Contour Detection and Characterization for Asynchronous Event Sensors
Supplementary Material

Francisco Barranco* [1,2], Ching L. Teo* [1], Cornelia Fermüller[1], Yiannis Aloimonos[1]
[1] Computer Vision Lab, University of Maryland (USA) [2] CITIC, University of Granada (Spain)
{barranco, cteo, fer, yiannis}@umiacs.umd.edu

1. Event-based Visual Features

We illustrate in Fig. 1 the visual features extracted from DVS data and presented in §3.1 of the paper. Fig. 1 (top panel) shows a 3D view of the last timestamp for every location after a short period of 20 ms, and a highlighted (boxed) patch that illustrates the extracted features used in the training. As mentioned in the main paper, we use four groups of features, and we illustrate a subset of them in Fig. 1 (bottom panel) with respect to the highlighted patch:

1. *Event temporal information*. We show in Fig. 1-A the timestamp of the last event triggered for every pixel, measured in terms of *relative* time (ms) with respect to the onset of the event.

2. *Event-based orientation*. The events are grouped into eight discrete spatial orientations (from 0 to π). Fig. 1-B shows the map of orientations for different spatial locations. For every new event, its timestamp is first compared to the average timestamp of the events in the neighborhood. If the difference exceeds 10 ms, the event is considered an outlier and is discarded. A winner-takes-all strategy is then used to obtain the most likely orientation for the new event which we admit if the difference between the new orientation and previous orientation exceeds 2 orientation bins.

3. *Event-based motion estimation*. Following [1], we used a function $\mathcal{T}_e$ that assigns to every position the timestamp of its last event. This function locally defines a surface of size 5×5 pixels. The spatial derivatives of this surface provide the speed and direction of the local motion. Specifically, the gradient vector $\nabla\mathcal{T}_e = \left(v_x^{-1}, v_y^{-1}\right)^T$ gives the inverse of the image velocity. In practice, the function $\mathcal{T}_e$ is approximated by fitting a local plane $\mathcal{P}$ (with normal vector $\vec{n}$) to the last timestamp for every location, as illustrated in Fig. 1-C.

   Additionally, a regularization of the data is performed simultaneously along with the plane fitting process. For each new event that is reasonably close ($<$ 0.2 pixels), $\mathcal{P}$ is updated within a time interval of 7.5 ms.

4. *Event-based time-texture*. Instead of intensity texture gradients as used on images, we use a map of the timestamps of the last event triggered at every pixel. This map defines a *time-texture* surface and we apply a bank of Gabor filters, using 6 orientations and 3 scales. The three feature maps depicted in Fig. 1-D correspond to the maximum response over all orientations at every location, for each of the three spatial scales considered.

All these feature maps are estimated using short time intervals of 20 ms. As mentioned in the main paper, all feature processing is event-driven: with every new event, all the feature maps and their timestamps are updated with respect to the new event.

2. Experiments

We show here more results that illustrate boundary and and border ownership predictions from experiments described in §4 of the paper. Fig. 2 shows our prediction results using all features compared to the baseline for some example DVS data for datasets exhibiting different predominant motions: “Rotation”, “Translation”, and “Zoom”. We show 9 testing examples (out of the 18) for each dataset.

*– indicates equal contribution

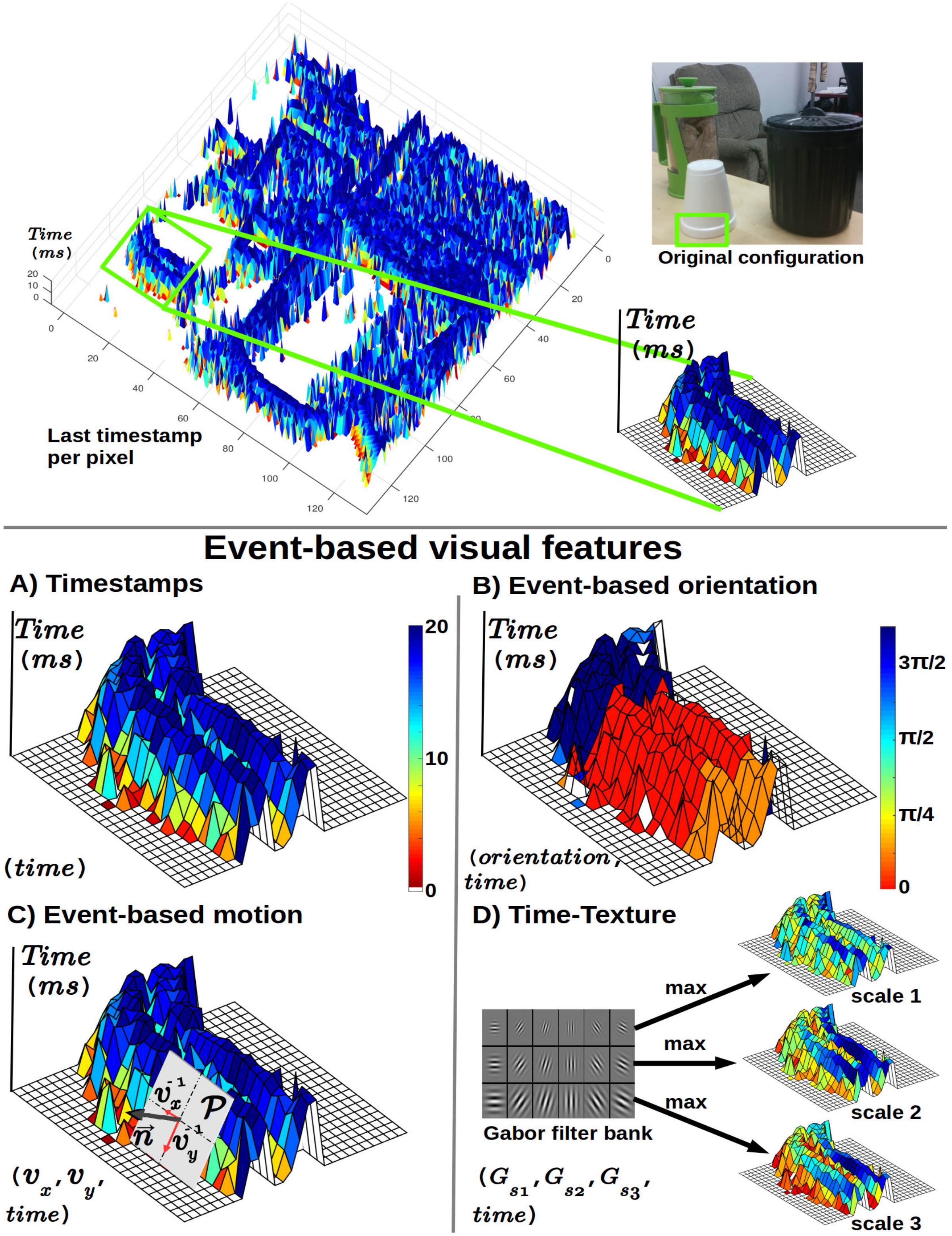

Figure 1. Event-based visual features. (Top panel) A 3D spatial representation that encodes the timestamp of the last event in the z axis (after 20 ms) for every pixel in the DVS sensor. The image on the top-left shows the original configuration of the scene (captured with a conventional camera). We show more features derived from the highlighted patch (boxed) in the bottom panel: A) the last timestamp (time); B) event-based orientation (orientation, time); C) event-based motion estimation, $\nabla \mathcal{T}_e$, computed by fitting local 5×5 planes to the surface $\mathcal{T}_e$ (horizontal component of the motion v_x, vertical component of the motion v_y, time); D) event-based time-texture, obtained from the maximum responses per scale of a bank of Gabor filters with 6 orientations and 3 scales (max response at 1^{st} scale G_{s1}, max response at 2^{nd} scale G_{s2}, max response at 3^{rd} scale G_{s3}, time).

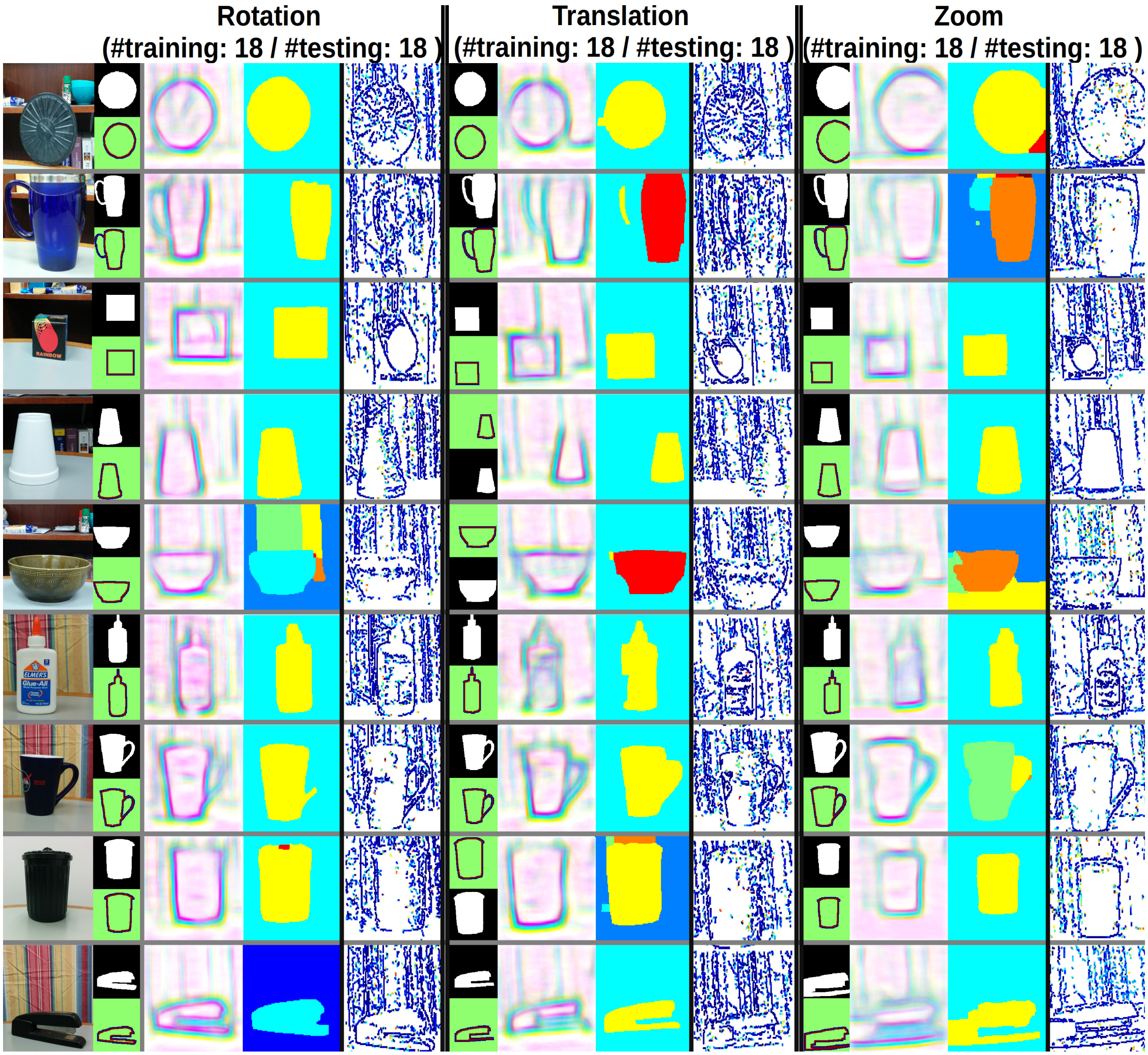
Rotation
(#training: 18 / #testing: 18)
Translation
(#training: 18 / #testing: 18)
Zoom
(#training: 18 / #testing: 18)

Figure 2. Example results of datasets separated by predominant motion: (L-R) “Rotation”, “Translation” and “Zoom”. The first image per row is the original scene configuration captured with a conventional camera. For every dataset, we show (from left to right): Hand annotated segmentation and border ownership groundtruths; predicted boundaries (blue) and ownership (red: foreground, yellow: background) from DVS data; predicted segmentation from DVS data; baseline contour results.

Fig. 3 presents results for the remaining two datasets: “Complex”, and “NewObj-NewBG”. The “Complex” dataset consists of rigid motions (rotation + translation + zoom) with a maximum of 3 objects inducing 3 motion layers (excluding the background). In addition, “NewObj-NewBG” is a held out testing set that contains random objects and complex backgrounds not encountered during training to determine the performance of the approach in such difficult scenarios.

We note that qualitatively, our predictions are much cleaner and closer to the groundtruth than the baseline.

3. Video results for DVS sequence in “NewObj-NewBG”

This supplementary material also contains a video file (“dvsBdBo.mp4”) that shows boundary and border ownership prediction for a continuous sequence from the “NewObj-NewBG” dataset. In addition, the video shows the segmentation

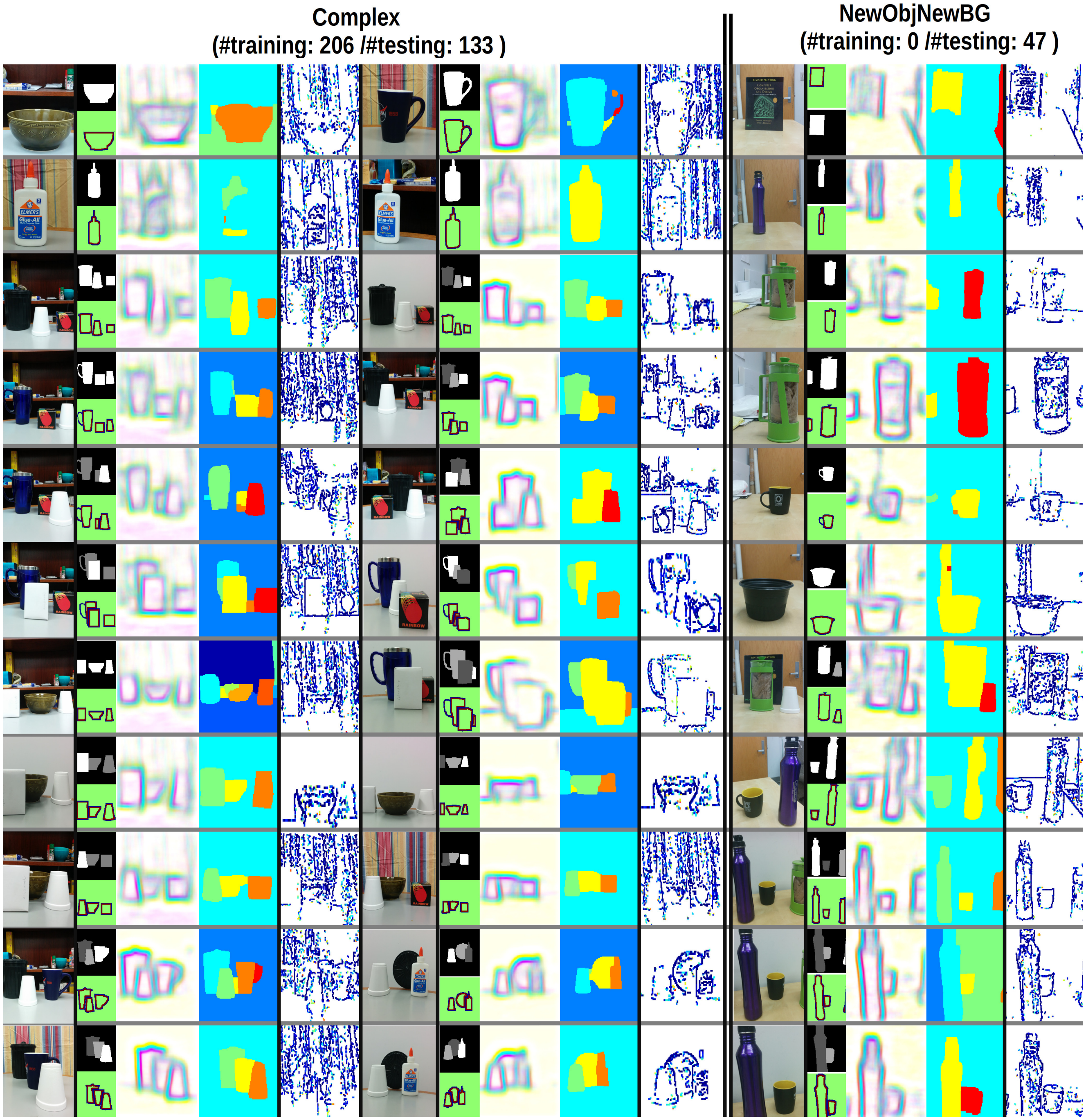
Complex
(#training: 206 /#testing: 133)
NewObjNewBG
(#training: 0 /#testing: 47)

Figure 3. Example results for datasets: (L-R) “Complex” and “NewObj-NewBG”. For every dataset, the first image per row is the original scene configuration captured with a conventional camera. For each dataset, we show (from left to right): Hand annotated segmentation and border ownership groundtruths; predicted boundaries (blue) and ownership (red: foreground, yellow: background) from DVS data; predicted segmentation from DVS data; baseline contour results. Note that for the “Complex” dataset we are showing two groups of results (22 examples), and for “NewObj-NewBG” one group of results (11 examples).

results for several key frames and the original configuration of the corresponding scenes. Note that since the original image is captured with a conventional camera, it will not correspond exactly with our results obtained from the DVS, but it gives an idea about the scene for a better understanding of the predicted results.

References

[1] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi. Event-based visual flow. *IEEE Trans. on Neural Networks and Learning Systems*, 25(2):407–417, 2014. 1