Telluride 2018 Neuromorphic Cognition Engineering Workshop

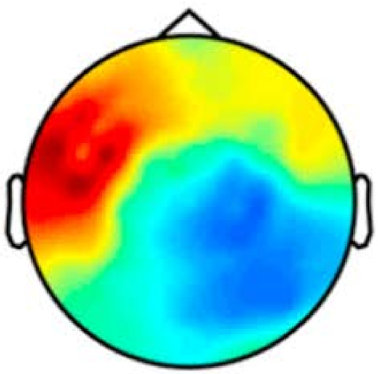
July 1-20, 2018. Telluride, Colorado, USA

Telluride 2018 Neuromorphic Cognition Engineering Workshop

July 1-20, 2018. Telluride, Colorado, USA

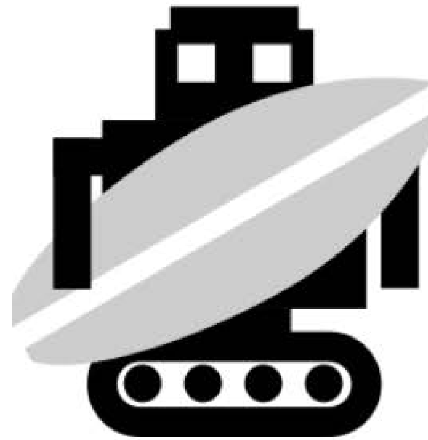
Topic Areas

AUD18: Prediction and Surprise in Natural Sound Processing: Comparing DNNs to the human brain

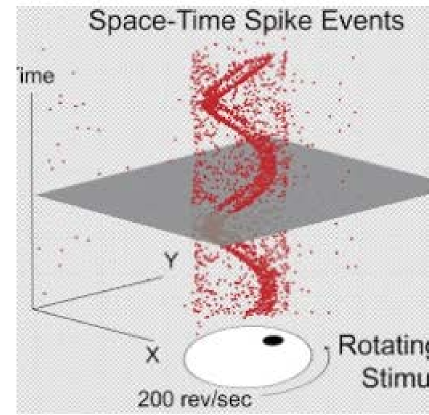


BRD18: Building applications with Braindrop —a novel neuromorphic chip for embodied perception and action

CAL18: Cognitive Agents that Learn in the Wild



ESP18: Fundamentals of Event Sensor Signal Processing



NSF

National Science Foundation

THE KAVLI FOUNDATION

Oticon Fonden

intel AI

PROPHESEE
METAVISION FOR MACHINES

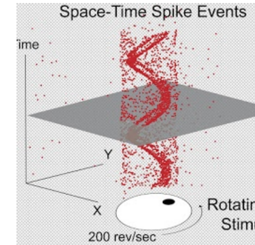
BRAIN VISION LLC

Google

iniVation

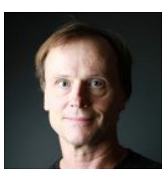
Telluride 2018 Neuromorphic Cognition Engineering Workshop

July 1-20, 2018. Telluride, Colorado, USA



ESP18: Fundamentals of Event Senor Signal Processing

- 1 . Can we lay a practical mathematical foundation that allows deriving efficient event-driven signal processing algorithms, analogous to the Z-**transform** of DSP?
- 2. Can we find better **noise reduction (NR)** algorithms than existing ones?
- 3. Can we find general methods for **adaptively controlling sensor parameters** like threshold, bandwidth, and refractory period?
- 4. Can we find **better input representations** for event cameras data for CNN?
- 5. What can we do to combine DVS events with **color vision**?

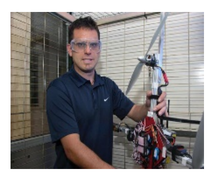


Tobi Delbruck
ETH

Ryad Benosman
UMPC

Garrick Orchard
NUS

Cornelia Fermuller
Univ.Maryland



David Mascarenas
LANS

Yiannis Andreopoulus
UCL

Francisco
Univ, Grenada

Alex Zhu
Univ, Penn.

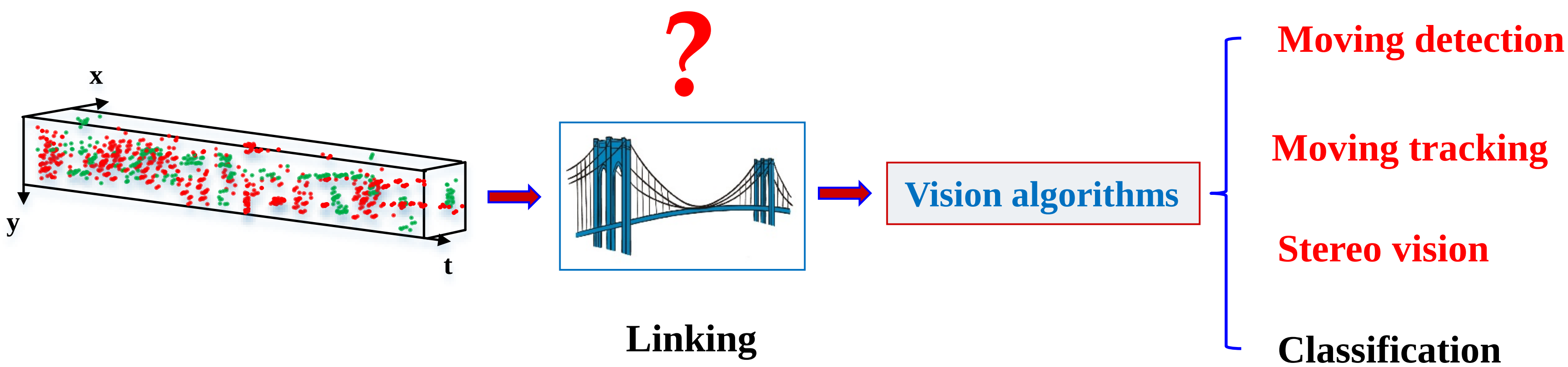
How to find better input representations for event-based camera data?

Jianing Li Spiking Computing Group lijianing@pku.edu.cn

Sep. 28, 2018

Motivation

- **Sparse and asynchronous spatial-temporal events**
  - High temporal resolution
  - Low spatial resolution



Overview

- **Introduction**
  - Event-based sensors
  - Related works
- **Image representations**
  - Steering prediction, CVPR 2018
- **Time surface representations**
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- **Feature representations**
  - Bag of Events, TNNLS 2017
- **End-to-end SNN**
  - STDP, TNNLS 2014
- **Discussion**
  - Better input representations for CNN
  - Event-based sensors future

Overview

- **Introduction**
  - Event-based sensors
  - Related works
- **Image representations**
  - Steering prediction, CVPR 2018
- **Time surface**
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- **Feature representations**
  - Bag of Events, TNNLS 2017
- **End-to-end SNN**
  - STDP, TNNLS 2014
- **Discussion**
  - Better input representations for CNN
  - Event-based sensors future

Event-based sensors

Milestones

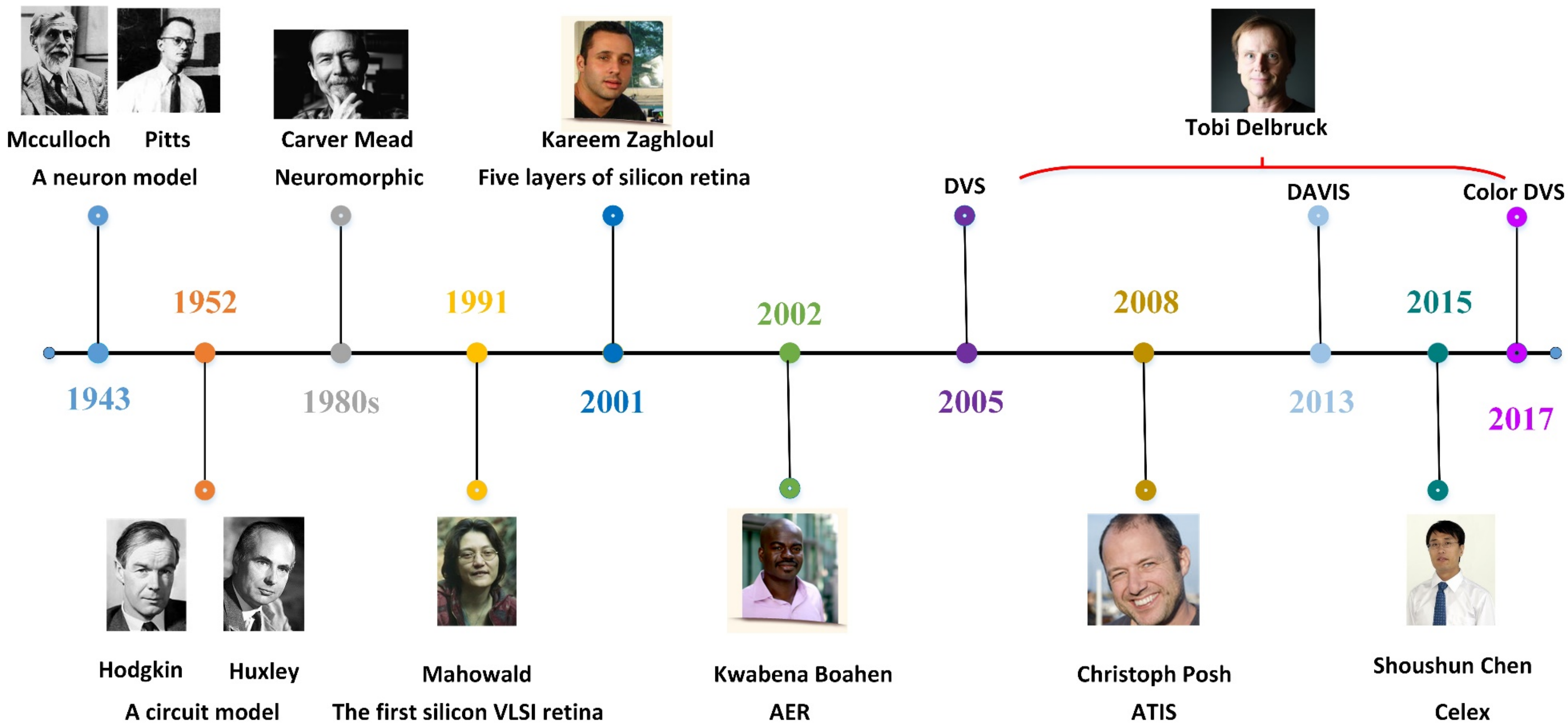


Fig.1 The time diagram for vision devices [1]

[1] Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output, C Posch. *Proceedings of IEEE*, 2014.

Event-based sensors

Bioinspired vision

The advantage

(1) High temporal resolution
(2) Low redundancy
(3) High dynamic range

The disadvantage

(1) Sensitive to noise
(2) Low spatial resolution
(3) Spatio-temporal sparse

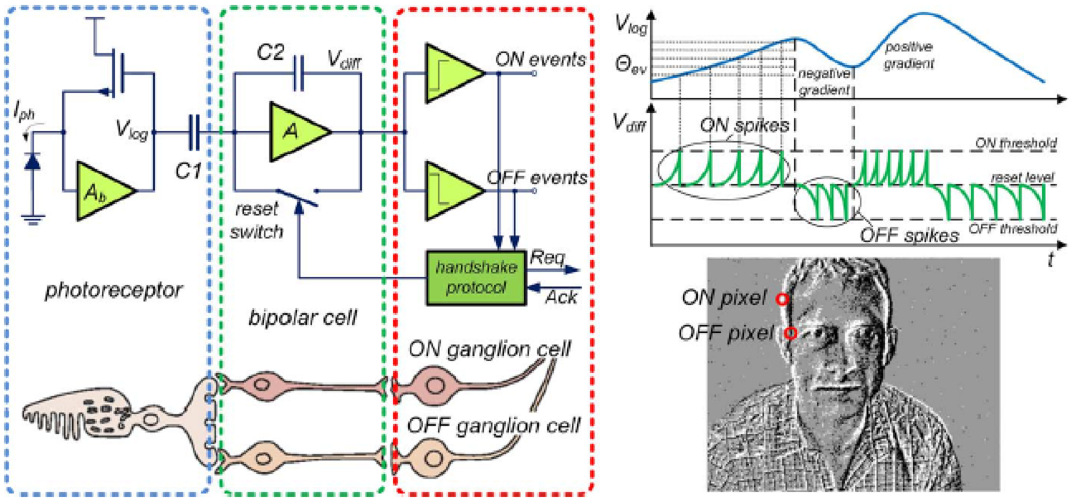


Fig.2 Three-layer model of silicon retina and DVS [1]

$$ev_i = [X_i, t_i, p_i]^T, i \in N$$

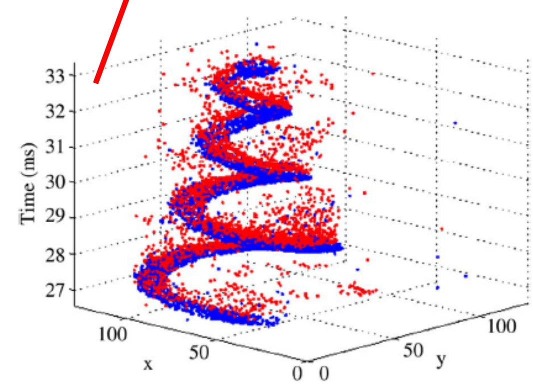


Fig.3 Illustration of DVS output [1]

[1] Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output, C Posch. *Proceedings of IEEE*, 2014.

Event-based sensors

Cameras

Camera	DVS	ATIS	DAVIS	Celex
Function	Dynamic event	Dynamic event + Intensity	Dynamic event + image capture	Dynamic event + image capture
First release year	2005	2008	2013	2017
Fixed Pattern noise	2.1%	0.25% intensity	0.5%APS, DVS 3.5%	0.38%
Power consumption	24mW	175mW (high activity) 50mW(low activity)	14mW(high activity) 5mW(low activity)	700mW
Array size	128*128	304*240	240*180	**1280*720**
Pixel Size(um^2)	40*40	30*30	18.5*18.5	30*30
Latency	15us@1klux	4us@1klux	3us@1klux	6us@1klux
Dynamic range	**120dB**	**125dB**	**130dB DVS** **51dB APS**	**120dB**
Commercialization	Commercialized (DVS128)	Commercialized (ATIS304)	Commercialized (DAVIS240)	Prepared

Tab.1 Papers of event-based vision in related topics [2,3]

[2] A Review of Bioinspired Vision Sensors and Their Applications, D Cho et al. ***Sensors & Materials*** , 2015.
[3] A Dynamic vision with direct logarithmic output and full-frame picture-on-demand, M Guo. ***PHD***, 2016.

Event-based sensors

☐ DVS VS standard camera

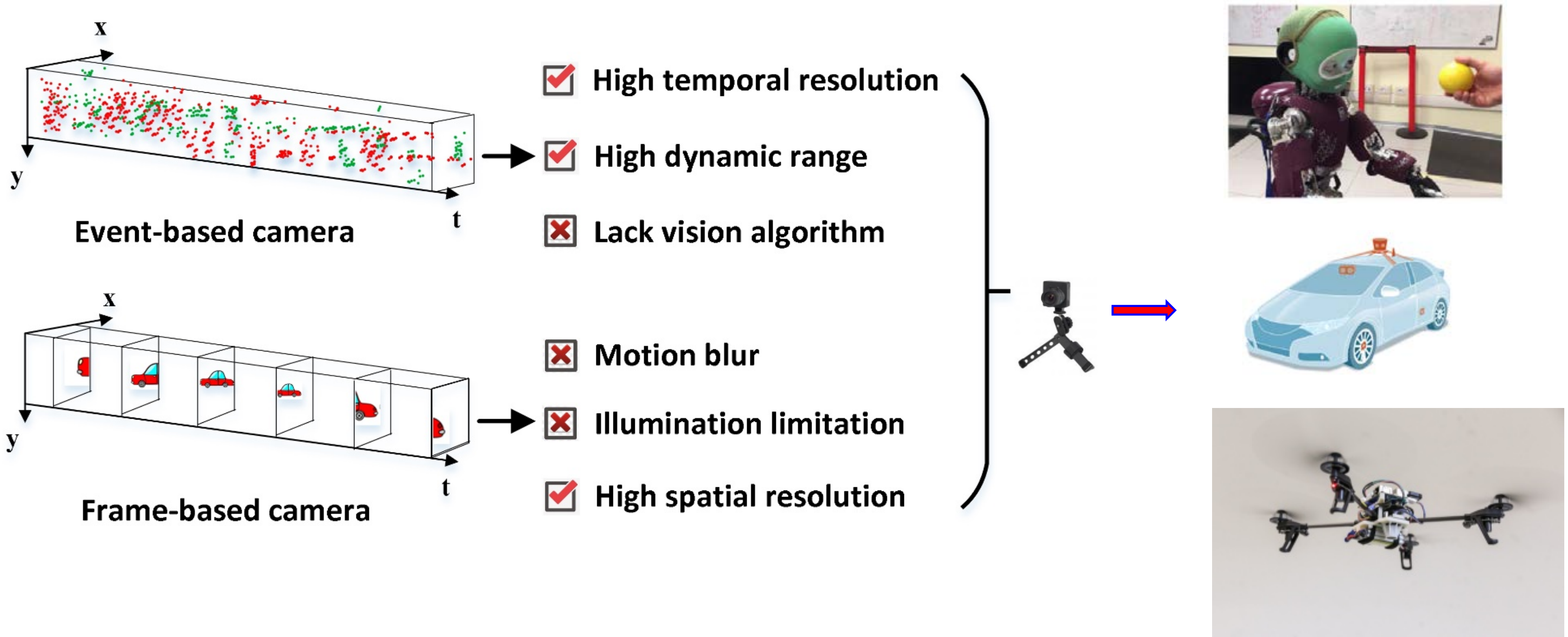


Fig.4 Event-based and frame-based cameras in applications

Event-based sensors

- **Embedded neuromorphic chip**
  - To mimic neural network architecture of biological brains
  - Low-power

TureNorth [4]
IBM

SpiNNaker [5]
Uni. Manchester

Loihi [6]
Intel

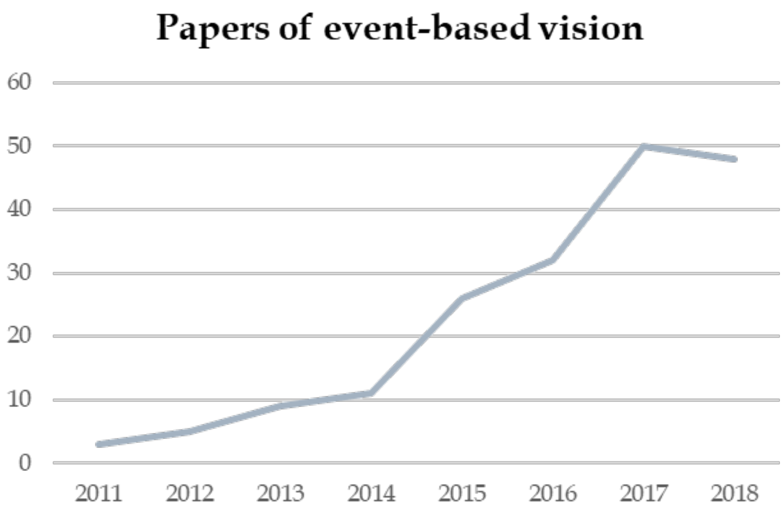
Fig.5 Bioinspired neuromorphic chips

[4] A million spiking-neuron integrated circuit with a scalable communication network and interface, Paul A. Merolla et.al, ***Science***, 2014.
[5] The SpiNNaker Project, Steve B. Furber et.al, ***The Proceedings of IEEE***, 2014.
[6] Loihi: A neuromorphic manycore processor with on-chip learning, Mike Davies, ***IEEE micro***, 2018.

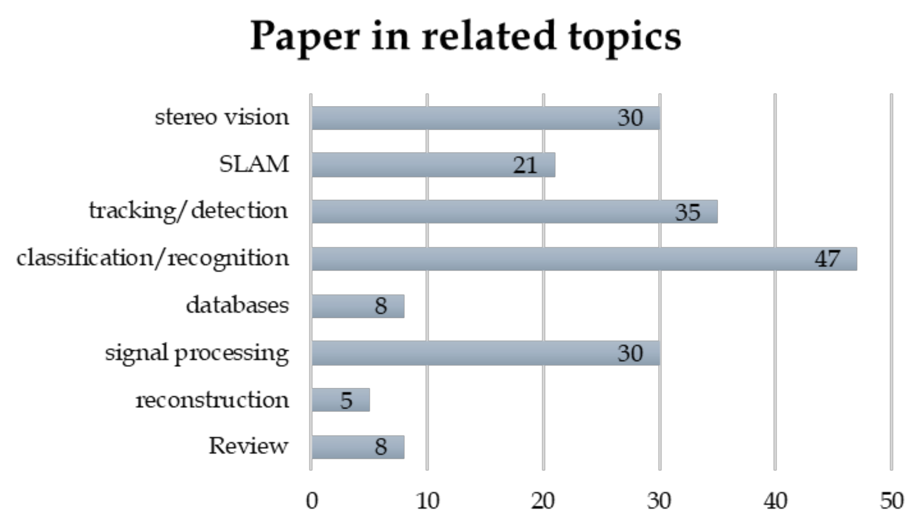
Related works

Event-based vision

- Research papers in recent years
- Related topics, mainly in vision applications



Tab.2 Papers of event-based vision in recent years



Tab.3 Papers of event-based vision in related topics

Tips: all statistical papers mainly are about DVS, ATIS and DAVIS.

Overview

- Introduction
  - Event-based sensors
  - Related works
- **Image representations**
  - Steering prediction, CVPR 2018
- Time surface representations
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- Feature representations
  - Bag of Events, TNNLS 2017
- End-to-end SNN
  - STDP, TNNLS 2014
- Discussion
  - Better input representations for CNN
  - Event-based sensors future

Event-based vision meet deep learning on steering prediction for self-driving cars

Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcia,
Davide Scaramuzza *

CVPR, 2018

1 Introduction

Motivation

- Challenging illumination conditions
- Fast motion

Contributions

- **Deep learning to event-based vison on regression task**
- Show that possible transfer learning from pre-trained CNN
- Outperforming state-of-art systems

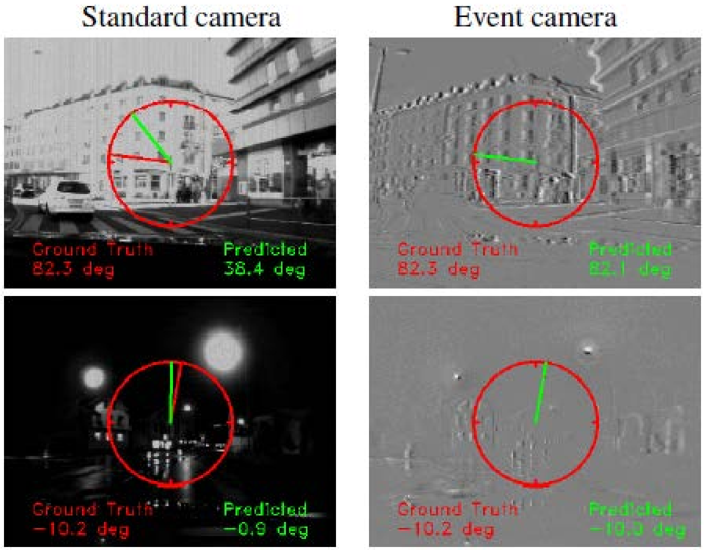


Fig.1 Steering angle performance on frames and event camera.

2 Framework

Methodology

- **Event-to-Frame conversation**
- Network architecture

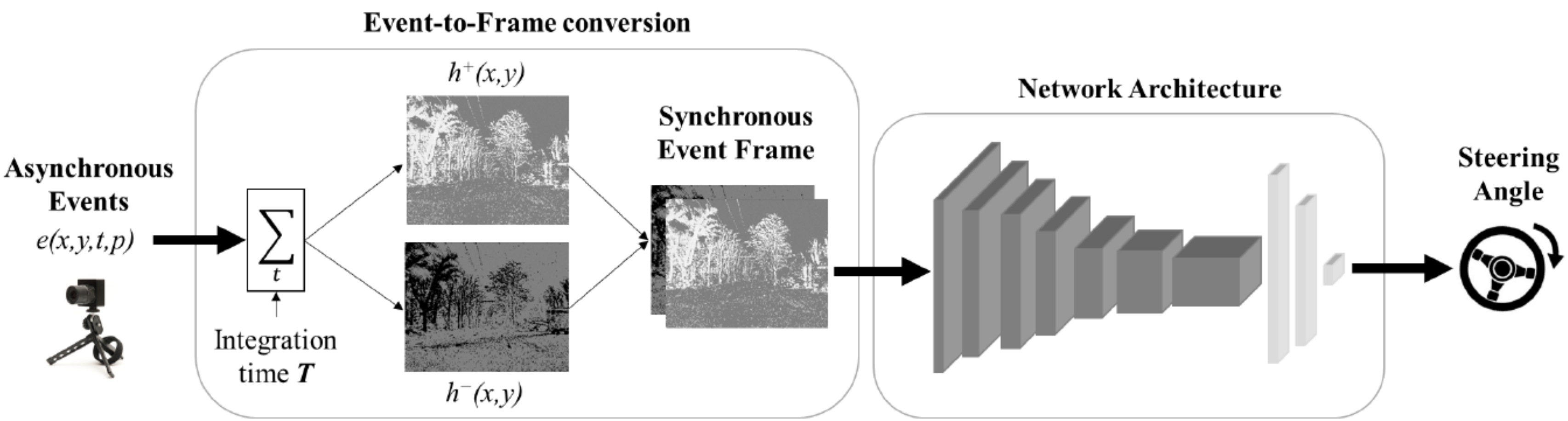


Fig.2 The framework of steering angle prediction based on event-based camera.

3 Integration time for events

Performance metrics

- RMSE
- EVA

$$\mathrm{RMSE} \doteq \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{\alpha}_j-\alpha_j)^2}.$$

$$\mathrm{EVA} \doteq 1-\frac{\mathrm{Var}(\hat{\alpha}-\alpha)}{\mathrm{Var}(\alpha)}.$$

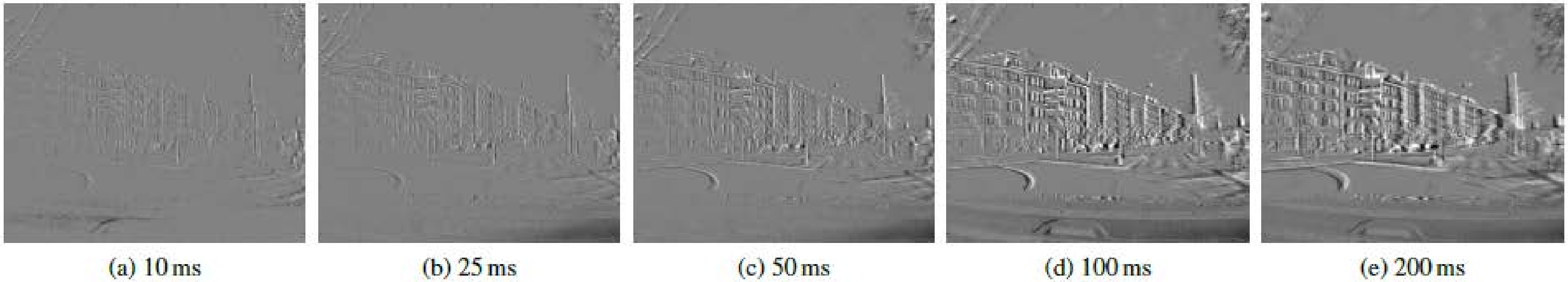


Fig.3 Events collected for different durations of the interval.

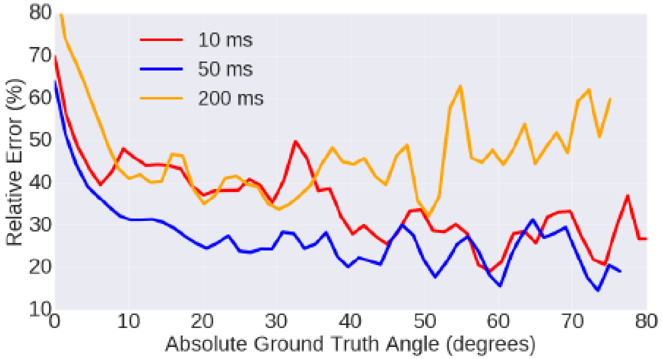


Fig.4 The relative error in steering angle prediction

Integration time T	EVA	RMSE
10 ms	0.790	11.53°
25 ms	0.792	10.42°
50 ms	**0.805**	**9.47°**
100 ms	0.634	13.43°
200 ms	0.457	15.87°

Tab.1 Comparison performances for different integration times

4 Experiments

Datasets

- DDD17 [1]

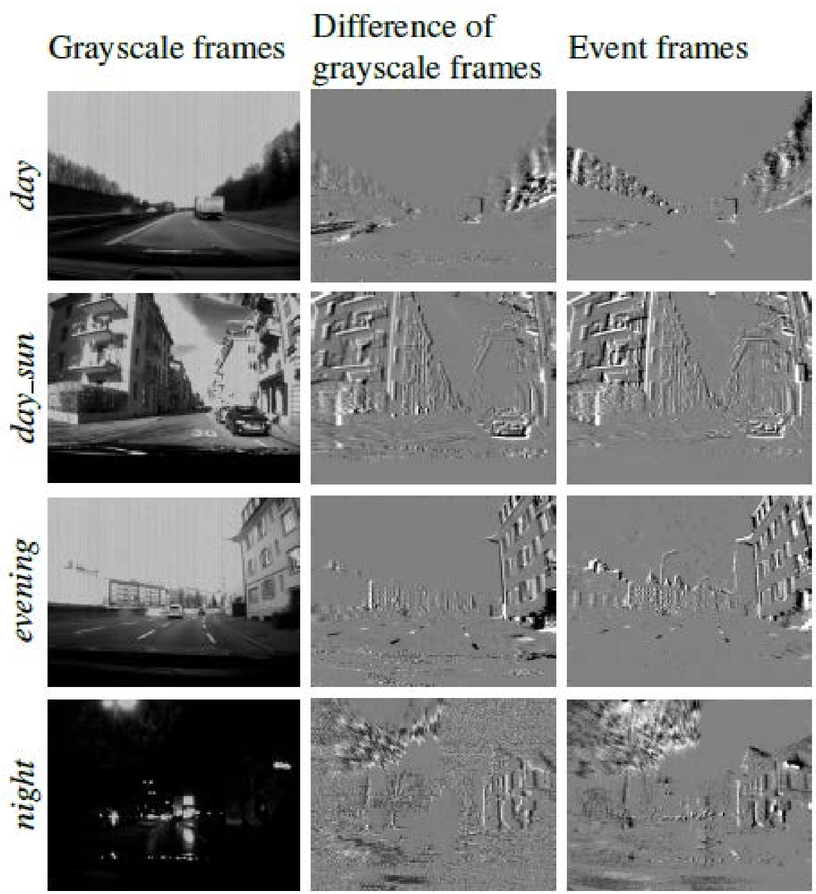


Fig.5 DDD17 dataset for four lighting conditions

Architecture	Grayscale EVA	Grayscale RMSE	Grayscale diff. EVA	Grayscale diff. RMSE	Events EVA	Events RMSE
ResNet18	0.047	4.57°	0.329	3.65°	**0.551**	**2.99°**
ResNet50	0.449	3.31°	0.653	2.62°	**0.728**	**2.33°**

Tab.2 Results for day subset

Architecture	Grayscale EVA	Grayscale RMSE	Grayscale diff. EVA	Grayscale diff. RMSE	Events EVA	Events RMSE
ResNet18	0.125	20.07°	0.729	11.53°	**0.742**	**10.87°**
ResNet50	0.383	16.85°	0.802	9.62°	**0.805**	**9.47°**

Tab.3 Results for day_sun subset

Architecture	Grayscale EVA	Grayscale RMSE	Grayscale diff. EVA	Grayscale diff. RMSE	Events EVA	Events RMSE
ResNet18	0.172	7.23°	0.183	7.19°	**0.518**	**5.45°**
ResNet50	0.360	6.37°	0.418	6.07°	**0.602**	**5.01°**

Tab.4 Results for evening subset

Architecture	Grayscale EVA	Grayscale RMSE	Grayscale diff. EVA	Grayscale diff. RMSE	Events EVA	Events RMSE
ResNet18	0.181	6.96°	0.449	5.73°	**0.654**	**4.51°**
ResNet50	0.418	5.88°	0.621	4.73°	**0.753**	**3.82°**

Tab.5 Results for night subset

[1] DDD17: End-to-end DAVIS driving dataset, Jonathan Binas et.al. ***ICML workshops***, 2017.

5 Outlook

- **1** **Adaptive integration time** to convert into images?
- **2** Generating feature maps based on **SNN**?
- **3** **Joint** frame-based and event-based in predicting steering angle?
- **4** How to use high temporal information?

Overview

- Introduction
  - Event-based sensors
  - Related works
- Image representations
  - Steering prediction, CVPR 2018
- **Time surface representations**
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- Feature representations
  - Bag of Events, TNNLS 2017
- End-to-end SNN
  - STDP, TNNLS 2014
- Discussion
  - Better input representations for CNN
  - Event-based sensors future

HATS: histograms of averaged time surfaces for robust event-based object classification [1]

Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, **Ryad B. Benosman ***

CVPR, 2018

[1] HOTS: A hierarchy of event-based time-surfaces for pattern recognition. Xavier Lagorce et.al . *PAMI* 2017.

1 Related works

Time surface

- Event streams

$$ev_i = [\mathbf{x_i}, t_i, p_i]^T, \quad i \in \mathbb{N}$$

- Time context

$$\mathcal{T}_i(\mathbf{u}, p) = \max_{j \leq i}\{t_j \,|\, \mathbf{x_j} = (\mathbf{x_i} + \mathbf{u}),\, p_j = p\}$$

- Computing time surface

$$\mathcal{S}_i(\mathbf{u}, p) = e^{-(t_i - \mathcal{T}_i(\mathbf{u}, p))/\tau}.$$

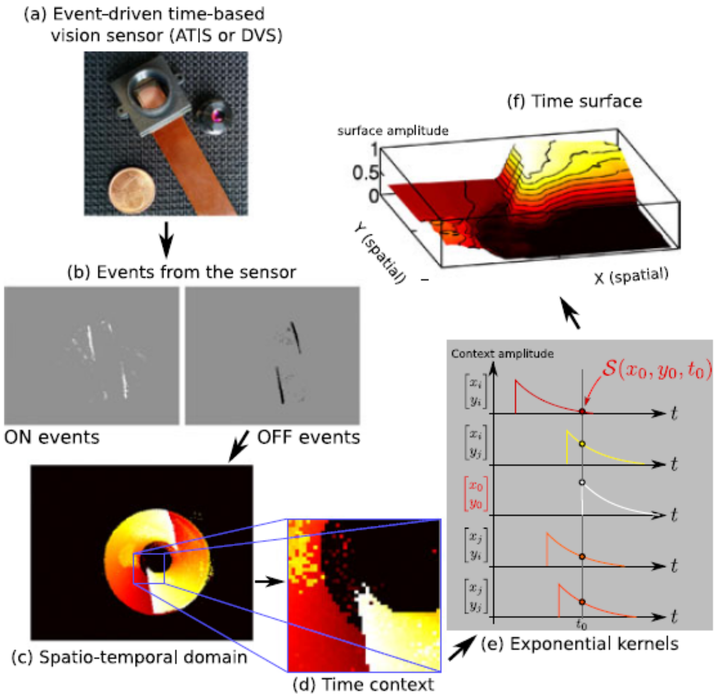


Fig.1 Time surface from the spatiotemporal events

1 Related works

- **Feature representations**
  - Online clustering of time-surfaces
  - Extracting features

$$feat_i = [x_i, y_i, t_i, k_i]^T,$$

Algorithm 1. Online Clustering of Time-Surfaces

Ensure: N cluster centers $\mathbf{C}_n$, $n \in [\![1, N]\!]$
  Use the first N events' time-surfaces as initial values for $\mathbf{C}_n$, $n \in [\![1, N]\!]$
  Initialize $p_n \leftarrow 1$, $n \in [\![1, N]\!]$
  for every incoming event ev_i **do**
    Compute time-surface $\mathcal{S}_i$
    Find closest cluster center $\mathbf{C}_k$
    $\alpha \leftarrow 0.01/(1 + p_k/20000)$
    $\beta \leftarrow \mathbf{C}_k \cdot \mathcal{S}_i/(||\mathbf{C}_k|| \cdot ||\mathcal{S}_i||)$
    $\mathbf{C}_k \leftarrow \mathbf{C}_k + \alpha(\mathcal{S} - \beta\mathbf{C}_k)$
    $p_k \leftarrow p_k + 1$
  end for

Tab.1 Online clustering of event streams based on time-surfaces

1 Related works

Framework

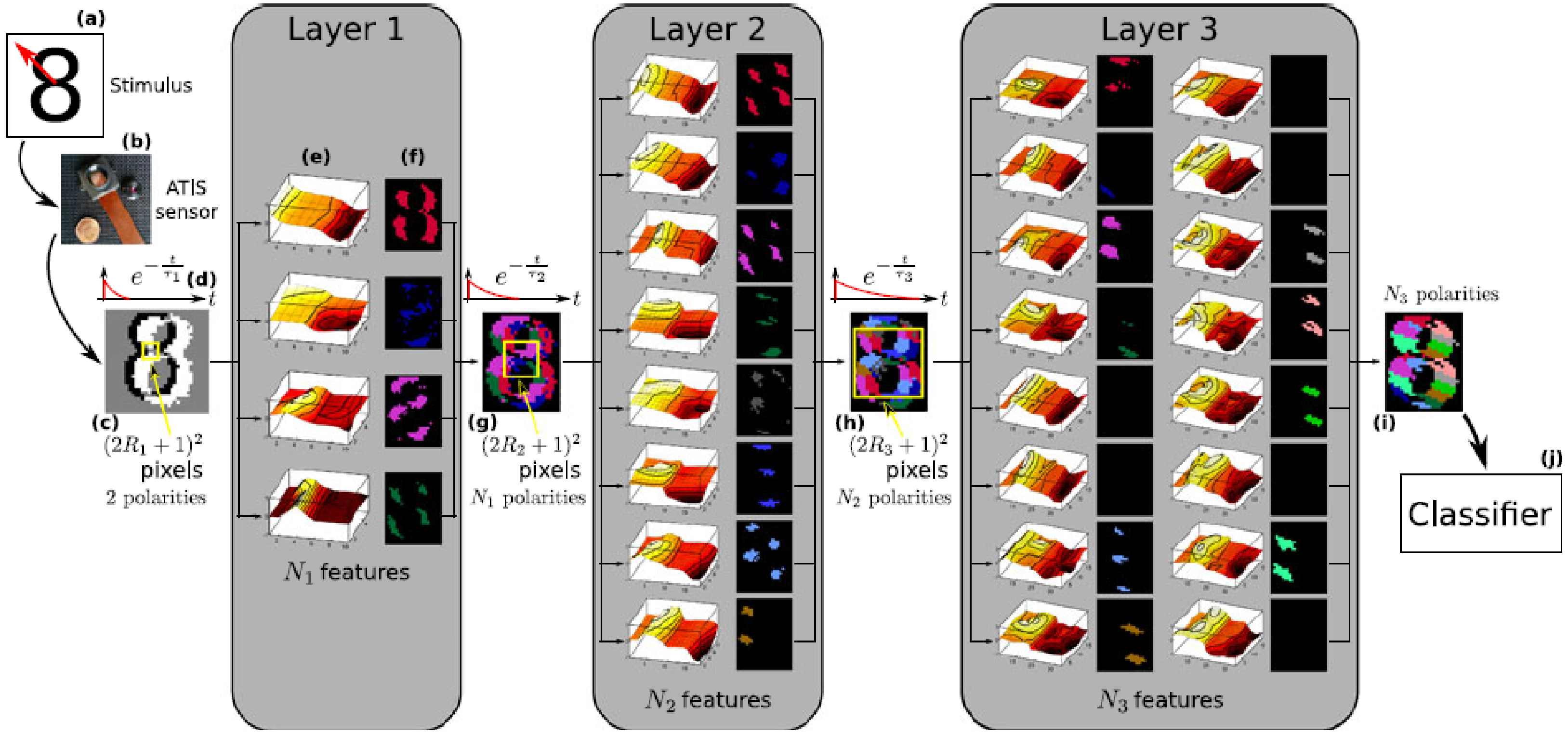


Fig.2 The proposed hierarchical framework based on time-surfaces

2 Introduction

- **Motivation**
  - Overcoming noisy events
  - Real-world event-based dataset
- **Contribution**
  - Local memory time surfaces
  - HATS—Histograms of averaged time surfaces
  - N-CARS dataset

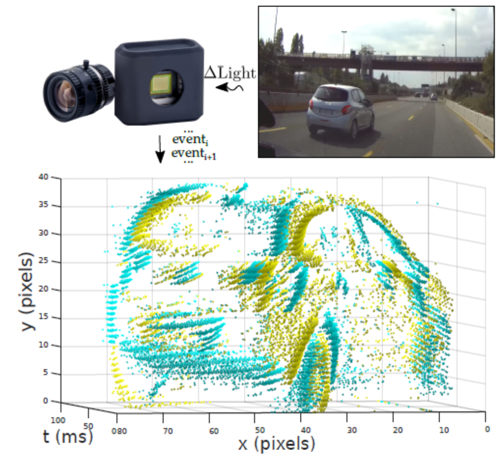


Fig.3 N-CARS dataset

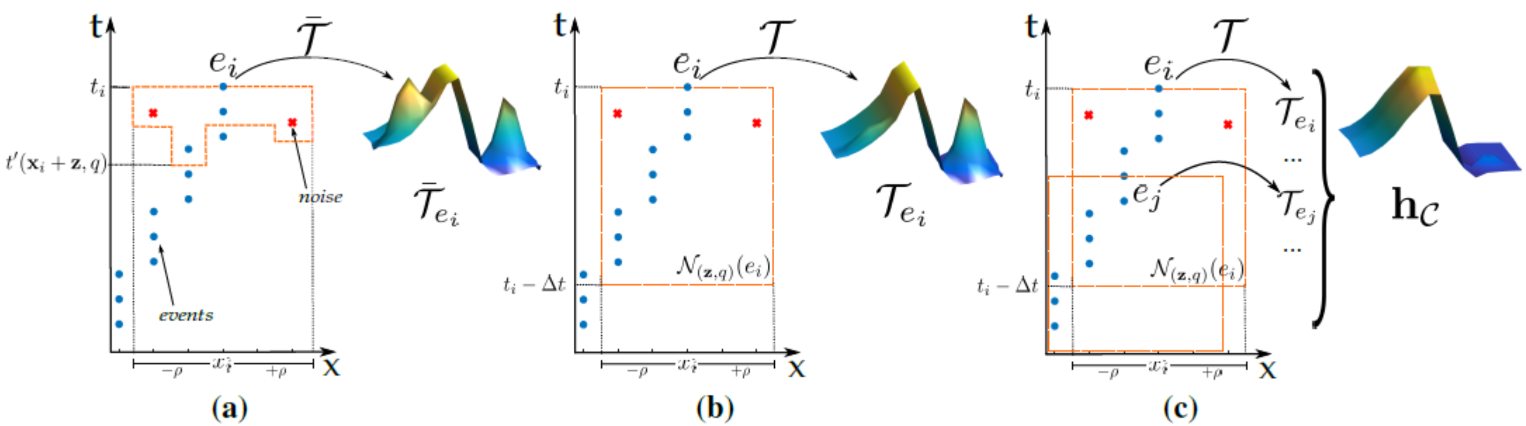


Fig.4 Time surface computation around an event, in presence of noise. (a)time surfaces; (b)local memory time surfaces; (c)HATS

3 Method

- **Local memory time surfaces**
  - Temporal window Δt

$$\mathcal{T}_{e_i}(\mathbf{z},q)=\begin{cases}\sum_{e_j\in\mathcal{N}_{(\mathbf{z},q)}(e_i)} e^{-\frac{t_i-t_j}{\tau}} & \text{if } p_i=q\\ 0 & \text{otherwise,}\end{cases}$$

$$\mathcal{N}_{(\mathbf{z},q)}(e_i)=\{e_j:\mathbf{x}_j=\mathbf{x}_i+\mathbf{z},t_j\in[t_i-\Delta t,t_i),p_j=q\}$$

- **Histograms of averaged time surfaces**
  - Averaged histogram

$$\mathbf{h}_{\mathcal{C}}(\mathbf{z},p)=\frac{1}{|\mathcal{C}|}\bar{\mathbf{h}}_{\mathcal{C}}(\mathbf{z},p)=\frac{1}{|\mathcal{C}|}\sum_{e_i\in\mathcal{C}}\mathcal{T}_{e_i}(\mathbf{z},p)$$

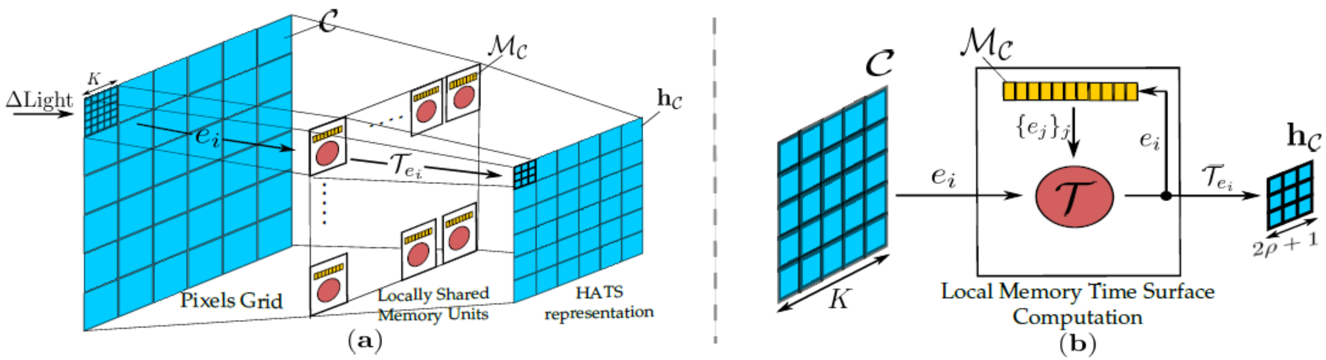


Fig.5 Overview of the proposed architecture. (a)Cells C; (b)Local memory time surface computation

3 Method

Algorithm

Algorithm 1 *HATS* with shared memory units

Input: Events $\mathcal{E} = \{e_i\}_{i=1}^{I}$ Parameters: $\rho, \Delta t, \tau, K$

Output: *HATS* representation $\mathbf{H}(\{e_i\})$

Initialize: $\mathbf{h}_{\mathcal{C}_l} = \mathbf{0},\ |\mathcal{C}_l| = 0,\ \mathcal{M}_{\mathcal{C}_l} = \emptyset$, for all l

for $i = 1, \ldots, I$ **do**

    $\mathcal{C}_l \leftarrow \mathrm{getCell}(x_i, y_i)$

    $\mathcal{T}_{e_i} \leftarrow \mathrm{computeTimeSurface}(e_i, \mathcal{M}_{\mathcal{C}_l})$

    $\mathbf{h}_{\mathcal{C}_l} \leftarrow \mathbf{h}_{\mathcal{C}_l} + \mathcal{T}_{e_i}$

    $\mathcal{M}_{\mathcal{C}_l} \leftarrow \mathcal{M}_{\mathcal{C}_l} \cup e_i$

    $|\mathcal{C}_l| \leftarrow |\mathcal{C}_l| + 1$

return $\mathbf{H} = [\mathbf{h}_{\mathcal{C}_1}/|\mathcal{C}_1|, \ldots, \mathbf{h}_{\mathcal{C}_L}/|\mathcal{C}_L|]^\intercal$

Tab.2 The algorithm of histograms average time-surfaces

4 Experiments

Classification accuracies

- database

Table 1 Compared methods for database

	N-MNIST	**N-Caltech101**	**MNIST-DVS**	**CIFAR10-DVS**
H-First [50]	0.712	0.054	0.595	0.077
HOTS [30]	0.808	0.210	0.803	0.271
Gabor-SNN	0.837	0.196	0.824	0.245
HATS (this work)	**0.991**	**0.642**	**0.984**	**0.524**
Phased LSTM [46]	0.973	-	-	-
Deep SNN [33]	0.987	-	-	-

Complexity analysis

- N-CARS

Table 2 Complexity analysis for N-CARS

N-CARS	Average Comp. Time per Sample (ms)	Kev/s
HOTS [30]	157.57	25.68
Gabor-SNN	285.95	14.15
HATS (this work)	**7.28**	**555.74**

5 Outlook

- **1** Decreasing **complexity**, rather than based on single spike?
- **2** **Local feature** representations?
- **3** **End-to-end architecture** used in spatial-temporal spike stream?

Overview

- Introduction
  - Event-based sensors
  - Related works
- Image representations
  - Steering prediction, CVPR 2018
- Time surface representations
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- **Feature representations**
  - Bag of Events, TNNLS 2017
- End-to-end SNN
  - STDP, TNNLS 2014
- Discussion
  - Better input representations for CNN
  - Event-based sensors future

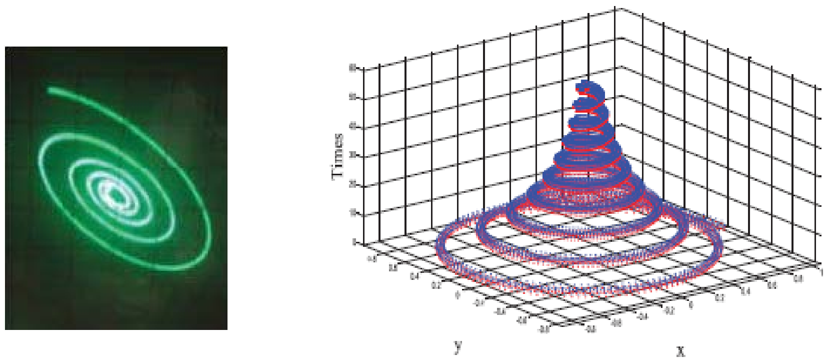
Bag of events: an efficient probability-based feature extraction method for AER image sensors

Xi Peng, Bo Zhao, Rui Yan, **Huajin Tang ***, Zhang Yi

TNNLS, 2017

1 Introduction

- **Challenges**
  - A sequence of events
  - Asynchronous and sparse



(a) Conventional camera (b) DVS

Fig.3 Event camera VS conventional camera.

- **Contribution**
  - BOE—feature extraction method based on probability theory
  - Online learning algorithm
  - Simple and competitive performance

2 Method

☐ Framework

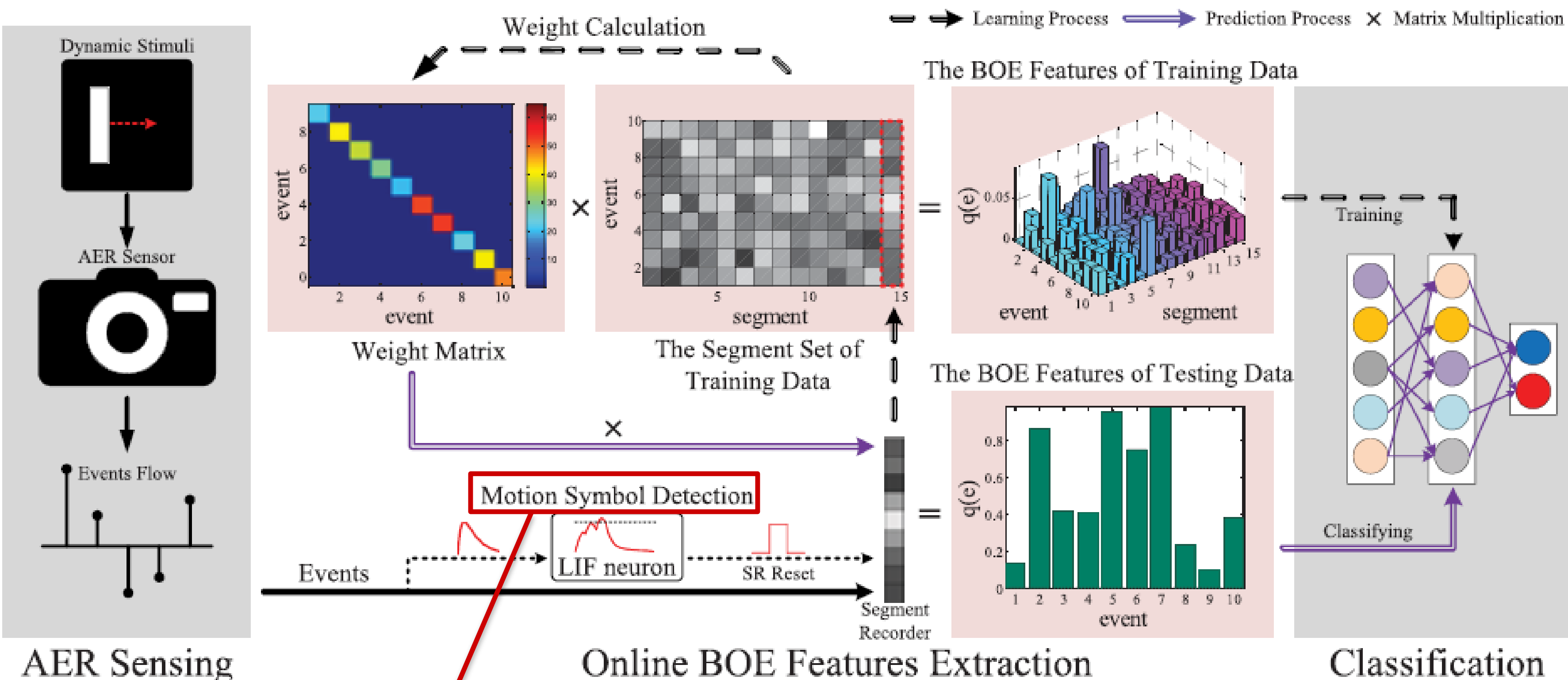


Fig.2 Architecture of the proposed system

Generating bags

2 Feature extracting

- **Bag of events**
  - LIF neural model

$$\mathcal{K}(t_i) = \exp\left(-\frac{t - t_i}{\tau}\right)$$

$$\mathcal{K}(t) = \sum_{t_i \in [t-1, t]} \mathcal{K}(t_i)$$

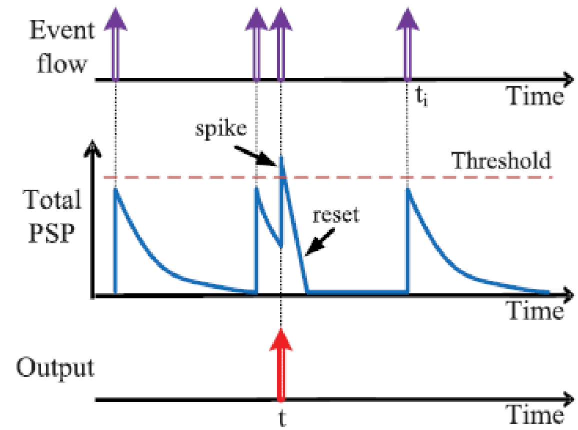


Fig.3 Dynamics of an LIF neuron

- **Event-based feature extracting**
  - Segments

$$\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$$

  - Bag of events

$$\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$$

  - Joint probability distribution

$$s_j = P(e_1, e_2, \ldots, e_m)$$

$$\boxed{[f_{1j}, f_{2j}, \ldots, f_{mj}]}$$

  - Feature representations

$$q_{ij} = w_i f_{ij}$$

$$\boxed{w_i = -\log \frac{n}{n_i}}$$

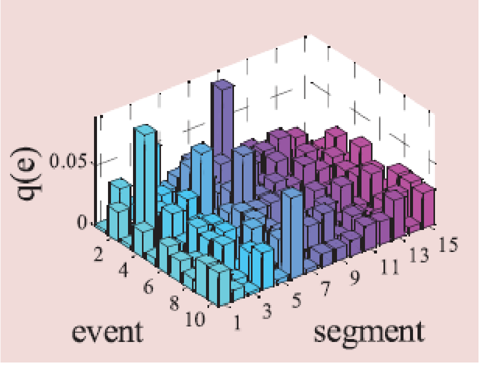


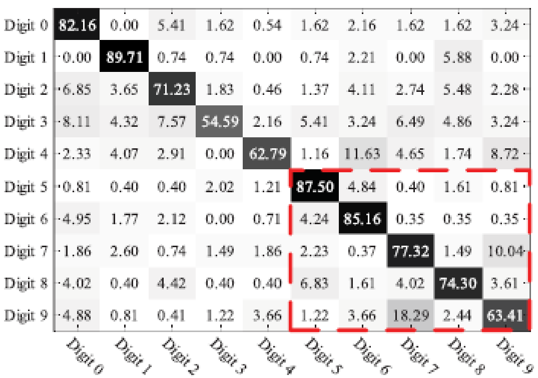
Fig.4 The BOE features

3 Experiments

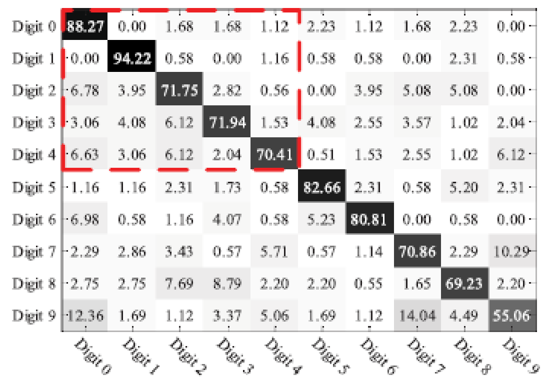
Classification accuracies

- MNIST-DVS

Tab. 1 Compared methods for MNIST-DVS database



BOE: 75.09%



Zhao et.al[1]: 73.35%

Chen et.al[2]: 61.23%

Complexity analysis

- Feature extraction and classification

Tab. 2 Compared methods for complexity analysis

Algorithms	Feature Extraction					Classification				
	training(s)	testing(s)	total(s)	fps	tpe(s)	training(s)	testing(s)	total(s)	fps	tpe(s)
BOE	**27.89**	**27.28**	**55.17**	**402.65**	**8.28E-06**	**3.63**	**0.12**	**3.75**	**5926.63**	**5.62E-07**
Zhao's [18]	8601.10	955.68	9556.78	1.87	1.17E-03	204.11	26.93	231.05	77.23	2.82E-05
Chen's [15]	1208.38	134.26	1342.64	16.69	2.00E-04	-	7691.26	7691.26	2.91	1.14E-03

[1] Feed-forward categorization on AER motion events using cortex-like features in a spiking neural network, Bo Zhao et.al, ***TNNLS*** 2015.
[2] Efficient feedforward categorization of objects and human postures with address-event image sensors, Shoushun Chen et.al, ***PAMI***, 2012.

4 Outlook

- 1 **Temporal information** can be feature representations?
- 2 **Local feature** representations?
- 3 **End-to-end SNN** used in spatial-temporal spike stream?

Overview

- Introduction
  - Event-based sensors
  - Related works
- Image representations
  - Steering prediction, CVPR 2018
- Time surface representations
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- Feature representations
  - Bag of Events, TNNLS 2017
- **End-to-end SNN**
  - STDP, TNNLS 2014
- Discussion
  - Better input representations for CNN
  - Event-based sensors future

Unsupervised learning of digit recognition using spike-timing-dependent plasticity

Peter U. Diehl *, and Matthew Cook

TNNLS, 2014

1 Introduction

Leaky-integrate-and-fire, LIF

- Firing model

$$\tau_m \frac{du}{dt} = -u(t) + RI(t)$$

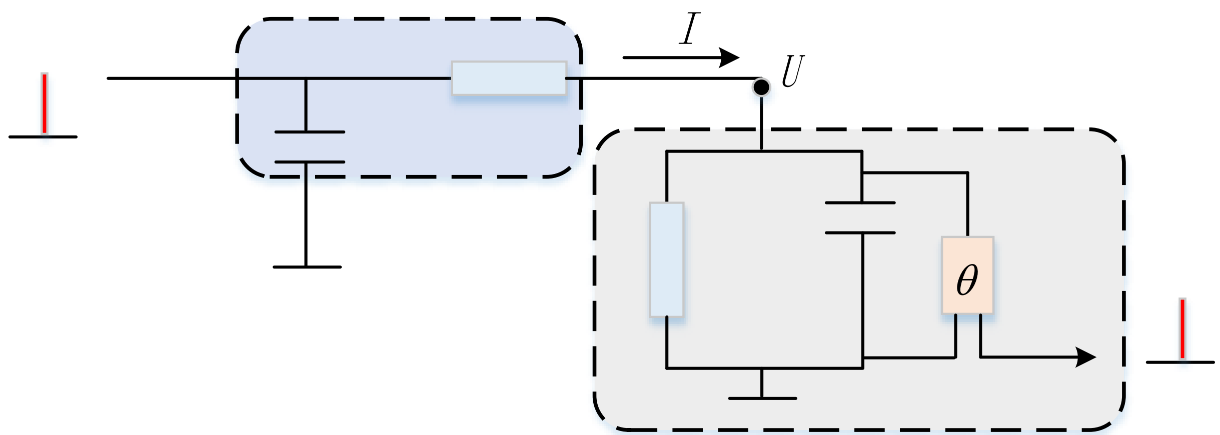


Fig.1 Leaky-integrate-and-fire model

1 Introduction

Spike-timing dependent plasticity, STDP

- Synapse plasticity

$$\left.\begin{aligned} \frac{dx_{pre}}{dt} &= -\frac{x_{pre}}{\tau_{pre}} \\ \frac{dx_{post}}{dt} &= -\frac{x_{post}}{\tau_{post}} \end{aligned}\right\} \quad \Delta w = \eta O(x_{pre}, x_{post})$$

- Synapse weight

$$\Delta w = \sum_{t_{pre}} \sum_{t_{post}} f(t_{post} - t_{pre})$$

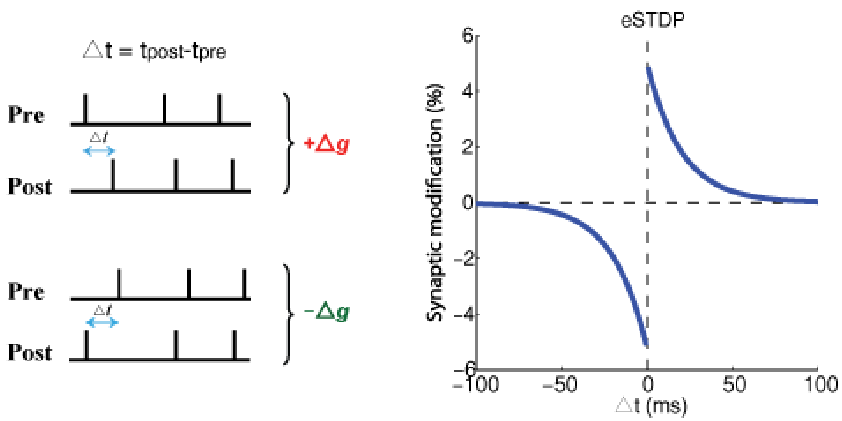


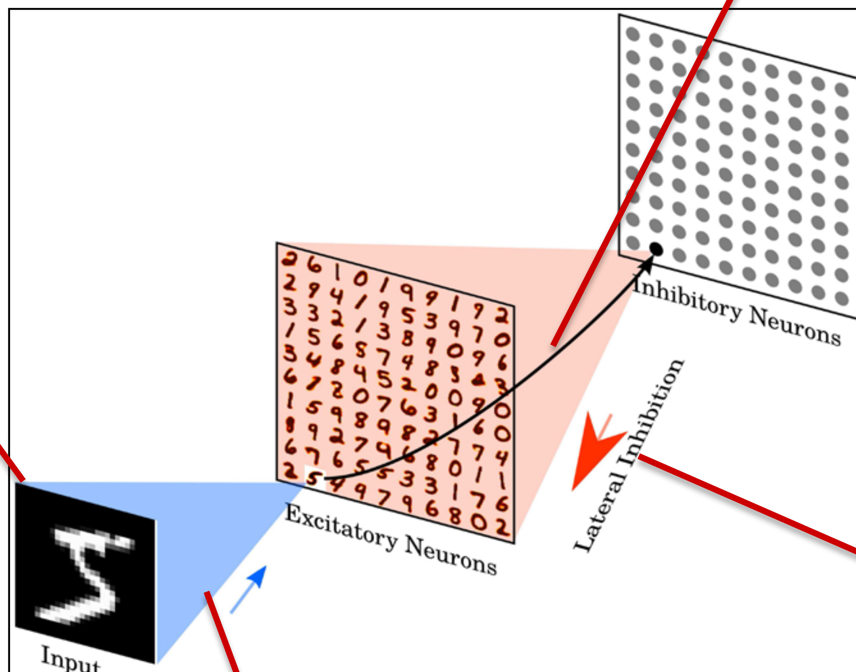
Fig.2 e-STDP learning function

2 Method

- **Network architecture**
  - **STDP called unsupervised learning**
  - How to use labels

One to one connections

Possion-spike with firing rates



Connected all except one

Fig.3 Two layers network

STDP learning rule

2 Method

Neuron and synapse model

- **LIF model**, the membrane voltage V [1]

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_e(E_{ext} - V) + g_i(E_{inh} - V)$$

Learning rule

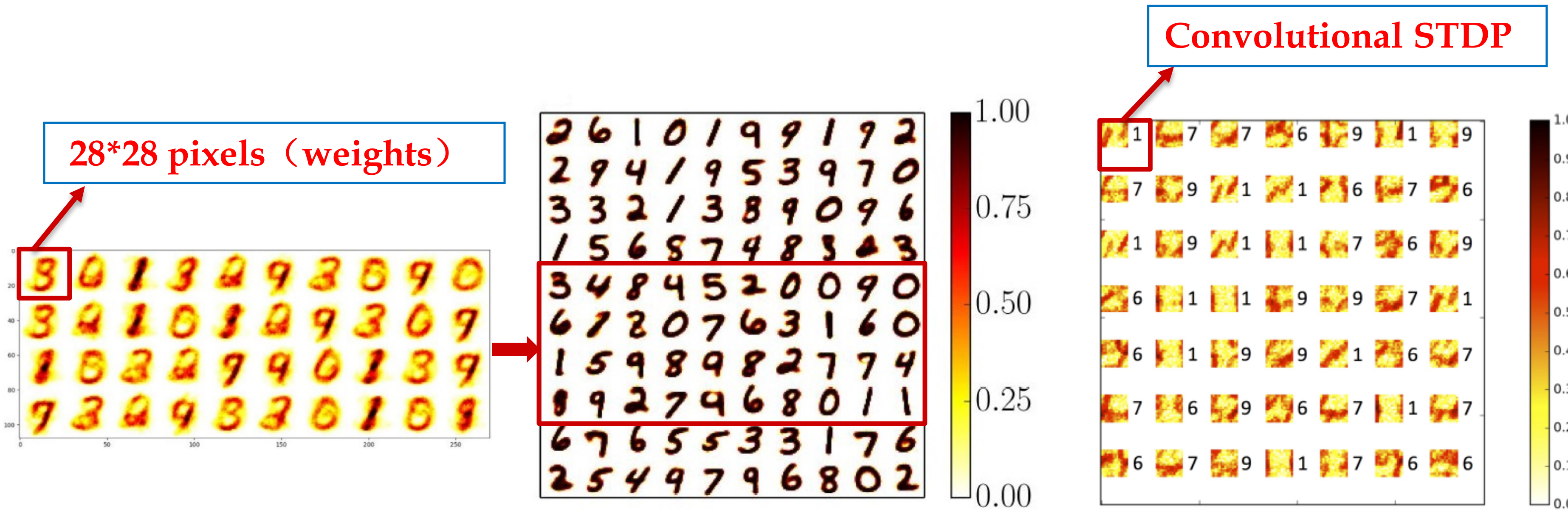
- Weight change

$$\Delta w = \eta\left(x_{pre} - x_{tar}\right)(w_{max} - w)^{\mu}$$

[1] Brian2 link: http://brian2.readthedocs.io/en/stable/index.html

2 Method

- **Train**
  - **Assigned excitatory neurons**



(a) 4*10 excitatory neurons (b) 10*10 excitatory neurons(400) (c) 7*7 excitatory neurons [2]

Fig.4 2D receptive fields

[2] Deep spiking convolutional neural network trained with unsupervised spike timing dependent plasticity. Chankyu Lee., TCDS, 2018.

3 Experiments

Results

Event-based data?

Architecture	Preprocessing	Training-type	(Un-)supervised	Learning-rule	Performance
Dendritic neurons (Hussain et al., 2014)	Thresholding	Rate-based	Supervised	Morphology learning	90.3%
Spiking RBM (Merolla et al., 2011)	None	Rate-based	Supervised	Contrastive divergence, linear classifier	89.0%
Spiking RBM (O'Connor et al., 2013)	Enhanced training set to 120,000 examples	Rate-based	Supervised	Contrastive divergence	94.1%
Spiking convolutional neural network (Diehl et al., 2015)	None	Rate-based	Supervised	Backpropagation	99.1%
Spiking RBM (Neftci et al., 2013)	Thresholding	Rate-based	Supervised	Contrastive divergence	92.6%
Spiking RBM (Neftci et al., 2013)	Thresholding	Spike-based	Supervised	Contrastive divergence	91.9%
Spiking convolutional neural network (Zhao et al., 2014)	Scaling, orientation detection, thresholding	Spike-based	Supervised	Tempotron rule	91.3%
Two layer network (Brader et al., 2007)	Edge-detection	Spike-based	Supervised	STDP with calcium variable	96.5%
Multi-layer hierarchical network (Beyeler et al., 2013)	Orientation-detection	Spike-based	Supervised	STDP with calcium variable	91.6%
Two layer network (Querlioz et al., 2013)	None	Spike-based	Unsupervised	Rectangular STDP	93.5%
Two layer network (this paper)	None	Spike-based	Unsupervised	Exponential STDP	95.0%

Tab.1 Classification accuracy of SNN on MNIST

4 Extended works

Experiments

- DVS-MNIST dataset

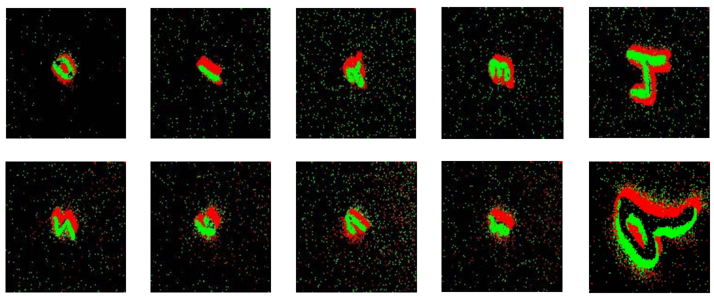


Fig.5 N-MNIST dataset

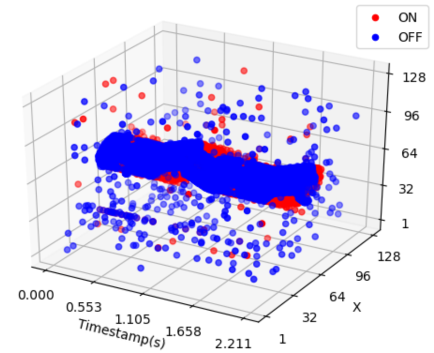


Fig.6 event streams

- Results comparison

Rate-based coding

Tab.2 Event-based classification on N-MNIST dataset

Methods	Author	Now work
STDP [NNLS, 2015][3]	-	0.913
HATS [CVPR, 2018][4]	**0.984**	<u>0.972</u>

[3] Unsupervised learning of digit recognition using spike-timing-dependent plasticity, Peter U. Diehl et. al, *TNNLS* 2014.
[4] HATS: Histograms of averaged time surfaces for robust event-based object classification, Amos Sironi, et. al, *CVPR* 2018.

5 Outlook

- 1 **No Brian2 , No Nest Simulation platform**, but in deep Architecture, such as Pytorch [5]?

- **2 End-to-end SNN** can be applied in complex event-based vision tasks?

- **3** Sparse lattice networks [6] used in spatial-temporal spike stream?

[5] Direct training for spiking neural networks: faster, larger, better, Yujie Wu et. al, ***arXiv*** 2018.
[6] Hnnge Su et.al . SPLATNet: Sparse Lattice Netorks for Point Cloud Processing. Hang su et.al, ***CVPR*** 2018.

6 Summary

Representations	Disadvantages	Advantages
Image	**Lack of temporal information**	**Deep learning**
Time surface	**Complexity & Local feature**	**Temporal information**
Feature	**Multi-steps**	**Complex vision tasks**
End-to-end SNNs	**Neural model + Framework**	**Temporal information**
End-to-end CNNs	**Lack of datasets** ***Waiting …***	**Complex vision tasks**

Tab.3 Representations for event-based camera data

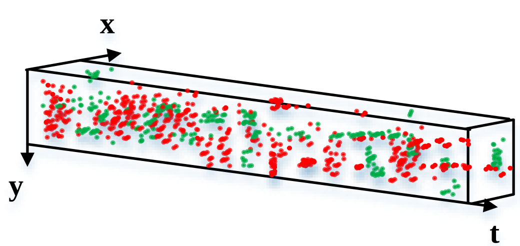
Overview

- Introduction
  - Event-based sensors
  - Related works
- Image representations
  - Steering prediction, CVPR 2018
- Time surface representations
  - HOTS, PAMI 2017
  - HATS, CVPR 2018
- Feature representations
  - Bag of Events, TNNLS 2017
- End-to-end SNN
  - STDP, TNNLS 2014
- **Discussion**
  - Better input representations for CNN
  - Event-based sensors future

Discussion

- **Better input representations for CNN**
  - Point process, such as **PointNet** [1]
  - **Lack of training dataset**

- **The future of event-based cameras**
  - Sparse and asynchronous events
  - Point process



[1] PointNet++: Deep hierarchical feature learning on point sets in a metric space. Charles R. Qi., NIPS, 2017.

Q&A?

Thanks !