Event feature detection

In the context of computer vision and image processing, image features are distinctive landmarks in the images, preferably unsusceptible to point-of-view, scale, and the aperture problem. Features are important as they provide information on the image, which can be used for recognition, matching, reconstruction, and tracking, among many other applications. Many types of features can be considered, such as edges, corners, blobs, ridges, and shapes.

The process of identifying features in an image is called feature detection, and multiple detectors have been described in the literature, dependent on the features of interest, such as Canny and Sobel detectors (for edges), Hough transform (for shapes), Laplacian operator (for blobs), and Harris detector (for corners).

For event-based cameras, new types of features, as well of detectors, are being proposed, as classical techniques are not easily transferable in most cases, or result in a non-negligible performance decrease, due to conversion overhead from asynchronous events to frames. However, corners seem to prove as interesting features to use, as not only can they be used to match features between the event stream and full frame images, allowing for sensor fusion, native event algorithms are available, which leverage the potential of events, namely speed and event independence.

Classic corner detection

The typical example for a classic corner detector is the Harris Corner Detector. It works using the following steps:

1. Compute the x-wise and y-wise partial image derivatives
2. Compute the second-order derivatives and , and cross-derivatives
3. Compute the second-moment matrix
4. Compute the Harris score
5. Detect local extrema whose Harris score is greater than the set threshold

The partial derivatives are computed by applying a Sobel derivative kernel (usually 3x3 or 5x5 kernels) to the whole image, producing the x-wise and y-wise partial image derivatives. From these derivatives, we can define the vector and the second-moment matrix for each pixel (), defined as , where is a Gaussian weighting function centred around , which controls the “sharpness” of the edge.

Then, we can compute the Harris score as

Where k is an empirical value, .

Finally, if , we consider the pixel as a corner.

This will produce corner blobs. In order to select a single pixel to represent the corner, we select the local extrema (the pixel with the highest Harris score).

Another option to evaluate the presence of corners is to analyse the eigenvalues of M. If both eigenvalues are low, no interesting features are detected. If one is low, but the other is high, we are in the presence of and edge. Lastly, if both eigenvalues are high, the pixel is likely a corner. As such, one interpretation of the Harris detector is that corners are identified by finding the intersection of edges.

With this in mind, an alternative for the corner analysis, is to check the value of the lowest eigenvalue of M (), through the approximation

And then this value is used as a corner criterion.

Event-based corner detection

Due to the nature of events, gradient operators are not possible (at least directly applied to the event stream), since there is no image on which to apply them, and multiple techniques have been proposed, of which three will be presented.

Space-time detection

This method relies on the space-time properties of events, and creates a 3D representation, containing the spatial position of an event , as well as the time it was received. In this representation, edges moving with uniform linear speed create planes (stack of lines at different instants), and corner movement creates lines (stack of points at different instants).

As such, this technique tracks moving edges by fitting planes in this 3D representation, implicitly estimating the speed of the moving edge (optical flow). Each new event is matched to the previously estimated planes, and the estimates are updated.

A close up of a logo

Description automatically generated

The way this technique identifies (and tracks corners) is by detecting intersection between these planes, as these intersections correspond to the corner movement through a period of time.

A close up of a map

Description automatically generated

Event-based Harris Corner Detector

This technique relies on the Surface of Active Events (SAE), a representation system for events, which keeps track of the timestamp of the most recent event for any given pixel, regardless of polarity. Indeed, it is a spatial representation ( coordinates, corresponding to each pixel), which can be discretized by assigning a value to each pixel based on its timestamp. The image shows an example of the SAE, where the events are represented from white to grey, as they go from more recent to older.

A picture containing food, white, sugar

Description automatically generated

Since this discretized representation is now a frame in the classical sense, we can apply the Harris Corner Detector directly to the SAE and identify the corners from these results.

A more efficient implementation relies on considering only the neighbouring region of an event as it arrives, instead of the whole SAE. As such, only a subset of the SAE is analysed. Since each event, and consequently, each subset, is independent on the other subsets (provided the subsets do not overlap), parallel implementations are possible, and also improve speed.

SAE-based corner detector

This technique also relies on the SAE representation of events but does not perform any computations. Rather, it performs only comparison operations on a local neighbourhood around the relevant event.

As each event is received, its timestamp is compared with the neighbouring pixels using circular segments (for isotropic response and efficiency), and checked if patterns similar to the one in the image are present (contiguous pixels with decreasing timestamps), as these are typical corner patterns.

Though this method is not as effective, it is much faster, as no computations are performed, and each event can be processed independently (and concurrently in a parallel fashion).

A picture containing toy

Description automatically generated

Concepts

Feature - features are distinctive landmarks in the images. They are important as they provide information on the image, which can be used for recognition, matching, reconstruction, and tracking, among many other applications. Many types of features can be considered, such as edges, corners, blobs, ridges, and shapes.

Corner – an example of a feature. Can be identified in both classical cameras, as well as event-based cameras, allowing for comparisons between the two.

Harris detector – a corner detector which computes the x-wise and y-wise derivatives in the image to detect corners, resulting from the intersection of edges.

SAE - a map with the timestamp of the latest event at each pixel

References

Paper Harris - A Combined Corner and Edge Detector

Paper Benosman - Asynchronous event-based corner detection and matching

Paper event Harris - Fast event-based Harris corner detection exploiting the advantages of event-driven cameras

Paper Scaramuzza - Fast Event-based Feature Detection