Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks

Tianfan Xue*  Jiajun Wu*  Katie Bouman  Bill Freeman

* Indicates equal contribution
Task: future frame prediction

Frame 1

Frame 2
Deterministic predictions fail to model uncertainty
Deterministic predictions fail to model uncertainty
Sampling a motion field from a prior distribution

Only a few motion fields are consistent with the input image.

\[ \sim P_{\text{prior}}(\text{motion}) \]
Related work

Deterministic prediction

Sample from prior distribution

Motion prediction: [Pintea et al., 2014], [Walker et al. 2015]
Visual feature prediction: [Vondrick et al., 2014]
Future frame synthesis: [Mathieu et al., 2014]

Image prior: [Simoncelli 2001], [Zoran 2012]
Motion prior: [Weiss & Adelson, 1998], [Fleet 2000]
Image synthesis: [Portilla and Simoncelli, 2000], [Kingma and Welling, 2014], [Radford 2015], [Oord 2016]

Probabilistic prediction: [Walker et al., 2016]
Related work

Deterministic prediction

Sample from prior distribution

Input frame

Sampled future frames

Our approach
Task: sample future frames consistent with the input
Segment-based synthesis

Input frame

Sampled future frame

Outline  Main idea  Network structure  What the network learns  Result
Segment-based synthesis

Input frame → Segments → Transformed segments → Sampled future frame
Synthesize using different transformations

Input frame  ➔  Segments  ➔  Transformed segments  ➔  Another sampled future frame

Input random motion vector $z \sim p_z(z)$
Synthesis network

Main idea

Network structure

What the network learns

Result
Sample different future frames

Input random motion vector $z \sim p_z(z)$

Outline

Main idea

Network structure

What the network learns

Result
Sample different future frames

Input frame → Synthesis network → Sampled future frame

Input random motion vector

\[ z \sim p_z(z) \]
Sample different future frames

Input random motion vector $z \sim p_z(z)$

Outline
Main idea
Network structure
What the network learns
Result
Training

Input frame → Encoding network → Motion vector \( z \) → Synthesis network → Sampled future frame

Future frame (ground truth)
Training

Future frame (ground truth)

Input frame

Encoding network

Motion vector $z$

Synthesis network

Future frame (prediction)

Training samples (Label-free)
Training

Objective function:
\[ \left\| I_{syn} - I_{gt} \right\| + D_{KL}(z \| N(0, I)) \]

Reconstruction loss

Main idea

Network structure

What the network learns

Result
Training

Future frame $I_{gt}$ (ground truth)

Input frame

Future frame $I_{syn}$ (prediction)

Objective function:

$$||I_{syn} - I_{gt}|| + D_{KL}(z||N(0, I))$$

KL-divergence loss

Variational Autoencoder
[Kingma and Welling, 2014]
Testing

Future frame $I_{gt}$ (ground truth)

Input frame

Future frame $I_{syn}$ (prediction)

Real output from our network

Outline

Main idea

Network structure

What the network learns

Result
How do we design the synthesis network?

Main idea

Network structure

Outline

What the network learns

Result
Synthesize by transforming segments

Input random motion vector $z$

Input frame ➔ Find segments ➔ Synthesis network ➔ Transform segments ➔ Future frame
Synthesize by transforming segments

Input random motion vector $z$

Find segments

Transform segments

Input frame

Image segments

Future frame

Outline

Main idea

Network structure

What the network learns

Result
Synthesize by transforming segments

Input frame

Future frame

Evaluate motion vector $z$

Find segments

Image segments

Transform segments

Convolution

Outline

Main idea

Network structure

What the network learns

Result
Movement can be synthesized through convolution

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Main idea

Network structure

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Outline

What the network learns

Result
Transforming segments vis Cross-convolution

Input random motion vector $z$

Find segments

Image segments

Convolutions

Motion kernels

Transform segments

Input frame

Future frame

Outline

Main idea

Network structure

What the network learns

Result
Applying motion to each segment

Input random motion vector \( z \)

Main idea

Outline

What the network learns

Result

Network structure

Input frame → Find segments → Segment 1 → Transform segments → Future frame

Motion kernels for segment 1
Applying motion to each segment

Input random motion vector $z$

Input frame $\rightarrow$ Find segments $\rightarrow$ Segment 2 $\rightarrow$ Transform segments $\rightarrow$ Future frame

Motion kernels for segment 2

Main idea

Network structure

Outline

What the network learns

Result
Applying motion to each segment

Input random motion vector $z$

Decoding net

Motion kernels for segment 3

Convolutions

Segment 3

Find segments

Transform segments

Input frame

Future frame

Outline

Main idea

Network structure

What the network learns

Result
Applying motion to each segment

The decoding network generates a motion kernel for each corresponding segment

[Brabandere et al. 2016]
[Finn et al. 2016]
Synthesize by transforming segments

Outline

Main idea

Network structure

What the network learns

Result
What is encoded in the motion vector?

**Input frame** → Encoding network → **Motion vector $z$** → **Synthesis network** → **Future frame**
Each dimension encodes a type of motion

Motion vector $z$  
Upward motion when changing this dimension
Each dimension encodes a type of motion

Motion vector $z$  Leg motion when changing this dimension
Results: toy example

• Simulated shapes

• Training samples
Network automatically detects segments

Input

Learned segments

Triangles

Circles
Network learns the correlation between appearance and motion

Input

Sampled next frame

Ground truth distribution

Sample distribution

Outline  Main idea  Network structure  What the network learns  Result
Results: real-world images

Input

Sampled future frames
Challenge: large motion

Input

Two sampled future frames

Artifacts appear when motion is large
Mechanical Turk study to assess synthesis quality

Instructions
Each time you will see two animated GIFs. One is taken from a real video, and the other is synthesized. Your goal is to click on the GIF that you think is real.

Which looks real? Just click on it.

Labeled as real

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: Transfer flow</td>
<td>25.5 %</td>
</tr>
<tr>
<td>Our method</td>
<td>31.3 %</td>
</tr>
</tbody>
</table>

Ideal synthesis algorithm achieves 50%
Contributions

• Sample multiple future frames that are consistent with the input

• Synthesize frames by transforming segments

• Learn a motion representation without supervision
Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks

Tianfan Xue*  Jiajun Wu*  Katie Bouman  Bill Freeman

http://visualdynamics.csail.mit.edu