A Appendix

A.1 Preprocessing schematic

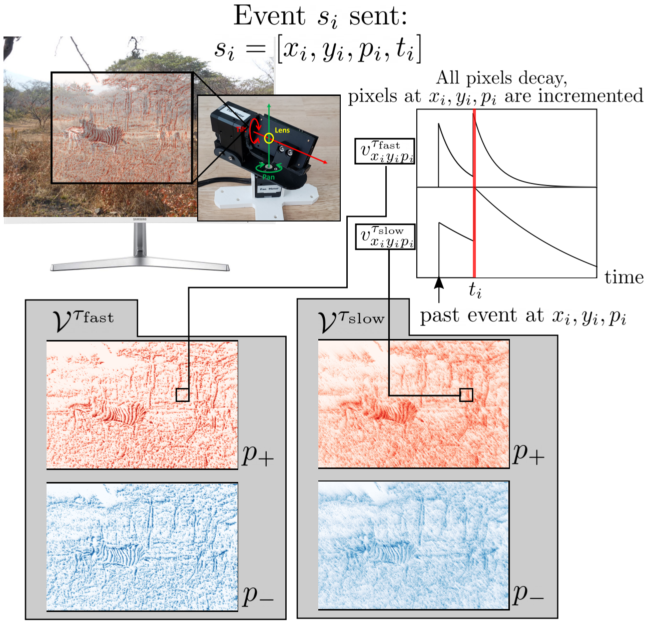
Event s_i sent:
$s_i = [x_i, y_i, p_i, t_i]$
All pixels decay,
pixels at x_i, y_i, p_i are incremented
Lens
Pan
$v^{\tau_{\text{fast}}}_{x_i y_i p_i}$
$v^{\tau_{\text{slow}}}_{x_i y_i p_i}$
time
t_i
past event at x_i, y_i, p_i
$\mathcal{V}^{\tau_{\text{fast}}}$
$\mathcal{V}^{\tau_{\text{slow}}}$
p_+
p_-
p_+
p_-

Fig. 1: Preprocessing pipeline. (1) Pan tilt stage moves dynamic vision sensor (DVS) in front of computer monitor presenting static COCO images. (2) Event s_i is sent with its spatial index x_i, y_i, polarity index p_i and time t_i. (3) $\mathcal{V}^{\tau}$ is updated by first decaying all pixels according to the time elapsed since the last event s_{i-1} and time constant τ. The pixel corresponding to event s_i is incremented by 1. (right) Two different τ decay profiles are shown for a single pixel, v_{x_i,y_i,p_i}. We use two τ's in our pipeline, 10 ms and 20 ms, referred to as fast and slow respectively. (bottom) The four images of leaky integrators, one for each τ and polarity pair.

A.2 τ selection

We argue that in order to capture a range of pixel speeds $[0, R]$ with spatial filter of width w pixels, the pair of $\tau = \{\tau_{\text{slow}}, \tau_{\text{fast}}\}$ should be selected such that:

$$\frac{\tau_{\text{slow}} - \tau_{\text{fast}}}{\tau_{\text{slow}}\tau_{\text{fast}} \log(\tau_{\text{slow}}/\tau_{\text{fast}})} = \frac{2R}{w}$$

This equation arises from calculating the time at which the difference between two exponential functions reaches its maximum.

A.3 Event-based update of leaky integrating images

To make our preprocessing event-based (rather than updating every pixel using every time step), we note that the elements of $\mathcal{V}^{\tau p}(t)$ only need updating if t is the time of an event. For a sequence of K events $\mathcal{S}$ with monotonically non-decreasing event times $\{t_i\}_{i=1}^{K}$, the value at a pixel $v_{xy}^{\tau p}$ in our image of leaky integrating pixels $\mathcal{V}^{\tau p}$ can be updated sequentially:

$$v_{xy}^{\tau p}(t_i) = v_{xy}^{\tau p}(t_{i-1}) e^{(t_{i-1} - t_i)/\tau} + \delta_{x,x_i} \delta_{y,y_i} \delta_{p,p_i}$$

where δ is the Kronecker delta function. The first term controls the decay and the last term increments only the leaky pixels corresponding to the event location and polarity of event s_i.

A.4 Data collection

Our DVS is mounted on a pan-tilt actuated stage as shown in Supplementary Figure 1 in front of a computer monitor. This monitor displays an image from the COCO dataset one at a time. For each image, our DVS records for 15 seconds while our pan-tilt stage generates saccade-like motions. A saccade beginning at time t_0 with a pan angle $\theta_{t_0}^{P}$ and starting tilt angle $\theta_{t_0}^{T}$ moves along the path of minimal distance toward an ending point $(\theta_{t_1}^{P}, \theta_{t_1}^{T})$ with a randomly selected maximum speed $|V_{max}|$. The ending point $(\theta_{t_1}^{P}, \theta_{t_1}^{T})$ is selected uniformly at random from the range of θ within the monitor boundaries: $\theta_{t_1}^{a} \sim U(\theta_{min}^{a}, \theta_{max}^{a})$, where $a \in \{P, T\}$. Because the range of angles is small relative to the whole sphere, the problem of sampling uniformly from a sphere is negligible. The time length of the saccade, $\varDelta t = t_1 - t_0$ is unconstrained. As a result, there are a variable number of saccades per image, but occur roughly once per second.

We use a randomly selected subset of 1300 COCO images, 1000 for training and 300 for testing. From these videos, we subselect a set of time points for training and testing, $T = \{t_0, \ldots, t_N\}$, such that the marginal distribution of ω_{pan} and ω_{tilt} are approximately uniformly distributed in the range of $[-60, 60]$ and $[-40, 40]$, respectively.

A.5 Pan-tilt motor control

A DVS data capture system is presented in this section. A pan-tilt camera is implemented using two DYNAMIXEL (XH430-V350-R) motors as shown in the picture (Supplementary Fig. 1). Each pan and tilt axes are aligned with camera focal point so that there is no translational motion during the data capturing. Pan-tilt angles and velocities are uniformly randomized within boundaries that keep the field of view remains in the monitor screen. Positions and velocities of each motor are recorded with approximately 110Hz while DVS data is recorded with approximately 330Hz.

A curved monitor (Samsung CF391) is used to minimize an image distortion due to camera motion, and the data is captured in a dark box to eliminate a flickering noise from external light sources.

A.6 Tilt neuron tuning

See Supplementary Figure 2. Tilt neurons follow a similar tuning pattern to pan neurons as described in the main text.

A.7 Contrast Maximization

This approach takes a spatial and temporal window of events $\mathcal{S} = \{s_i\}_{i=0}^{K}$ and warps their location x_i, y_i by a proposed optical flow u, v to a new location x'_i:

$$x'_i = x_i - (t_i - t_0)u, \quad y'_i = y_i - (t_i - t_0)v \tag{1}$$

And then computes H, the image of warped events:

$$H(x, y; u, v) = \sum_{k=0}^{K} \delta_{xx'_i}\delta_{yy'_i}$$

where δ is the Kronecker delta. Finally, they compute the (empirical) variance, σ^2 of H as a function of u, v:

$$\sigma^2(H(X, Y, u, v)) = \frac{1}{N_p}\sum_{xy}(h_{xy} - \mu_H)^2$$

In their paper, they perform gradient ascent to find u, v that maximize σ^2. Conceptually, this approach assumes events in a spatiotemporal window have spatially uniform and temporally constant optical flow. Then, by warping events backward in time according to a proposed optical flow and the time past since the start of the window, all the events corresponding to the same stimulus should stack up in the same location. If this is correct, the resulting warped image will have some high values and many zeros, and thus a high variance.

In order to compare local CM predictions with our own network, we also extend their method to include heuristic confidence metrics, as well as using their method with our own confidence approach. In particular, for CM predictions $\widetilde{u}_{xy}(t), \widetilde{v}_{xy}(t)$ in a 15×15 window centered spatially at x, y, and in time at t, we define the CM global prediction as:

$$\widetilde{u}_{\text{global}}(t) = \sum_{xy}\frac{w^u_{xy}(t)}{\sum_{mn} w^u_{mn}(t)}\widetilde{u}_{xy}(t)$$

where $w^u_{xy}(t)$ is the heuristic weight for optical flow u at location x, y at time t. We compare a few different strategies to calculate $w^u_{xy}(t)$. We use the mean $w^u_{xy}(t) = \mu_{H_{xy}(t)}$, the variance $w^u_{xy}(t) = \sigma^2(H(X, Y, \widetilde{u}_{xy}(t), \widetilde{v}_{xy}(t)))$, and our own confidence scores $w^u_{xy}(t) = c^u_{xy}(t)$ and will refer to them in the comparison results section. For confident local predictions of CM using mean or variance (in parenthesise in Table 1) we use warped image regions which are in the top 10% of either score. This was selected by brute force optimization. The objective of these confidence heuristics is to identify if the features calculated by CM easily provide a confidence metric, or if additional calculations would be necessary to identify accurate CM regions.

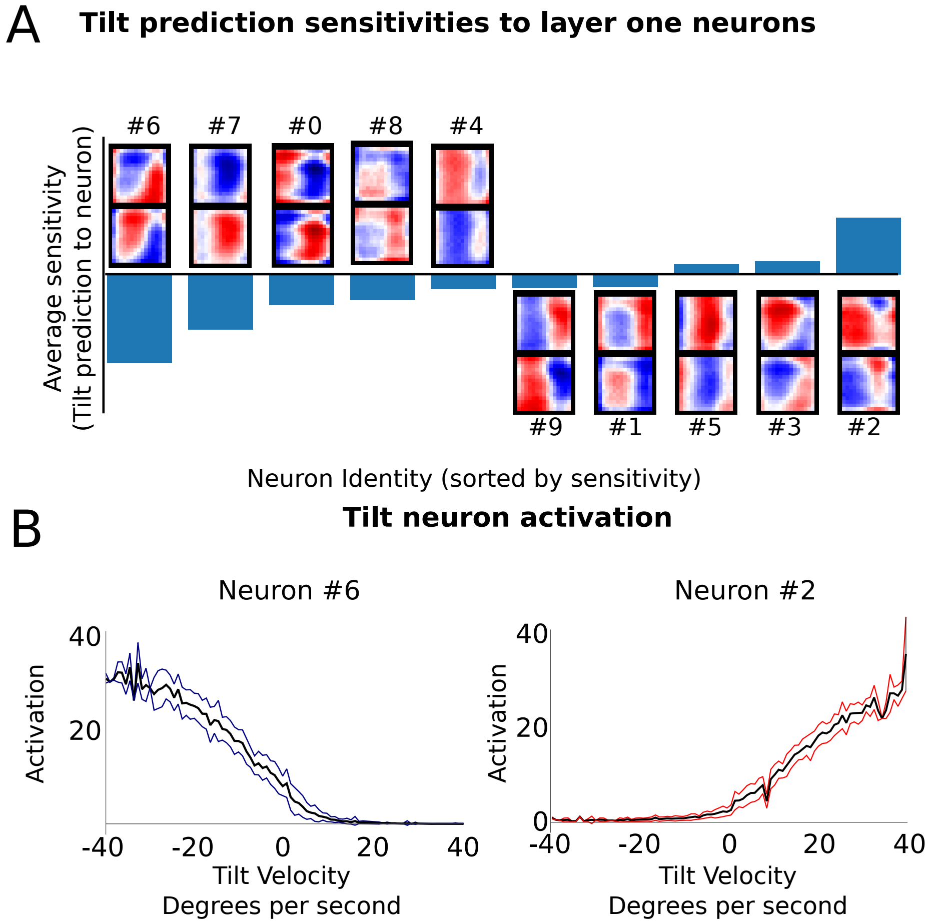

Fig. 2: Analysis of visual motion network. Panel A shows the sensitivity of tilt predictions to the activity of all ten neurons in layer 1. Accompanying each sensitivity is a heatmap of the spatial weights of that neuron. There are two spatial filters for each neuron, one for each time constant. Mean activation across the DVS-COCO testing set is shown in black, with shading to show one standard deviation above and below the mean. Color indicates negative velocity (blue) or positive velocity (red).