Multi-level Motion-Informed Approach for Video Generation with Key Frames

Zackary P. T. Sin(✉), Peter H. F. Ng, Simon C. K. Shiu, Fu-lai Chung, and Hong Va Leong

The Hong Kong Polytechnic University, Hung Hom, Hong Kong
{csptsin, cshfng, csckshiu, cskchung, cshleong}@comp.polyu.edu.hk

Abstract. Observing that a motion signal is decomposable into multiple levels, a video generation model which realizes this hypothesis is proposed. The model decomposes motion into a two-level signal involving a global path and local pattern. They are modeled via a latent path in the form of a composite Bezier spline along with a latent sine function respectively. In the application context, the model fills the research gap in its ability to connect an arbitrary number of input key frames smoothly. Experimental results indicate that the model improves in terms of the smoothness of the generated video. In addition, the ability of the model in separating global and local signal has been validated.

Keywords: Global motion path · Local motion pattern · Video generation with key frames · Latent path · Periodic latent function

1 Introduction

Motion could be modeled in different ways, among which optical flow is one well-known approach. More recently, Tulyakov et al. proposed using a string of motion codes to represent motion in the latent space [1]. In this paper, we observe that the motion signal could be decomposed and be represented as global and local signals. When a person moves, s/he moves from one place to another with a global trajectory, while exhibiting some repeating motion locally, such as arm and leg swinging. There may also be subtle movements for fingers and hair. To model this hierarchical motion structure, we propose to decompose the motion into a multi-level signal spanning from the top global level to the fine local level.

In this paper, we validate our motion decomposition approach with two levels: a global path signal and a local pattern signal. The former represents the motion that makes an object move in the environment while the latter represents the motion by a local part of the moving object. This is analogous to computer graphics concepts: the global path resembles the translation of an object while the local pattern its rotation.

In order to achieve a separation of global and local motion, some key problems need to be addressed. We model the properties in a latent space [1]. Since a global path is representing a global movement to model an object going from one place to another, an intuitive modeling could be a latent path drawn in the latent space. A local pattern

M. Gavrilova et al. (Eds.): CGI 2019, LNCS 11542, pp. 190–202, 2019.
https://doi.org/10.1007/978-3-030-22514-8_16

represents a repeating motion, which could intuitively be modeled with periodic functions in the latent space (e.g. sine function).

Modeling motion as a global path and local pattern opens up a new avenue for applications. For a controllable global signal, an intuitive application comes from the animation industry. It is well accepted that senior animators draw out some key frames as a rough video. Junior animators will fill out the frames in-between. Global path could be easily adapted to this application by ensuring that the latent path passes through the key frames specified by the user. Not only is this an intuitive method to control video generation, it also fills the research gap on using key frames for generating video. We currently adopt Bezier spline in the latent space.

With a controllable local motion signal, it would be ideal that the local motion could be tuned regardless of the global motion. An effect somewhat like moonwalking could be achieved, where a person's leg movement is seemingly detached with the person's movement. More research is needed to accurately replicate this with our current model. In summary, the contributions of this paper include the recognition of the motion signals being decomposable into multiple levels. We decompose the signal to facilitate automatic video frame generation in the latent space. We propose models for the global and local motions in the latent space and evaluate via experiments.

2 Literature Review

With the introduction of variational autoencoder (VAE) [2] and generative adversarial networks (GAN) [3], image and video generation problems have become robustly solvable. For example, VGAN adopts GAN to generate videos [4], which also implies that spatial and temporal dimensions have identical properties. TGAN was proposed to separately generate temporal codes and images from the said codes [5]. Since videos can be viewed as sequences of coherent images, they could be processed via recurrent neural networks (RNN) such as LSTM [6]. MoCoGAN is a good example [1].

There are also works in video generation. Mathieu et al. [7] worked on a loss function to improve the fidelity. Walker et al. [8] used human pose information to act as a higher-level abstraction for GAN. Liang et al. [9] and Liu et al. [10] adopted optical flow as additional feature for the generative model. Chan et al. [11] proposed a motion transfer method for human subjects with stick figure as an intermediate representation to enable a dancer's motion to be transferred to another person via video.

There seems to be relatively few works on video generation models controllable by multiple key frame inputs. VGAN, TGAN and MoCoGAN allow a video to be generated by a conditional image input. Motion codes in MoCoGAN could also be transferred from one video to another, but it is unclear how easy it is to control and get the motion codes. Wang et al. [12] proposed a video-to-video synthesis model that could translate a video from one domain to another. Although the result is impressive, it requires another video to generate a new video. Controllability of the video generation process via a few specific video key frames as in animation context remains lacking.

3 Methodology

The key idea of the proposed model is to separate motion into its global and local components. We will first discuss how the global path could be modelled, followed by the local pattern. With the global path, we can draw a smooth path in the latent space for generating a video. How this latent path could fit the input key frames will then be discussed. For the local pattern, we can use a periodic function to model.

3.1 Global Motion

For the global motion path, we have chosen a VAE framework [2]. A GAN-based solution does not quite work with the idea of latent path, perhaps due to the fact that GAN does not explicitly model the distribution of the latent space. On the other hand, VAE explicitly models the distribution of the latent space and hence, the distance between the points is meaningful. This allows us to easily apply Euclidean geometry techniques such as Bezier curves in the latent space. However, we make no claim that this is the reason why GAN fails in our experiments.

In order to model latent paths, we propose that the input key frames (images) x be projected first into the latent space via an encoder F_E such that $(\boldsymbol{z_c}, \boldsymbol{z_g^{(t)}}) = F_E(x^{(t)})$ where $\boldsymbol{z_c}$ is the content code, $\boldsymbol{z_g}$ is the global motion code [1] and t is the time step. Content code models the content in a video frame and therefore should be consistent throughout all the frames of a video while motion code models the motion in a video and therefore represents the changes between frames. $\boldsymbol{z_c}$ and $\boldsymbol{z_g}$ are both sampled from prior distributions P_{z_c} and P_{z_g}. Similar to [1], we make a distinction between content and motion by fixing $\boldsymbol{z_c}$ for generating all frames (by picking a $\boldsymbol{z_c}$ from one of the encoded key frames). Then the path could be drawn such that it passes through the latent space projection $\boldsymbol{z_g} = \{..., \boldsymbol{z_g^{(t)}}, ...\}$ of all input key frames. Let us first consider the simplistic case where only the starting and ending frames are the inputs such that $\boldsymbol{z_g} = \{\boldsymbol{z_g^{(0)}}, \boldsymbol{z_g^{(T-1)}}\}$ (Fig. 1), where T is the total number of frames of the to-be-generated video. The latent path will simply be a line which samples the in-between global motion code $\widehat{z_g^{(t)}}$ such that $\widehat{z_g^{(t)}} = z_g^{(0)}\left(1 - \frac{t}{T-1}\right) + z_g^{(T-1)}\left(\frac{t}{T-1}\right)$. Each video frame could then be constructed by decoding the consecutive global motion code $\widehat{z_g} = \left\{\widehat{z_g^{(0)}}, \ldots, \widehat{z_g^{(t)}}, \ldots, \widehat{z_g^{(T-1)}}\right\}$ with the decoder F_D such that the video $v = F_D(z_c, \widehat{z_g})$.

For a three key frames scenario (Fig. 1), a quadratic Bezier curve in the latent space will be required. It is worth pointing out that although the first and last control points of the Bezier curve, $\boldsymbol{c^{(0)}}$, $\boldsymbol{c^{(2)}}$, lie on the curve, the middle one does not. Hence, the second control point $\boldsymbol{c^{(1)}}$ needs to be computed for the curve to pass through the second key frame $x^{(M_k)}$, where M_k is the time step of the second key frame. This second control point $\boldsymbol{c}^{(\mathbf{1})}$ is computed as in Eq. (1).

$$\boldsymbol{c}^{(1)} = -\frac{\boldsymbol{c}^{(0)}\left(1-\frac{M_k}{T-1}\right)^2+\boldsymbol{c}^{(2)}\left(\frac{M_k}{T-1}\right)^2-z^{(M_k)}}{2\left(1-\frac{M_k}{T-1}\right)\left(\frac{M_k}{T-1}\right)} \tag{1}$$

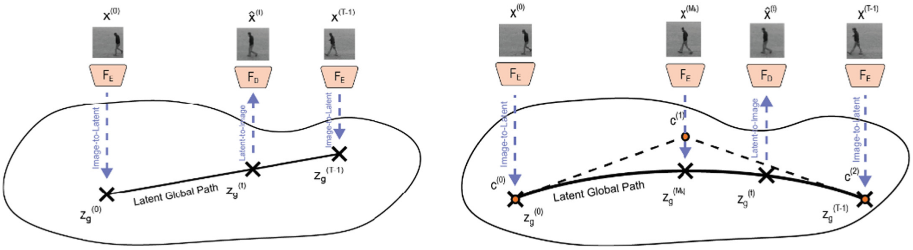

Fig. 1. Illustration of latent paths. The left and right are the latent paths drawn given two and three conditional key frames input respectively.

3.2 Local Motion Pattern

To model the local motion pattern, we propose using a periodic function in the latent space. Here, we have chosen the sine function. Although we will show that using a sine function is possible, it is not robust and requires a specific design in order to work. So, we will show here an architecture which we have found to be usable. That is, it is able to facilitate in the separation of the global and local motions.

Ideal Modeling. Since the original purpose of the periodic function in the latent space is to model the repeating local motion, ideally the local motion codes sampled from this function should be repeated every specific time interval. Therefore, the function should be temporal-based such that the latent local motion $\boldsymbol{z_l^{(t)}} = \boldsymbol{sin}\left(\frac{\boldsymbol{t}}{\boldsymbol{T}}\right)$. Using only one sine curve is too simplistic for modeling. Inspired by the Fourier transformation, we instead propose that there should be multiple sine curves. Internally, the neural network could be expected to combine the output of multiple sine curves (a sine curve for each dimension) to model more complex periodic functions.

In our first attempt, we allow the model to choose the amplitude, frequency and phase shift as in Eq. (2). Specifically for our case, ($\boldsymbol{z_c}$, $\boldsymbol{z_g^{(t)}}$, $\boldsymbol{z_s}$)= $F_E(\boldsymbol{x^{(t)}})$, where $\boldsymbol{z_s}$ is sampled from P_{z_s}. Our encoder now generates codes for the local motion, as shown in Eq. (2):

$$\boldsymbol{z_l^{(t)}} = \boldsymbol{z_a} sin\left(\boldsymbol{z_b}\left(\frac{t}{T}\right)+\boldsymbol{z_u}\right) \tag{2}$$

where $\boldsymbol{z_s}$={$\boldsymbol{z_a}$, $\boldsymbol{z_b}$, $\boldsymbol{z_u}$} and $sin(\cdot)$ is an element-wise sine function. The reason why the sine curve parameters $\boldsymbol{z_s}$ do not have a temporal component (t) is that the sine curve should remain the same throughout the entire video (i.e. picking a $\boldsymbol{z_s}$ from one of the encoded key frames, similar to how we treat z_c). Regardless, our experiments show that this ideal approach does not quite work as the model simply disregards the contribution from the local motion model. We suspect that the periodic nature of the sine curve could have led to local minima which could have trapped the search for the optimal solution during the training process.

Modeling with Forced Step Observing the deficiency of the ideal model with a number of free parameters, we would like to impose additional constraints, in our alternative model known as "forced steps" (see Fig. 2). This model tries to predict how far in a phase this time step should move with an RNN such that:

$$\boldsymbol{s}^{(t)} = \sigma\left(\mathbf{z}_{\boldsymbol{u}}^{(t)}\right) + \boldsymbol{s}^{(t-1)} \tag{3}$$

$$z_{\boldsymbol{l}}^{(t)} = z_{\boldsymbol{a}} sin\left(\boldsymbol{s}^{(t)}\right) \tag{4}$$

where z_s={$z_{\boldsymbol{a}}$, $z_{\boldsymbol{u}}^{(t)}$} and $\sigma(\cdot)$ is a sigmoid function. Here, the frequency component is dropped. This is because if the model controls which phase is to be sampled for each time step (Fig. 2), effectively it is also controlling the frequency of the curve. We found that this model is capable of separating the global and local motions which we will show later. This solution is based on the assumption that the ideal model experiences difficulties crossing the local minima induced by the sine curves. Instead, we encourage the model to move across the local minima with forced phase steps.

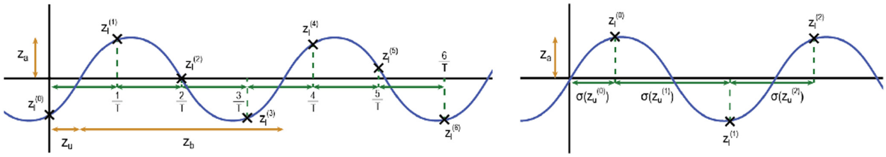

Fig. 2. The left is the ideal model while the right is the forced step model. In contrast to the ideal model which has fixed steps, the forced step model generates a step for each time step.

3.3 Latent Path

One of the key goals of this paper is to utilize a latent path to generate a smooth video given an arbitrary number of input key frames. It is possible that the latent path is immediately drawn based on the number of input key frames initially. For example, given five key frames we draw a quartic Bezier curve. However, this limits the flexibility of the latent path as the number of inputs needs to be known prior. Instead, it is proposed that it is better to use Bezier spline to draw the path. To utilize Bezier spline, we propose two strategies, extension and connection.

Extension. Given a path segment p_A and an additional key frame, we need to smoothly extend the latent path such that it could cross the latent representation of the newly added input key frame image $z_{\boldsymbol{g}}^{(\boldsymbol{T_A}+\boldsymbol{T_B}-1)}$, where T_B is the number of frames of the extended path p_B (Fig. 3). In order for the new composite path to be smooth throughout, there must not be sudden changes in the tangents of the path. Obviously, the point that would cause problem in this case is where the path is to extend. To ensure a smooth extension from that point, it is proposed that the extended path p_B be a quadratic Bezier curve and its second control point $\boldsymbol{c}_{\boldsymbol{B}}^{(\mathbf{1})}$ be computed as follows:

$$\boldsymbol{c}_{\boldsymbol{B}}^{(1)} = \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} + \frac{\boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-2\right)}}{\left|\boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-2\right)}\right|} \cdot \frac{\left|\boldsymbol{z}_{\boldsymbol{g}}^{T_A+T_B-1} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)}\right|}{2} \tag{5}$$

where M_{cA} is the number of control points of path segment p_A. Note that since p_A and p_B meet at $\boldsymbol{z}_{\boldsymbol{g}}^{(T_A-1)}$, $\boldsymbol{c}_{\boldsymbol{B}}^{(0)} = \boldsymbol{c}_{\boldsymbol{A}}^{(M_c-1)}$.

Connection. Given two path segments p_A and p_C, we need to smoothly connect the two latent paths by drawing an in-between path p_B. Similar to extension, we need to consider the tangents. However this time, we need to consider tangents at the end of p_A and the start of p_C. We will connect the two latent paths with a cubic Bezier curve (Fig. 3). Its second and third control points could be computed similar to extension as:

$$\boldsymbol{c}_{\boldsymbol{B}}^{(1)} = \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} + \frac{\boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-2\right)}}{\left|\boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-1\right)} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_A}-2\right)}\right|} \cdot \frac{\left|\boldsymbol{c}_{\boldsymbol{A}}^{(0)} - \boldsymbol{c}_{\boldsymbol{C}}^{(M_{cC}-1)}\right|}{2} \tag{6}$$

$$\boldsymbol{c}_{\boldsymbol{B}}^{(2)} = \boldsymbol{c}_{\boldsymbol{C}}^{(0)} + \frac{\boldsymbol{c}_{\boldsymbol{C}}^{(1)} - \boldsymbol{c}_{\boldsymbol{C}}^{(0)}}{\left|\boldsymbol{c}_{\boldsymbol{C}}^{(1)} - \boldsymbol{c}_{\boldsymbol{C}}^{(0)}\right|} \cdot \frac{\left|\boldsymbol{c}_{\boldsymbol{C}}^{(0)} - \boldsymbol{c}_{\boldsymbol{A}}^{\left(M_{c_C}-1\right)}\right|}{2} \tag{7}$$

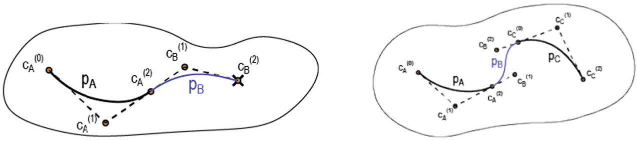

Fig. 3. The extensions (left) and connections (right) of latent paths could be achieved by drawing a new path p_B with control points without causing a sudden change at the ends of the paths.

3.4 Proposed Model

We combine all the methods described above to complete our proposed model. In our experimental prototype, the global motion code z_g is computed as in Sect. 3.1. The local motion code z_l is computed as in Sect. 3.2. When we attempt to generate a video, we adopt the strategies in Sect. 3.3. Video frames are generated by the decoder from latent inputs such that a video frame $v^{(t)} = F_D(z_c, \widehat{z_g^{(t)}}, z_l^{(t)})$. The training objective of our proposed VAE model with parameters θ is as follow:

$$J_\theta = \sum_{m=0}^{M_c-1} D_{KL}\left[q_\theta\left(z^{\left(k^{(m)}\right)}|k^{(m)}\right)||p_z(z)\right]) \cdot \lambda_{latent} \\ + E_{q_\theta(z^{(k)}|k)}\left[\sum_{t=0}^{T-1} L_2\left(F_{D_\theta}\left(R\left(z^{(k)}, \frac{t}{T-1}\right)\right), x\right)\right] \cdot \lambda_{rec} \tag{8}$$

where latent vector $z^{(t)}=\{z_c, z_g^{(t)}, z_l^{(t)}\}$, k is the set of key frames, $k^{(m)}$ is the m^{th} key frame, R is a latent function that computes the latent code for each time step and λs are hyperparameters. The former term is the latent loss while the latter term is the reconstruction loss. To obtain $z_u^{(t)}$ for each time step, a string of GRU [13] cells as an RNN is used with the initial state $h_0=\{z_c, z_u^{(0)}\}$. Each GRU cell at time step t will be fed with the tangent of the latent path $z_{\Delta g}^{(t)}$ and the previous prediction $z_u^{(t-1)}$. An illustration of the model is shown in Fig. 4.

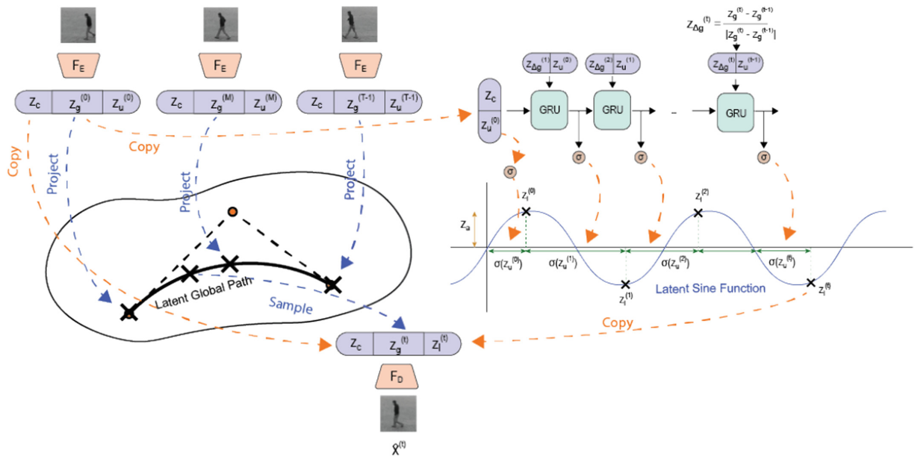

Fig. 4. The proposed model separates the global and local signals via a latent path and a latent sine curve. An encoder-decoder (F_E and F_D) pair is used to project an image to and from the latent space. A string of GRU cells is used to predict $z_u^{(t)}$ for each time step t.

4 Experiments

As our proposed model involves a global motion and a local motion component, we intend to evaluate them separately. The proposed model will be evaluated and compared with MoCoGAN [1] and VGAN [4]. Both MoCoGAN and VGAN models used spatio-temporal convolutional networks as discriminators. However, since the video generated does not have a fixed length here, we use a discriminator with spatial convolutional layers and GRU-based RNN for the temporal connection instead. Here, we also use two datasets in the evaluation, the walking videos in KTH action database [14] and the Weizmann Action database [15]. The KTH walking dataset consists of 75 videos with 25 people walking from left to right and right to left. Given a video length of 15, 11284 samples can be extracted from the dataset. The Weizmann Action dataset consists of 72 videos of 9 people performing 4 actions such as hand-waving and crouching. Given a video length of 15, 2069 samples can be extracted.

4.1 Three Conditional Images

We first evaluate the model given three conditional images, i.e. three input key frames. When evaluating, one of the questions we want to answer is whether the proposed model is able to generate a smooth video given arbitrary number of input frames. This is a question of interest as though models such as MoCoGAN or VGAN could only accept a fixed number of input images, we can still use them to generate video segments given two input key frames then merge all video segments together into a single video. However, it is hypothesized that since the models generate each segment independently, the transition before and after key frames will not be smooth. Thus, it is expected that the proposed model would produce an improved performance over models that do not consider arbitrary number of inputs.

Quantitatively, we make use of four metrics to evaluate the performance of our model. Average content distance (ACD) and mean-square error (MSE) are two common metrics. Nevertheless, since they are more useful in evaluating the content similarity between frames while we are more interested in evaluating the smoothness between frames, we propose two metrics, first-order optical flow distance (OFD) and angular-sensitive smoothness distance (ASD) to evaluate the smoothness of a video.

The OFD metric is defined as in Eq. (9):

$$OFD\left(\boldsymbol{f}^{(t)}\right) = \frac{1}{N_I}\sum\nolimits_i^{N_I} \left| f_i^{(t+1)} - f_i^{(t-1)} \right| \tag{9}$$

where $\boldsymbol{f}$ is the optical flow map, N_I the number of pixels and i the index of a pixel. OFD approximates the acceleration of pixel movements. The expectation is that this metric could detect sudden movement which may be perceptually viewed as unsmooth. However, the OFD leans towards measuring the change in speed. Two flow changes with the same magnitude, but different directions will be measured similarly. To make the direction carry a stronger influence on the metric, the ASD metric is introduced as in Eq. (10):

$$ASD\left(\boldsymbol{f}^{(t)}\right) = \frac{1}{N_I}\sum\nolimits_i^{N_I} \left(\frac{\left| f_{\theta_i}^{(t)} - f_{\theta_i}^{(t-1)} \right|}{\pi} + 1 \right) \left| f_{mi}^{(t)} - f_{mi}^{(t-1)} \right| \tag{10}$$

where $\boldsymbol{f_\theta}$ is the angles on a flow map and $\boldsymbol{f_m}$ is the magnitudes on a flow map.

To evaluate whether the content is consistent, we followed [1] in adapting the ACD metric as in Eq. (11):

$$ACD\left(\boldsymbol{I}^{(t)}\right) = \sqrt{\sum\nolimits_c \left(\frac{1}{N_I}\left(\sum\nolimits_i^{N_I} I_{i,c}^{(t)} - \sum\nolimits_i^{N_I} I_{i,c}^{(t-1)}\right)\right)^2} \tag{11}$$

where $\boldsymbol{I}$ is a video frame and c is the number of color channels. The MSE metric is similar to ACD. It compares each frame to the previous frame as in Eq. (12).

$$MSE\left(\boldsymbol{I}^{(t)}\right) = \frac{1}{N_I}\sum\nolimits_i^{N_I}\left(I_{i,c}^{(t)} - I_{i,c}^{(t-1)}\right)^2 \tag{12}$$

In our evaluation, we give the proposed model three key frames and use it to generate a video. We evaluate the smoothness of the video by using OFD and ASD to check whether the frames before and after key frames are smooth transitions. Given each key frame with time step t_k, we use OFD at time step t_k and ASD at t_k and t_k+ 1. To verify the content consistency, we adopt ACD and MSE for each frame consecutively. As shown in Tables 1 and 2, the proposed model has achieved its intended effect on generating a video, that is, smoother, since it generally achieves a better OFD and ASD. However, it does not achieve a better score in terms of content consistency (ACD and MSE) in general. This outcome is expected as GAN architectures generally perform better than VAE in terms of content fidelity. Qualitatively, we can also see that the proposed model could maintain a better smoothness before and after the key frames as shown in Fig. 5.

Table 1. Model scores when conditioned on 3 key frames from the KTH walking dataset.

	OFD	ASD	ACD	MSE
Proposed model	0.03366	0.03385	0.0008766	0.008037
VGAN	0.05214	0.04689	0.0005443	0.007557
MoCoGAN	0.04096	0.03769	0.0006116	0.010060

Table 2. Model scores when conditioned on 3 key frames from the Weizmann dataset.

	OFD	ASD	ACD	MSE
Proposed model	0.01359	0.01629	0.00004677	0.001511
VGAN	0.01599	0.01558	0.00014130	0.001895
MoCoGAN	0.01604	0.01722	0.00011010	0.001661

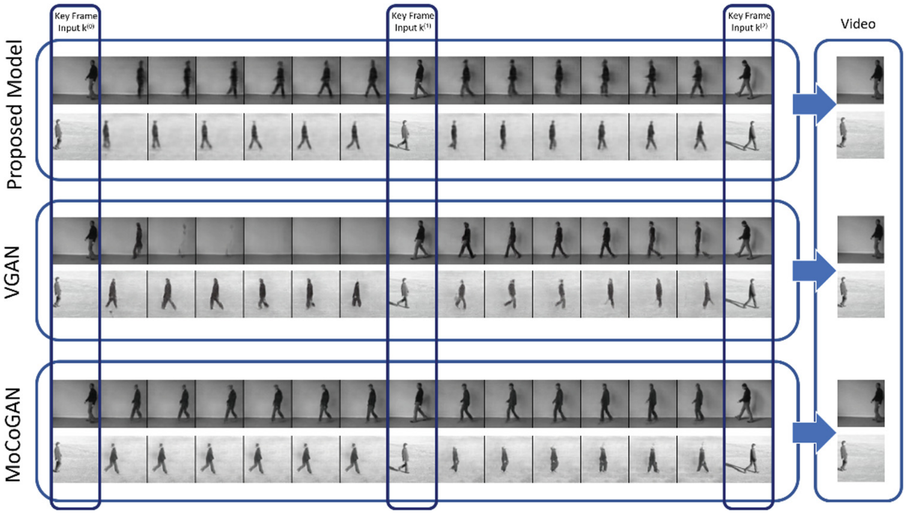
Key Frame Input $k^{(0)}$
Key Frame Input $k^{(1)}$
Key Frame Input $k^{(2)}$
Video
Proposed Model
VGAN
MoCoGAN

Fig. 5. Results of the three video generation models conditioned on three key frames. Each set contains two videos with the first, eighth and fifteenth frames as the key frames. From top to bottom, the sets represent videos generated by the proposed model, VGAN and MoCoGAN respectively. Readers can visit https://youtu.be/cj-HsAro_Zk for the video demonstration.

4.2 Extending and Connecting Latent Paths

Next, we evaluate the model with different number of input frames, two, three, four and five. For the four and five input key frames, we need to use the latent path strategies mentioned in Sect. 3.3. The results on extending and connecting the latent path are shown in Fig. 6.

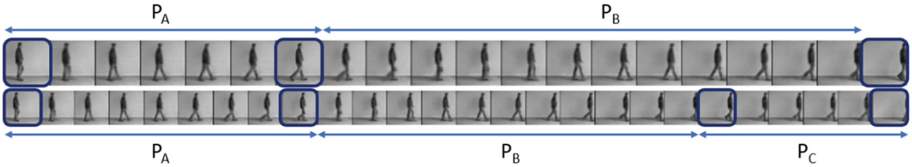

Fig. 6. Results of extending a latent path to pass through a key frame in the latent space (left) and connecting two latent paths by drawing an in-between latent path (right). The dark blue circle indicates the key frame inputs. (Color figure online)

As shown in Fig. 7, the proposed model is generally able to achieve a better smoothness given different number of input key frames. A noticeable exception is when the model is generating with only two key frames. This outcome is expected as the global path modeling only has an advantage when the number of input frames is more than two. This result shows that the latent path is a viable solution to the problem with an arbitrary number of input frames. There seems to be no trend between the number of inputs and the ASD. However, there seems to be a subtle negative correlation between the number of inputs and the OFD. We suspect that the reason is that given more conditional key frames, the in-between frames generated will become more constrained and therefore it will be more likely for the model to be able to generate video frames that could be coherent with all others.

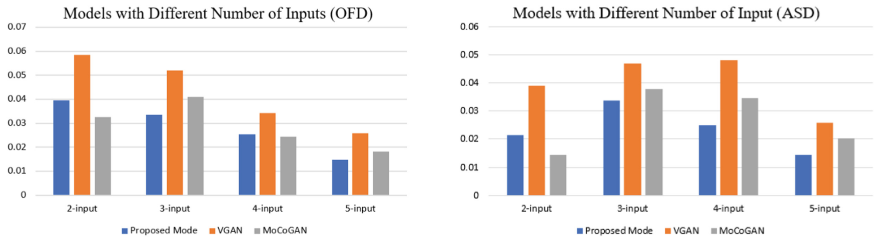

Fig. 7. The results when given different number of input key frames for the different models. The left is evaluated in OFD while the right is evaluated in ASD.

4.3 Separation of Global and Local Motion

To visualize the separation of the global and local motion, we have devised an experiment that manipulates the global and local latent space separately. Given a string of latent vector $\boldsymbol{z}^{(t)}$=$\{\boldsymbol{z_c}, \boldsymbol{z_g^{(t)}}, \boldsymbol{z_l^{(t)}}\}$, we generate two results where one locks $\boldsymbol{z_g^{(t)}}$ such that $v^{(t)}= F_D(z_c, \widehat{z_g^{(0)}}, z_l^{(t)})$, and the other locks $\boldsymbol{z_l^{(t)}}$ such that $v^{(t)}= F_D(z_c, \widehat{z_g^{(t)}}, z_l^{(0)})$. The results are depicted in Fig. 8. It is shown that our model is able to achieve a separation of the global and local motions. When $\boldsymbol{z_l}$ is locked, the local motion is missing while when $\boldsymbol{z_g}$ is locked, the global motion is missing.

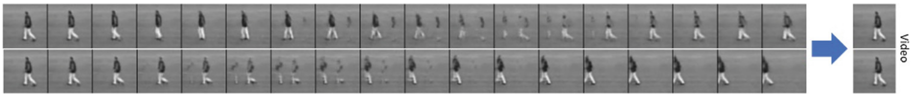

Fig. 8. Video generated with global motion (above) and local motion (below) locked respectively. It can be seen that when global motion is locked, the model tries to keep the person in the same place while still generating the local motion. Readers can visit https://youtu.be/PYGC0jMa9vw for the video demonstration.

We also conduct an experiment to see if the local motion could be tuned to create special effects like a person moonwalking. To achieve that, we multiply the value with $\sigma\left(\boldsymbol{z_u^{(t)}}\right)$ produced by the local motion component described in Sect. 3.2. Effectively, we have scaled the step to make it go further or vice versa. In Fig. 9, we multiply it by 1.5 and 0.5 to make it go faster and slower respectively. The results show that although the person indeed moves faster and slower, there are strong artefacts. To fully utilize local motion tuning for application, further work is needed.

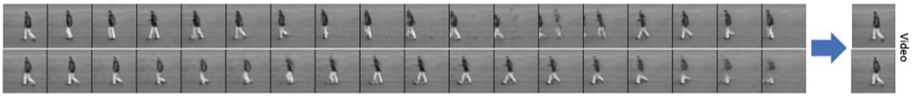

Fig. 9. The result of scaling with $\sigma\left(\mathbf{z}_u^{(t)}\right)$. The outcome is a video where the person walks faster or slower. The model tries to keep the person in the same global position regardless of local motion, but ultimately, the artefact is clearly visible. Readers can visit https://youtu.be/PYGC0jMa9vw for the video demonstration.

5 Conclusion and Future Work

Inspired by the observation that a motion signal could be decomposed into multi-level signals, we proposed a model to materialize our observation and conducted a feasibility study of the idea via decomposing a motion into a global path and a local pattern. We suggested that the global path could be represented as a latent Bezier spline while the local pattern could be represented as a latent sine function. To evaluate the smoothness of the generated video, two measurement metrics were also proposed. Our experiments showed that our proposed model was capable of generating a video given some conditional key frames with general improvement on smoothness. One of the experiments also demonstrated the decomposition of the global and local motion.

In the future, we will expand the number of motion levels to completely model the multi-level motion in representing a rich continuum that spans from the very global motion to the very local motion. This means that a hierarchy of motions could be formed to increase the controllability of the video generation. With human walking as an example, we could control from the movement of the whole body, to the movement of the legs, then to the ankles and further down to the feet, and so on to even more local parts. Instead of just a global motion parented to a local motion, each level of motion would be parented to another motion one level down in general.

References

1. Tulyakov, S., Liu, M.-Y., Yang, X., Kautz, J.: MoCoGAN: decomposing motion and content for video generation. In: CVPR Workshop (2017)
2. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of ICLR (2013)
3. Goodfellow, I.J., et al.: Generative adversarial nets. In: Proceedings of NIPS (2014)
4. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Proceedings of NIPS (2016)
5. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: Proceedings of ICCV (2017)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: Proceedings of ICLR (2016)

8. Walker, J., Marino, K., Gupta, A., Hebert, M.: Video forecasting by generating pose futures. In: Proceedings of ICCV (2017)
9. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion GAN for future-flow embedded video prediction. In: Proceedings of ICCV (2017)
10. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: Proceedings of ICCV (2017)
11. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: ECCV Workshop (2018)
12. Wang, T.-C., et al.: Video-to-video synthesis. In: Proceedings of NIPS (2018)
13. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of International Conference on Empirical Methods in NLP (2014)
14. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of International Conference on Pattern Recognition (2004)
15. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proceedings of ICCV (2005)