LAB ROTATION

IMAGE RECONSTRUCTION USING NEUROMORPHIC CAMERAS

Sudeshna Bora
Guided By - Prof. Dr. Guillermo Gallego
Bernstein Center for Computational Neuroscience, Berlin

November 4, 2020

Contents

I Abstract

Neuromorphic cameras (also called event-based cameras) are a paradigm shift from conventional cameras and provides potential advantages over conventional frame-based cameras (high speed, high dynamic range, low power consumption) [13]. Due to their different way of acquiring and storing visual data, the algorithms used for event-based processing are also markedly different. My lab rotation focuses on the task of image intensity reconstruction algorithms from event data. This is a quasi theoretical lab rotation which focuses on the following deliverable. First, I provide an introduction to event-based cameras (section II) and review the literature on image reconstruction algorithms in this paradigm (section III). Then I decide to focus on comparing two methods. For the first one, I implement the unsupervised sparse dictionary learning algorithm proposed by Barua et. al. [4] (subsection IV.1), since it is not publicly available. As second algorithm, I use the convolutional recurrent neural network proposed by Scheerlinck et. al. [34] (subsection IV.2). In conclusion, this report focuses on our findings from the implementation of the first algorithm and on a comparative analysis of the two algorithms (section V).

II Introduction

Neuromorphic (i.e., event-based) cameras are biologically inspired visual sensors that unlike traditional cameras do not capture entire intensity of the visual space at a given time. Instead, the pixels of these sensors work independently (there is no global or rolling shutter) and capture any change in brightness (or "intensity") that is sensed at the photoreceptor (actually, the logarithm of brightness) if it is greater than a certain threshold [6]. These pixel-wise intensity changes are called "events" and they are timestamp with very high temporal resolution (in the order of microseconds). Figure 1 illustrates the different working principles of frame-based and event-based cameras. The asynchronous nature and sensitivity to scene dynamics makes neuromorphic event cameras superior to conventional cameras in terms of low latency, high temporal resolution, high dynamic range and low power consumption.

The first neuromorphic camera was developed by Mahowald and Mead [20]. The Dynamic Vision Sensor (DVS) [19], the Asynchronous Time-based Image Sensor (ATIS) [28] and the Dynamic and Active Pixel Vision Sensor (DAVIS) [7] are the three main event camera designs. The DVS is inspired by the transient visual pathway till the ganglion cells (dorsal stream). The ATIS, has in addition to the DVS output, grayscale events that are inspired by the "what" visual pathway through the parvo layer of the brain (ventral stream) [29]. The DAVIS combines a DVS with a synchronous frame-based camera, on the same pixel array. The workings of the event camera can be juxtaposed to the flow of stimuli from visual receptive field to deeper brain layers with the help of neurons that fire if the incoming stimuli is above a threshold. Similarly, spikes (events) are produced by pixels of the event camera when a threshold is crossed. This information then travels from the first layer to deeper layers of the camera circuitry.

II.1 Working Principle of Event-based Cameras

This subsection develops a mathematical model to describe how an event-based camera works (and the output that it produces).

As stated above, event-based cameras respond to brightness changes in the scene asynchronously and independently for every pixel. Each pixel memorizes the log intensity (log photocurrent $L = \log(I)$) each time it

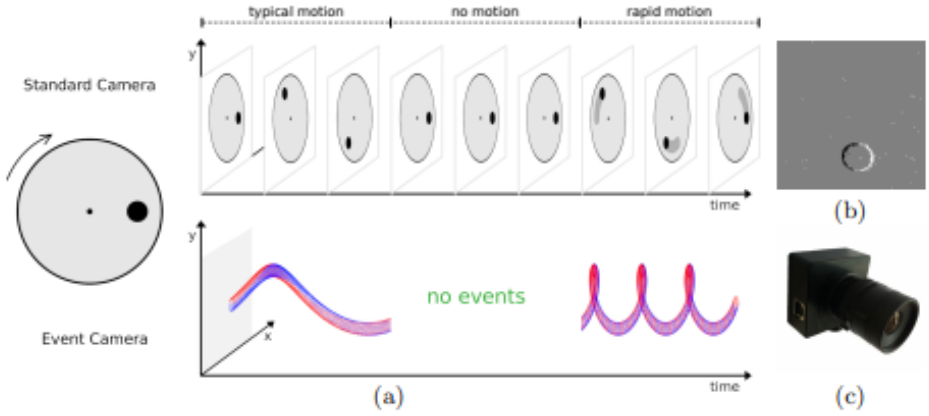


Figure 1: Difference between traditional camera (top) and event-based camera (bottom) when they both view a small black circle on a rotating disk. The traditional (video) camera acquires frames at a constant rate regardless of what happens in the scene. By contrast, in the event-based camera only the pixels that *change intensity* (i.e., those corresponding to the moving edges of the black circle) produce an output; in this case, a "spiral" of events in space-time. Image courtesy of [17].

sends an event, and continuously monitors for a change of sufficient magnitude from this memorized value to trigger the next event. Ideally in the noiseless case, an event e_k, which is described by a tuple (X_k, t_k, p_k), is triggered at pixel location $X_k = (x_k, y_k)^\top$ at time t_k as soon as the brightness increment since the last event at that pixel,

$$\Delta L(X_k, t_k) \doteq L(X_k, t_k) - L(X_k, t_k - \Delta t_k), \tag{1}$$

crosses a threshold $C > 0$:

$$\Delta L(X_k, t_k) = p_k C. \tag{2}$$

Here, Δt_k is the time since the last event, $p_k \in \{+1, -1\}$ is the polarity signifying the sign of brightness change [19] (positive if there was a brightness increase –also called ON event–, and negative if there was a brightness decrease –an OFF event–). The threshold or contrast sensitivity C is determined by the pixel bias currents [35, 26] which set the speed and threshold voltages of the change detector. Positive and negative events may be triggered according to different thresholds.

Linearized Event Generation Model. For small Δt_k, Equation 1 can be approximated using Taylor's expansion $\Delta L(X_k, t_k) \approx \frac{\partial L}{\partial t}(X_k, t_k)\Delta t_k$, thus the events can be interpreted as the temporal derivative:

$$\frac{\partial L}{\partial t}(X_k, t_k) \approx \frac{p_k C}{\Delta t_k} \tag{3}$$

Approximating Equation 2 under brightness constancy assumption and small Δt_k gives us

$$\Delta L \approx -\nabla L \cdot v \Delta t_k. \tag{4}$$

This equation states that the brightness change was produced by a brightness gradient $\nabla L(X_k, t_k) = (\partial_x L, \partial_y L)^\top$ (i.e., an edge) moving with velocity v on the image plane during a time Δt_k, that is, over a displacement $\Delta x = v\Delta t_k$. No events are generated when the motion is parallel to the edge (as implied by the dot product in the formula) and fastest events (minimal time) are generated if the motion is perpendicular to the edge [8].

Noise Model. The above equations are idealized and do not account for sensor noise and transistor mis-

match. A more realistic model accounts for the stochasticity of the model and can be represented by a probability function of local illumination level and sensor parameters. The DVS [19] paper suggests a noise model described by a Gaussian distribution centered at the contrast threshold C. There are other, more complicated, probabilistic models that account for the likelihood of event generation as being proportional to the magnitude of the image gradient [9] or being modeled by a mixture distribution in order to be robust to sensor noise [14].

II.2 Event Representation

The asynchronous event stream produced by event-based cameras is often transformed into alternative, more familiar representations that are compatible with non-spiking algorithms and facilitate the extraction of meaningful information to solve the task at hand. In this subsection we discuss some of the forms in which event data are represented. Specifically we focus on two event types that will be used in section IV.

The simplest representation is the individual events $e_k = \{X_k, t_k, p_k\}$. As these do not provide much information on the event/visual field; it is used conjointly with probabilistic filters or SNN which has additional events information stored [14]. In order to aggregate the spatial and temporal information from multiple events, event packets and frame/2D histogram are used. Events $\mathscr{E} = \{e_k\}_{k=1}^{N_e}$ in a spatio-temporal neighbourhood are processed together in an event packet. These packets are then converted to an image to form a 2D histogram. Based on spatio-temporal neighbourhood characterisation, other forms of representations have been developed like Time Surfaces[11], 3D point sets [5] or 2D point sets on the image plane [25].

Event data can be represented by summing the number of events at each pixel (event frame or 2D polarity histogram, also called brightness increment image). However this representation decreases the performance in case of large motions (i.e., large event accumulation). Event frames can be motion-compensated [15] when combined with a motion hypothesis. To mitigate the above decrease in performance (i.e., to decrease information loss due to accumulation), Zhu et.al [38] and Ye et. al [36] proposed to combine even frames (which count the number of events per pixel) and time surfaces (which record the timestamp of the last event at each pixel). This combined representation was used to train artificial neural networks that learned to estimate these motion (optical flow) in self- and unsupervised manners.

Building on top of it, Zhu. et. al [39] proposed event tensors or "voxel grids". Voxel grids or 3D histograms are a better representation than 2D histograms because they preserve better temporal information. A voxel grid is a 3D histogram where each voxel represents a particular pixel and timestamp. Each event's polarity can be accumulated on a voxel or spread among closest voxels using a kernel κ (weighted accumulation). Barua et. al's algorithm (subsection IV.1) used m consecutive event frames to represent the event data, which is equivalent to a zero-th order interpolated voxel grid with m time slices.

If there are N input events and B temporal bins (slices) in the voxel grid, the timestamps of the events are scaled to the rage $[0, B-1]$ using the simple formula $t_i \mapsto b_i$:

$$b_i = (B-1)\frac{t_i - t_1}{t_N - t_1}. \tag{5}$$

Now each event has integer coordinates (x_i, y_i), and a fractional coordinate $b_i \in [0, B-1]$. Events are accumu-

lated in the voxel grid (volume) V according to formula:

$$V(x,y,b) = \sum_{i=1}^{N} p_i \, \delta(x - x_i) \, \delta(y - y_i) \, \kappa(b - b_i), \tag{6}$$

$$\kappa(a) = \max(0, 1 - |a|) \tag{7}$$

Where δ is the Kronecker delta. If the x_i, y_i coordinates of the events were real numbers, due to, e.g., a camera lens undistortion procedure, then the kernel κ would be applied to these spatial coordinates, too. Otherwise, it is not necessary since they are already integer values and the Kronecker delta suffices. In the formula above, V is a discretized volume, of dimensions $W \times H \times B$ (image with, image height, number of time bins). In the tensors, the time domain is treated as channels of traditional images. 2D convolution are performed across the x, y spatial dimensions. As we will mention in section IV, Scheerlinck et. al. [34] use voxel grids to represent event data that is fed to their artificial neural network.

III Literature Review of Image Reconstruction from Event Cameras

Reconstructed brightness images act as an interface between event cameras and conventional computer vision algorithms. Rebecq et al. [31] have shown that conventional, frame-based algorithms when applied on reconstructed images perform superior to dedicated event algorithms in classification and visual inertia odometry. Additionally, reconstruction of image aids human with visualization and interpretation. This section deals with the history of development of image reconstruction algorithms with an aim to provide chronological view of the development in this area.

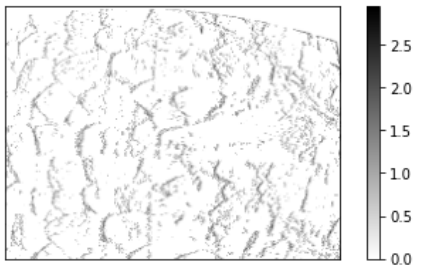
The initial algorithms [10] [16] for image reconstruction worked under the motion restriction of rotational camera motion, static scene and brightness constancy. These algorithms employed message passing between pixels in a network to jointly estimate characteristics. Kalman filter was used on the pixels to estimate brightness gradient and then poisson reconstruction was used to give absolute brightness. Motion restriction was replaced by using variational frameworks with penalty terms. These regularising assumptions [21] enabled reconstruction for motions and scenes. Temporal smoothing [33] for reconstruction and fusion of events and frames did away with spatial regularization and also reduced noise and artefacts. Currently, data driven deep learning algorithms [32] [23] are used in reconstruction. They have been able to produce more natural looking images and mitigate visual artefacts.

IV Description of the Model and Results

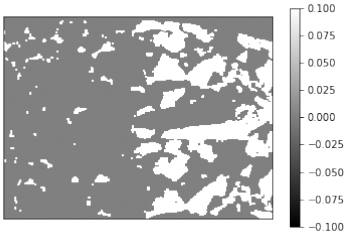
IV.1 Image Reconstruction Using Sparse Dictionary Learning[1]

The first algorithm [4] is a sparse dictionary learning algorithm. It develops a patch-based model for event streams by training an over-complete dictionary. For the input dataset, we generated events using the event camera simulator, ESIM [30]. The following algorithm [4] (additionally with an offset) was also used to create events for each pixel from a reference image.

[1] url for implementation : https://github.com/SudeshnaBora/Computer_Vision/tree/master/code



(a) Image of the positive event voxel grid where events are created by ESIM



(b) Image of the negative event voxel grid where events are created by Algorithm

Figure 2: Sample time slices of voxel grids of events. The texture corresponds to some poster of rocks on a flat wall, as in the [24] dataset. The edges that we see represent the borders of those rocks.

Input: threshold τ, number of frames m, pixel intensity for the original image v_0
$r = v_0$
for i = 1 to m do
  if $v_i > r$ then
    $E_{p_i} = \lfloor \log_\tau \frac{v_i + \text{offset}}{r} \rfloor$
    $r = \tau^{E_{p_i}} r$
  else
    $E_{n_i} = \lceil \log_\tau \frac{v_i + \text{offset}}{r} \rceil$
    $r = \tau^{E_{n_i}} r$
  end if
end for
Output: event data E_p and E_n

The events were converted into a voxel grid. The polarity in a voxel grid was determined by linear voting of adjacent voxels. Figure 2 shows the results of the event generation algorithm (Figure 2a) used in ESIM and by the above stated subsection IV.1, for one of the time slices of the voxel grid. Moving forward we would be using the events generated from the simulator ESIM [30].

In the training phase, we computed the x and y gradients of the logarithm of the intensity image. The gradients were computed by using two 1-D filters (i.e., applying spatial convolution):

$$f_1 = [-1, 1], \quad f_2 = f_1^T \tag{8}$$

The input dataset contains 100000 patches (X) from the gradient and the event frames. The first $2n$ rows contain the gradients and the remaining $2mn$ rows contain the events from m frames, where n is the number of pixels in each patch. We used the K-SVD algorithm [3] to train the sparse patch-based dictionary. That is, we solved the problem:

$$D^* = \arg\min_{D,\alpha} ||X - D\alpha||_2^2 \quad \text{s.t.} \quad ||\alpha||_0 \leq K_s,$$

$$D = \begin{bmatrix} D_g \\ D_e \end{bmatrix}$$

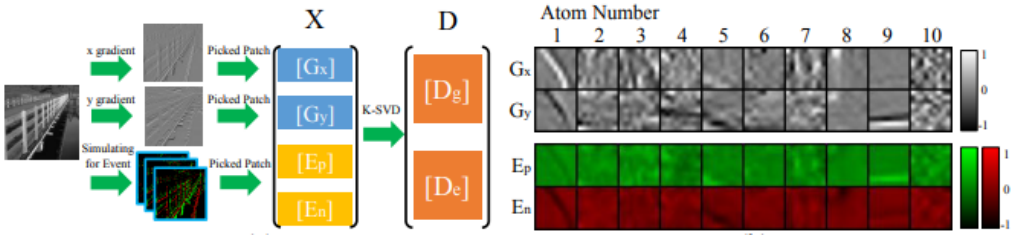


Figure 3: Training phase using a Dataset. A dictionary of patches D is learned from synthetic data. The dictionary consists of atoms. Each atom has a spatial component (the part of the patch corresponding to the x and y image gradients, G_x, G_y) and a temporal component (the part of the patch corresponding to the event frames E_p, E_n). The dictionary is learned by applying the K-SVD clustering algorithm on a large collection of input patches X.

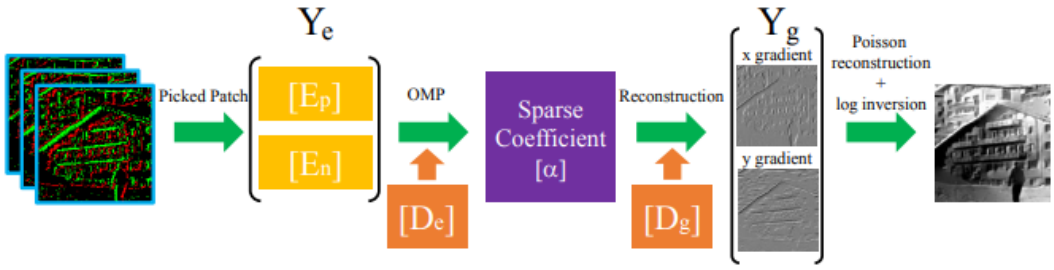


Figure 4: Testing phase (i.e., inference): reconstruct an intensity image from raw event data. Positive and negative events are converted into a grid-like representation (event frames) Y_e. Then, using the part of the dictionary corresponding to the events (D_e), the sparse coefficient α is computed using the OMP (Orthogonal Matching Pursuit) algorithm. Multiplying this coefficient α by the part of the dictionary corresponding to the spatial gradients (D_g) yields gradient patches Y_g, with are collected into two images (the x and y gradients). Finally the image gradients are integrated using a Poisson solver to produce the reconstructed absolute intensity.

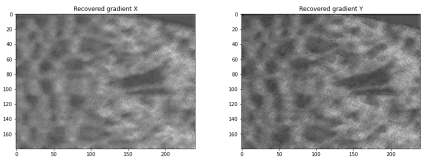
where α is vector of sparse coefficient. $K_s = 8$. Figure 3 shows the flow of execution for training an over-complete representation of the data.

The learned dictionary D is then used to reconstruct new raw data from the DVS camera. The Orthogonal Matching Pursuit (OMP) algorithm [27] was used on the event data vector to get the sparse coefficient α for a new input event data Y_e:

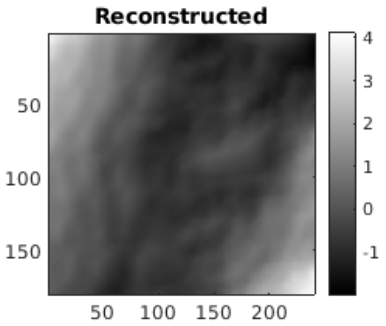
$$\alpha^* = \arg\min_{\alpha} \|Y_e - D_e\alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_0 \leq K_0. \tag{9}$$

This coefficient vector is a "sparse code" representing the input event data in terms of the atoms of the learned event dictionary. Next, the sparse coefficient is multiplied with the gradient portion of the dictionary to get logarithmic intensity gradient data. Poisson reconstruction [1] [2] is then used to get the recovered logarithmic intensity image from the gradient data, and this image when raised to the power of threshold is used to get the absolute intensity image in linear scale. Figure 4 shows the reconstruction procedure from raw event data via the learned dictionary.

As can be seen from Figure 5 , we were not able to do a perfect reconstruction of the testing dataset. We accord this noisy reconstruction to our learnt dictionary. We believe using a more robust, varied natural images dictionary would have fetched a better result. However, due to time constraint we decided to move forward to the next algorithm.



(a) Reconstructed gradient images



(b) Final Reconstructed Image

Figure 5: Sample of the reconstructed x and y gradients and the final reconstructed image.

IV.2 Fast Image Reconstruction with Convolutional Neural Network [2]

The previous algorithm [4] is considered to be the first learning-based method for image reconstruction from event data. The current algorithm by Scheerlinck et. al [34] is a much smaller convolutional network (smaller computational cost) that achieves comparable results with respect to the current state of the art architecture E2VID [32]. The algorithm, which is referred to as FireNet, is a fully convolutional recurrent neural network. Figure 6a shows the architecture of the network. All the layers are 3×3 convolutions except the prediction layer which is 1×1 convolutions and use ReLU (REctified Linear Unit) activation function. The input is a $H \times W \times B$ event tensor with $B = 5$ temporal bins, and sensor height and width as H and W respectively. They use linear voting in time to populate the tensor / voxel grid (Equation 6 and Equation 7). The 5-channel event tensor is fed into the Head layer that has 16 channels. This layer is followed by 16 channel convolution gated recurrent (G1, G2) units and 16 channels residual blocks (R1, R2) with skip connections. The output is one image per input event tensor.

The loss function used to train this artificial neural network was a weighted sum of reconstruction and temporal loss over L ($L = 20$) consecutive images.

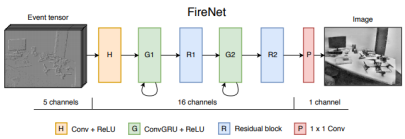
$$\mathscr{L} = \sum_{k=0}^{L} \mathscr{L} + \lambda_{TC} \sum_{k=L_0}^{L} \mathscr{L}_k^{TC} \tag{10}$$

The reconstruction loss is calculated by using Perceptual similarity [37] to ground truth image. The temporal loss is the photometric error [32] between two consecutive images. The algorithm utilised default ADAM [18] with learning rate 10^{-4} for 1000 epochs and λ_{TC} and $L_0 = 10$.Figure 6b shows the reconstructed image from event data. AS can be seen, this output shows lesser noise and a more accurate reconstruction.

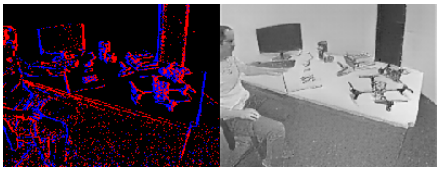
V Discussions and Conclusions

Discussion. The two publications [4, 34] do not show results on the same datasets. Therefore, it is not straightforward to compare their performance. Notably, FireNet [34] is based on the latest results on deep learning, and therefore it is expected to perform better than the 4-year earlier work [4].

[2] Personal implementation is not maintained in github as it was used for understanding purpose. Url for authors's implementation : https://github.com/cedric-scheerlinck/rpg_e2vid/tree/cedric/firenet



(a) rchitecture of the FireNet artificial neural network. Image courtesy of [34].



(b) Final Reconstructed Image

Figure 6: Firenet architecture and output

From a methodology point of view, Barua et al. [4] perform image reconstruction in two steps: First a gradient image is obtained from the events using the learned dictionary and then the gradient is upgraded to absolute intensity using Poisson integration. The first step is the learning phase, whereas the second step is deterministic. By contrast, FireNet [34] performs image reconstruction in one "single step", i.e., one artificial neural network; there is no intermediate obvious interpretation of the hidden layers of the neural network; only the output matters.

From a training point of view, both are trained using simulated data. FireNet uses supervised learning (the loss function measures the error between the predicted image and the ground truth one). On the other hand, the method of Barua et al. is based on learning a dictionary using K-SVD, which is a clustering technique (i.e., unsupervised). FireNet was trained on a larger and probably more diverse dataset, as in [32], so it has better data to generalize to unseen scenes.

From an architecture point of view, FireNet has recurrent connections, that is, some sort of memory of past events and reconstructions that can use to produce a better output that if it had no memory. By contrast, Barua's method has no memory: once the dictionary has been learned, every m event frames are processed independently, without leveraging past events.

Barua's method [4] has 10000 atoms and each atom has 300 dimensions. Thus it has a total of 300k parameters. Whereas, Firenet [34] has 38k parameters.

A comparison in terms of speed depends on the particular implementations of the methods. As far as we know, Barua et al. [4] do not report the runtime of their method. We used python for prototyping Barua's method and it took several seconds to infer the gradient maps from the input events and the learned dictionary. By contrast, FireNet is deployed on efficient code that runs on the GPU, so in this regard, the current implementation of FireNet is faster than our re-implementation of Barua's method.

In **conclusion**, the lab rotation helped me gain knowledge in a completely different field of study as well as in ROS and event simulators. Implementing and comparatively analysing the two algorithms have helped me gain knowledge into niche image processing domains of image reconstruction algorithms, sparse dictionary learning, poisson reconstruction etc.

Looking forward, there are other areas that can be looked onto. One such area is the use of kernelised matrix factorization [12] for learning the dictionary at Barua et. al [4]. Additionally, we can use layer relevance propagation [22] to visualise which pixels of the input dataset have higher relevance in reconstruction.

VI Bibliography

[1] AGRAWAL, A., CHELLAPPA, R., AND RASKAR, R. An algebraic approach to surface reconstruction from gradient fields. In *Int. Conf. Comput. Vis. (ICCV)* (2005), pp. 174–181.

[2] AGRAWAL, A., RASKAR, R., AND CHELLAPPA, R. What is the range of surface reconstructions from a gradient field? In *Eur. Conf. Comput. Vis. (ECCV)* (2006), pp. 578–591.

[3] AHARON, M., ELAD, M., AND BRUCKSTEIN, A. M. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process. 54*, 11 (2006), 4311–4322.

[4] BARUA, S., MIYATANI, Y., AND VEERARAGHAVAN, A. Direct face detection and video reconstruction from event cameras. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)* (2016), pp. 1–9.

[5] BENOSMAN, R., CLERCQ, C., LAGORCE, X., IENG, S.-H., AND BARTOLOZZI, C. Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst. 25*, 2 (2014), 407–417.

[6] BOAHEN, K. Neuromorphic microchips. *Scientific American 292*, 5 (2005), 56–63.

[7] BRANDLI, C., BERNER, R., YANG, M., LIU, S.-C., AND DELBRUCK, T. A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits 49*, 10 (2014), 2333–2341.

[8] BRYNER, S., GALLEGO, G., REBECQ, H., AND SCARAMUZZA, D. Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (2019).

[9] CENSI, A., AND SCARAMUZZA, D. Low-latency event-based visual odometry. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (2014), pp. 703–710.

[10] COOK, M., GUGELMANN, L., JUG, F., KRAUTZ, C., AND STEGER, A. Interacting maps for fast visual interpretation. In *Int. Joint Conf. Neural Netw. (IJCNN)* (2011), pp. 770–776.

[11] DELBRUCK, T. Frame-free dynamic digital vision. In *Proc. Int. Symp. Secure-Life Electron.* (2008), pp. 21–26.

[12] FAN, J., YANG, C., AND UDELL, M. Robust non-linear matrix factorization for dictionary learning, denoising, and clustering, 2020.

[13] GALLEGO, G., DELBRUCK, T., ORCHARD, G., BARTOLOZZI, C., TABA, B., CENSI, A., LEUTENEGGER, S., DAVISON, A., CONRADT, J., DANIILIDIS, K., AND SCARAMUZZA, D. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).

[14] GALLEGO, G., LUND, J. E. A., MUEGGLER, E., REBECQ, H., DELBRUCK, T., AND SCARAMUZZA, D. Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Trans. Pattern Anal. Mach. Intell. 40*, 10 (Oct. 2018), 2402–2412.

[15] GALLEGO, G., AND SCARAMUZZA, D. Accurate angular velocity estimation with an event camera. *IEEE Robot. Autom. Lett. 2*, 2 (2017), 632–639.

[16] KIM, H., HANDA, A., BENOSMAN, R., IENG, S.-H., AND DAVISON, A. J. Simultaneous mosaicing and tracking with an event camera. In *British Mach. Vis. Conf. (BMVC)* (2014).

[17] KIM, H., LEUTENEGGER, S., AND DAVISON, A. J. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Eur. Conf. Comput. Vis. (ECCV)* (2016), pp. 349–364.

[18] KINGMA, D. P., AND BA, J. L. Adam: A method for stochastic optimization. *Int. Conf. Learn. Representations (ICLR)* (2015).

[19] LICHTSTEINER, P., POSCH, C., AND DELBRUCK, T. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits 43*, 2 (2008), 566–576.

[20] MAHOWALD, M., AND MEAD, C. The silicon retina. *Scientific American 264*, 5 (May 1991), 76–83.

[21] MANDERSCHEID, J., SIRONI, A., BOURDIS, N., MIGLIORE, D., AND LEPETIT, V. Speed invariant time surface for learning to detect corner points with event-based cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2019).

[22] MONTAVON, G., BINDER, A., LAPUSCHKIN, S., SAMEK, W., AND MÜLLER, K.-R. *Layer-Wise Relevance Propagation: An Overview*. Springer International Publishing, Cham, 2019, pp. 193–209.

[23] MUEGGLER, E., BARTOLOZZI, C., AND SCARAMUZZA, D. Fast event-based corner detection. In *British Mach. Vis. Conf. (BMVC)* (2017).

[24] MUEGGLER, E., REBECQ, H., GALLEGO, G., DELBRUCK, T., AND SCARAMUZZA, D. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research 36*, 2 (2017), 142–149.

[25] NI, Z., BOLOPION, A., AGNUS, J., BENOSMAN, R., AND RÉGNIER, S. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Trans. Robot. 28*, 5 (2012), 1081–1089.

[26] NOZAKI, Y., AND DELBRUCK, T. Temperature and parasitic photocurrent effects in dynamic vision sensors. *IEEE Trans. Electron Devices 64*, 8 (Aug. 2017), 3239–3245.

[27] PATI, Y. C., REZAIIFAR, R., AND KRISHNAPRASAD, P. S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proc. 27th Asilomar Conf. Signals, Systems and Computers* (1993), pp. 40–44.

[28] POSCH, C., MATOLIN, D., AND WOHLGENANNT, R. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE J. Solid-State Circuits 46*, 1 (Jan. 2011), 259–275.

[29] POSCH, C., SERRANO-GOTARREDONA, T., LINARES-BARRANCO, B., AND DELBRUCK, T. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proc. IEEE 102*, 10 (Oct. 2014), 1470–1484.

[30] REBECQ, H., GEHRIG, D., AND SCARAMUZZA, D. ESIM: an open event camera simulator. In *Conf. on Robotics Learning (CoRL)* (2018).

[31] REBECQ, H., RANFTL, R., KOLTUN, V., AND SCARAMUZZA, D. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2019).

[32] REBECQ, H., RANFTL, R., KOLTUN, V., AND SCARAMUZZA, D. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

[33] ROSE, A. *Vision: Human and Electronic.* Plenum Press, New York, 1973.

[34] SCHEERLINCK, C., REBECQ, H., GEHRIG, D., BARNES, N., MAHONY, R., AND SCARAMUZZA, D. Fast image reconstruction with an event camera. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)* (2020).

[35] XU, H., GAO, Y., YU, F., AND DARRELL, T. End-to-end learning of driving models from large-scale video datasets. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2017), pp. 3530–3538.

[36] YE, C., MITROKHIN, A., PARAMESHWARA, C., FERMÜLLER, C., YORKE, J. A., AND ALOIMONOS, Y. Unsupervised learning of dense optical flow and depth from sparse event data. *arXiv e-prints* (2019).

[37] ZHANG, R., ISOLA, P., EFROS, A. A., SHECHTMAN, E., AND WANG, O. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2018).

[38] ZHU, A. Z., YUAN, L., CHANEY, K., AND DANIILIDIS, K. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)* (2018).

[39] ZHU, A. Z., YUAN, L., CHANEY, K., AND DANIILIDIS, K. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2019).