Supplementary Materials for

Dynamic obstacle avoidance for quadrotors with event cameras

Davide Falanga*, Kevin Kleber, Davide Scaramuzza

*Corresponding author. Email: falanga@ifi.uzh.ch

DOI: 10.1126/scirobotics.aaz9712

The PDF file includes:

- Fig. S1. Monodimensional example to explain the working principle of event-based detection of moving obstacles.
- Fig. S2. Time statistics of the events belonging to static and dynamic regions.
- Fig. S3. The quadrotor platform we used in our outdoor experiments.
- Fig. S4. Ego-motion compensation computation time as function of the number of events.
- Fig. S5. Clustering computation time as function of the pixels count.
- Fig. S6. Detection of objects having different sizes and shapes.
- Fig. S7. Detection of multiple objects simultaneously.
- Fig. S8. Sequence of detection.
- Fig. S9. Obstacle ellipsoid.
- Fig. S10. Repulsive potential.
- Fig. S11. Attractive potential.
- References (56, 57)

Other Supplementary Material for this manuscript includes the following:

(available at robotics.sciremarg.org/cgi/content/full/5/40/eaaz9712/DC1)

- Movie S1 (.mp4 format). Outdoor dynamic experiments.
- Movie S2 (.mp4 format). Explanation of the working principle of the event-based detection algorithm.
Supplementary Materials

Time Statistics of Events to Detect Moving Obstacles

To provide an intuitive example of how and why our algorithm successfully classifies static and dynamic events, Fig. S1 shows the simplified case of a mono-dimensional event camera (i.e., an event camera having only one row), rotating in a plane while observing both a static and a dynamic object. The dynamic object (in red) moves from left to right, while the event camera rotates counter-clockwise.

In the center of the figure, we consider a time window spanning from an initial time \( t_1 \) to a final time \( t_5 \), and we discretize this interval into five time instants to visualize the sequence of events generated by both the motion of the camera and the dynamic object. Let us assume that at time \( t_1 \), both objects generate an event due to the motion of the camera: the static object fires an event at pixel \( p_1 \), the dynamic object at pixel \( p_2 \). At time \( t_2 \), the motion of the dynamic object causes another event at pixel \( p_3 \), while at time \( t_3 \) the motion of the camera generates events at pixels \( p_2 \) (static) and \( p_4 \) (dynamic). The same concept applies to times \( t_4 \) and \( t_5 \). After collecting all these events, if we motion-compensate them to remove the effects of the motion of the camera, we obtain a situation like the one depicted at the bottom of the center part of the image, where multiple events get back-projected into the same pixel location.

On the right side of the figure, we report the time statistics of the event project into pixels \( p_1 \) to \( p_4 \), which are the only ones having motion-compensated events. As one can see, the events belonging to the static part of the scene are equally spread across the time window, while the events fired due to the motion of the dynamic object are concentrated either at the beginning, the center, or the end of the window. If we now compute the mean timestamp of all the events falling in each pixel, subtract the mean of all the events, and normalize it by the length of the time window, we obtain a score for each pixel spanning between \(-1\) and \(1\). We expect events belonging to the static part of the scene to have a score of approximately zero since they contain events spread across the entire window more or less uniformly. On the contrary, events belonging to the dynamic part of the scene have scores that can span between \(-1\) and \(1\), depending on where they are concentrated within the time window. In particular, the events generated by the dynamic object at the beginning of the window have a score of \(-1\), those fired at the center of the window have a zero score, while those generated at the end of the window have a score of approximately \(1\). Since we are interested only in the latest position of the dynamic obstacles, we discard non-positive scores, taking into account only events with a score above zero. Fig. S2 confirms the expected pattern in the statistics of the events in a time window on real data: the first row shows the mean timestamp of a region belonging to the static part of a scene, where the histogram clearly highlights an equal distribution of the events across the entire window; the second row shows the same data for a region belonging to the dynamic part of a scene, where the events are concentrated towards the two ends of the time window.

It is important to notice that, for the sake of making this example simple enough, we only considered one type of event (either positive or negative), while in a real case, each object
generated both positive and negative events, simultaneously. However, the principle can be easily extended to events with polarity. Additionally, we invite the reader to notice that not every motion can be compensated, but rather only rotations and roto-translations with respect to a planar scene. Indeed, these motions can be modeled as homographies, and the events they generate can be motion-compensated. However, since we only consider the events that fire within a very short time window, the majority of the scene moves by a very small amount of pixels. Therefore, we can approximate the camera motion as a homography. The mathematical description of the motion compensation algorithm is provided in Sec. Ego-Motion Compensation of the Events.

**Impact of Using a Simplified Ego-Motion Estimation Algorithm**

The ego-motion estimation and compensation algorithm used in our system (Sec. Ego-Motion
Compensation of the Events) assumes the optical flow and, therefore, the events to be only due to the rotation of the camera. In order to account for the linear motion of the camera, one would need some sort of depth estimation, which would render the entire algorithm significantly slower. Nevertheless, using a rotational model to explain the events generated by the ego-motion of the camera is sufficiently accurate to guarantee good overall performance, as shown by the tables in Sec. Accuracy and Success Rate. From a theoretical perspective, this is justified by the fact that translational and rotational optical flow are often very similar and, therefore, very hard to tell apart. As shown in Fig. 15.7 of (56), translation along the X-axis is almost like a rotation around the Y-axis. One needs a long observation time and a smaller focal length (wider field of view) to better distinguish them. Thanks to such a similarity, one can fit a rotational motion to the translational flow to explain the events generated by the latter. This intuition was corroborated in (57), which shows in the supplementary video the impact of ignoring the linear velocity of the motion. This video was generated using only the IMU, without any depth estimation: one can see that the motion-compensated images obtained when the IMU data is taken into account look quite sharp everywhere on the image plane.
(a) A scene without moving objects. The patch highlighted in red in the left mean timestamp image belongs to a static part of the scene, and is reported in the center figure. On the right side, we show the histogram of all the ego-motion events belonging in such patch.

(b) A scene with one moving object. In this case, we selected a patch belonging to a dynamic part of the scene, namely a ball thrown through the field of view of the camera and moving from left to right in the frame. As one can notice, several pixels report a high mean timestamp, and the histogram of all the ego-motion compensated events belonging to the patch confirms this trend.

Fig. S2. **Time statistics of the events belonging to static and dynamic regions.** A figure reporting the statistics of the events within a single time window for two cases: no dynamic object in the scene (top row) and one dynamic object in the scene (bottom row). For each row, we report: on the left, the mean timestamp image, with color-code shown on the right side representing the mean of the timestamps of all the events back-projected to each pixel location; in the center, a $4 \times 4$pxl patch belonging to a static part (top row) or dynamic part (bottom row) of the scene, taken from the region highlighted in red in the mean timestamp image; on the right, the distribution of the events belonging to that patch. As one can notice, the events in a patch belonging to the static part of the scene report a fairly uniform distribution of their timestamps within the window. Conversely, the events belonging to a dynamic object are very concentrated towards one side of the window (in this case, the end).
Fig. S3. The quadrotor platform we used in our outdoor experiments. The quadrotor platform we used in our outdoor experiments. The following components are highlighted in the picture: (1) the Nvidia Jetson TX2, running the obstacle detection and avoidance algorithm, as well as the high-level controller; (2) the Lumenier F4 AIO Flight Controller; (3) the two Insightness SEES1 cameras, in a vertical stereo setup; (4) the Qualcomm Snapdragon Flight board, used for state estimation.
Fig. S4. Ego-motion compensation computation time as function of the number of events. Time necessary to perform the ego-motion compensation as a function of the number of events generated.

Fig. S5. Clustering computation time as function of the pixels count. Time necessary to perform the clustering of the scene’s dynamic part, depending on the amount of pixels belonging to moving objects.
Fig. S6. Detection of objects having different sizes and shapes. Our algorithm is able to detect different kinds of objects, as shown in this figure. Each row shows the detection of different objects, depicted in the pictures in the first column. From top to bottom: a small-sized ball, a box, a whiteboard marker, a frisbee, a quadrotor, and a bowling pin. These objects were detected using our stereo setup, without any prior information about their shape or size. As one can notice, the frame provided by the on-board camera (second column) presents some motion blur due to the speed of the object, which however is not a problem for our event-based detection algorithm (last column).
Fig. S7. Detection of multiple objects simultaneously. An example of our algorithm detecting and clustering multiple moving obstacles. (S7a) The frame from the on-board camera, where three moving obstacles, manually circled in red, are visible. (S7b) The mean-timestamp image. (S7c) The mean-timestamp image after thresholding: green represents the static part of the scene, purple indicates events belonging to dynamic obstacles. (S7d) Clustering of the events belonging to different dynamic obstacles.
**Fig. S8. Sequence of detection.** A sequence captured during one of our experiments, where the quadrotor is hovering indoors and an object is thrown towards it with the purpose of evaluating the sensing pipeline. Each column represents a different time, more specifically: $t_0 = 0\text{s}$ (first column), $t_1 = 0.05\text{s}$ (second column), $t_2 = 0.10\text{s}$ (third column), $t_3 = 0.15\text{s}$ (fourth column). The first row reports the frame captured by the on-board camera. The second row shows the events, generated by both the motion of the vehicle and the moving obstacle, collected within the last time window of size $\delta t = 10\text{ms}$, where blue represents positive events, and red represents negative events. The third row shows the ego-motion compensated events belonging only to the dynamic part of the scene, obtained applying the algorithm described in Sec. Ego-Motion Compensation of the Events.
Fig. S9. **Obstacle ellipsoid.** Construction of the obstacle’s ellipsoid in the world’s frame of reference from the clustered data in the image plane. A minimal volume ellipsoid is fitted around the six projected points using an iterative approach.

Fig. S10. **Repulsive potential.** Plots illustrating the two different types of repulsive forces described in this work.
Fig. S11. Attractive potential. Illustration of the attractive force for different values of $\gamma_a$. 