Applications, Software and Hardware for Event-Based Vision

Neuromorphic Vision Systems

Kynan Eng

Prelude

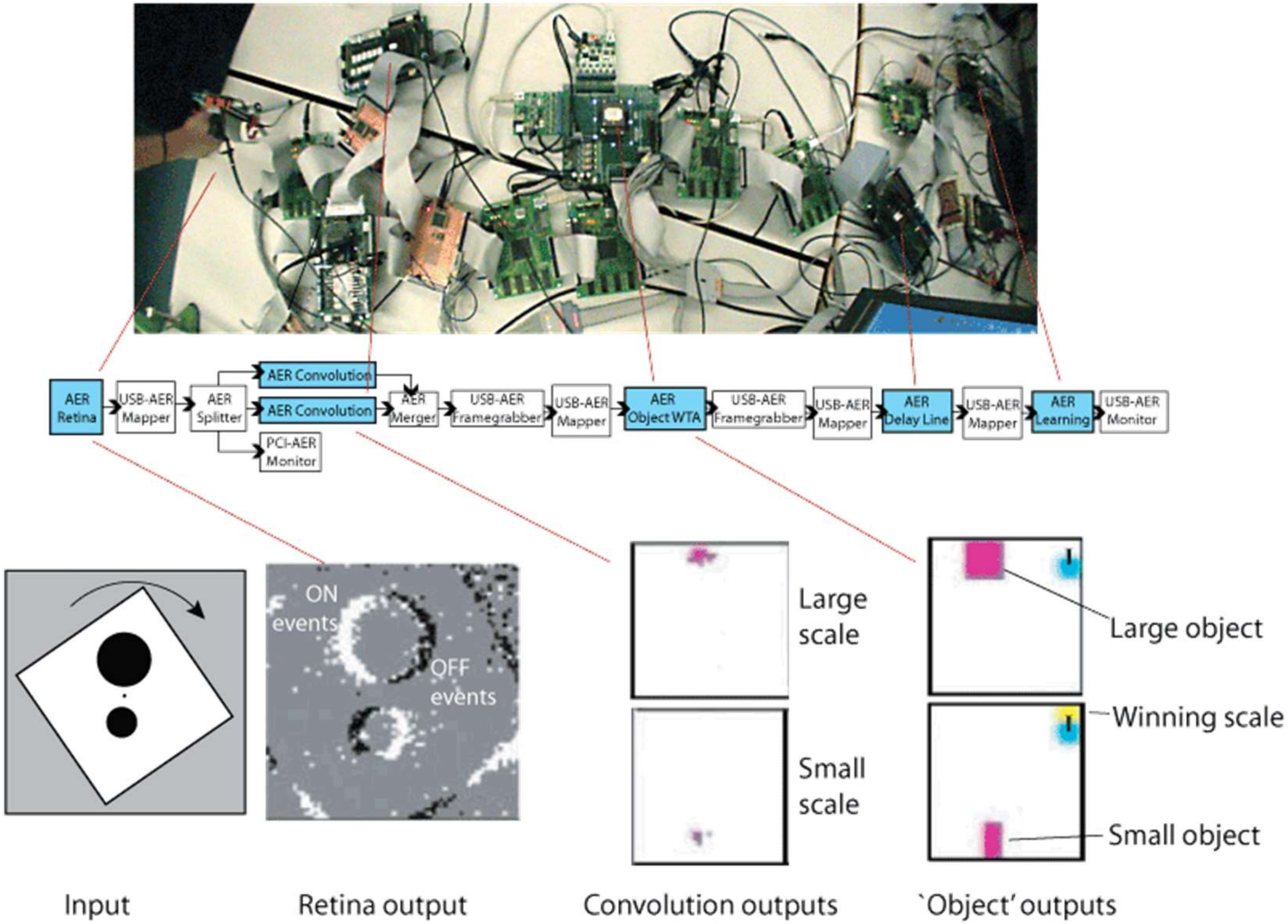
Will Event-Based Vision Change the World (of vision)?

If yes, then we need:

1. A community of developers
   (including workshops like this one)

2. Methods for quantifying benefits

Event-Based Vision, 2005

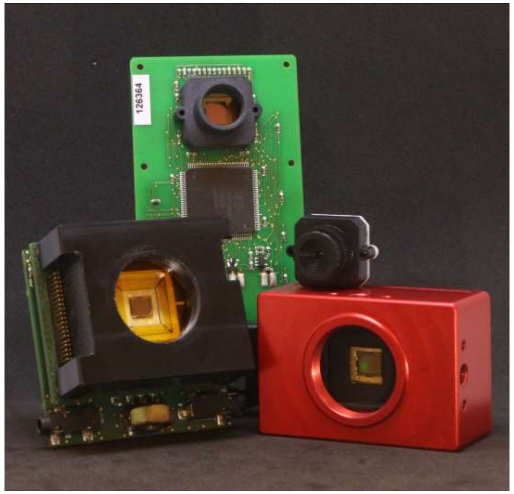


Delbruck, Liu, et al. 2005. CAVIAR EU project

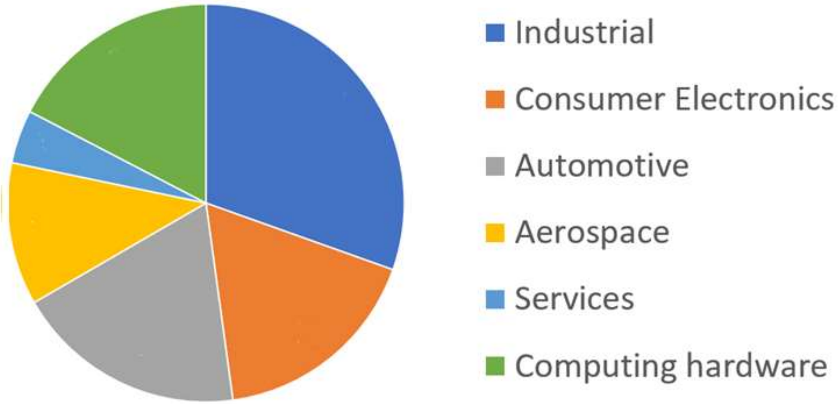
Since 2008

+Bias generator
+USB cable

2008
DVS128



Resolutions
Form factors
Greyscale / Color



2019
>250 organizations

Today’s Talk

Software
- Making it easier

Applications
- Does it make sense to do this with DVS?

Hardware
- How to maximize DVS benefits

Software

Open DVS Development

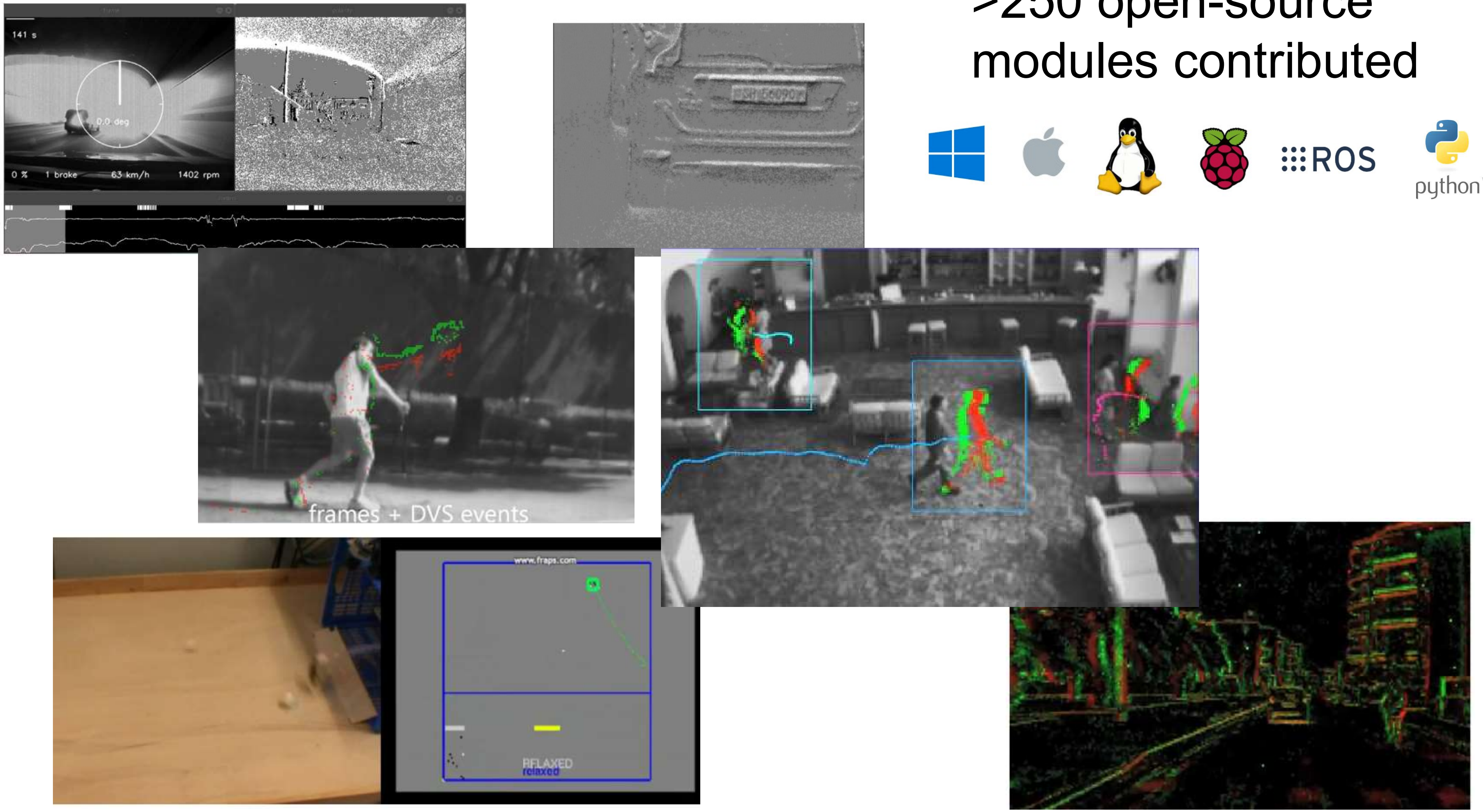
Problems with DVS Development

Problems
- It’s too hard
- I can’t use OpenCV

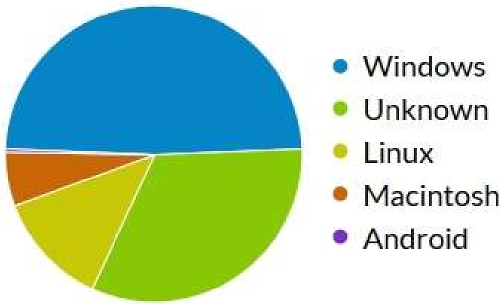
Solutions
- Development environment
- Pre-built modules
- Interface to everything

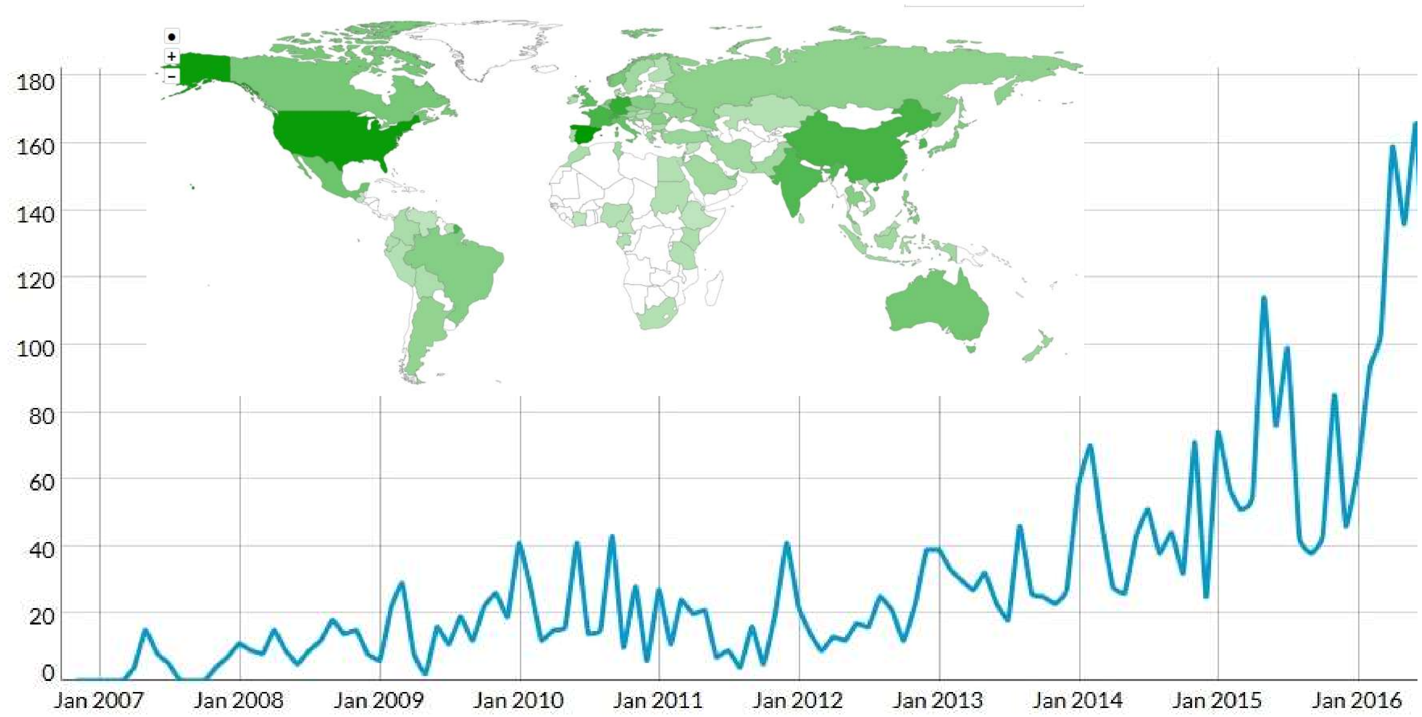
Step 1: Prototyping with jAER + libcaer

>250 open-source modules contributed



Step 1: Prototyping with jAER



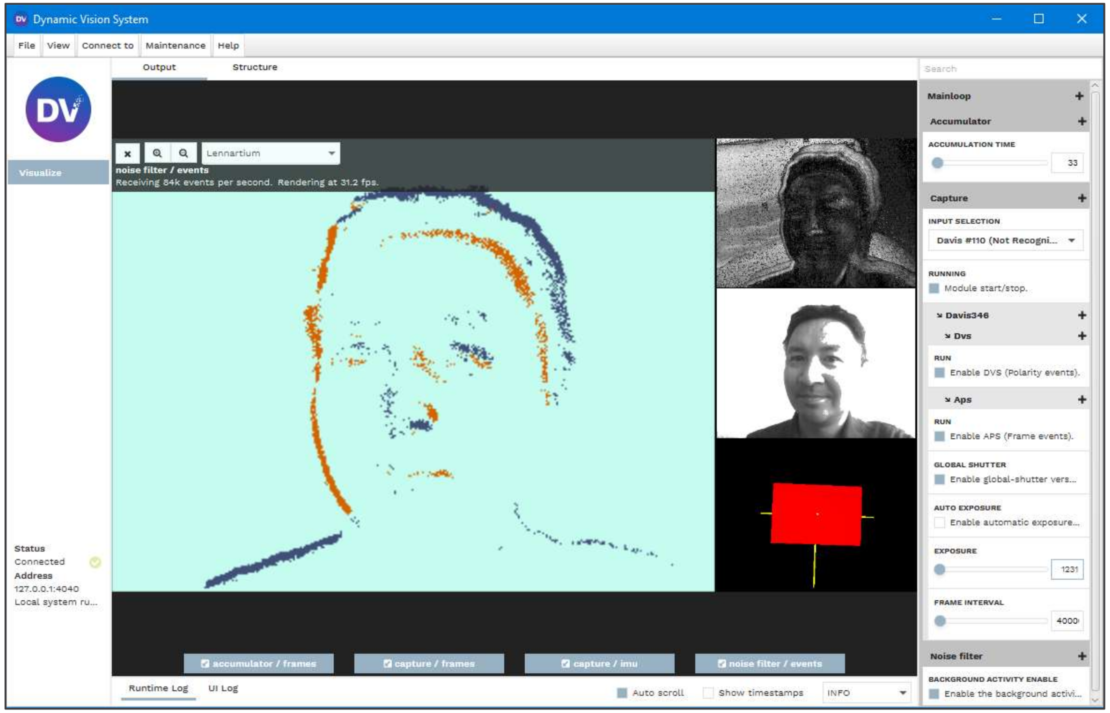


Up to 2016: >4000 downloads (Sourceforge, before switch to Git)

Step 2: Robust Solution

- High-performance C/C++
- Open API
- Decoupled GUI and engine
- Cross-platform
- Works with different DVS cameras
- Interfaces to CNN hardware/software

Live Development



Live Viewer

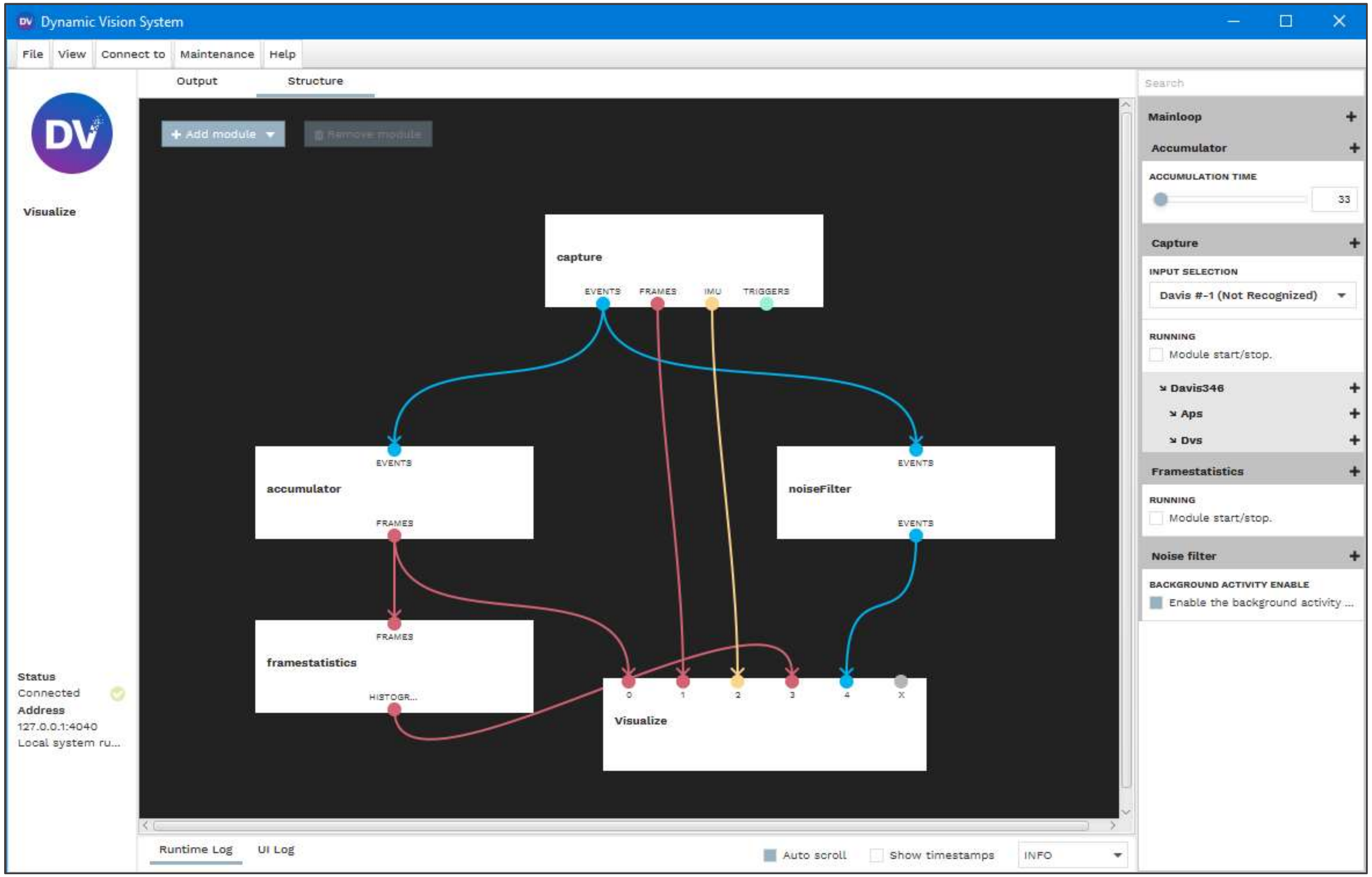
Network socket

Deployment

DV Instance
(local or rmote)

DVS

Data Flow Definition



+ pre-built high-level modules

Applications

How to select DVS applications

Questions & Assumptions

Where does DVS work best?

How can I decide?

Possible answers

- Low Latency
- HDR
- Energy efficiency?

Questions & Assumptions

Where does DVS work best?

How can I decide?

Single events vs Event frames vs Diff image

Can we quantify how much DVS can beat frames?

Problem Formulation

Input Variables

F = Target frame rate

P_s =Total system power available

Constants

E_d = Energy per frame diff image

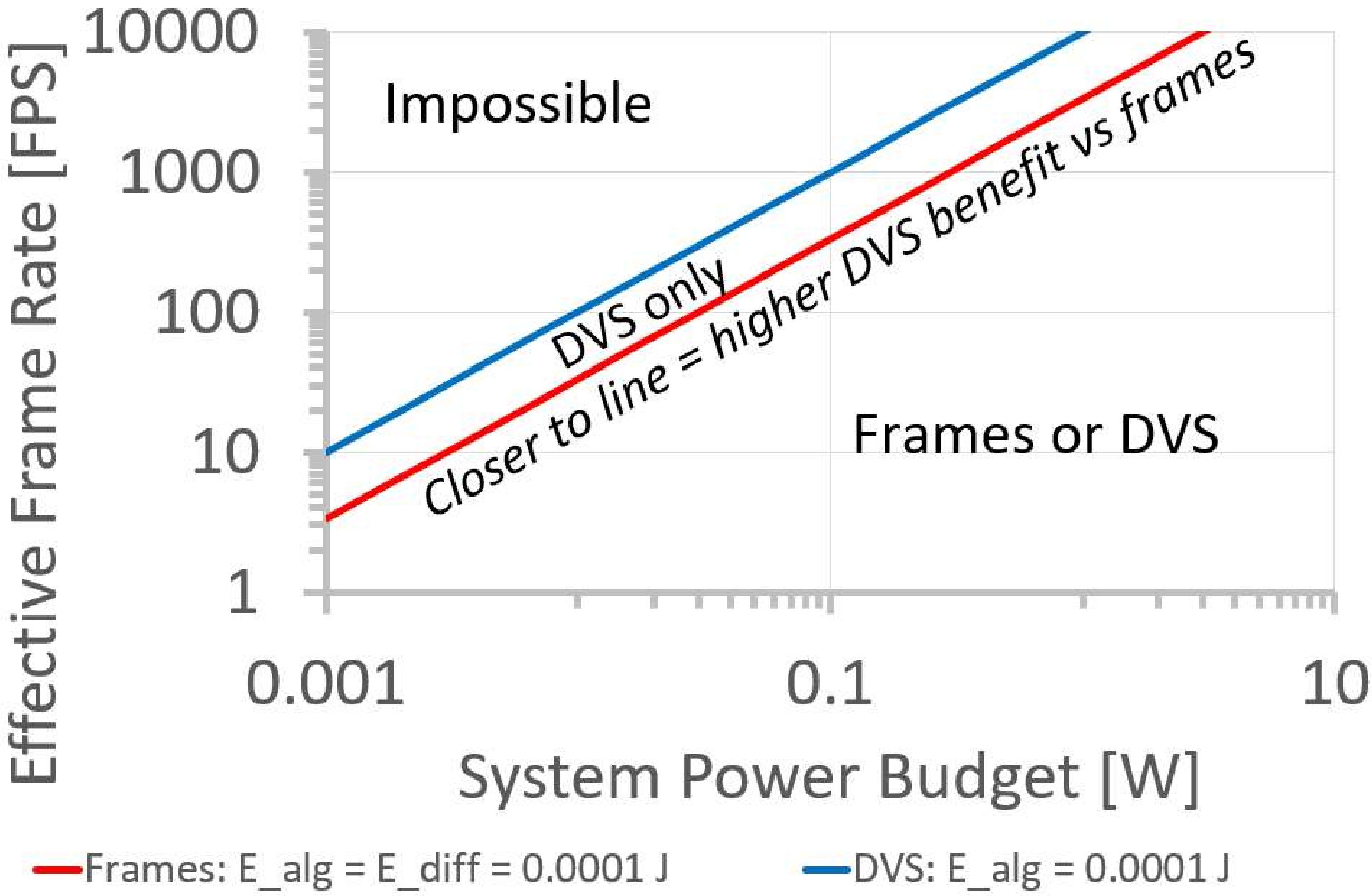
E_a = Energy per processing algorithm update step

(note: assume algorithms need the same energy)

Output

P_a = Power margin available per frame for algorithms

Result: Where DVS Works Best



Hardware

Maximizing DVS Benefits

Questions, Principles

Question

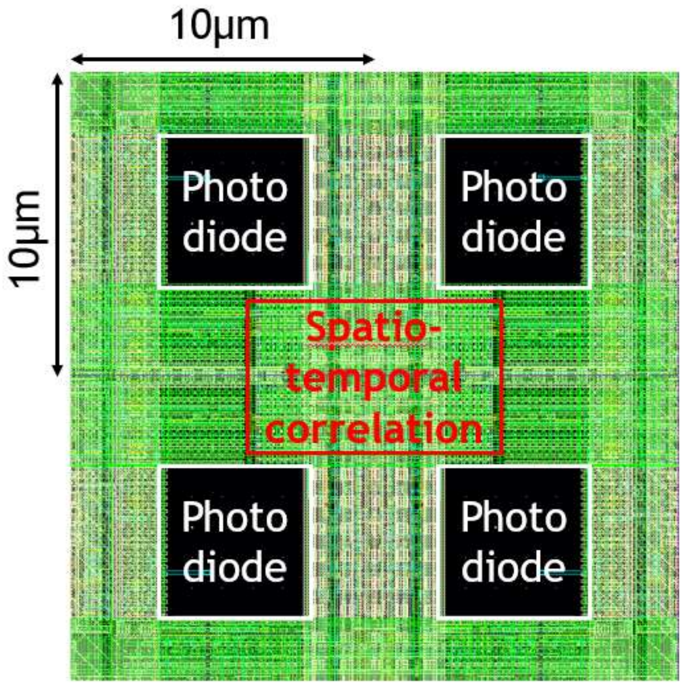
What hardware works best with DVS?

Principles

Minimal power budget ASICs

Activity-dependent computation

In-Array Noise Filtering



2×2 pixels

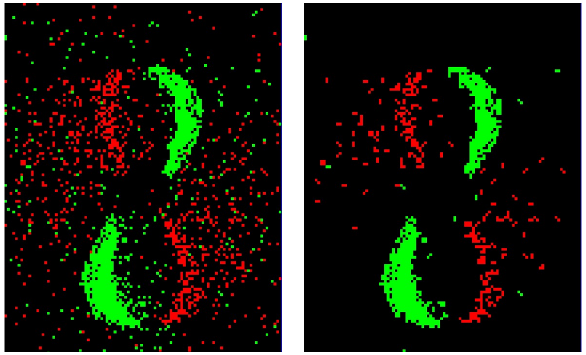
10µm pitch, 20% fill factor,
65nm 1P9M (non-CIS)

Technology		**65nm**
Pixel size (µm)		**10**
Max Readout Speed (Meps)		**180**
Readout Efficiency (event/clock)		**Best: 4, worst: 0.25**
Power Supply (V)		**1.2**
Power (mW)	High activity	**180Meps: 4.9**
	Low activity	**100keps: 0.25**
Normalized	Dynamic energy (pJ/event)	**26**
	Static power (nW/pixel)	**18**

Li C, IISW 2019

In-Array Noise Filtering

Raw data | Noise



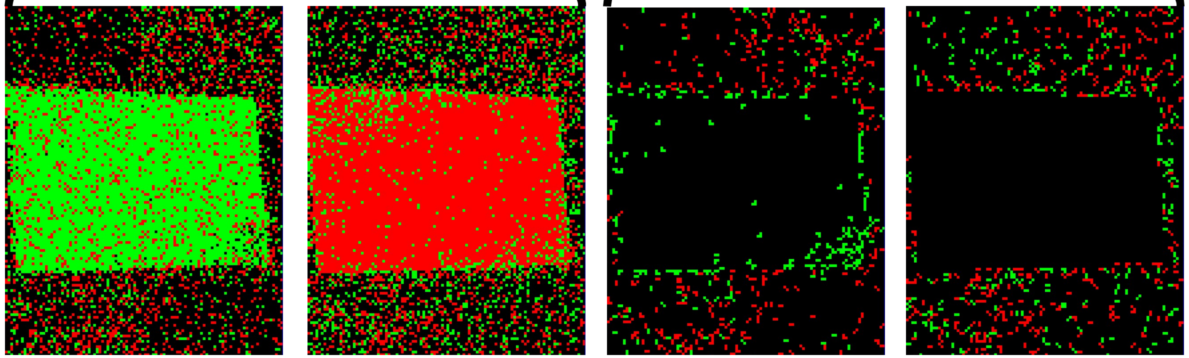
1804 events | 1070 events

40% less data

1800 rpm USB fan
1 kefps
300 lux

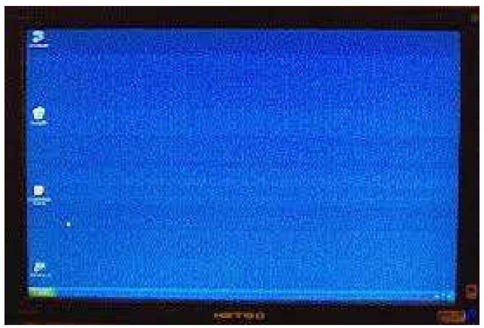
Li C, IISW 2019

Raw data | Noise and spatial redundancy suppressed



8199 events | 8764 events | 1034 events | 1067 events

87% less data



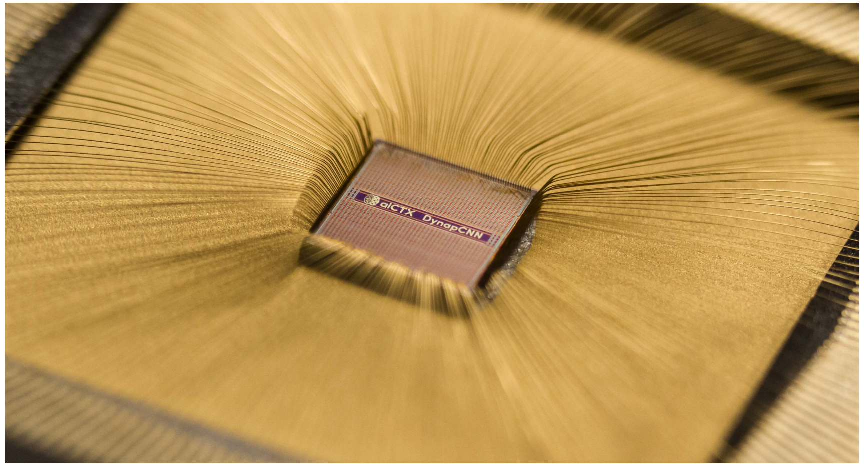
Flickering monitor
200 efps
300 lux

Dynap-SE by aiCTX

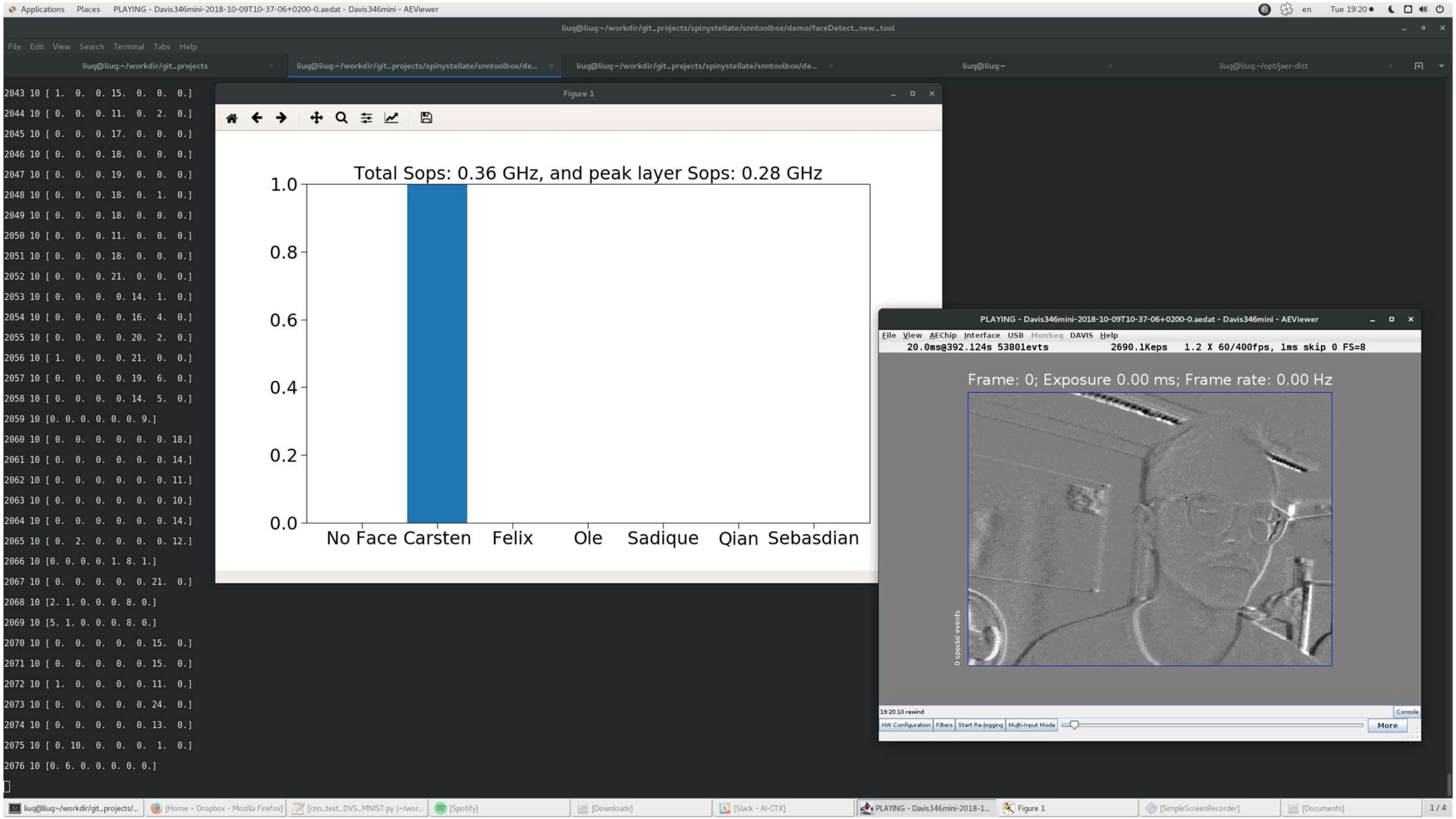
Technology	28 nm FDSOI
Supply Voltage	0.73 V
Neuron Type	**Analog**
Neurons per core	256
Core Area	0.36 mm^2
Fan In/Out	2k/8k
Synaptic Weight	(4+1) bits
Energy per SOP	<2 pJ[*2]
Energy per spike	<1.68 nJ[*3]

DynapCNN by aiCTX

Technology	22 nm
Neuron Type	**Digital**
Neurons	1 M
Core Area	12 mm^2
Parameters	4 M
Input sync	Fully event-driven
Power	< 1 mW (typical)



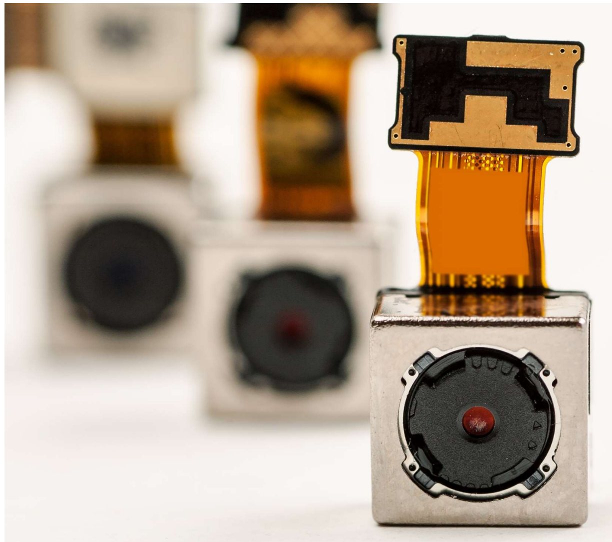
DynapCNN by aiCTX



DVS + DynapCNN

SPECK

Micropower
intelligent scene analysis
for mobile and IoT



- Announced CES 2019
- Single-chip DVS + CNN processor
- Ultra-low-power classification
- <1 mW power (typical)
- Low latency
- Sampling Q3 2019

Talk to Us

www.iniVation.com

Live demos @ CVPR: Booth 1554