Generating Synthetic Video Sequences by Explicitly Modeling Object Motion

S. Palazzo*, C. Spampinato*,§, P. D'Oro*, D. Giordano*, M. Shah§

* Pattern Recognition and Computer Vision (PeRCeiVe) Lab
University of Catania, Italy, www.perceivelab.com
§ Center for Research in Computer Vision
University of Central Florida, USA, http://crcv.ucf.edu

Abstract. Recent GAN-based video generation approaches model videos as the combination of a time-independent scene component and a time-varying motion component, thus factorizing the generation problem into generating background and foreground separately. One of the main limitations of current approaches is that both factors are learned by mapping one source latent space to videos, which complicates the generation task as a single data point must be informative of both background and foreground content. In this paper we propose a GAN framework for video generation that, instead, employs two latent spaces in order to structure the generative process in a more natural way: 1) a latent space to generate the static visual content of a scene (background), which remains the same for the whole video, and 2) a latent space where motion is encoded as a trajectory between sampled points and whose dynamics are modeled through an RNN encoder (jointly trained with the generator and the discriminator) and then mapped by the generator to visual objects' motion. Performance evaluation showed that our approach is able to control effectively the generation process as well as to synthesize more realistic videos than state-of-the-art methods.

1 Introduction

Generative Adversarial Networks (GANs) [1] are a recent trend in computer vision and machine learning that advanced the state of the art on image and video generation to unprecedented levels of accuracy and realism. New adversarial models [2–8] are proposed at an accelerating pace, both to increase the diversity and resolution of generated images and to tackle theoretical issues on training and convergence. GANs have been applied mainly to image generation, and naively extending image generation methods to videos is not sufficient, as it jointly attempts at handling both the spatial component of the video, describing object and background appearance, and the temporal one, representing object motion and consistency across frames. Building on these considerations, recent generative efforts [9, 10] have attempted to factor the latent representation of each video frame into two components that model a time-independent background of the scene and the time-varying foreground elements. We argue that the main limitation of these methods is that both factors are learned by mapping a single point of a source latent space (sampled as random noise) to a whole video. This, indeed, overcomplicates the generation task as two videos depicting the same scene with different

object trajectories or the same trajectory on different scenes are represented as different points in the latent space, although they share a common factor (in the former case the background, in the latter case object motion). To address this limitation, in this paper we propose a GAN-based generation approach that employs two latent spaces (as shown in Fig. 1) to improve the video generation process: 1) one latent space to model the static visual content of the scene (background), and 2) a foreground latent space to learn object motion dynamics. In particular, these dynamics are modeled as point trajectories in the second latent space, with each point representing the foreground content in a scene and each latent trajectory ensuring regularity and realism of the generated motion across frames. Variations in the scene latent space result in different scenes, while variations in the trajectories of the foreground latent space result in different object motion. We demonstrate the effectiveness of the proposed approach by extensively evaluating the realism of the generated videos and compared the videos generated by state of the art methods [9, 10], which, conversely to our method, learn a mapping between a single latent space and video data distribution instead of learning to generate specific motion and eventually object behaviour.

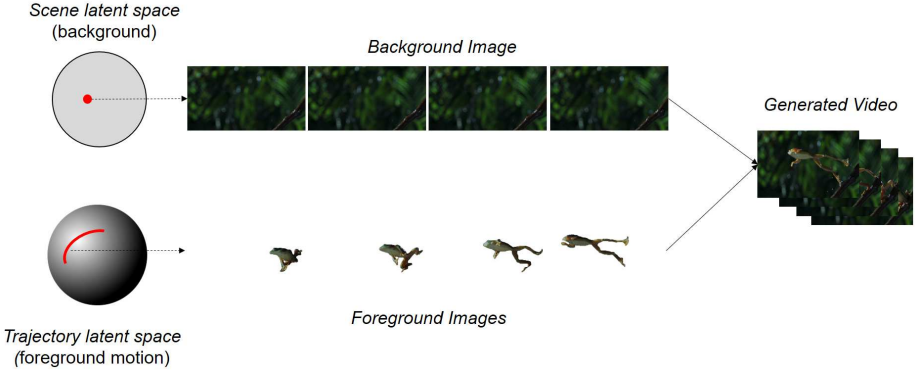

Fig. 1: **Video Generation in VOS-GAN**: we employ a scene latent space to generate background and a foreground latent space to generate object appearance and motion.

2 Video Generation Model

The video generation architecture presented in this work is based on a GAN framework consisting of the following two modules:

– a *generator*, implemented as a hybrid deep CNN-RNN, that receives two kinds of input: 1) a noise vector from a latent space that models scene background; 2) a sequence of vectors that model foreground motion as a trajectory in another latent space. The output of the generator is a video with its corresponding foreground mask.

– a *discriminator*, implemented as a deep CNN, that receives an input video and predicts whether it is real or not.

The architecture of the generator, inspired by the two-stream approach in [9], is shown in Fig. 2. Specifically, our generation approach factorizes the process into separate background and foreground generation, on the assumption that the world is generally stationary and the presence of informative motion can be constrained only to a set of objects of interest in a semi-static scenery. However, unlike [9], we separate the latent spaces for scene and foreground generation, and explicitly represent the latter as a temporal quantity, thus enforcing a more natural correspondence between the latent input and the frame-by-frame motion output.

Hence, the generator receives two inputs: $z_C \in \mathcal{Z}_C = \mathbb{R}^d$ and $z_M = \{z_{M,i}\}_{i=1}^t$, with each $z_{M,i} \in \mathcal{Z}_M = \mathbb{R}^d$. A point z_C in the latent space $\mathcal{Z}_C$ encodes the general scene to be applied to the output video, and is mainly responsible for driving the *background stream* of the model. This stream consists of a cascade of transposed convolutions, which gradually increase the spatial dimension of the input in order to obtain a full-scale background image $b(z_C)$ that is used for all frames in the generated video.

The set of $z_{M,i}$ points from the latent space $\mathcal{Z}_M$ defines the objects motion to be applied in the video. The latent sequence is obtained by sampling the initial and final points and performing a spherical linear interpolation (SLERP [11]) to compute all intermediate vectors, such that the length of the sequence is equal to the length (in frames) of the generated video. Using an interpolation rather than sampling multiple random points should enforce temporal coherency between appearances in the generated foreground. The list of latent points is then encoded through a recurrent neural network (LSTM) in order to provide a single vector (i.e., the LSTM's final state) summarizing a representation of the whole motion. The input to the *foreground stream* is then a concatenation of the vector coming out of the LSTM and z_C, so that the generated motion can take into account the scene to which it will be applied. After a cascade of spatio-temporal convolutions (i.e., with 3D kernels that also span the time dimension), the foreground stream provides a set of frames $f(z_C, z_M)$ with foreground content and binary masks $m(z_C, z_M)$ defining motion pixel location.

The two streams are finally combined as

$$G(z_C, z_M) = m(z_C, z_M) \odot f(z_C, z_M) + (1 - m(z_C, z_M)) \odot b(z_C) \qquad (1)$$

Foreground generation can be directly controlled acting on z_M. Indeed, varying z_M for a fixed value of z_C results in videos with the same background and different foreground appearance and motion. Thus, z_C can be seen as a condition for the foreground stream, in a similar way to conditional generative adversarial networks for restricting generation process to a specific class.

The primary goal of the discriminator network is to distinguish between generated and real videos, in order to push the generator towards more realistic outputs. The architecture of our discriminator follows a standard architecture for video discrimination [9]. The input to the model is a video clip (either real or produced by the generator), that goes first through a series of convolutional layers, encoding the video dynamics in a more compact representation, which is provided to a *discrimination stream* (bottom),

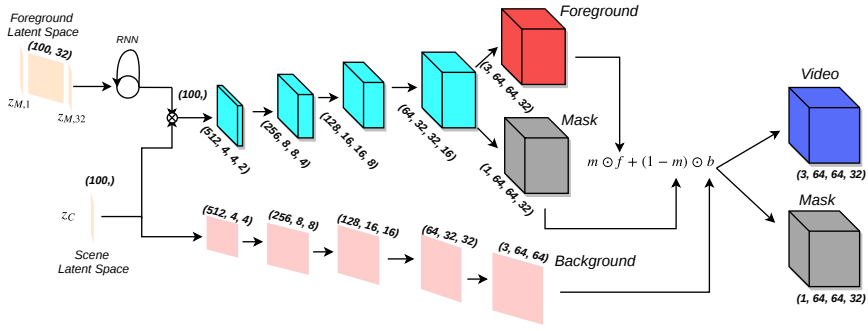

Fig. 2: **Generator architecture:** the *background stream* (bottom) is conditioned by a latent vector defining the general scene of the video, and produces a background image; the *foreground stream* (top) processes a sequence of latent vectors, obtained by spherically interpolating the start and end points, and the scene latent vector to generate frame-by-frame foreground appearance and motion masks. Information about dimensions of intermediate outputs is given in the figure by (*channels, height, width, duration*) tuples.

which applies a 3D convolution to the intermediate representation and then makes a prediction on whether the input video is real or fake.

We jointly train the generator and the discriminator in a GAN framework, with the former trying to maximize the probability that the discriminator predict fake outputs as real, and the latter trying to minimize the same probability.

The discriminator loss is then defined as follows (for the sake of compactness, we will define $z = (z_C, z_M)$):

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{real}}} \left[\log D_{\text{adv}}\left(x\right)\right] - \mathbb{E}_{x \sim p_z} \left[\log\left(1 - D_{\text{adv}}\left(G\left(z\right)\right)\right)\right] \tag{2}$$

In the equation above, the first line encodes the adversarial loss, which pushes the discriminator to return high likelihood scores for real videos and low ones for the generated videos.

The generator loss is, more traditionally, defined as:

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z} \left[\log D_{\text{adv}}\left(G\left(z\right)\right)\right] \tag{3}$$

In this case, the generator tries to push the discriminator to increase the likelihood of its output being real.

During training, we follow the common approach for GAN training, by sampling real videos (from an existing dataset) and generated videos (from the generator) and alternately optimizing the discriminator and the generator.

3 Performance Analysis

Our video generation model was trained on the "golf course" videos (over 600,000 videoclips) of the dataset proposed in [9]. For testing the video generation capabilities

we performed quantitative evaluation. In particular, we evaluated separately the quality of generated background, foreground, and motion using the following metrics:

- **Foreground Content Distance (FCD)**. This score aims at assessing the consistency between visual appearance of foreground objects in consecutive figures and is measured by computing the average L2 distance between visual features, extracted from a fully-connected layer of a pre-trained Inception network [14], of foreground objects in two consecutive figures. The input to the Inception model is the bounding box containing the foreground region, defined as the discriminator's segmentation output.
- **Motion coherency (MC)**. While the previous score describes the quality of the generated visual appearance of moving objects, this one aims at evaluating how realistic the generated motion is, and is computed as the KL-divergence between magnitude/orientation histograms of optical flows of real and generated videos.
- **Inception score (IS)** [15] is the most adopted metric in GAN literature. In our case, we compute the Inception score by sampling a random frame from each video of a pool of generated ones.

During GAN training, we performed gradient-descent using ADAM, with an initial learning rate of 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and batch size of 16 for 25 epochs.

FCS, MC and IS scores were computed on a set of 50,000 videos generated by the compared models trained on "golf course" [9], and on the same number of random real videos as a baseline.The results in Tab. 1 shows that our approach significantly outperformed VGAN and TGAN on the three metrics, achieving closer values to those yielded by real videos, indicating a higher realism in scene appearance and object motion. Samples of generated videos on for VGAN, TGAN and our method are shown in Fig. 3.

	FCD	MC	IS
VGAN [9]	10.61	0.017	1.74
TGAN [10]	3.74	0.011	2.02
Our approach	**4.80**	**0.002**	**2.90**
Real videos	4.59	0.0001	4.59

Table 1: Quantitative evaluation of video generation capabilities measured by foreground content distance (FCD), motion coherency (MC) and Inception Score (IS).

4 Conclusion

We propose a novel GAN-based video generation approach that employs two input latent spaces: one for modeling the background, and one to model foreground motion and appearance. Extensive experimental evaluation showed that our VOS-GAN outperforms significantly existing GAN-based methods, VGAN [9] and TGAN [10], on the video generation process, by creating videos with more realistic motion measured quantitatively.

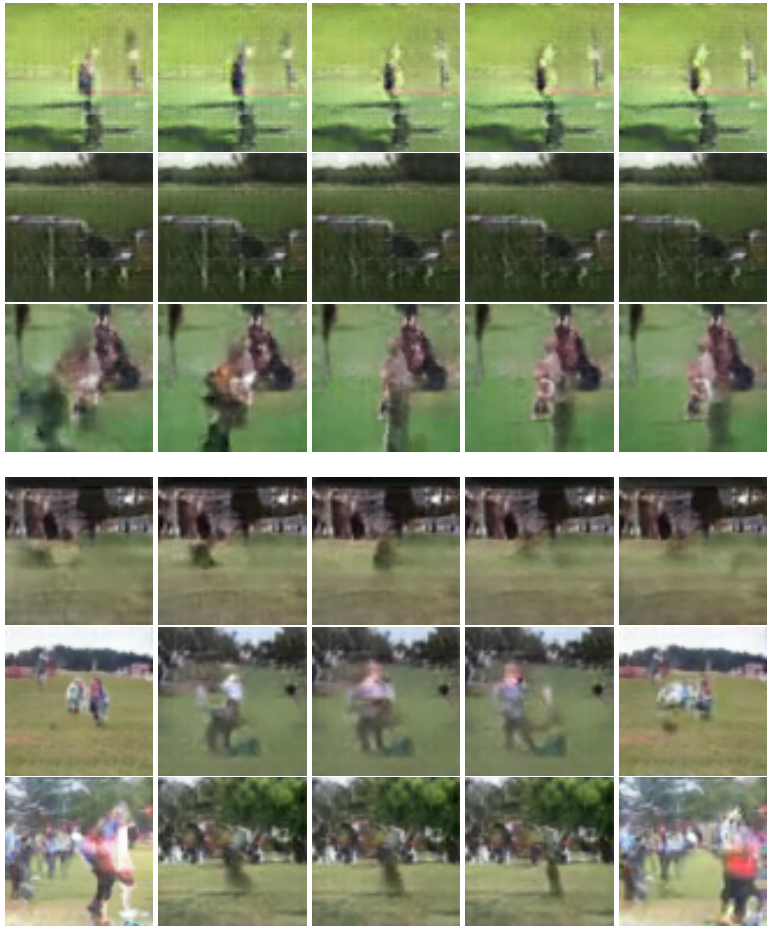

Fig. 3: **Frame samples**. (First and forth row) VGAN-generated video figures show very little object motion, while (second and fifth row) TGAN-generated video figures show motion, but the quality of foreground appearance is low. Our approach (third and sixth row) generates video figures with a good compromise between object motion and appearance.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 2672–2680
2. Denton, E.L., Chintala, S., szlam, a., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 1486–1494
3. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)
4. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
5. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In Precup, D., Teh, Y.W., eds.: Proceedings of the 34th International Conference on Machine Learning. Volume 70 of Proceedings of Machine Learning Research., International Convention Centre, Sydney, Australia, PMLR (06–11 Aug 2017) 214–223
6. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
7. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems 30. Curran Associates, Inc. (2017) 2018–2028
8. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
9. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 613–621
10. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
11. Shoemake, K.: Animating rotation with quaternion curves. SIGGRAPH Comput. Graph. **19**(3) (July 1985) 245–254
12. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
13. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Proceedings of the 13th Scandinavian Conference on Image Analysis. SCIA'03, Berlin, Heidelberg, Springer-Verlag (2003) 363–370
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR). (2015)
15. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 2234–2242