Supplemental Material of Eventfulness for Interactive Video Alignment

JIATIAN SUN, Cornell University, USA
LONGXIULIN DENG, Cornell University, USA
TRIANTAFYLLOS AFOURAS, Meta AI, USA
ANDREW OWENS, University of Michigan, USA
ABE DAVIS, Cornell University, USA

In this supplemental material, we discuss the synthetic data generation and network training details.

ACM Reference Format:
Jiatian Sun, Longxiulin Deng, Triantafyllos Afouras, Andrew Owens, and Abe Davis. 2023. Supplemental Material of Eventfulness for Interactive Video Alignment. *ACM Trans. Graph.* 42, 4 (August 2023), 3 pages. https://doi.org/10.1145/3592118

1 LEARNING VISUAL EVENTFULNESS

1.1 Synthesizing Training Data for Eventfulness

1.1.1 Synthesizing Motions. Figure 3 of the main paper summarizes the synthetic video generation process. As shown in the figure, after sampling a finite set of event timing $E = \{e_0, e_1, e_2, \ldots, e_n\} \in R$, we will generate motion discontinuities at those events. These discontinuities are introduced by making large changes in the direction of linear velocity.

To do so, first, we sample a 3D polyline trajectory for each moving object. As an object tracks its trajectory, it always moves in a straight line until an event occurs. Thus, events only occur at vertices of a polyline. To ensure large changes in velocity at all events, all angles of the polyline are above 30 degree.

Next, we replace the line segment trajectories in between events with low-curvature splines, so that we could cover a larger family of smooth motions without losing the directional changes at the events.

1.1.2 Label Generation. In the synthetic data set, eventfulness is proportional to the number of motion discontinuities at each frame. We consider a moving object has motion discontinuity at one frame, if an event e occurs in between this frame and its previous frame. Thereafter, we count all the objects with motion discontinuity at each frame and raise it to the power of 0.7 to make it into the eventfulness label. Eventually, to facilitate the regressive training process, we blur the eventfulness label with a Gaussian kernel at run time.

The other dimensions of the motion descriptor, the camera-space linear velocity and acceleration's projection on positive and negative direction of x and y axis of the screen, are evaluated with finite difference methods. Specifically, the linear velocity is evaluated with backward difference scheme and acceleration is evaluated with central difference scheme.

Table 1. Architecture details for the eventfulness spatio-tempotal CNN model. The model is based on the (2+1)D ResNet-18 video architecture [Tran et al. 2018] by removing the temporal strides and adaptation for dense regresssion. Batch Normalization and ReLU activation are added after every convolutional layer. Shortcut connections are also added at each layer, except for the first layer of every residual block (the ones with stride > 1).

Layer	# Channels	Kernel	Stride	Padding	Output dimensions
video input	3	-	-	-	$T \times 100 \times 100 \times 3$
$conv_{1,1}$	64	(3,7,7)	(1,2,2)	(1,3,3)	$T \times 50 \times 50 \times 64$
$conv_{1,2}$	64	(3,3,3)	(1,1,1)	(1,1,1)	$T \times 50 \times 50 \times 64$
$conv_{1,3}$	64	(3,3,3)	(1,1,1)	(1,1,1)	$T \times 50 \times 50 \times 64$
$conv_{2,1}$	128	(3,3,3)	(1,2,2)	(1,1,1)	$T \times 25 \times 25 \times 128$
$conv_{2,2}$	128	(3,3,3)	(1,1,1)	(1,1,1)	$T \times 25 \times 25 \times 128$
$conv_{3,1}$	256	(3,3,3)	(1,2,2)	(1,1,1)	$T \times 12 \times 12 \times 256$
$conv_{3,1}$	256	(3,3,3)	(1,1,1)	(1,1,1)	$T \times 12 \times 12 \times 256$
$conv_{4,1}$	512	(3,3,3)	(1,2,2)	(1,1,1)	$T \times 6 \times 6 \times 512$
$conv_{4,1}$	512	(3,3,3)	(1,1,1)	(1,1,1)	$T \times 6 \times 6 \times 512$
avgpool	512	(1,6,6)	(1,1,1)	(0,0,0)	$T \times 512$
fc_1	1	-	-	-	$T \times N$

1.1.3 Randomization. We experimented with multiple domain randomization strategies to improve our model's robustness to videos in the wild. To recapitulate, we randomize our synthetic data in following ways:

(1) **Geometry and Texture:** We have 5 basic geometry primitives in total, including a cube, a sphere, a cylinder, a thin rod and a humanoid (obtained from Unity's asset store). For each synthesized video, we generate a displacement vector for each vertex of the geometry to add randomness to object shapes. Then, we sample a random texture from ImageNet for each object. Random texture sample from the same dataset is also attached to the background image plane of the scene.
(2) **Lighting and Shading:** For each scene we would randomly sample 5 types of lights (spot lights, directional lights, point

Authors' addresses: Jiatian Sun, js3623@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850; Longxiulin Deng, ld469@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850; Triantafyllos Afouras, afourast@robots.ox.ac.uk, Meta AI, New York City, New York, USA; Andrew Owens, ahowens@umich.edu, University of Michigan, Ann Arbor, Michigan, USA; Abe Davis, abedavis@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850.

0730-0301/2023/8-ART $15.00
https://doi.org/10.1145/3592118

Table 2. **Ablation Study on Randomization Strategies and Data Augmenetation.**We use average precision to compare the performance different randomization and data augmentation methods.

Random Texture	Random Geometry	Color Jitter	Spatial Cropping	Screenspace Camera Shake	Greatest Hit	Bouncing Ball	Videos in the Wild
✓	✓	✓	✓	✓	**0.14**	**0.56**	**0.31**
✓	X	✓	✓	✓	**0.14**	0.52	**0.31**
X	✓	✓	✓	✓	0.13	**0.59**	0.23
✓	✓	✓	X	✓	**0.14**	**0.58**	**0.31**
✓	✓	X	✓	✓	**0.14**	0.54	0.24
✓	✓	✓	✓	X	**0.14**	0.36	0.24
✓	✓	X	✓	X	**0.14**	0.35	**0.30**
✓	✓	✓	X	X	0.13	0.34	**0.30**
✓	✓	X	X	✓	**0.14**	0.42	**0.29**
✓	✓	X	X	X	0.13	0.47	**0.32**

lights, rectangular lights, and disc lights), and randomize their color and position. Lights with orientation are pointed toward the center of the scene to increase the probability that objects are exposed to their illumination.

(3) **Objects:** We randomly initialize the position, orientation and scale of each object. The number of objects in the scene is also randomized.

(4) **Camera Motion and Shake:** To help the network become robust to camera motion, we simulate a synthetic camera shake being applied to the camera. This shake is generated by sampling a random force to be applied to the camera in every frame while applying a elastic force that pulls the camera back to the target position.

1.2 Eventfulness Prediction Model

The complete network architecture is summarized in Table 1. We trained eventfulness with adam optimization and learning rate l listed in Table 3.

1.3 Training

1.3.1 Data augmentations. To reduce overfitting and improve generalization to real data, we also experimented with different data augmentations during training. In particular, we perform random spatial cropping, and random color jittering (brightness, contrast, hue, and saturation). We also add an artificial camera shake effect, by applying a random displacement to the crop location at every single frame while also applying a displacement to drag the crop center back to the original position.

1.4 Ablation Study

We have proposed multiple randomization strategies and data augmentation methods to make the network more robust. To validate the effectiveness of those strategies, we conduct an ablation study on synthetic data generation and data augmentation.

In Table 2, we demonstrate the ablation study results by comparing the average precision of adaptive eventfulness trained with different combinations of domain randomization and data augmentation methods.

From the ablation study, we observe that the network trained with all randomization and data augmentation method except for spatial cropping performs generally well for all three evaluation datasets. Spatial cropping in training isn't helpful for improving the network's performance on these three datasets since all the videos in these three sets always keep the object of interest in view and unoccluded. This makes it difficult to test if spatial cropping increases the network's performance with a plethora of occluded motion.

Since spatial cropping doesn't influence the network's performance on real-world videos in the Greatest Hit and Videos in the Wild datasets, we use the more general network with spatial cropping to generate all of the results included in the main paper and the supplemental material.

Table 3. Network Training Parameters

Parameters	#	Explanation
T	72	The input video frame number
Target FPS	24	Each input video is resampled to this frame rate.
M	128	Each input video frame is resized to $M \times M$.
N_{acc}	4	# of linear acceleration related labels.
N_{vel}	4	# of linear velocity related labels.
N_{blur}	4	# of labels is a Gaussian blur of the eventfulness label.
σ_{min}	0.1	The minimum standard deviation of the Gaussian kernel used for blurring eventfulness.
σ_{max}	1	The maximum standard deviation of the Gaussian kernel used for blurring eventfulness.
d	7	The diameter of the Gaussian kernel in # of frames.
l	10^{-6}	learning rate

1.5 Evaluating Eventfulness

Eventfulness is evaluated by running a sliding window of size T through the frames of the video. The stride between 2 consecutive windows is $T/3$ and we concatenate the eventfulness value of the middle $T/3$ frames of each window to formulate the eventfulness of an entire video.

REFERENCES

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. 6450–6459. https://doi.org/10.1109/CVPR.2018.00675