samples with other clothing types (e.g., trousers, tshirts). Each sample in the original Cloth 3D dataset is a motion sequence. We randomly choose one of the frames in each sequence as a random human body pose. The chosen frame, body and outfit, defines the initial conditions of our cloth simulation. We use Blender [13] to run the simulations. To this end, we choose a random wind direction in the image plane with constant wind force, and simulate the cloth dynamics while the underlying body remains still. Each simulation output is rendered from a fixed viewpoint with a predefined lighting setup. We apply random checkerboard texture patterns to some garments and assign a uniform color material to others. In addition to RGB output, we also render the corresponding surface normal maps and segmentation masks (body, cloth and background). Figure 4 shows examples from our dataset. We simulate each sample for 250 frames at 30 fps. We observe that the garment drapes on the body in roughly the first 50 frames of the sequence and later starts blowing in the wind. It is not trivial to guarantee the resulting garment animation is cyclic in such a physically based simulation setup. Hence, we process the resulting animations with the method of Liao et al. [23] which detects loops in an input video. After this step, we obtain animation sequences of length 150 frames which we use as the duration of loops, i.e., $T = 150$.

In addition to synthetic data, we test our method on real samples from the Deep Fashion dataset [25] as well as additional stock images to test generalization. To evaluate if the predictions obtained on real samples contain plausible cloth dynamics, we capture a small set of real examples. Specifically, we ask a human subject wearing different types of garments to hold a still pose next to an oscillating fan while we record a short video sequence with a fixed camera mounted on a tripod. We record 50 such videos demonstrating 8 different outfit types. Similar to synthetic data, we process each video with the method of Liao et al. [23] to obtain looped animations. Figure 5 shows some real samples.

Evaluation Metrics. We evaluate our method and baselines on synthetic data where we can access ground truth image and animation pairs. First, we adopt metrics that focus on pixel-level similarity. Specifically, we report per-pixel mean average error (MAE), mean squared error (MSE), root of mean squared error (RMSE), and PSNR. In addition we report metrics that focus on more structural (SSIM [39]) and perceptual similarities (LPIPS [45]). For DeepFashion samples we do not have ground truth video data. Hence, in order to evaluate the plausibility of the generated animated sequences we use Frechet Video Distance (FVD [36]) against the real data we have captured.

Baselines. We compare our method to two baselines. First, we compare with the work of Mahapatra et al. [27], which extends the original Eulerian motion fields approach [20] to a controllable setup. Since this method is a flow based approach and uses optical flow information to be provided in the dataset, we train it with the looped RGB videos in our synthetic dataset where optical flow can be more reliably estimated using off-the-shelf methods [34]. For each looped sequence, we extract a mask denoting the region where motion is observed and a sparse set of motion directions from the estimated optical flow. We also compare our method to LIA [38], a state-of-the-art single image-based controllable video generation framework. Since LIA requires a target video sequence to specify the desired animation, we provide the ground truth animation sequences as targets both during training and testing. While it is not possible to use this configuration in a real setup, it provides the best possible results. Outperforming LIA under this configuration means outperforming it under any other.

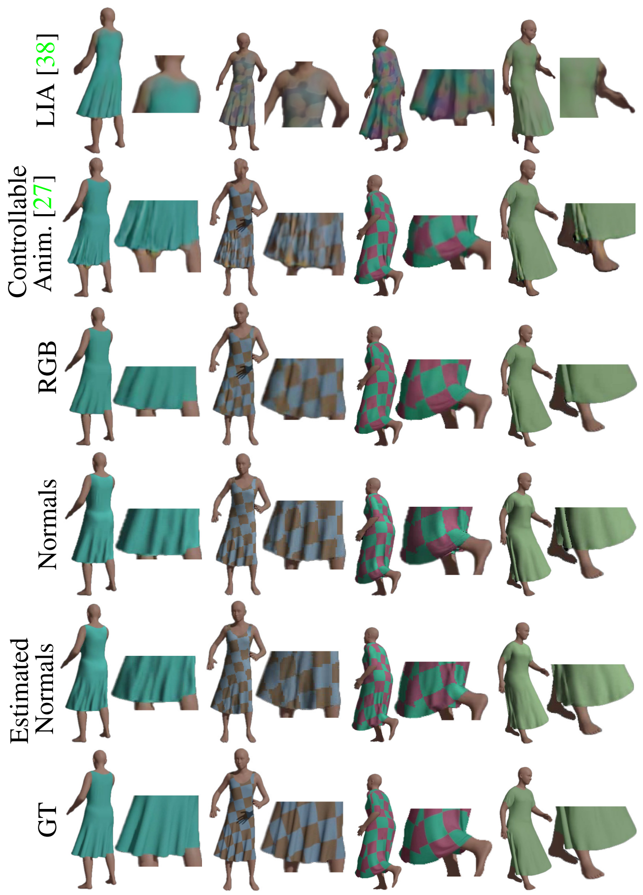


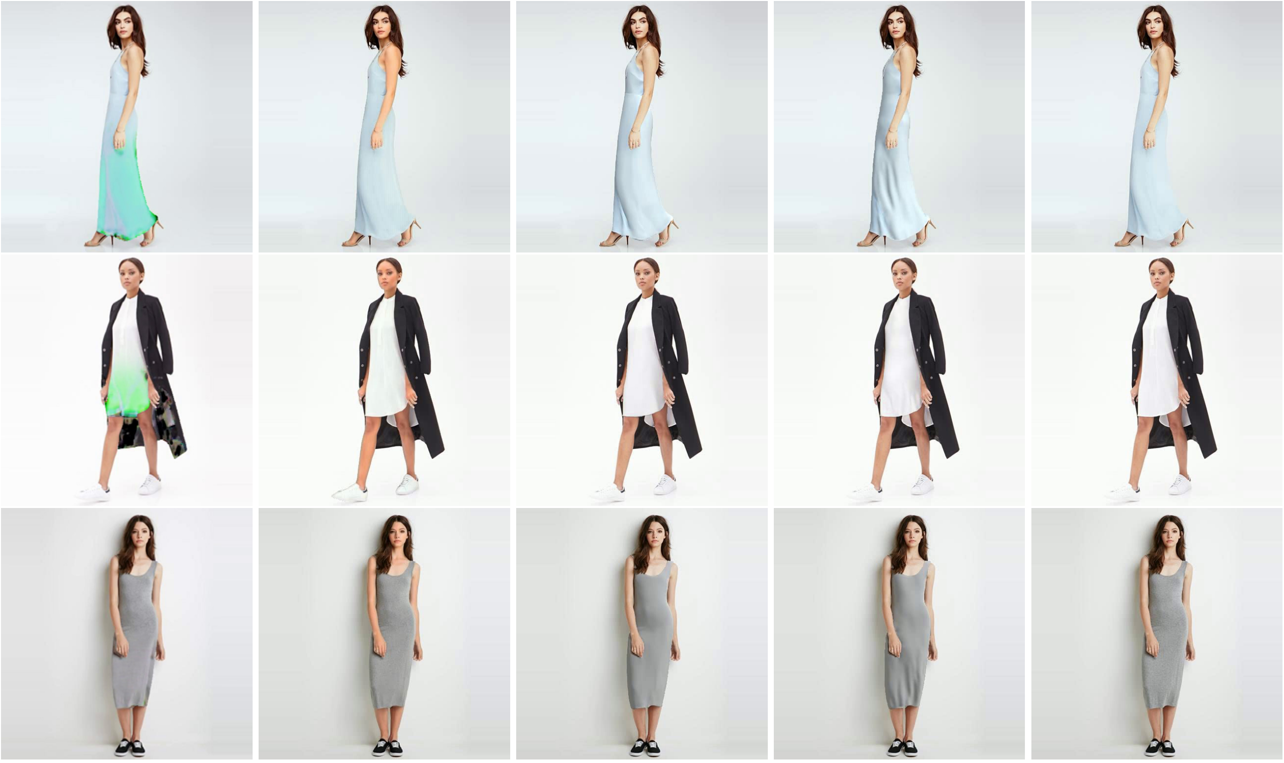
Figure 6. Ablation study. We evaluate qualitatively the image synthesis capacity of each methodology in our synthetic dataset, namely the state-of-the-art baselines LIA [38] and the work on controllable animations of [27] and the different variations of our approach, trained on different data modalities. Normal-based solutions require a re-shading step (Sec. 3.2). We observe SOTA solutions (warping-based) are sub-optimal for our setting. Ours is able to generate consistent images with plausible wrinkles.

Ablation. To analyze the effectiveness of the design choices, we test additional baselines which are variants of our method. On one hand, we train our CycleNet using RGB data directly, as opposed to the proposed pipeline in which we operate in the normal space. Next, we analyze the performance of the proposed methodology trained using ground truth normal maps. Nonetheless, there is still a domain gap between synthetic normal maps and normal maps estimated from real data. For this reason, we also train our method using normal maps estimated from synthetic RGB data as input, while using ground truth normal maps as labels. Since our eventual goal is to generate cinemagraphs from real images, we also assess the effectiveness of each approach on real data. Finally, we add as a reference, a *Toy* baseline, in which we *generate* the output cinemagraph $\boldsymbol{V}$ for a given input image $\boldsymbol{I}$ as $\boldsymbol{V} = \{\boldsymbol{I}, \boldsymbol{I}, ..., \boldsymbol{I}\}$. This baseline generates high quality videos since it uses the original image but lacks any motion. This baseline evaluates how informative the different metrics we use are for our task.

Tab. 1 shows the quantitative evaluation of the different baselines. Supervised metrics are computed using the test split of our synthetic dataset. Then, to measure quantitatively the capacity to generalize to real data, we test using DeepFashion samples as input and compare to the small set of real samples we have captured using FVD [36]. We show qualitative results of each approach in Fig. 6 and 7. LIA [38] is a solution tailored for human face animation, which exploits some characteristics of the domain, such as the structural and motion similarities of different faces performing the same action/expression (e.g., smiling, talking, etc.). Due to domain differences, we observe its performance is poor when trained on our task. It is unable to produce meaningful motions and often generates an average image for each synthetic animation sequence (best seen in the 2nd and 3rd column in Fig. 6). This is reflected in the quantitative metrics as well. The work of [27] is designed to work for images of fluids, under the simplifying assumption that a video sequence has a constant flow. While this assumption works well for natural images of fluids, it does not hold for the domain of garments. This method can generate consistent images, but with unrealistic motion and artifacts due to its warping based architecture. We see this in the second sample of Fig. 6. Furthermore, as expected, when trained with only synthetic data, neither of these solutions are able to generalize to real data.

	Synthetic Data						Real Data
Experiment	MAE	MSE	RMSE	SSIM	PSNR	Perceptual	FVD
LIA [38]	23.77	1131.62	32.45	0.11	25.50	400.26	873.86
Controllable Anim. [27]	9.37	302.26	15.89	0.48	34.49	**220.86**	625.45
Toy	**6.52**	197.73	12.36	**0.61**	**40.76**	**218.39**	643.42
RGB	7.64	**175.24**	**12.21**	**0.56**	**37.64**	230.40	623.32
Normals	15.75	563.76	22.63	0.41	32.11	231.29	633.12
Estimated normals	17.32	674.62	24.31	0.34	31.81	231.74	**613.26**

Table 1. Quantitative evaluation of the different methods tested including state-of-the-art baselines, LIA [38] and controllable animations [27], and the different variants of our method trained using different data modalities. Normal-based solutions require a re-shading step (Sec. 3.2)



(a) Contr. Anim. [27] (b) RGB (c) Normals (d) Estimated Normals (e) Input

Figure 7. We analyze the behaviour of each model during generalization using real samples from DeepFashion [25]. As can be seen, the work of [27] shows color artifacts. Then, our model shows a tradeoff between reconstruction fidelity and wrinkle generation. The RGB solution generates images where color is faithfully maintained, but shows no wrinkles or motion except for very few specific samples. On the other hand, normal-based solutions are not always able to generate the same color distribution due to limitations in the re-shading algorithm (Sec. 3.2). Additionally, we observe only the model trained on estimated normals is able to generalize properly to real samples. We omit LIA [38] from this comparison as it is unable to generalize at all to real samples.

Next, we analyze the different variations of our method. Unsurprisingly, the RGB solution is the best performing in the synthetic dataset according to the quantitative metrics. Predicting normal map sequences followed by a re-shading step impacts the pixel level reconstruction accuracy. Finally, using estimated normals as input slightly hinders performance w.r.t. using ground truth normals. As observed, the *Toy* baseline performs very well in comparison, while we know it is not generating any motion. This suggests the classical metrics used for image reconstruction have limitations for evaluating the solutions of our specific problem. The motion we want to generate is localized and subtle. The difference between a plausible motion and a static prediction may be smaller than the reconstruction error of an RGB auto-encoder. The solutions based in normal maps require a re-shading step that might produce a slight shift in pixel color. While this does not hurt the quality of the dynamics, it increases the reconstruction error. The increase in perceptual error from RGB to normal-based solutions is not comparable to that of the other metrics. This suggests that the perceived quality of the generated images is comparable. Next, we evaluate the performance in real samples. For this case, the behaviour we observe is different. While the RGB solution is able to faithfully reproduce the input image with a slight color shift, a large majority of predictions do not show any motion. We observe a similar behaviour with the model trained on ground truth synthetic normals. Due to the domain gap, very few input normal maps produce dynamics in the output. This, added to the color shift due to the re-shading step increases the value of the FVD. Finally, we observe the model trained using predicted normals as input is able to generate plausible dynamics for all the input normals estimated from real samples resulting in the best FVD score. Note how in this case, the *Toy* solution, which *generates* static videos, has the worst FVD score. In, Fig. 8, we show additional sequence results for our model trained on estimated normals. For each sequence, we show frames sampled uniformly every 25 frames. We also add close-up looks of the wrinkles. Our method is able to generate visually appealing results with plausible wrinkles. Finally, we further test the generalization of our method by testing with an image of a hanging garment. This can be seen in Fig. 9.

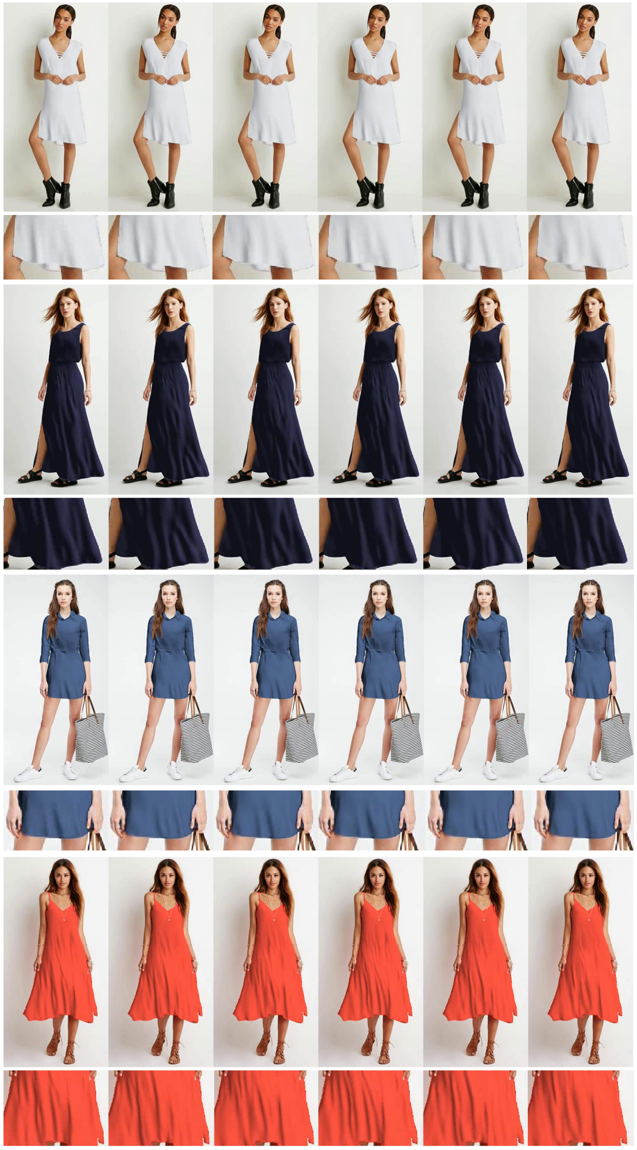


Figure 8. Qualitative results using real samples from DeepFashion [25]. We obtain this results with the model corresponding to the last row of Tab. 1. Our approach is able to generate consistent images with plausible wrinkles. We omit LIA [38] from this comparison since it is unable to generalize at all.

Finally, due to the limitations of the quantitative metrics, we complement the evaluation with a qualitative user study. We show random samples of generated animations both on synthetic and real data with the different methods. We ask the users to rate if the animations are plausible or not. Variants of our method are rated as plausible more than 90% of the time on synthetic data. For real images, our method trained on predicted normals is rated as plausible around 60% of the time whereas our method trained on RGB and ground truth normals is rated as plausible around 20% and 30% of the time respectively. The strongest baseline [27] is not perceived as plausible neither on real nor synthetic data. We also ask for an estimation of the perceived wind direction (left or right). Around $\sim$ 70% of users correctly identified the wind direction under which the sequence was generated.

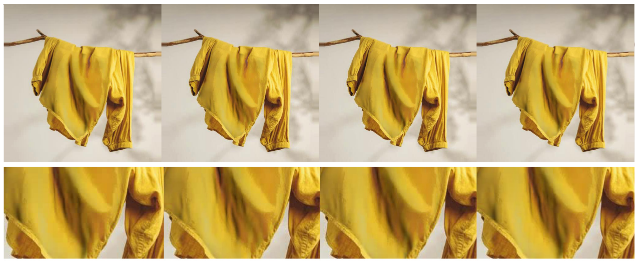


Figure 9. We further test the generalization capacity of our methodology by testing it with an image of a hanging garment. As observed, our approach can synthesize wrinkles on the cloth.

5. Conclusion

We introduced a method to generate human cinemagraphs from single RGB images. Our main contribution is a cyclic neural network that produces looping video clips. We demonstrated it is possible to train the network with synthetic data and generalize to real data. To do so, we propose working in the image normal space to close the gap between the different data distributions.

While generating plausible results, our method has some limitations. Intrinsic image decomposition which we use to synthesize back RGB images is a challenging problem and often lacks the high-frequency texture details of the original garments, which are lost during the re-shading step. In the future we would like to tailor a generic solution towards the type of fabric materials and textures that we are interested. Further, we would like to extend our setup to jointly optimize for normal estimation and motion prediction steps, in an end-to-end fashion. Finally, a challenging next step would be to add movement on hair strands that can add significant realism to the cinemagraphs. Since hair simulator is far from being a solved problem, it may be worth rethinking the setup to directly learn a neural (hair) simulator from real video footage, thus closely bringing together research in neural simulators and conditional generative models.

Acknowledgements. This work has been partially supported by the Spanish projects PID2019-105093GB-I00, TED2021-131317B-I00 and PDC2022-133305-I00. This work is partially supported by ICREA under the ICREA Academia programme. We also thank Aniruddha Mahapatra for his valuable assitance and guidance in the development of this work.

References

[1] Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. Panoramic video textures. In *ACM SIGGRAPH*, SIGGRAPH '05, page 821–827, 2005. 2

[2] Rajat Arora and Yong Jae Lee. Singan-gif: Learning a generative video model from a single gif. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1310–1319, 2021. 3

[3] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[4] Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. Selectively de-animating video. *ACM Trans. Graph.*, 31(4):66–1, 2012. 2

[5] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014. 4

[6] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: Clothed 3d humans. In *European Conference on Computer Vision (ECCV)*, page 344–359, 2020. 4, 5

[7] Kiran S. Bhat, Steven M. Seitz, Jessica K. Hodgins, and Pradeep K. Khosla. Flow-based video synthesis and editing. *Transactions on Graphics (TOG)*, 23(3):360–363, aug 2004. 2

[8] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis, 2021. 3

[9] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5171–5181, 2021. 3

[10] Katherine L. Bouman, Bei Xiao, Peter Battaglia, and William T. Freeman. Estimating the material properties of fabric from video. In *International Conference on Computer Vision (ICCV)*, pages 1984–1991, 2013. 2

[11] Chia-Chi Cheng, Hung-Yu Chen, and Wei-Chen Chiu. Time flies: Animating a still image with time-lapse video as reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5641–5650, 2020. 2

[12] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H. Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. *ACM Trans. Graph.*, 24(3):853–860, jul 2005. 2

[13] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[14] Vincent Couture and Sébastien Langer, Michael S. and Roy. Panoramic stereo video textures. In *International Conference on Computer Vision (ICCV)*, pages 1251–1258. IEEE, 2011. 2

[15] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G. Derpanis, and Björn Ommer. Stochastic image-to-video synthesis using cinns, 2021. 2

[16] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: Self-supervised learning of decoupled motion and appearance for single-image video synthesis. *Transactions on Graphics (TOG)*, 38(6), nov 2019. 2

[17] Siming Fan, Jingtan Piao, Chen Qian, Kwan-Yee Lin, and Hongsheng Li. Simulating fluids in real-world still images. *arXiv preprint arXiv:2204.11335*, 2022. 2

[18] Tavi Halperin, Hanit Hakim, Orestis Vantzos, Gershon Hochman, Netai Benaim, Lior Sassy, Michael Kupchik, Ofir Bibi, and Ohad Fried. Endless loops: Detecting and animating periodic patterns in still images. *Transactions on Graphics (TOG)*, 40(4), Aug. 2021. 2

[19] Aleksander Holynski, Brian L. Curless, Steven M. Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5810–5819, June 2021. 2

[20] Aleksander Holynski, Brian L Curless, Steven M Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5810–5819, 2021. 5

[21] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *European Conference on Computer Vision (ECCV)*, September 2018. 2

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[23] Jing Liao, Mark Finch, and Hugues Hoppe. Fast computation of seamless video loops. *Transactions on Graphics (TOG)*, 34(6), nov 2015. 2, 5

[24] Zicheng Liao, Neel Joshi, and Hugues Hoppe. Automated video looping with progressive dynamism. *Transactions on Graphics (TOG)*, 32(4), jul 2013. 2

[25] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5, 7, 8

[26] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *European Conference on Computer Vision (ECCV)*, August 2020. 2

[27] Aniruddha Mahapatra and Kuldeep Kulkarni. Controllable animation of fluid elements in still images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3667–3676, 2022. 2, 5, 6, 7, 8

[28] Makoto Okabe, Ken Anjyo, Takeo Igarashi, and Hans-Peter Seidel. Animating pictures of fluid using video examples. *Computer Graphics Forum*, 28(2):677–686, 2009. 2

[29] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[30] Ekta Prashnani, Maneli Noorkami, Daniel Vaquero, and Pradeep Sen. A phase-based approach for animating images using video examples. *Comput. Graph. Forum*, 36(6):303–311, sep 2017. 2

[31] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Computer Vision (ICCV), IEEE International Conference on*, 2019. 3

[32] Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa. Video textures. In *ACM SIGGRAPH*, pages 489–498, 2000. 2

[33] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019. 2

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 5

[35] James Tompkin, Fabrizio Pece, Kartic Subr, and Jan Kautz. Towards moment imagery: Automatic cinemagraphs. In *CVMP*, pages 87–93, 2011. 2

[36] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 5, 6

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 3

[38] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 5, 6, 7, 8

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[40] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[41] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 2

[42] Mei-Chen Yeh and Po-Yi Li. An approach to automatic creation of cinemagraphs. In *ACM MM*, page 1153–1156, New York, NY, USA, 2012. Association for Computing Machinery. 2

[43] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *European Conference on Computer Vision (ECCV)*, pages 300–315, 2020. 2

[44] Meng Zhang, Tuanfeng Wang, Duygu Ceylan, and Niloy J Mitra. Deep detail enhancement for any garment. In *Computer Graphics Forum*, volume 40, pages 399–411. Wiley Online Library, 2021. 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5