Supplementary Material for the Paper
"Speed Invariant Time Surface for Learning to Detect Corner Points with Event-Based Cameras"

Jacques Manderscheid[1], Amos Sironi[1], Nicolas Bourdis[1], Davide Migliore[1] and Vincent Lepetit[2]
[1]Prophesee, Paris, France, [2]University of Bordeaux, Bordeaux, France
{jmanderscheid, asironi, nbourdis, dmigliore}@prophesee.ai, vincent.lepetit@u-bordeaux.fr

In this appendix, we report additional results and details which did not fit in our submission due to space limitations. In Section 1, we provide more details about the training dataset introduced in of Sec. 4.4 of the main submission. In Section 2, we describe the acquisition protocol and the content of the HVGA ATIS Corner dataset introduced in Sec. 5. Then, in Section 3, we explain the details of the nearest neighbor matching used for the corner tracking also introduced in Sec. 5. Finally, in Section 4, we present detailed results on the HVGA ATIS Corner dataset and in Section 5, the results on the Event-Camera Dataset [1].

Attached to this appendix, we also provide results video sequences comparing our method and the baselines on the two datasets used in our experiments.

1. Train Dataset

Our method relies on a binary classifier to discriminate patches of the Speed Invariant Time Surface as generated by a moving corner or not. The classifier is trained on a sequence acquired by a HVGA ATIS sensor that is returning at the same time the events and the graylevel measurements of the scene. For a given event, the Harris detector is run on the graylevels at its location. We use the resulting score to classify the event as corner or non corner event. An example of the resulting classification is shown in Fig. 1. Once the events have been labeled, we extract corresponding patches of the Speed Invariant Time Surface around them. This operation is done per time slice of 1ms every 100ms and results in 109220 positive and 710177 negative samples.

The entire training set comes from a video of the patern shown in Fig. 1. Geometrical shapes were chosen to ensure the reliability of the Harris detector which leads to the quality of the labels in training set.

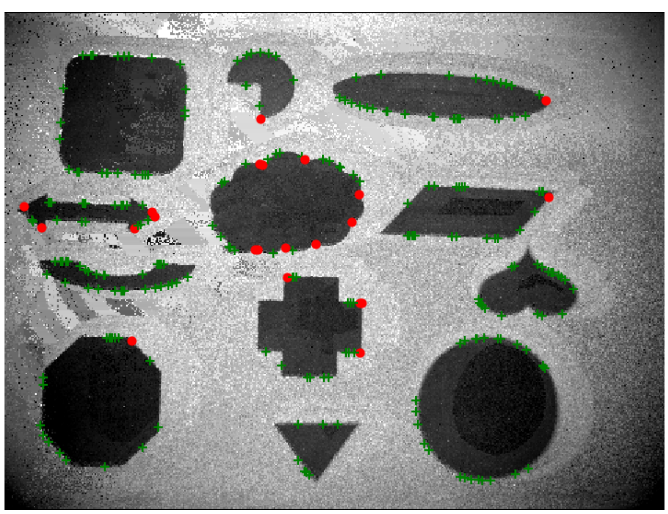

Figure 1: Snapshot of the graylevels with 1ms of corresponding events used in the training set. Red: corner events, Green: non corner events.

2. HVGA ATIS Corner Dataset

The HVGA ATIS Corner dataset is composed of 7 sequences recorded on a HVGA ATIS sensor filming four different planar posters. The sequences comes form the following posters:

- 1 sequence was recorded using a chessboard poster, denoted `chessboard`;
- 2 sequences were recorded from a poster of the Picasso's Guernica painting, denoted `guernica_1` and `guernica_2`;
- 2 sequences from a urban graffiti, denoted `graffiti_1` and `graffiti_2`;
- 2 sequences from a picture of the city of Paris, denoted `paris_1` and `paris_2`;

The chessboard and Guernica are composed of geometric patterns and provide simple cases to evaluate a corner detector. The Paris and Graffiti sequences are more challenging and recreate more realistic environments to evaluate corner detectors. Fig. 2 shows the original images used to generate the dataset. While snapshots of the events obtained by recording them are given in Fig. 3.

Figure 2: Original posters of the HVGA ATIS Corner dataset. **(Top Left)** `chessboard` poster, **(Top Right)** `Guernica` poster, **(Bottom Left)** `Paris` poster, **(Bottom Right)** `Graffiti` poster.

Each sequence last around 150 seconds with no camera movement during the first 60 seconds. This time is used for camera synchronization purposes. Furthermore, the location of the four outside corners of the plane are provided during the entire sequence. This information is used to restrict the evaluation of the homography reprojection error to points contained in the planar pattern.

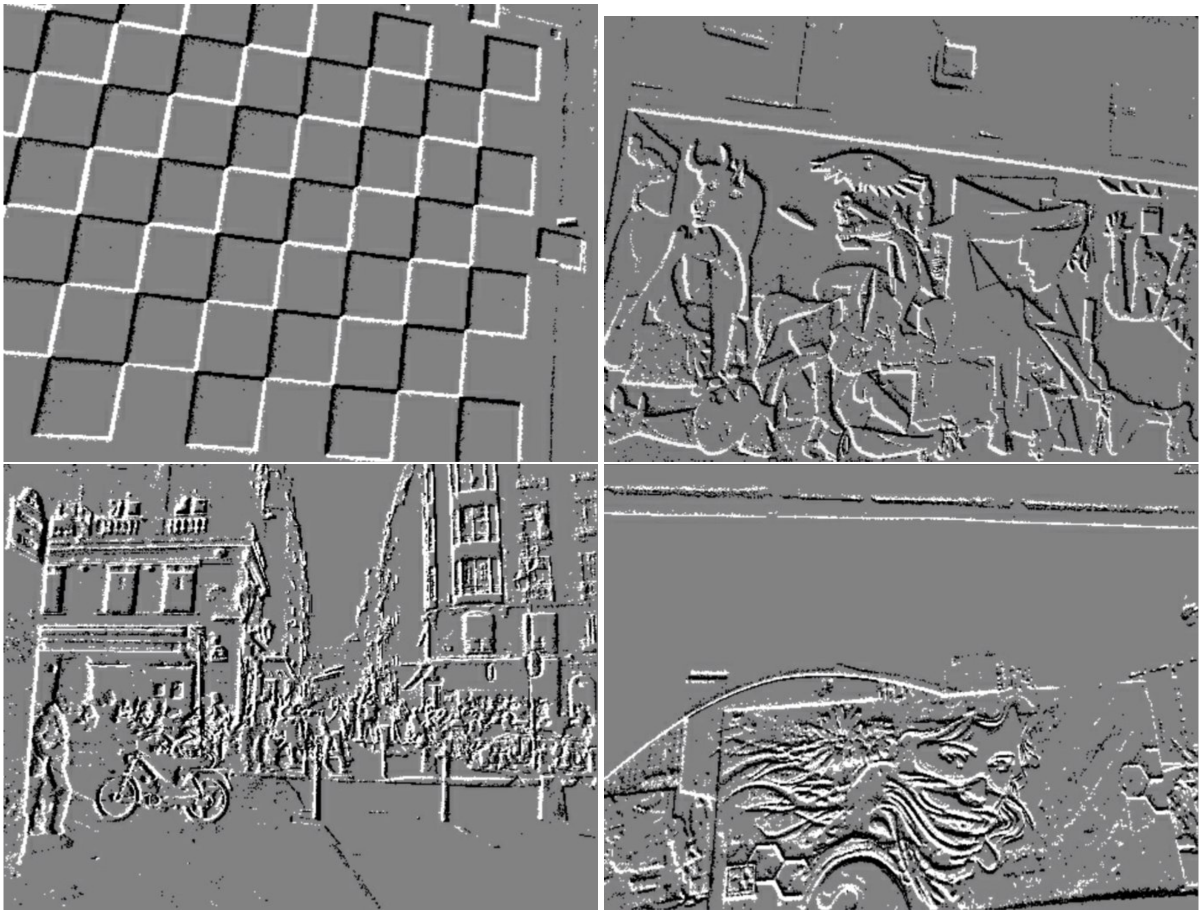

Figure 3: Snapshots of 5ms of events of the HVGA ATIS Corner dataset. **(Top Left)** `Chessboard`, **(Top Right)** `Guernica`, **(Bottom Left)** `Paris`, **(Bottom Right)** `Graffiti`.

3. Corner Tracking

A nearest neighbor tracking algorithm is used to evaluate the accuracy of the corner detectors. It associates corner events to existing tracked features. This algorithm considers a radius in time and space around the detected corner event. If no feature is found in this radius, a new feature is created. If one or multiple features are found, this corner event is associated to the feature with the longest track in term of number of events. We chose this metric to favor stable features against noisy corner events. Moreover, in the case of a perfect event corner detector, only one feature should be available in a small region around the detected corner event.

In order to remove some noise, the tracking is not returning the first n corner events that are associated to a feature. Furthermore, the tracking returns the average of the last n corner events with the timestamp of the last corner event as the position of the feature. This increases the accuracy on the position of the feature if the corner detector detects corners in a radius around the true position of the corner. The pseudocode for the algorithm is given in Algorithm 1.

Algorithm 1 Corner Tracking

Parameters: r, t_0, n
Output: Tracked Corner Events, with feature ID
Feature Event list = struct
    queue : queue of events of size n
    ID : int
    t : timestamp
    count : int
Initialization: $Tracks(x, y) \leftarrow$ Feature event list for all (x, y)
For each incoming event (x, y, t),
$Max \leftarrow 0; dx_{max} \leftarrow 0; dy_{max} \leftarrow 0;$
for $-r \leq dx \leq r$ **do**
    for $-r \leq dy \leq r$ **do**
        if $t - Tracks(x + dx, y + dy).t < t_0$ and $Tracks(x + dx, y + dy).count) > Max$ **then**
            $Max \leftarrow Tracks(x + dx, y + dy).count$
            $dx_{max} \leftarrow dx; dy_{max} \leftarrow dy$
if $Max > 0$ **then**
    $Tracks(x, y) \leftarrow Tracks(x + dx_{max}, y + dy_{max})$
    $Tracks(x + dx_{max}, y + dy_{max}) \leftarrow$ New Feature Event list
else
    $Tracks(x, y) \leftarrow$ New Feature Event list
$Tracks(x, y).t \leftarrow t; Tracks(x, y).count \leftarrow Tracks(x, y).count + 1; Tracks(x, y).queue.append((x, y))$
if $Tracks(x, y).count \geq n$ **then**
    $x_{mean}, y_{mean} \leftarrow mean(Tracks(x, y).queue)$
    Return $(x_{mean}, y_{mean}, Tracks(x, y).ID)$

4. Results on the HVGA ATIS corner dataset

In this section, we provide the detailed results on the HVGA ATIS Corner dataset. Extracts of the videos of each sequence with each corner detector are provided together with this appendix. In Fig. 4 we plot the reporjection as a function of the time interval used to estimate the homography for a Random Forest trained on the standard Time Surface and on our Speed Invariant Time Surface (SLIC).

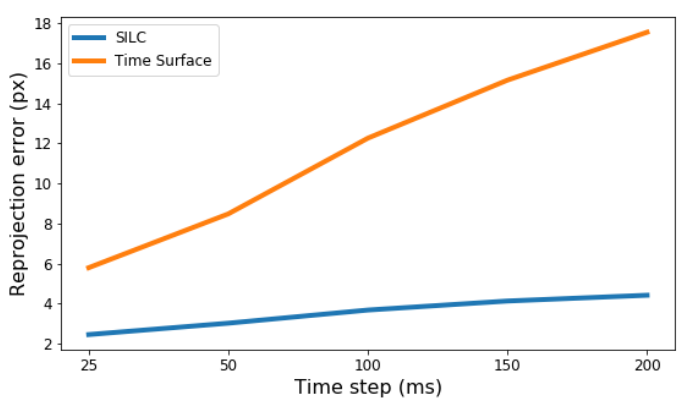

Figure 4: Mean reprojection error on the HVGA ATIS Corner dataset for SILC and for a Random Forest learned of the standard Time Surface, as a function of the Δt used to compute the homography. Using our Speed Invariant Time Surface formulation (SILC) consistently gives lower error.

We provide the complete numbers for the reprojection error on the entire dataset in Table. 1 and Table. 2.

Chessboard	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	1.24	1.53	2.03	2.53	3.08
Arc	1.88	2.06	2.38	2.74	3.13
evFast	1.36	1.65	2.09	2.48	2.81
SLIC	2.16	2.57	3.13	3.67	3.93
Time Surface	2.24	2.96	4.05	5.34	5.93

Guernica_1	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	1.99	2.48	3.11	3.55	3.75
Arc	3.29	4.40	5.77	6.53	7.10
evFast	1.69	1.99	2.42	2.80	2.89
SLIC	2.09	2.55	3.22	3.68	4.09
Time Surface	5.11	7.30	10.14	12.24	13.85

Guernica_2	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	1.95	2.45	3.02	3.47	3.80
Arc	3.30	4.42	5.68	6.55	7.15
evFast	1.69	2.01	2.44	2.76	3.05
SLIC	2.10	2.55	3.08	3.52	3.79
Time Surface	5.13	7.34	10.35	12.51	14.39

Graffiti_1	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	2.59	3.48	4.54	5.22	5.59
Arc	3.72	5.24	7.15	8.34	9.29
evFast	2.16	2.68	3.22	3.59	3.91
SLIC	2.35	2.90	3.48	3.85	4.08
Time Surface	6.32	9.32	13.47	16.52	18.92

Graffiti_2	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	3.03	4.12	5.34	6.00	6.74
Arc	4.34	6.23	8.47	10.06	11.09
evFast	2.66	3.41	4.17	4.81	5.08
SLIC	2.76	3.47	4.29	4.85	5.06
Time Surface	6.92	10.29	14.95	18.37	21.39

Paris_1	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	3.46	4.78	6.39	7.28	8.50
Arc	4.95	7.22	10.15	11.59	13.17
evFast	2.57	3.21	3.64	3.65	3.69
SLIC	2.83	3.51	4.13	4.40	4.62
Time Surface	7.29	10.88	16.12	20.02	23.76

Paris_2	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	3.73	5.41	7.64	9.53	10.95
Arc	5.14	7.60	10.97	13.57	15.50
evFast	2.69	3.43	4.26	4.89	5.31
SLIC	2.90	3.60	4.43	4.94	5.35
Time Surface	7.53	11.28	16.75	21.11	24.61

Table 1: Mean reprojection error in pixels on the HVGA ATIS corner dataset for each method in function of the Δt (in milliseconds) used to compute the homography **(Part 1/2)**.

Average	$\Delta t = 25$	$\Delta t = 50$	$\Delta t = 100$	$\Delta t = 150$	$\Delta t = 200$
evHarris	2.57	3.46	4.58	5.37	6.06
Arc	3.80	5.31	7.22	8.48	9.49
evFast	2.12	2.63	3.18	3.57	3.82
SLIC	2.45	3.02	3.68	4.13	4.42
Time Surface	5.79	8.48	12.26	15.16	17.55

Table 2: Mean reprojection error in pixels on the HVGA ATIS Corner dataset for each method in function of the Δt (in milliseconds) used to compute the homography **(Part 2/2)**.

We also rport in Table. 3 the lifetime of the 100 longest features for each method for each sequence.

Sequence	`Chessboard`	`Guernica_1`	`Guernica_2`	`Graffiti_1`	`Graffiti_2`	`Paris_1`	`Paris_2`
evFast	0.90	0.76	0.70	0.74	0.62	0.57	0.57
Arc	1.37	0.96	0.95	0.83	0.74	0.77	0.77
evHarris	0.87	0.72	0.68	0.76	0.72	0.72	0.70
SILC	2.59	1.03	1.13	0.97	0.77	0.67	0.65

Table 3: Average lifetime (in seconds) of the first 100 longest tracked features.

5. Results on the Event Camera dataset

In this section we report complete results on the Event-Camera Dataset. We use the metrics described in Sec. 5 of our main submission. We report in Fig. 5 the reprojection error and in Fig. 6 the percentage of valid tracks, for all the methods and all the sequences of the dataset. Since the Random Forest used by our method on this dataset (SLIC DAVIS) was trained on the `shapes`, this sequences are excluded from the average evaluation. However, we report for completeness the results on the `shapes` for the baseline methods and our method trained on the HVGA ATIS dataset.

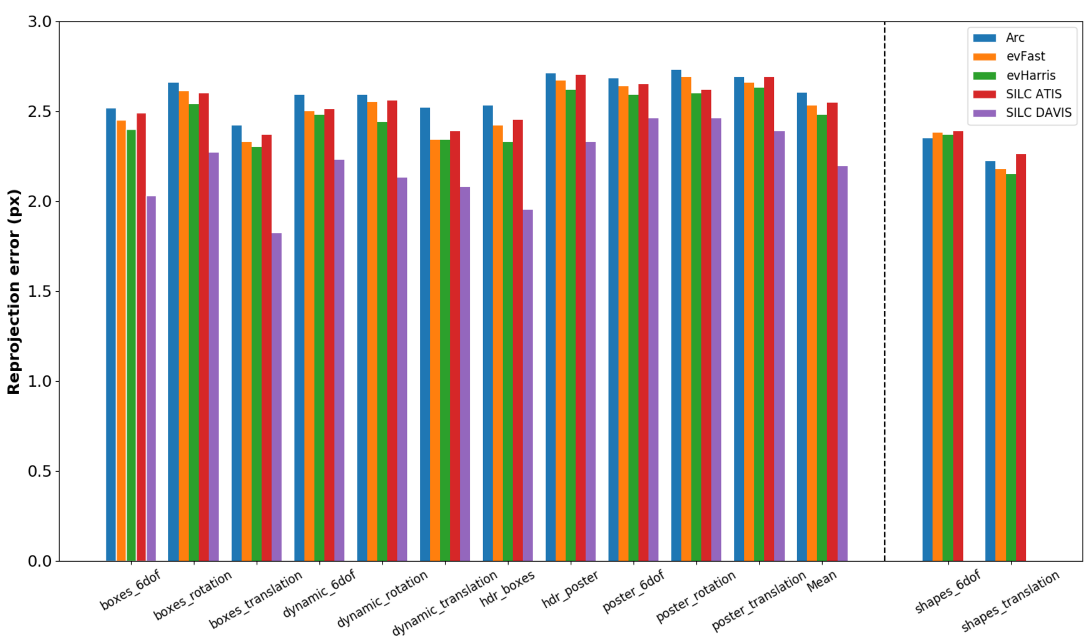

Figure 5: Reprojection error in pixel on the Event-Camera dataset. The lower the better. We don't report results for SLIC DAVIS on the `shapes` sequences since this method was trained on them.

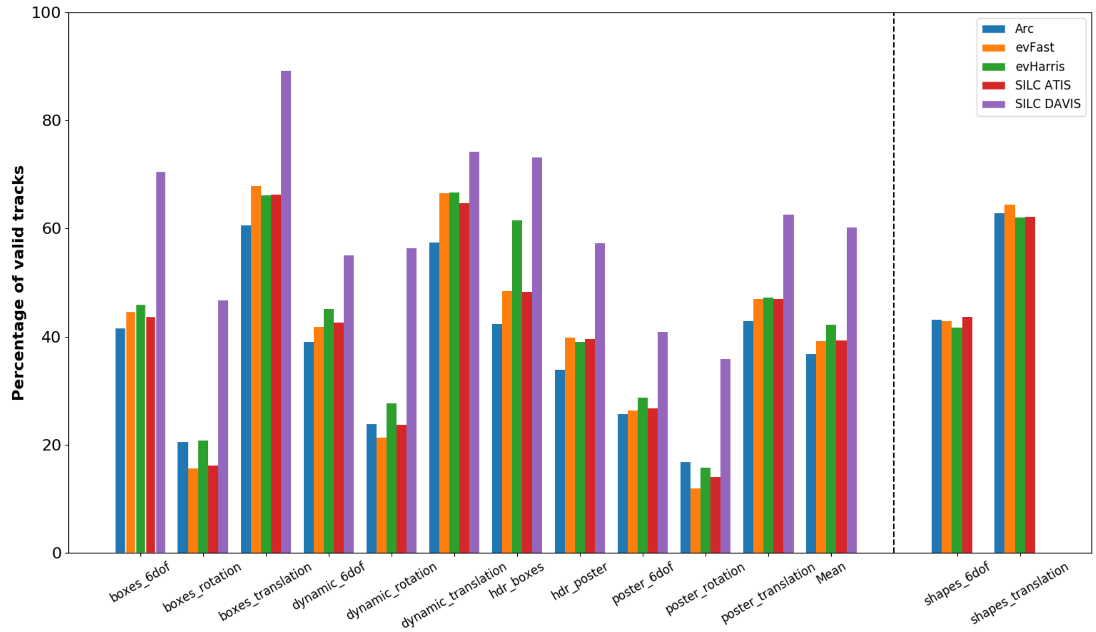

Figure 6: Percentage of valid tracks on the Event-Camera dataset. The higher the better. We don't report results for SLIC DAVIS on the `shapes` sequences since this method was trained on them.

References

[1] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza. The Event-Camera Dataset and Simulator: Event-Based Data for Pose Estimation, Visual Odometry, and SLAM. *The International Journal of Robotics Research*, 2017.