Synthesizing Dynamic Patterns by Spatial-Temporal Generative ConvNet

Jianwen Xie, Song-Chun Zhu, Ying Nian Wu
University of California, Los Angeles (UCLA), USA

IEEE 2017 Conference on Computer Vision and Pattern Recognition

Abstract

1

Video sequences contain rich dynamic patterns, such as dynamic texture patterns that exhibit stationarity in the temporal domain, and action patterns that are non-stationary in either spatial or temporal domain. We show that a spatial-temporal generative ConvNet can be used to model and synthesize dynamic patterns. The model defines a probability distribution on the video sequence, and the log probability is defined by a spatial-temporal ConvNet that consists of multiple layers of spatial-temporal filters to capture spatial-temporal patterns of different scales. The model can be learned from the training video sequences by an "analysis by synthesis" learning algorithm that iterates the following two steps. Step 1 synthesizes video sequences from the currently learned model. Step 2 then updates the model parameters based on the difference between the synthesized video sequences and the observed training sequences. We show that the learning algorithm can synthesize realistic dynamic patterns.

Conclusion

5

We propose a spatial-temporal generative ConvNet model for synthesizing dynamic patterns, such as dynamic textures and action patterns. Our experiments show that the model can synthesize realistic dynamic patterns. Moreover, it is possible to learn the model from video sequences with occluded pixels or missing frames.

Acknowledgment

6

The work is supported by NSF DMS 1310391, DARPA SIMPLEX N66001-15-C-4035, ONR MURI N00014-16-1-2007, and DARPA ARO W911NF-16-1-0579.

Reproducibility

7

http://www.stat.ucla.edu/~jxie/STGConvNet/STGConvNet.html

Spatial-temporal generative ConvNet

2

Notation

Let $\mathbf{I}(x,t)$ be an image sequence of a video, where $x \in D$ indexes the coordinates of pixels, and $t \in T$ indexes the frames in the video. Let $F_k^{(l)}$ be the k-th spatial-temporal filter at layer $l \in \{1,2,\dots,L\}$ in the spatial-temporal ConvNet. Let $[F_k^{(l)} * \mathbf{I}](x,t)$ be the filter response at pixel x and time t.

Model

It is an energy-based model defined on the image sequence $\mathbf{I}$ with the form of

$$p(\mathbf{I};w) = \frac{1}{Z(w)}\exp[f(\mathbf{I};w)]q(\mathbf{I}),$$

where the scoring function $f(\mathbf{I};w)$ is

$$f(\mathbf{I};w) = \sum_{k=1}^{K}\sum_{x\in D_L}\sum_{t\in T_L}\left[F_k^{(L)} * \mathbf{I}\right](x,t),$$

where w consists of all the weight and bias terms that define the filters $\left(F_k^{(L)}, k=1,\dots,K=N_L\right)$ at layer L, and q is the Gaussian white noise model, i.e.,

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|D\times T|/2}}\exp\left[-\frac{1}{2\sigma^2}\|\mathbf{I}\|^2\right],$$

where $|D\times T|$ counts the number of pixels in the domain $D\times T$. Without loss of generality, we shall assume $\sigma^2 = 1$.

Sampling and learning algorithm

(1) Sampling by Langevin dynamics

One can sample from $p(\mathbf{I};w)$ by

$$\mathbf{I}_{\tau+1} = \mathbf{I}_\tau - \frac{\epsilon^2}{2}\left[\mathbf{I}_\tau - \frac{\partial}{\partial \mathbf{I}}f(\mathbf{I}_\tau;w)\right] + \epsilon Z_\tau,$$

where τ indexes the time steps, ϵ is the step size, and $Z_\tau \sim N(0,1)$.

$\frac{\partial}{\partial \mathbf{I}}f(\mathbf{I}_\tau;w) = \boldsymbol{B}_{w,\delta(\mathbf{I}_\tau;w)}$ is an auto-encoding reconstruction process, where binary activation pattern $\delta(\mathbf{I}_\tau;w)$ is computed by a bottom-up convolutional process, where w plays a role of filters; $\boldsymbol{B}_{w,\delta}$ is computed by a top-down deconvolutional process, where w plays a role of bases. The dynamics is driven by the reconstruction error $\mathbf{I}_\tau - \boldsymbol{B}_{w,\delta(\mathbf{I}_\tau;w)}$.

(2) Learning by Maximum Likelihood

The learning of w from training image sequences $\{\mathbf{I}_m, m=1,\dots,M\}$ can be accomplished by the MLE. Let $L(w) = \sum_{m=1}^{M}\log p(\mathbf{I}_m;w)/M$,

$$\frac{\partial L(w)}{\partial w} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial w}f(\mathbf{I}_m;w) - \mathrm{E}_w\left[\frac{\partial}{\partial w}f(\mathbf{I};w)\right].$$

The expectation can be approximated by Monte Carlo samples $\{\tilde{\mathbf{I}}_m, m=1,\dots,\tilde{M}\}$ generated by Langevin dynamics

$$\mathrm{E}_w\left[\frac{\partial}{\partial w}f(\mathbf{I};w)\right] \approx \frac{1}{\tilde{M}}\sum_{m=1}^{\tilde{M}}\frac{\partial}{\partial w}f(\tilde{\mathbf{I}}_m;w).$$

Update $w^{(t+1)} \leftarrow w^{(t)} + \eta_t\frac{\partial L(w)}{\partial w}$, with step size η_t.

Recovery algorithm

3

The model can learn from videos with occluded pixels. It simultaneously accomplishes: (1) recover the occluded pixels of the training video sequences, (2) synthesize new video sequences from the learned model, (3) learn the model by updating the model parameters using the recovered sequences and the synthesized sequences.

Input:

(1) Training image sequences with occluded pixels; (2) Binary masks indicating the locations of the occluded pixels in the training image sequences; (3) Number of learning iterations T.

Output

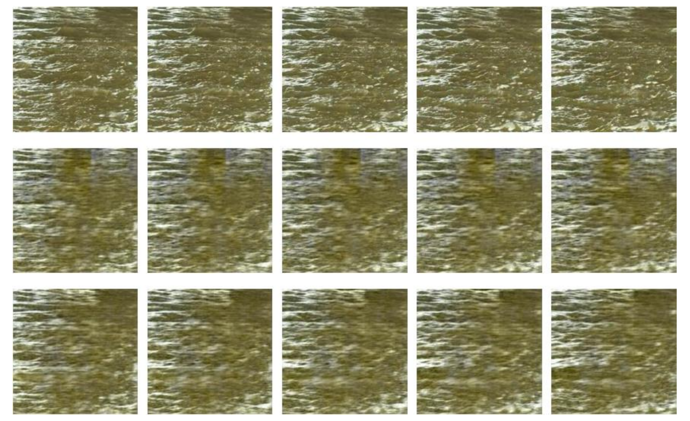
(1) Estimated parameters w; (2) Synthesized image sequences $\{\tilde{\mathbf{I}}_m, m=1,\dots,\tilde{M}\}$; (3) Recovered image sequences $\{\mathbf{I}'_m, m=1,\dots,M\}$.

[1] Let $t \leftarrow 0$, initialize $w^{(0)}$.
[2] Initialize $\tilde{\mathbf{I}}_m$, for $m=1,\dots,\tilde{M}$.
[3] Initialize $\mathbf{I}'_m$, for $m=1,\dots,M$.
Repeat
[4] For each m, starting from the current $\mathbf{I}'_m$, run k steps of Langevin dynamics to recover the occluded region of $\mathbf{I}'_m$.
[5] For each m, starting from the current $\tilde{\mathbf{I}}_m$, run l steps of Langevin dynamics to update $\tilde{\mathbf{I}}_m$.
[6] Compute $H^{\mathrm{obs}} = \sum_{m=1}^{M}\frac{\partial}{\partial w}f(\mathbf{I}'_m;w^{(t)})/M$
[7] Compute $H^{syn} = \sum_{m=1}^{\tilde{M}}\frac{\partial}{\partial w}f(\tilde{\mathbf{I}}_m;w^{(t)})/\tilde{M}$.
[8] Update $w^{(t+1)} \leftarrow w^{(t)} + \eta_t(H^{\mathrm{obs}} - H^{syn})$
[9] Let $t \leftarrow t+1$
Until $t = T$

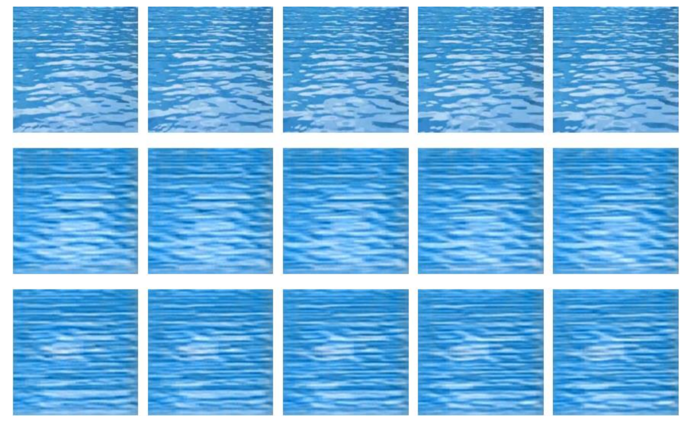
Experiments

Exp 1: Generating dynamic textures with both spatial and temporal stationarity

The first row displays the frames of the observed sequence, and the second and third row displays the corresponding frames of two synthesized sequences.



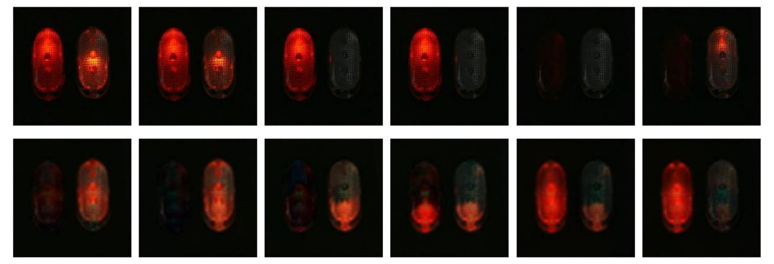
(1) river



(2) ocean

Exp 2: Generating dynamic textures with only temporal stationarity

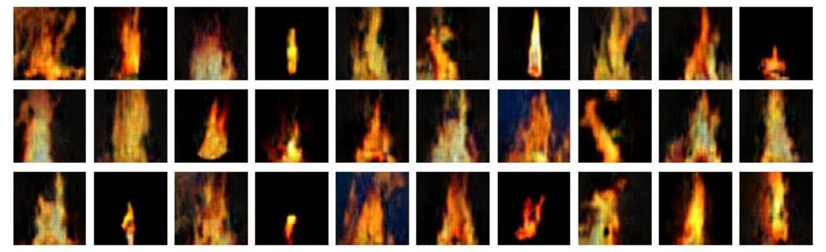
4



(1) flashing lights

(2) burning fire heating a pot

(a) 21-st frame of 30 observed sequences

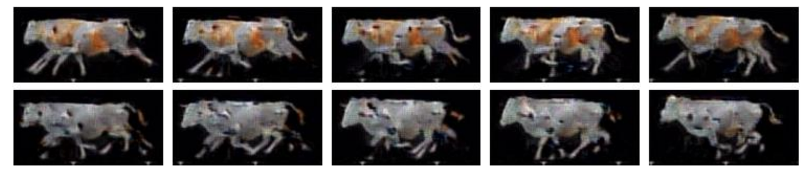


(b) 21-st frame of 30 synthesized sequences

(3) fire

Exp 3: Generating action patterns without spatial or temporal stationarity

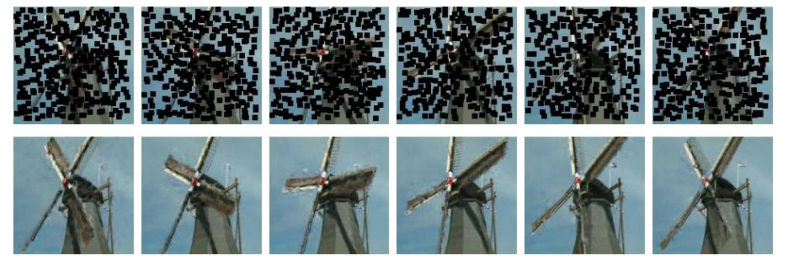
observed sequences



synthesized sequences

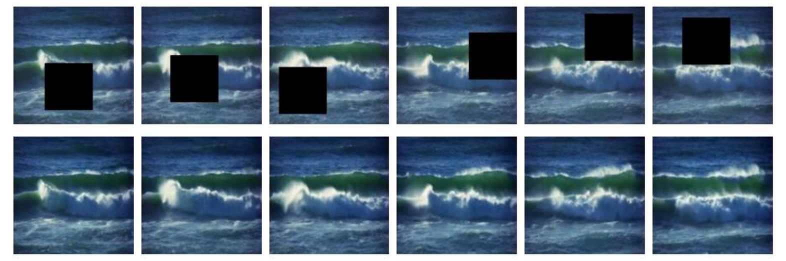
Exp 4: Learning from incomplete data

The first row shows a segment of the occluded sequence with black masks. The second row shows the corresponding segment of the recovered sequence.



(a) 50% salt and pepper masks

(c) 50% missing frames



(b) single region masks

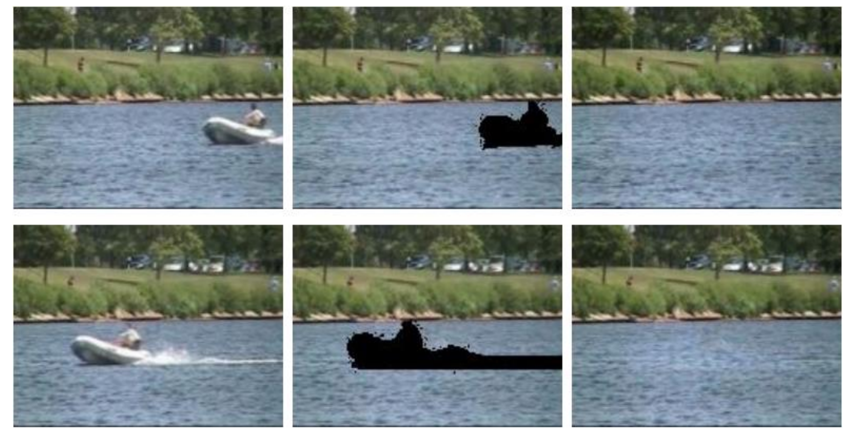
Type	ours	MRF-L1	MRF-L2
(a)	**5.3048**	10.1071	12.8272
(b)	**6.3484**	12.2856	13.1836
(c)	**6.7285**	15.0085	13.7746

Table 1: recovery errors in occlusion experiments

Exp 5: Background inpainting

If a moving object in the video is occluded in each frame, it turns out that the recovery algorithm will become an algorithm for background inpainting of videos, where the goal is to remove the undesired moving object from the video.

original *masked* *inpainted*



(1) removing a moving boat in the lake

original *masked* *inpainted*

(2) removing a walking person in front of fountain