A Low Power, High Throughput, Fully Event-Based Stereo System

Alexander Andreopoulos*, Hirak J. Kashyap*,†, Tapan K. Nayak, Arnon Amir, and Myron D. Flickner
IBM Research

Abstract

We introduce a stereo correspondence system implemented fully on event-based digital hardware, using a fully graph-based non von-Neumann computation model, where no frames, arrays, or any other such data-structures are used. This is the first time that an end-to-end stereo pipeline from image acquisition and rectification, multi-scale spatio-temporal stereo correspondence, winner-take-all, to disparity regularization is implemented fully on event-based hardware. Using a cluster of TrueNorth neurosynaptic processors, we demonstrate their ability to process bilateral event-based inputs streamed live by Dynamic Vision Sensors (DVS), at up to 2,000 disparity maps per second, producing high fidelity disparities which are in turn used to reconstruct, at low power, the depth of events produced from rapidly changing scenes. Experiments on real-world sequences demonstrate the ability of the system to take full advantage of the asynchronous and sparse nature of DVS sensors for low power depth reconstruction, in environments where conventional frame-based cameras connected to synchronous processors would be inefficient for rapidly moving objects. System evaluation on event-based sequences demonstrates a $\sim 200\times$ *improvement in terms of power per pixel per disparity map compared to the closest state-of-the-art, and maximum latencies of up to 11ms from spike injection to disparity map ejection.*

1. Introduction

Sparsity and parallel asynchronous computation are two key principles of information processing in the brain. They allow to solve complex tasks using a tiny fraction of the energy consumed by stored-program computers [64]. While the successful artificial neural networks may not operate the same way as the brain, both of them utilize highly parallel and hierarchical architectures that gradually abstract input data to more meaningful concepts [8, 51, 16]. However, event-based computation has not been equally adopted [4]. Another barrier for sparse computation are traditional sensors, such as frame-based cameras, which provide regular inputs. For autonomous vehicles, drones, and satellites, energy consumption is a challenge [6]. Event-based processing dramatically reduces power consumption by computing only what is new while omitting unchanged input parts.

Recently developed event-based cameras such as Dynamic Vision Sensor (DVS) [37, 10] and ATIS [50], inspired by the biological retina, encode pixel illumination changes as events. These sensors solve two major drawbacks of frame-based cameras. First, temporal resolution of frame-based applications is limited by the camera frame rate, usually 30 frames per second. Event-based cameras can generate events at microsecond resolution. Second, consecutive frames in videos are usually highly redundant, which waste downstream data transfer, computing resources and power. Since events are sparse, event-based cameras lead to better downstream resource usage. Moreover, event-based cameras have high dynamic range (~ 100 dB), which is useful for real world variations in lighting conditions.

To achieve the low energy and high temporal resolution benefits of event-based inputs, computations must be performed asynchronously. To benefit from sparse and asynchronous computation, neuromorphic processors have been developed [44, 24, 30, 9, 56]. These processors represent input events as spikes and process them in parallel using a large neuron population. They are stimulus-driven and the propagation delay of an event through the neuron layers is usually a few milliseconds, suitable for many real-time applications. For example, the TrueNorth neuromorphic chip [44] has been used for high throughput Convolutional neural networks (CNNs) [22], character recognition [53], optic flow [11], saliency [3], and gesture recognition [2].

Depth perception is an important task for autonomous mobile agents to navigate in the real world. The speed and low power requirements of these applications can be effectively met using event-based sensors. Event-based stereo provides additional advantages over other depth estimation methods that increase accuracy and save energy, such as high temporal resolution, high dynamic range, and robustness to interference with other agents.

Several methods have been proposed to solve event-

*equal contribution. †Work done as an intern at IBM Research - Almaden. Cognitive Anteater Robotics Lab (CARL), University of California, Irvine

based stereo correspondence. Most global methods [40, 17, 49, 45] are derived from the Marr and Poggio cooperative stereo algorithm [42]. The algorithm assumes depth continuity and often event-based implementations are not tested with objects tilted in depth. Local methods can be parallelized and find corresponding events using either local features over a spatiotemporal window or event-to-event features [13, 58, 52, 32, 57]. However, most approaches use non-event-based hardware, such as CPU or DSP.

We propose a fully neuromorphic event-based stereo disparity algorithm. A live-feed version of the system running on nine TrueNorth chips is shown to calculate 400 disparity maps per second, and the ability to increase this up to 2,000 disparities per second (subject to certain trade-offs) is demonstrated, for use with high speed event cameras, such as DVS. The main advantages of the proposed method, compared to the related work [17, 49, 45, 52, 57], are simultaneous end-to-end neuromorphic disparity calculation, low power, high throughput, low latency (9-11 ms), and linear scalability to multiple neuromorphic processors for larger input sizes. Compared to frame-based computation, in the asynchronous, event-based computation supported by TrueNorth, at each time cycle, in general only neurons that have input spikes are computed, and only spike events "1" are communicated. When the data in a cycle is sparse, as is the case with a DVS sensor, most neurons would not compute for most of the time, resulting in low active power [44]. This processing differs from traditional architectures that use frame-buffers and other conventional data structures; where same memory fetching and computation is repeated for each pixel every frame, independent of scene activity.

The proposed event-based disparity method is implemented using a stereo pair of DAVIS sensors [10] (a version of DVS) and nine TrueNorth NS1e boards [53]. However, the method is applicable to other spiking neuromorphic architectures, and it is also tested offline on larger models using a TrueNorth simulator. Input rectification, spatiotemporal scaling, feature matching, search for best matches, morphological erosion and dilation, and bidirectional consistency check are all performed on TrueNorth, for a fully neuromorphic disparity solution. With respect to the most relevant state-of-the-art approach [17], our method uses $\sim 200\times$ less power per pixel per disparity map. We also release the event-based stereo dataset used, which includes Kinect-based registered ground-truth.

2. Related work

Frame-based stereo disparity methods calculate matching cost using a spatial similarity metric [25, 27, 29] or a cost function learned from a dataset (see reviews [62, 55, 34, 63]). CNNs [35] have been used to learn stereo matching cost [66, 46]. Ground truth disparity maps from benchmark frame-based datasets [27, 54, 26, 43] are used to train these

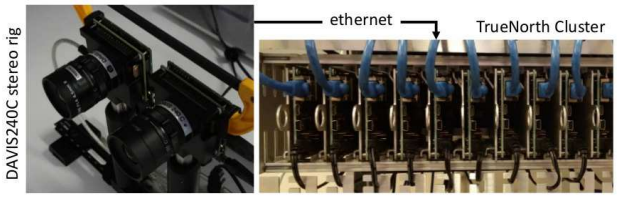

Figure 1. The time-stamp synchronized stereo rig is connected to a cluster of TrueNorth chips via ethernet.

models, followed by sparse-to-dense conversions [18, 5]. Feature based matching techniques, such as color, edge, histogram, and SIFT [39] based matching, produce sparse disparity maps [28, 38, 21, 61].

In contrast, event-based stereo correspondence literature is relatively new. Mahowald and Delbrück [41] implemented the Marr and Poggio cooperative stereo algorithm [42], a global approach, in an analog VLSI circuit. The algorithm converges well when object surfaces are fronto-parallel and candidate matches injected to the network are close together [40, 17]. Later Mahowald [40] modified the VLSI embodied algorithm to solve tilted depth maps using a network of analog valued disparity units, which linearly interpolates the cooperative network output.

However, most of the recent event-based implementations of the cooperative algorithm do not consider depth gradients [47, 48, 23, 17]. Piatkowska et al. [49] inject neighborhood similarity of candidate matches into the cooperative network. Dikov et al. [17] use six SpiNNaker [24] processor boards to implement the cooperative network for 106×106 pixels of stereo event data. Osswald et al. [45] propose an FPGA based implementation of spiking neurons as the nodes of the cooperative network. Xie et al. [65] employ message passing on a Markov Random Field with depth continuity for a global solution.

Local event-based stereo correspondence approaches are area-based or time-based. Area-based methods assume that object shapes appear identically on left and right sensors. Camuñas-Mesa et al. [13, 12] propose to match edge orientations in event frames accumulated over 50 ms. Schraml et al. [60, 58] propose DSP implementation of a spatiotemporal similarity method using two live event sensors [37]. Belbachir et al. [7] use a rotating pair of event-based line (vertical) sensors in static scenes and render events from each rotation to an edge map [33], which is subsequently processed using a frame-based panoramic stereo algorithm [36].

Time-based methods utilize event timestamps for matching. Although spike dynamics vary among pixels and sensors [52] and events cannot be matched based on exact timestamps. Rogister et al. [52, 14] propose to use event-to-event constraints for calculating matching cost, such as time window, distance to the epipolar line, ordering constraint, and polarity. Kogler et al. [32, 31] calculate similarity as the

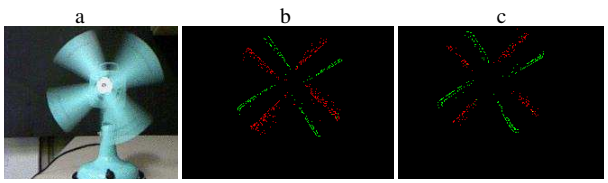

Figure 2. Frames-based (a) and event-based (b-left DAVIS and c-right DAVIS) camera output for a rotating fan. Green dots are positive events, i.e. increase in pixel intensity, and red dots are negative events.

inverse of temporal distance and average them within each depth plane. The proposed method and its FPGA implementations [20, 19] are equivalent to the cooperative stereo algorithm [42] with noisy time difference inputs. Schraml et al. [59, 57] propose a cost function for the rotating stereo panorama setup in [7] based on temporal event difference.

3. Event-based hardware

Our implementation uses a pair of synchronized DAVIS240C cameras [10], connected via Ethernet to a cluster of TrueNorth NS1e boards (Fig. 1). The use of DAVIS sensors improve speed, power, dynamic range, and computational requirements. As shown in Fig. 2, fast moving objects are more challenging for frame-based cameras.

The IBM TrueNorth is a reconfigurable, non-von Neumann neuromorphic chip containing 1 million spiking neurons and 256 million synapses distributed across 4096 parallel, event-driven, neurosynaptic cores [44]. Cores are tiled in a 64×64 array, embedded in a fully asynchronous network-on-chip. The chip consumes 70mW when operating at a 1 ms computation tick and normal workloads. Depending on event dynamics and network architecture, faster tick period is possible, which we take advantage of in this work to achieve as low as 0.5 ms per tick, thus doubling the maximum throughput achievable. Each neurosynaptic core connects 256 inputs to 256 neurons using a crossbar of 256×256 binary synapses with a lookup table of weights for 8 bits of precision, plus a sign bit. A neuron state variable, called membrane potential, integrates synaptically weighted input events with an optional leak decay. Each neuron can generate an output event deterministically, if the membrane potential $V(t)$ exceeds a threshold; or stochastically, with a probability that is a function of the difference between the membrane potential and its threshold [2, 15]. The membrane potential is updated at each tick t to $V(t) = V(t-1) + \frac{\partial V(t)}{\partial t}$, followed by the application of an activation function $\mathbf{a}_n(V(t))$ where

$$\mathbf{a}_n(V(t)) = \begin{cases} 1, & \text{if } V(t) \geq n \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Each neuron is assigned an initial membrane potential $V(0)$. Furthermore, upon producing an event, a neuron is reset to a user-specified value. Unless specified otherwise, we assume initial membrane potentials and reset values of zero. TrueNorth programs are written in the Corelet Programming Language — a hierarchical, compositional, object-oriented language [1].

4. Stereo correspondence on TrueNorth

The proposed local event-based stereo correspondence algorithm is implemented end-to-end as a neuromorphic event-based algorithm. This consists of systems of equations defining the behavior of TrueNorth neurons, encased in modules called corelets [1], and the subsequent composition of the inputs and outputs of these modules. Fig. 3 depicts the sequence of operations performed by the corelets using inputs from stereo event sensors.

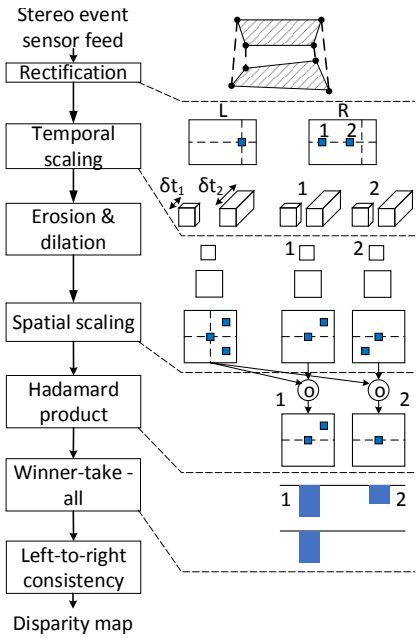

Figure 3. The pipeline of execution using input events generated by left and right sensors. A toy example of main operations performed is demonstrated side-by-side in a single spatiotemporal scale, with an event on the left image and its two candidate corresponding events on the right image. Standard morphological operations and left-to-right consistency check are not demonstrated.

4.1. Rectification

The stereo rectification is defined by a pair of functions $\mathcal{L}$, $\mathcal{R}$ which map each pixel in the left and right sensor's rectified space to a pixel in the left and right sensor's native resolution respectively. On TrueNorth, this is implemented using $|H| \cdot |W|$ splitter neurons per sensor & polarity channel, arranged in an $|H| \times |W|$ retinotopic map. The events at each rectified pixel $p \in H \times$

$W \times \{\mathcal{L}, \mathcal{R}\} \times \{+, -, \{+, -\}\}$ are generated through splitter neurons which replicate corresponding sensor pixels. Their membrane potential $V_p^{spl}(t)$ is defined by $\frac{\partial V_p^{spl}(t)}{\partial t} = I(t-1; p')$ where $I(t; p') \rightarrow \{0, 1\}$ denotes whether a sensor event is produced at time t and the sensor pixel p' corresponding to the rectified pixel p. $\mathbf{a}_1(V_p^{spl}(t))$ defines the activation of the corresponding neuron. Potentials are initialized to zero and set to also reset to zero upon spiking.

4.2. Multiscale temporal representation

The event rate of an event-based sensor depends on factors, such as scene contrast, sensor bias parameters, and object velocity. To add invariance across event rates, we accumulate spikes over various temporal scales through the use of temporally overlapping sliding windows. These temporal scales are implemented through the use of splitter neurons which cause each event to appear at its corresponding pixel multiple times, depending on the desired temporal scale, or through the use of temporal ring buffer mechanisms, which lead to lower event rates. The ring buffer is implemented by storing events in membrane potentials of memory cell neurons in a circular buffer, and through the use of control neurons which spike periodically to polarize appropriate memory cell neurons. Buffers can encode the input at various temporal scales. For example at a scale $T = 5$ the buffer denotes if an event occurred at the corresponding pixel during the last 5 ticks (logical disjunction).

A control neuron that produces events with period T and phase ϕ is defined by a neuron $\mathbf{a}_T(V_\phi^{ctrl})$ that satisfies $\frac{\partial V_\phi^{ctrl}(t)}{\partial t} = 1, V(0) = \phi$ and resets to zero upon producing an event. Through populations of such neurons one can also define $\mathbf{a}_T(V_{[\phi,\theta]}^{ctrl})$ corresponding to phase intervals $[\phi, \theta]$ (where $\theta - \phi + 1 \leq T$), defining periodic intervals of events. Such control neurons are used to probe (prb) or reset (rst) neuron membrane potentials. A memory cell neuron is a recurrent neuron which accepts as input either its own output (so that it does not lose its stored value whenever the neuron is queried for its stored value), input axons to set the neuron value and control axons for resetting and querying the memory cell. In more detail the output at index $r \in \{0, ..., T+1\}$ of a $T+2$ size memory cell ring-buffer at a given pixel p, is multiplexed via two copies ($m \in \{0, 1\}$) and is defined as $\mathbf{a}_2(V_{p,m,r}^{mem})$ where

$$\begin{aligned}\frac{\partial V_{p,m,r}^{mem}(t+1)}{\partial t} &= [\, -\mathbf{a}_{T+2}(V_{\hat{s}+1}^{rst}(t)) \\ &+([\mathbf{a}_1(V_p^{spl}(t))]_{\hat{r}}^r \vee [\mathbf{a}_2(V_{p,m,r}^{mem}(t-1))]_{\hat{t}}^m) \\ &+[\mathbf{a}_{T+2}(V_{[3-r,T+2-r]}^{prb}(t))]_{\hat{t}}^m \\ &-[\mathbf{a}_{T+2}(V_{[2-r,T+1-r]}^{rst}(t))]_{\hat{t}}^{1-m}]_+ \end{aligned} \tag{2}$$

where probe/reset (prb/rst) control neurons are used, $\hat{r} = t \bmod (T+2)$, $\hat{s} = T+2-r \bmod (T+2)$, $\hat{t} = t \bmod 2$, $\vee$ is logical disjunction[1],

$$[\mathbf{x}]_{\hat{r}}^r = \begin{cases} \max\{0, \mathbf{x}\}, & \text{if } r = \hat{r} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and $[\mathbf{x}]_+ \stackrel{\text{def}}{=} [\mathbf{x}]_1^1$ defines a ReLU function. Eq. 2 defines a ring-buffer with $T+2$ memory cells, where probe pulses periodically and uniformly query T of the $T+2$ cells for the stored memory contents at each tick, where $m = 0$ neurons are probed at odd ticks and $m = 1$ neurons are probed at even ticks. Reset pulses control when to reset one of the $T+2$ memory cells to zero in preparation of a new input. Notice that new inputs ($\mathbf{a}_1(V_p^{spl}(\cdot))$) are always routed to the cell r that was reset in the previous tick. The probe pulses result in the creation of an output event if during the last T ticks $\mathbf{a}_1(V_p^{spl}(\cdot))$ produced an event. After a probe event, a reset event decrements the previous $+1$ membrane potential increase, followed by the restoring of the memory event output during the last probe ($\mathbf{a}_2(V_{p,m,r}^{mem}(t-1))$).

4.3. Morphological erosion and dilation

Binary morphological erosion and dilation is optionally applied on the previous module's outputs to denoise the image. Given a 2-D neighborhood $N(p)$ centered around each pixel p, the erosion neuron's membrane potential V_p^e is guided by the system of equations $\frac{\partial V_p^e(t)}{\partial t} = [1 - |N(p)| + \sum_{q \in N(p)} \sum_m \bigvee_r \mathbf{a}_2(V_{q,m,r}^{mem}(t-1))]_+$ and uses an $\mathbf{a}_1$ activation function. Similarly, dilation neurons V_p^d with receptive fields $N(p)$ evolve according to $\frac{\partial V_p^d(t)}{\partial t} = \sum_{q \in N(p)} \mathbf{a}_1(V_q^e(t-1))$ where $\mathbf{a}_1$ is also used as the dilation neurons' activation function. The neuron potentials are initialized to zero and set to also reset to zero upon producing a spike. In practice 3×3 pixel neighborhoods are used. At each tick, erosion and dilation neurons output the minimum and maximum value respectively, of their receptive fields. Cascades of erosion and dilation neurons, are used to denoise retinotopic binary inputs (Fig. 3).

4.4. Multiscale spatiotemporal features

Each feature extracted around a rectified pixel p is a concatenation of event patches, extracted from one or more spatiotemporal scales. Spatial scaling consists of spatially sub-sampling each output map of the temporal scale phase (Sec. 4.2/4.3), as specified in the corelet parameters, to apply the window matching (Sec. 4.5) on the sub-sampled data. This results in spatiotemporal coordinate tensors $\mathcal{X}_{L,p}, \mathcal{X}_{R,p}$ defining the coordinates where events form feature vectors. The i^{th} of these coordinates is represented by neuron activations $\mathbf{a}_1(V_{\mathcal{X}_{L,p}^{(i)}}^{L\{+,-\}}(t))$ and $\mathbf{a}_1(V_{\mathcal{X}_{R,p}^{(i)}}^{R\{+,-\}}(t))$ in

[1] disjunction is implemented by sending input events to the same neuron input axon, effectively merging any input events to a single input event.

the left and right sensor's positive (+) or negative (−) polarity channel.[2]

4.5. Hadamard product for matching

Given a pair of spatiotemporal coordinate tensors $\mathcal{X}_{L,p}$, $\mathcal{X}_{R,q}$ centered at coordinates p, q in the left and right rectified image respectively and representing K coordinates each, we calculate the binary Hadamard product $\mathbf{f}_L(p,t) \circ \mathbf{f}_R(q,t)$ associated with the corresponding patches at time t, where $\mathbf{f}_L(p,t) = \prod_i \{\mathbf{a}_1(V^L_{\mathcal{X}^{(i)}_{L,p}}(t))\} \in \{0,1\}^K$ and $\mathbf{f}_R(q,t) = \prod_i \{\mathbf{a}_1(V^R_{\mathcal{X}^{(i)}_{R,q}}(t))\} \in \{0,1\}^K$. The product is calculated in parallel across multiple neurons, as K pairwise logical AND operations of corresponding feature vector entries, resulting in $(\mathbf{a}_1(V^{dot}_{p,q,1}), ..., \mathbf{a}_1(V^{dot}_{p,q,K}))$ where $\frac{\partial V^{dot}_{p,q,i}(t)}{\partial t} = [\mathbf{a}_1(V^L_{\mathcal{X}^{(i)}_{L,p}}(t-1)) + \mathbf{a}_1(V^R_{\mathcal{X}^{(i)}_{R,q}}(t-1)) - 1]_+$ The population code representation of the Hadamard product output is converted to a thermometer code[3], which is passed to the winner-take-all circuit described below.

4.6. Winner-Take-All system

The winner-take-all (WTA) system is a feed-forward neural network that takes as input D thermometer code representations of the Hadamard products for D distinct candidate disparity levels, and finds the disparity with the largest value, at every tick. For designing a scalable and compact WTA system on a neuromorphic hardware, we introduced a novel encoding technique for inputs. In a binary event-based system, numbers can be efficiently coded using base-4 representation where each digit is encoded using a 3-bits thermometer code. We denote it as *Quaternary Thermometer Code* (QTC). Note that a thermometer code of length 2^n bits can be represented by a QTC of length $3 * \lceil n/2 \rceil$ bits. For example, values between 0–255 are represented by a QTC of 12 bits. While it takes a few more bits than an 8 bits binary code, it allows designing a feed-forward WTA network comprising only four cascaded subnetworks, compared to eight for a binary representation, requiring fewer hardware resources as well as half the latency. Latency is further improved with larger bases, but the growth in thermometer code length for each digit results in consuming more hardware resources. Table 1 shows binary, base-4 and QTC representation of different decimal numbers.

We assume a maximum thermometer code length of $4^{B+1} \geq K$ for some $B \in \mathbb{N}$. Then for any $\alpha \in \{0,1,2\}$, $\beta \in \{0,1,...,B\}$, we define the conversion of candidate disparity level $d \in \{0,...,D-1\}$ to a QT-coded membrane potential $V^{CNV}_{\alpha,\beta,d}(t)$ as

[2]for notational simplicity we henceforth drop the $+,-$ superscripts: the left and right sensors could produce distinct event streams based on event polarity, or could merge events in a single polarity-agnostic stream.

[3]e.g., given a population code $(1,1,0,1,0)$ for value 3, its thermometer code is the right-aligned juxtaposition of all events: $(0,0,1,1,1)$.

Decimal	Binary	Base-4	QTC
126	01-11-11-10	1-3-3-2	001-111-111-011
174	10-10-11-10	2-2-3-2	011-011-111-011
33	00-10-00-01	0-2-0-1	000-011-000-001
167	10-10-01-11	2-2-1-3	011-011-001-111
26	00-01-10-10	0-1-2-2	000-001-011-011

Table 1. Decimal, binary, base-4 and QTC representation of five example numbers.

Value	W_0	Stage-0	W_1	Stage-1	W_2	Stage-2	W_3	Stage-3
126	1	~~001~~	0		0		0	
174	1	011	1	011	1	111	1	011 ✓
33	1	~~000~~	0		0		0	
167	1	011	1	011	1	~~001~~	0	
26	1	~~000~~	0		0		0	
stage max		011		011		111		011

Table 2. Winner selection process for QT-coded inputs ($B = 3$)

$$\frac{\partial V^{CNV}_{\alpha,\beta,d}(t)}{\partial t} = [\sum_{i \in U(\beta)} v^i_d(t-1) - \sum_{i \in U(\beta+1)} 4\, v^i_d(t-1) - \alpha]_+ \tag{4}$$

where $v^i_d(t)$ is the i-th element of the input thermometer code[4] for d^{th} disparity level at time t and $U(\beta) = \{n \in \mathbb{N} : n \equiv 0 \pmod{4^\beta},\ 1 \leq n < 4^{B+1}\}$. All the conversion neurons use an $\mathbf{a}_1$ activation function and reset to 0 membrane potential upon spiking. Notice that $(\mathbf{a}_1(V^{CNV}_{2,\beta,d}(t)), \mathbf{a}_1(V^{CNV}_{1,\beta,d}(t)), \mathbf{a}_1(V^{CNV}_{0,\beta,d}(t)))$ is a length-3 thermometer code representation of a value in $\{0,1,2,3\}$, representing the β^{th} digit in the base-4 representation of $v_d(t-1)$.

For a set of QT-coded inputs, the WTA system is realized by a cascade of $(B+1)$ feed-forward pruning networks where each of the pruning networks process only 3-bits of the QT codes and prune the inputs not equal to the bit-wise maximum of corresonding 3-bits thermometer codes from all inputs. Now starting from the most significant bits, all the inputs smaller than the maximum will be pruned at different stages and only the winner(s) will survive at the output of the last cascade network. The membrane potential $V^{WTA}_{\beta,d}$ of stage β and disparity index d is given by,

$$\frac{\partial V^{WTA}_{\beta,d}(t)}{\partial t} = [4 \cdot W_{\beta,d}(t-1) + \sum_{\alpha=0}^{2} [\mathbf{a}_1(V^{CNV}_{\alpha,B-\beta,d}(t-\beta)) - \max_{\bar{d} \in \{d' | W_{\beta,d'}(t-1) > 0\}} \{\mathbf{a}_1(V^{CNV}_{\alpha,B-\beta,\bar{d}}(t-\beta))\}] - 3]_+, \tag{5}$$

$$\text{where} \quad W_{\beta,d}(t) = \begin{cases} \mathbf{a}_1(V^{WTA}_{\beta-1,d}(t)), & \forall \beta > 0, \\ 1, & \text{if } \beta = 0 \end{cases} \tag{6}$$

Note that the function $W_{\beta,d}(t)$ represents the candidate status of the d-th input at the end of β-th stage. Initially all the

[4]the i variable indexing (v^i_d) starts from the right of the thermometer code v_d of $(\mathbf{a}_1(V^{dot}_{p,q,1}), ..., \mathbf{a}_1(V^{dot}_{p,q,K})) \in \{0,1\}^K$. The dependence of v_d and d on pixels p, q is implicit and is not shown to simplify notation.

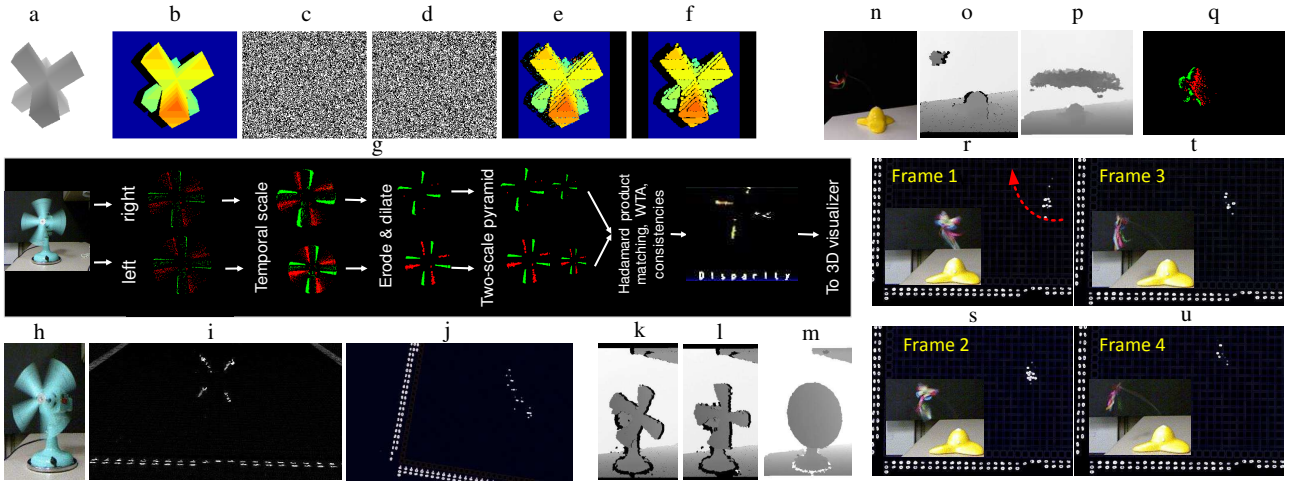

Figure 4. Experimental results obtained using TrueNorth. a) Example synthetic depth pattern, b) ground truth disparity map, c-d) RDS for left and right view, e) disparity map obtained from corelet implementation, f) corelet result after erosion and dilation post-processing, g) fan sequence input received from the left-right DAVIS cameras and results generated by each layer of corelets from this input, h) example frame with fan rotating in a particular orientation, i) 3D reconstruction by the proposed method as seen from an angled front view (screen capture from the 3D visualizer), j) 3D reconstruction from a top view, k-l) Kinect depth maps with static fan blades, m) merged Kinect depth map, n) example frame with the butterfly rotating around the spring base, o) Kinect depth map for a butterfly frame, p) merged Kinect depth map, q) left DAVIS output for a 3 ms time window, r-u) top view of 3D reconstruction from the 3D visualizer for four consecutive frames in the sequence as the butterfly rotates clockwise. See the supplementary material for example videos.

inputs are winning candidates ($W_{0,d}(t) = 1$) and the status changes after the input is pruned at any stage indicating it is out of the competition and the selection process continues with remaining candidates. As an illustration, winner is computed from the example set of numbers in Table 1 and the winner selection process is shown in Table 2.

4.7. Bidirectional consistency check

A left-right consistency check is then performed to verify that for each left-rectified pixel p matched to right-rectified pixel q, it is also the case that right-rectified pixel q gets matched to left-rectified pixel p. This is achieved using two parallel WTA streams. Stream 1 calculates the winner disparities for left-to-right matching, and stream 2 calculates the winner disparities of right-to-left matching. The outputs of each stream are represented by D retinotopic maps expressed in a fixed resolution ($\mathbf{D}^{v}_{i,j,d}(t)$, $d \in \{0, ..., D-1\}$, $v \in \{L, R\}$), where events represent the retinotopic winner disparities for that stream. The streams are then merged to produce the disparity map $\mathbf{D}^{L,R}_{i,j,d}(t) = \mathbf{a}_1(V^{L,R}_{i,j,d}(t))$ where

$$\frac{\partial V^{L,R}_{i,j,d}(t)}{\partial t} = [\mathbf{D}^{L}_{i,j,d}(t-1) + \mathbf{D}^{R}_{i,j-d,d}(t-1) + \mathbf{a}_1(V^{spl}_{(i,j,\mathcal{L},\cdot)}(t-\hat{t})) - 2]_{+} \quad (7)$$

where $\hat{t}$ is the propagation delay of the first layer splitter output events until the left-right consistency constraint merging takes place. This enforces that an output disparity is produced at time-stamp t and pixel (i, j) only for left-rectified pixel (i, j), where an event was produced at $t - \hat{t}$.

5. Experiments

5.1. Datasets

We evaluate the performance of the system on sequences of random dot stereograms (RDS) representing a rotating synthetic 3D object (Fig. 4a-f), and two real world sets of sequences, consisting of a fast rotating fan (Fig. 4g-m) and a rotating toy butterfly (Fig. 4n-u) captured using the DAVIS stereo cameras. The synthetic dataset provides dense disparity estimates, which are difficult to acquire with the sparse event based cameras. The dataset is generated by assigning to each left sensor pixel a random event with a 50% probability per polarity. Similarly, each right sensor pixel is assigned a value by projecting it to the 3D scene and reprojecting the corresponding data-point to the left camera coordinate frame to find the closest pixel value. Self-occluded pixels are assigned random values.

For the non-synthetic datasets, a Kinect [67] is used to extract ground truth of the scene structure. This also entails a calibration process for transforming the undistorted Kinect coordinate frame to the undistorted DAVIS sensor coordinate frame. The fan sequence is useful for testing the ability of the algorithm to operate on rapidly moving objects. Varying orientations of the revolving fan add continuously varying depth gradient to the dataset. Ground truth

is extracted in terms of the plane in 3D space representing the blades' plane of rotation (Fig. 4m). The butterfly sequence tests the ability of the algorithm to operate on non-rigid objects which are rapidly rotating in a circular plane approximately perpendicular to the y-axis. Ground truth is extracted in terms of the coordinates of the circle spanned by the rotating butterfly (Fig. 4p). Nine Fan sequences (3 distances $\times$ 3 orientations) and three Butterfly sequences (3 distances) are used. The dataset, with Kinect ground-truth, is at: http://ibm.biz/StereoEventData.

5.2. Evaluation

On the synthetic dataset we measure the average absolute disparity error, and the average recall, which is defined as the fraction of pixels where a disparity measurement was found. On the non-synthetic data, performance is measured in terms of precision, which is defined as the median relative error $\frac{\|x-x'\|}{\|x'\|}$ between each 3D coordinate x extracted in the DAVIS frame using the neuromorphic algorithm, and the corresponding ground coordinate x' in the aligned Kinect coordinate frame. Performance is also reported in terms of the recall, defined herein as the percentage of DAVIS pixels containing events, where a disparity estimate was also extracted. We tested a suite of sixty stereo disparity networks generated with ranges of spatiotemporal scales, denoising parameters, kernel match thresholds, with/without left-right consistency constraints etc.

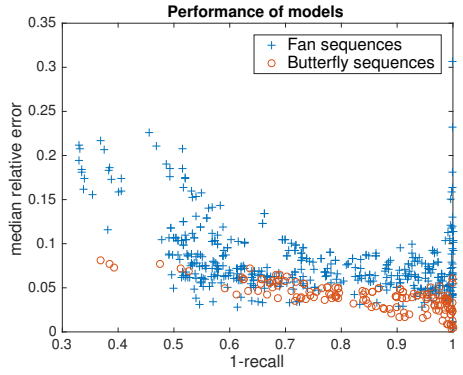

Figure 5. Performance of sixty models on the nine Fan sequences and the three Butterfly sequences.

5.3. Power measurement

Power is measured using the same process described in [2]. We calculate the power consumed by an n-chip system by measuring power on a single TrueNorth chip model running on an NS1t board with a high event rate input generated by the fan sequence. This board has circuitry to measure the power consumed by a TrueNorth chip. We multiply the power value by n to extrapolate the power consumed by an n-chip system. Measurements are reported at supply voltages of 0.8V, 1.0V. Total chip power is the sum of passive power, computed by multiplying the idle power by the fraction of the chip's cores under use, and active power — computed by subtracting idle power from the total power measured when the system is accepting input events .

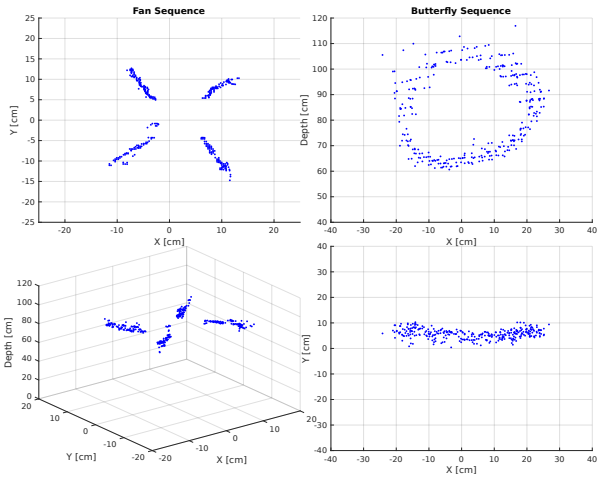

Figure 6. Depth reconstruction of the fan (first column) and butterfly sequence (second column), each shown from two viewpoints. Each point in the butterfly sequence shown is the median coordinate estimate of the butterfly location at a distinct time instant.

5.4. Results

The RDS is tested on a model using 3×5 spatial windows, left-right consistency constraints, no morphological erosion/dilation after rectification, and 31 disparity levels (0-30) plus a 'no-disparity' indicator (often occurring due to self-occlusions). We also experiment with a post-processing phase with erosion and dilation applied to output disparity maps in order to better regularize the output. Average disparity error and recall before regularization is 0.19/0.66 and post-regularization is 0.04/0.63. We observe major improvements due to the regularization, often occurring in self-occluded regions. Errors increase in slanted regions due to foreshortening effects. The left-right consistency constraint decreases false predictions in those regions.

The evaluation on the non-synthetic dataset was done under the practical constraints of the availability of a limited number of NS1e boards on which non-simulated models could be run, as well as the need to process the full DAVIS inputs at as high of a throughput as possible. The models that run on live DAVIS input are operated at spike injection rate of up to 2,000Hz (a new input every 1/2,000 seconds) and disparity map throughput of 400Hz at a 0.5ms tick period (400 distinct disparity maps produced every second) across a cluster of 9 TrueNorth chips. Single chip passive/active power on a characteristic model and input is 34.4mW/35.8mW (0.8V) and 82.1mW/56.4mW (1.0V).

Table 3. Comparison of event based depth estimation literature (a blank ' ' means feature not present, a '-' means unknown, a 'X' denotes the presence of the respective feature). As the baseline comparison datapoint, we use a system tested end-to-end with live camera feed and running in real-time on 9 TrueNorth boards. See the Experiments section for a discussion on other TrueNorth systems tested with different speed vs. power vs. input size tradeoffs. The relative error indicated for some papers is an approximation of the value, extrapolated from the reported data (the papers do not use the same evaluation dataset/metrics).

Approaches	Ours	Osswald [45]	Dikov[17]	Schraml[60]	Schraml[59]	Piatkowska[49]	Eibensteiner[20]	Mahowald[40]	Rogister[52]	Camuñas[13]
Features of disparity algorithm and implementation										
Fully neuromorphic disparity computation	X							X		
Neuromorphic rectification of input	X									
Real time depth from live sensor input	X			X	X				X	X
Multi-resolution disparity computation	X							X		
Bidirectional consistency check	X			X	X		X		X	
Scene-independent throughput & latency	X			X	X		X		X	X
Uses event polarity compatibility	X	X					X	NA	X	X
Tested on dense RDS data	X	X						X		
Tested on both fast and slow motions	X		X				X		X	
Implementation metrics										
Algorithm implementation hardware	Neuro	FPGA	CPU	DSP	CPU	CPU	FPGA	ASIC	CPU	FPGA
Energy consumption (mWatts/Pixel)	0.058	-	16	0.30	-	-	-	-	-	-
Disparity maps per second	400	151	500	200	-	-	1140	40	3333	20
System latency (ms)	9	6.6	2	5	-	-	0.87	25	0.3	50
Image size, per sensor, in real-time (pixels)	10800	32400	11236	16384	1.4 M	-	16384	57	16384	16384
Disparity levels in real-time	21	30	32	-	-	-	36	9	128	-
Relative error extrapolation	Fig.5	13.4-21%	-	6-10%	-	3-6%	6-16%	-	-	-

Running a model at the full 2,000Hz throughput comes at the expense of an increased neuron count. By adding a multiplexing spiking network to the network, we are able to reuse each feature-extraction/WTA circuit to process the disparities for 5 different pixels, effectively decreasing the maximum disparity map throughput from 2,000Hz to 400Hz, requiring fewer chips to process the full image (9 TrueNorth chips). We tested the maximum disparity map throughput achievable, by executing a one-chip model on a cropped input, with no multiplexing (one disparity map ejected per tick) at a 0.5ms tick period, achieving the 2,000Hz disparity map throughput. We tested sixty models on the TrueNorth simulator which provides a spike-for-spike equivalent behavior to the chip. We achieved best relative errors of $5 - 11.6\%$ and $7.3 - 8\%$ on the Fan and Butterfly sequence respectively (Fig. 5). We also observe qualitatively good performance (Fig. 6). It is observed that the temporal scale has a higher effect on accuracy than spatial scale. Left-right consistency constraints are typically present in the best performing fan-sequence models, but not so in the Butterfly sequences. Distance and orientation do not have a significant effect on performance. See supplementary materials for more details.

6. Discussion

We have introduced an advanced neuromorphic 3D vision system uniting a pair of DAVIS cameras with multiple TrueNorth processors, to create an end-to-end, scalable, event-based stereo system. By using a spiking neural network, with low-precision weights, we have shown that the system is capable of injecting event streams and ejecting disparity maps at high throughputs, low latencies, and low power. The system is highly parameterized and can operate with other event based sensors such as ATIS [50] or DVS [37]. Table 3 compares our approach with the literature on event based disparity. Comparative advantages are low power, multi-resolution disparity calculation, scalability to live sensor feed with large input sizes, and evaluation using synthetic as well as real world fast movements and depth gradients, in neuromorphic, non von-Neumann hardware. The implemented neuromorphic stereo disparity system achieves these advantages, while consuming $\sim 200\times$ less power per pixel per disparity map compared to the state-of-the-art [17]. The homogeneous computational substrate provides the first example of a fully end-to-end low-power, high throughput fully event-based neuromorphic stereo system capable of running on live input event streams, using a fully graph-based computation model, where no frames, arrays or other such data-structures are used.

References

[1] A. Amir, P. Datta, W. P. Risk, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, M. Flickner, R. Alvarez-Icaza, et al. Cognitive computing programming paradigm: a corelet language for composing networks of neurosynaptic cores. In *International Joint Conference on Neural Networks (IJCNN)*, 2013. 3

[2] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, et al. A low power, fully event-based gesture recognition system. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3, 7

[3] A. Andreopoulos, B. Taba, A. S. Cassidy, R. Alvarez-Icaza, M. Flickner, W. P. Risk, A. Amir, P. Merolla, J. V. Arthur, D. J. Berg, et al. Visual saliency on networks of neurosynaptic cores. *IBM Journal of Research and Development*, 59(2/3):9–1, 2015. 1

[4] A. Andreopoulos and J. K. Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891, 2013. 1

[5] J. T. Barron and B. Poole. The fast bilateral solver. In *European Conference on Computer Vision*, 2016. 2

[6] R. W. Beard, D. B. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. W. McLain, and M. A. Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *JACIC*, 2(1):92–108, 2005. 1

[7] A. N. Belbachir, S. Schraml, M. Mayerhofer, and M. Hofstätter. A novel hdr depth camera for real-time 3d 360 panoramic vision. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 425–432. IEEE, 2014. 2, 3

[8] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[9] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014. 1

[10] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1, 2, 3

[11] T. Brosch and H. Neumann. Event-based optical flow on neuromorphic hardware. In *proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) on 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 551–558. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016. 1

[12] L. Camunas-Mesa, T. Serrano-Gotarredona, B. Linares-Barranco, S. Ieng, and R. Benosman. Event-driven stereo vision with orientation filters. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 257–260, 2014. 2

[13] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco. On the use of orientation filters for 3d reconstruction in event-driven stereo vision. *Frontiers in neuroscience*, 8, 2014. 2, 8

[14] J. Carneiro, S.-H. Ieng, C. Posch, and R. Benosman. Event-based 3d reconstruction from neuromorphic retinas. *Neural Networks*, 45:27–38, 2013. 2

[15] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, et al. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. In *International Joint Conference on Neural Networks (IJCNN)*, 2013. 3

[16] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012. 1

[17] G. Dikov, M. Firouzi, F. Röhrbein, J. Conradt, and C. Richter. Spiking cooperative stereo-matching at 2 ms latency with neuromorphic hardware. In *Conference on Biomimetic and Biohybrid Systems*, pages 119–137. Springer, 2017. 2, 8

[18] S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier. Sparse stereo disparity map densification using hierarchical image segmentation. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 172–184. Springer, 2017. 2

[19] F. Eibensteiner, H. G. Brachtendorf, and J. Scharinger. Event-driven stereo vision algorithm based on silicon retina sensors. In *Radioelektronika (RADIOELEKTRONIKA)*, 2017. 3

[20] F. Eibensteiner, J. Kogler, and J. Scharinger. A high-performance hardware architecture for a frameless stereo vision algorithm implemented on a fpga platform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 623–630, 2014. 3, 8

[21] F. Ekstrand, C. Ahlberg, M. Ekström, and G. Spampinato. High-speed segmentation-driven high-resolution matching. In *Seventh International Conference on Machine Vision*, 2015. 2

[22] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, et al. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, 2016. 1

[23] M. Firouzi and J. Conradt. Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas. *Neural Processing Letters*, 43(2):311–326, 2016. 2

[24] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown. Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 62(12):2454–2467, 2013. 1, 2

[25] A. Fusiello, U. Castellani, and V. Murino. Relaxing symmetric multiple windows stereo using markov random fields. In *EMMCVPR*, pages 91–104. Springer, 2001. 2

[26] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[27] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[28] H. Huang and Q. Wang. A region and feature-based matching algorithm for dynamic object recognition. In *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, 2010. 2

[29] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze. A fast stereo matching algorithm suitable for embedded real-time systems. *Computer Vision and Image Understanding*, 114(11):1180–1202, 2010. 2

[30] G. Indiveri, E. Chicca, and R. Douglas. A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE transactions on neural networks*, 17(1):211–221, 2006. 1

[31] J. Kogler, M. Humenberger, and C. Sulzbachner. Event-based stereo matching approaches for frameless address event stereo data. *Advances in Visual Computing*, pages 674–685, 2011. 2

[32] J. Kogler, C. Sulzbachner, M. Humenberger, and F. Eibensteiner. Address-event based stereo vision with bio-inspired silicon retina imagers. In *Advances in theory and applications of stereo vision*. InTech, 2011. 2

[33] J. Kogler, C. Sulzbachner, and W. Kubinger. Bio-inspired stereo vision system with silicon retina imagers. *Computer Vision Systems*, pages 174–183, 2009. 2

[34] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos. Review of stereo vision algorithms: from software to hardware. *International Journal of Optomechatronics*, 2(4):435–462, 2008. 2

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[36] Y. Li, H.-Y. Shum, C.-K. Tang, and R. Szeliski. Stereo reconstruction from multiperspective panoramas. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):45–62, 2004. 2

[37] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *IEEE International Solid-State Circuits Conference*, 2006. 1, 2, 8

[38] J. Liu, X. Sang, C. Jia, N. Guo, Y. Liu, and G. Shi. Efficient stereo matching algorithm with edge-detecting. In *Optoelectronic Imaging and Multimedia Technology III*, 2014. 2

[39] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[40] M. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, California Institute of Technology, 1992. 2, 8

[41] M. Mahowald and T. Delbrück. Cooperative stereo matching using static and dynamic image features. *Analog VLSI implementation of neural systems*, 80:213–238, 1989. 2

[42] D. Marr, T. Poggio, et al. Cooperative computation of stereo disparity. *From the Retina to the Neocortex*, pages 239–243, 1976. 2, 3

[43] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 2

[44] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014. 1, 2, 3

[45] M. Osswald, S.-H. Ieng, R. Benosman, and G. Indiveri. A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Scientific reports*, 2017. 2, 8

[46] H. Park and K. M. Lee. Look wider to match image patches with convolutional neural networks. *IEEE Signal Processing Letters*, 2016. 2

[47] E. Piatkowska, A. Belbachir, and M. Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 45–50, 2013. 2

[48] E. Piatkowska, A. N. Belbachir, and M. Gelautz. Cooperative and asynchronous stereo vision for dynamic vision sensors. *Measurement Science and Technology*, 2014. 2

[49] E. Piatkowska, J. Kogler, N. Belbachir, and M. Gelautz. Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 2, 8

[50] C. Posch, D. Matolin, and R. Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011. 1, 8

[51] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999. 1

[52] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, 2012. 2, 8

[53] J. Sawada, F. Akopyan, A. S. Cassidy, B. Taba, M. V. Debole, P. Datta, R. Alvarez-Icaza, A. Amir, J. V. Arthur, A. Andreopoulos, et al. Truenorth ecosystem for brain-inspired computing: scalable systems, software, and applications. In *High Performance Computing, Networking, Storage and Analysis, SC16: International Conference for*, pages 130–141. IEEE, 2016. 1, 2

[54] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014. 2

[55] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 2

[56] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *IEEE International Symposium on Circuits and systems (ISCAS)*, 2010. 1

[57] S. Schraml, A. N. Belbachir, and H. Bischof. An event-driven stereo system for real-time 3-d 360 panoramic vision. *IEEE Transactions on Industrial Electronics*, 63(1):418–428, 2016. 2, 3

[58] S. Schraml, A. N. Belbachir, N. Milosevic, and P. Schön. Dynamic stereo vision system for real-time tracking. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010. 2

[59] S. Schraml, A. Nabil Belbachir, and H. Bischof. Event-driven stereo matching for real-time 3d panoramic vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 466–474, 2015. 3, 8

[60] S. Schraml, P. Schön, and N. Milosevic. Smartcam for real-time stereo vision-address-event based embedded system. In *VISAPP (2)*, pages 466–471, 2007. 2, 8

[61] K. Sharma, K.-y. Jeong, and S.-G. Kim. Vision based autonomous vehicle navigation with self-organizing map feature matching technique. In *International Conference on Control, Automation and Systems (ICCAS)*, 2011. 2

[62] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald. Review of stereo vision algorithms and their suitability for resource-limited systems. *Journal of Real-Time Image Processing*, 11(1):5–25, 2016. 2

[63] F. Tombari and F. Gori. Evaluation of stereo algorithms for 3d object recognition. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011. 2

[64] J. Von Neumann. *The computer and the brain*. Yale University Press, 2012. 1

[65] Z. Xie, S. Chen, and G. Orchard. Event-based stereo depth estimation using belief propagation. *Frontiers in Neuroscience*, 11:535, 2017. 2

[66] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 2

[67] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 6