Computational Neuroscience for Technology: Event-based Vision Sensors and Information Processing

Jörg Conradt

Neuroscientific System Theory
Center of Competence on Neuro-Engineering

Technische Universität München

`http://www.nst.ei.tum.de`

NEUROSCIENTIFIC SYSTEM THEORY
Technische Universität München

28.11.2014 Qualcomm Vienna

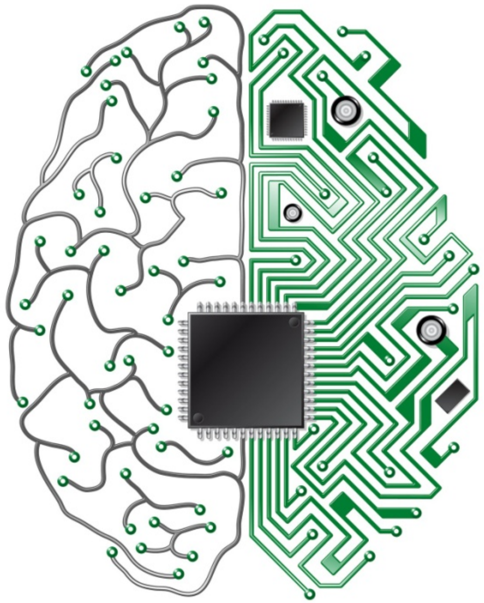
Neuromorphic Engineering

Neuro - morphic

"to do with neurons", i.e. neurally inspired

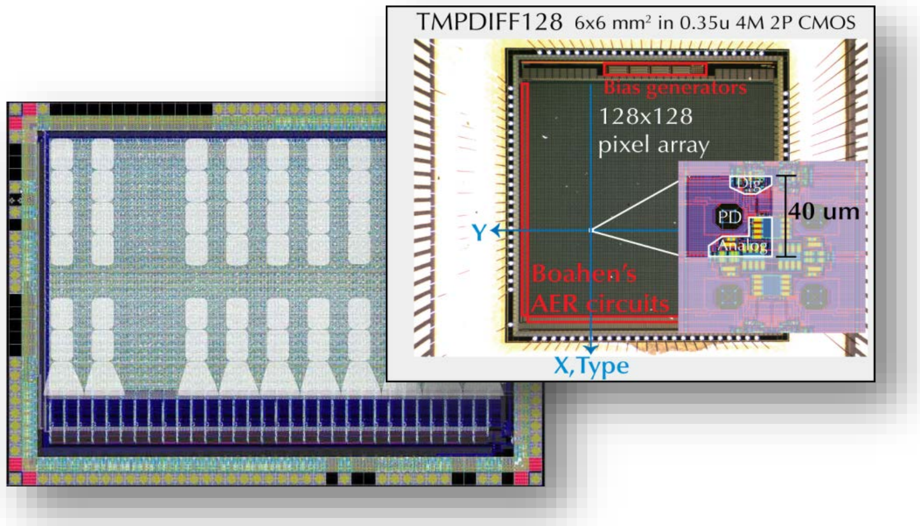
"structure or form"

Discovering key principles by which brains work and implementing them in technical systems that *intelligently* interact with their environment.



Examples:

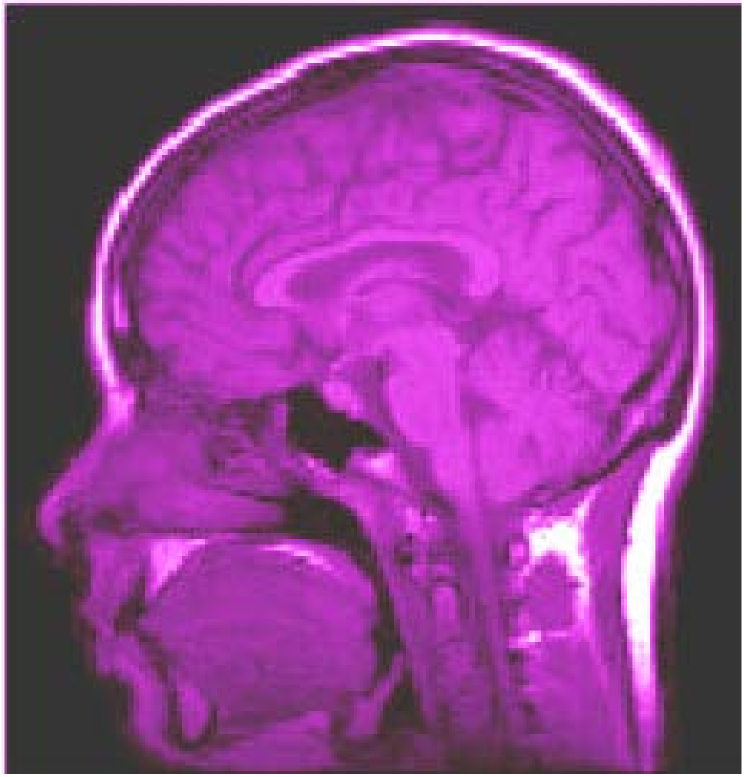
- analog VLSI Circuits to implement neural processing
- Sensors, such as Silicon Retinae or Cochleae
- Distributed Adaptive Control, e.g. CPG based locomotion
- Massively Parallel Self-Growing Computing Architectures
- Neuro-Electronic Implants, Rehabilitation



Computation in Brains

Getting to know your Brain

- 1.3 Kg, about 2% of body weight
- 10^{11} neurons
- Neuron growth:
  250.000 / min (early pregnancy)
  … but also … loss 1 neuron/second



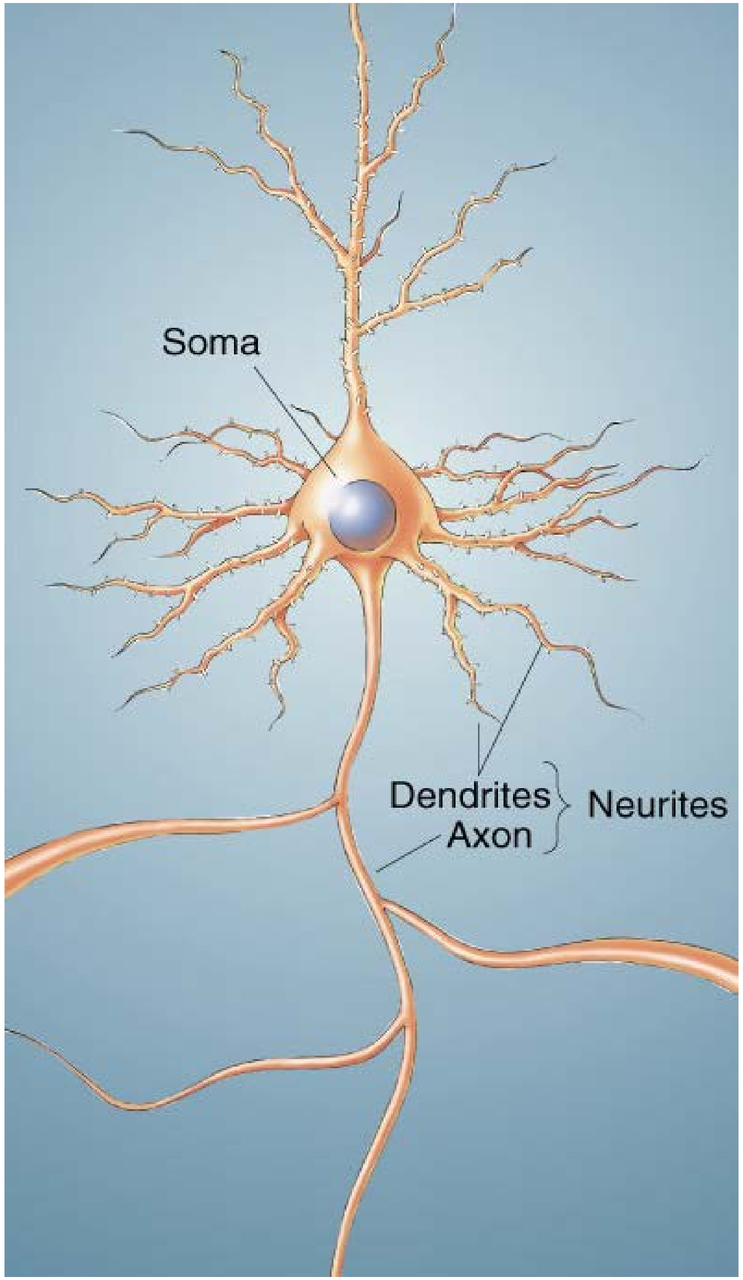
Computation in Brains

Getting to know your Brain

- 1.3 Kg, about 2% of body weight
- 10^{11} neurons
- Neuron growth:
  250.000 / min (early pregnancy)
  … but also … loss 1 neuron/second

"Operating Mode" of Neurons

- Analog leaky integration in soma
- Digital pulses (spikes) along neurites
- 10^{14} stochastic synapses
- Typical operating "frequency":
  ≤100 Hz, typically ~10Hz, asynchronous



Computation in Brains ... and Machines

Getting to know your Brain

- 1.3 Kg, about 2% of body weight
- 10^{11} neurons
- Neuron growth:
  250.000 / min (early pregnancy)
  … but also … loss 1 neuron/second

Getting to know your Computer's CPU:

- 50g, irrelevant for most applications
- 2 • 10^{9} transistors (Intel Itanium)
- ideally no modification over lifetime

"Operating Mode" of Neurons

- Analog leaky integration in soma
- Digital pulses (spikes) along neurites
- 10^{14} stochastic synapses
- Typical operating "frequency":
  ≤100 Hz, typically ~10Hz, asynchronous

"Operating Mode" of CPUs

- No analog components
- Digital signal propagation
- Reliable signal propagation
- Typical operating frequency:
  Several GHz, synchronous

Computation in Brains ... and Machines

Getting to know your Brain

- 1.3 Kg, about 2% of body weight
- 10^{11} neurons
- Neuron growth:
  250.000 / min (early pregnancy)
  … but also … loss 1 neuron/second

“Operating Mode” of Neurons

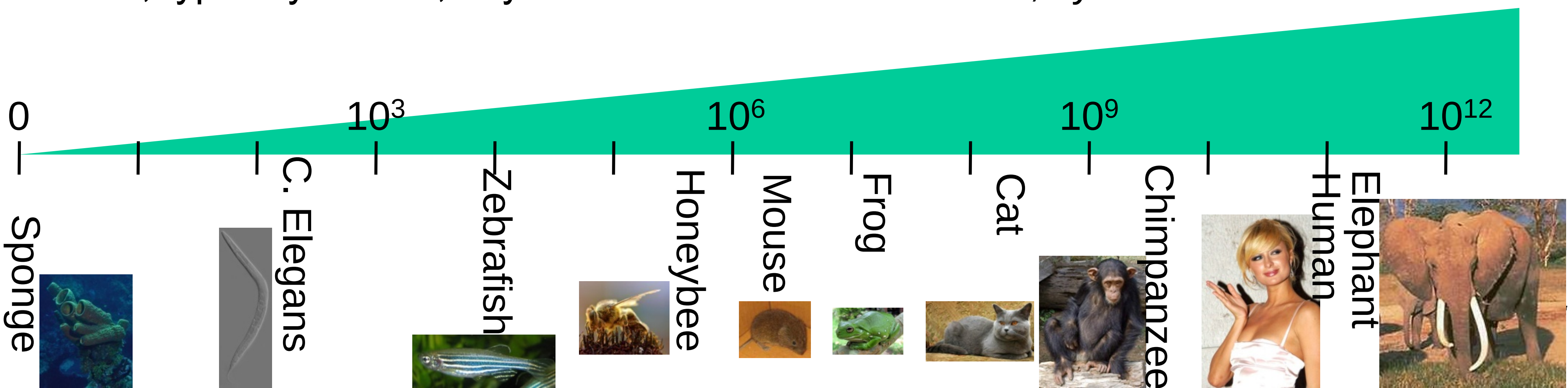
- Analog leaky integration in soma
- Digital pulses (spikes) along neurites
- 10^{14} stochastic synapses
- Typical operating “frequency”:
  ≤100 Hz, typically ~10Hz, asynchronous

Getting to know your Computer’s CPU:

- 50g, irrelevant for most applications
- 2 • 10^9 transistors (Intel Itanium)
- ideally no modification over lifetime

“Operating Mode” of CPUs

- No analog components
- Digital signal propagation
- Reliable signal propagation
- Typical operating frequency:
  Several GHz, synchronous



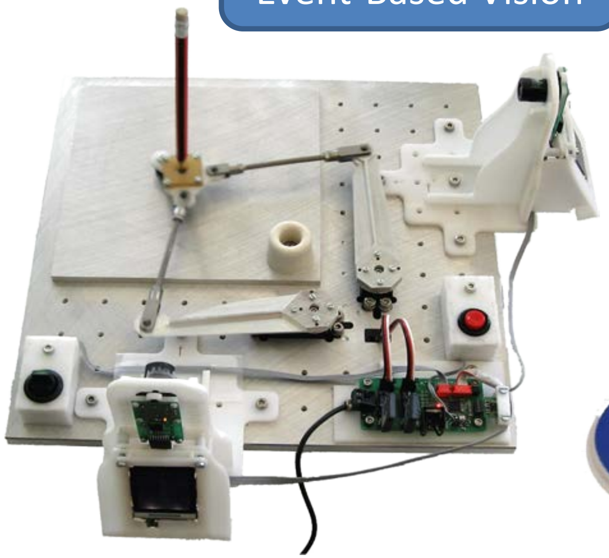
Research Area «Neuroscientific System Theory»

Neuronal-Style Information Processing in Closed-Control-Loop Systems

- Distributed Local Information Processing
- Growing and Adaptive Networks of Computational Units
- Neuromorphic Sensor Fusion and Distributed Actuator Networks
- Event-Based Perception, Cognition, and Action

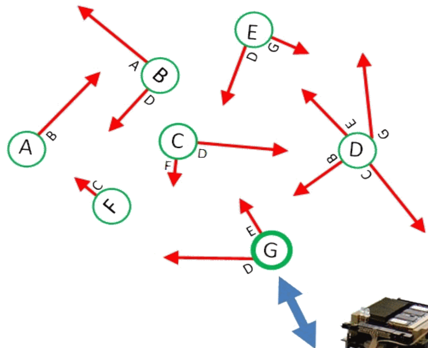
On-Board Vision-Based Control of Miniature UAVs

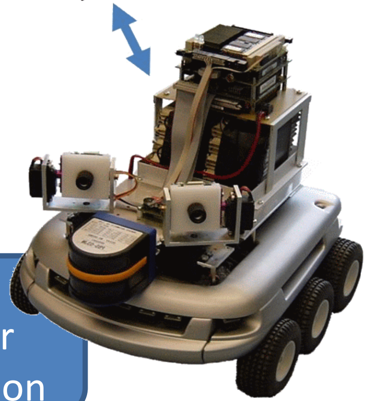
High-Speed Event-Based Vision



Event-Based Navigation

Distributed Sensing and Actuation

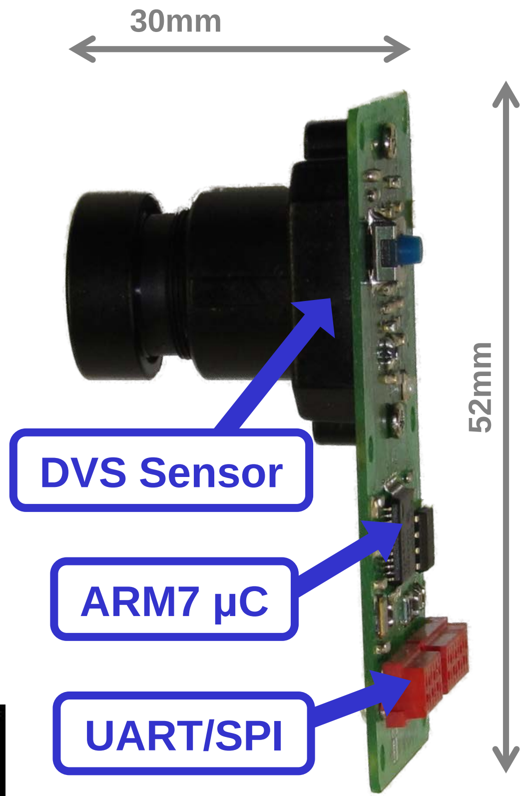


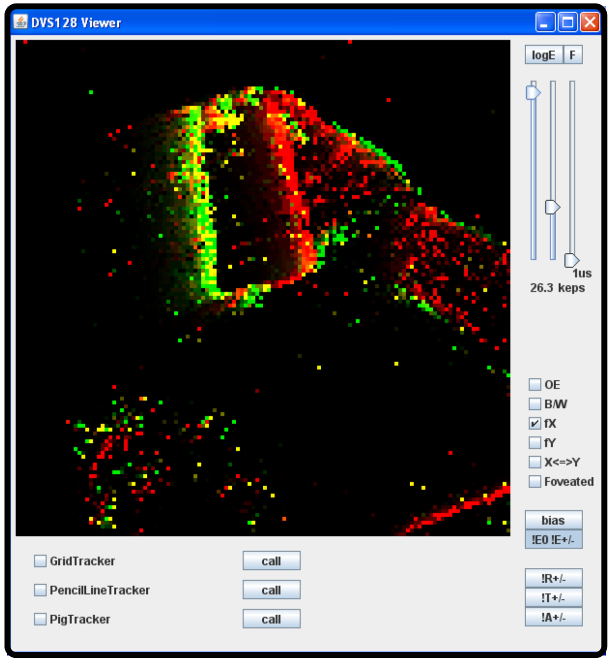


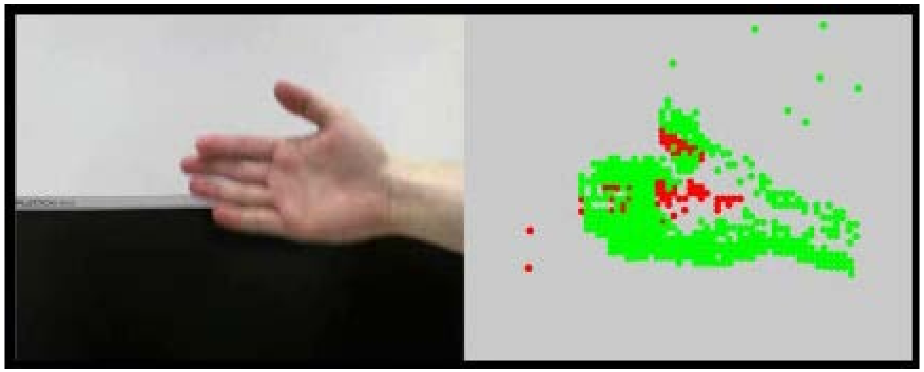
Distributed Local Cognitive Maps for Spatial Representation

Retinal Inspired Dynamic Vision Sensor

- 128 x 128 pixels, each signals
  temporal changes of illumination (“events”)
- Asynchronous updates (no image frames)
- 15 µs latency, up to 1Mevents/sec

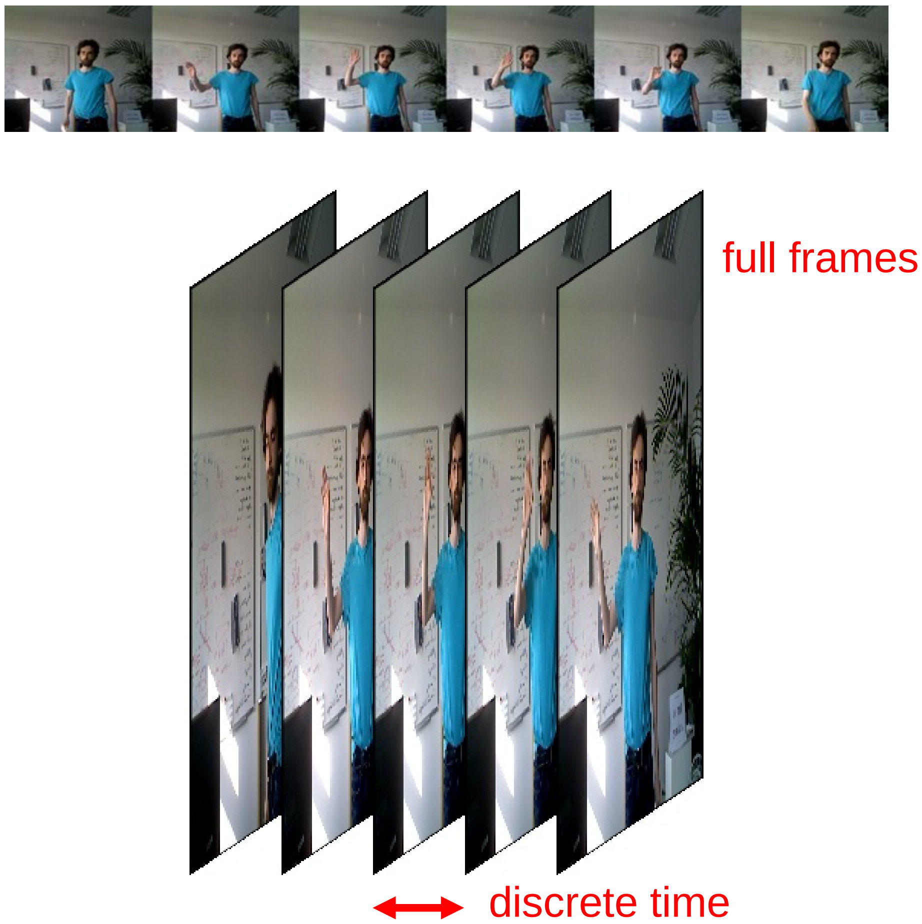




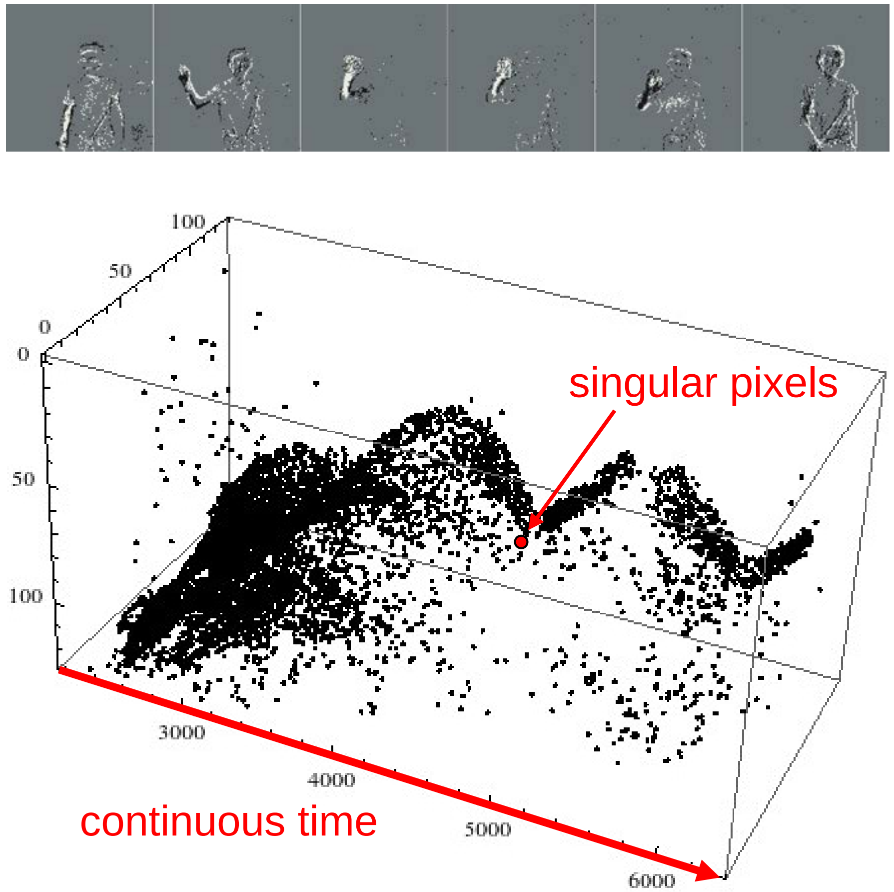


Event - Based Vision

Frame - based



Event - based



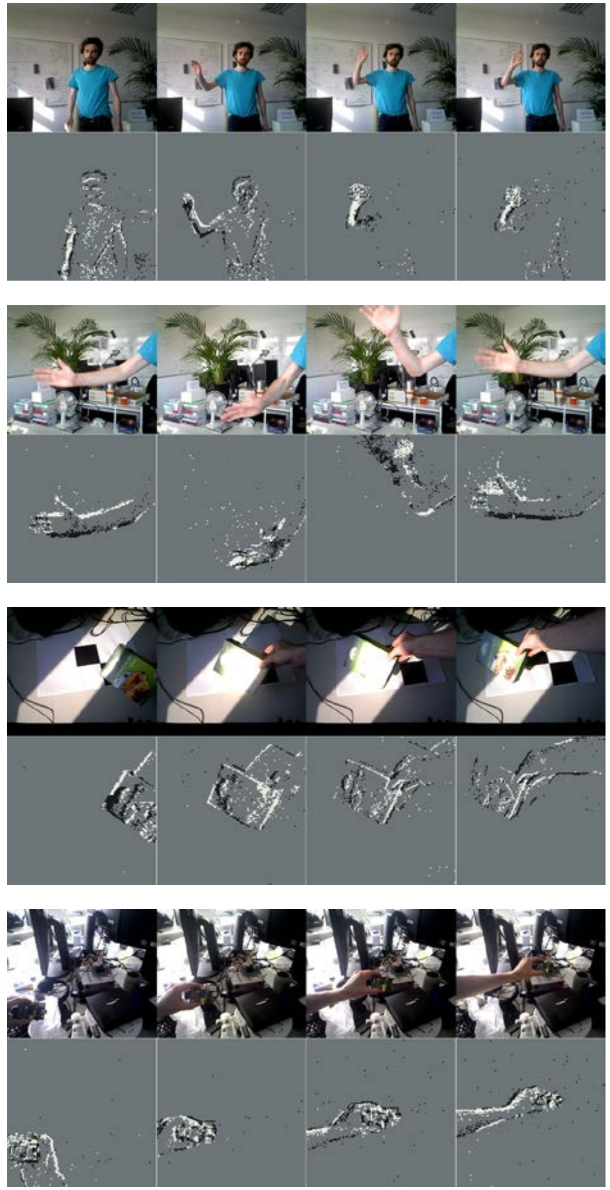
Event - Based Vision

Advantages

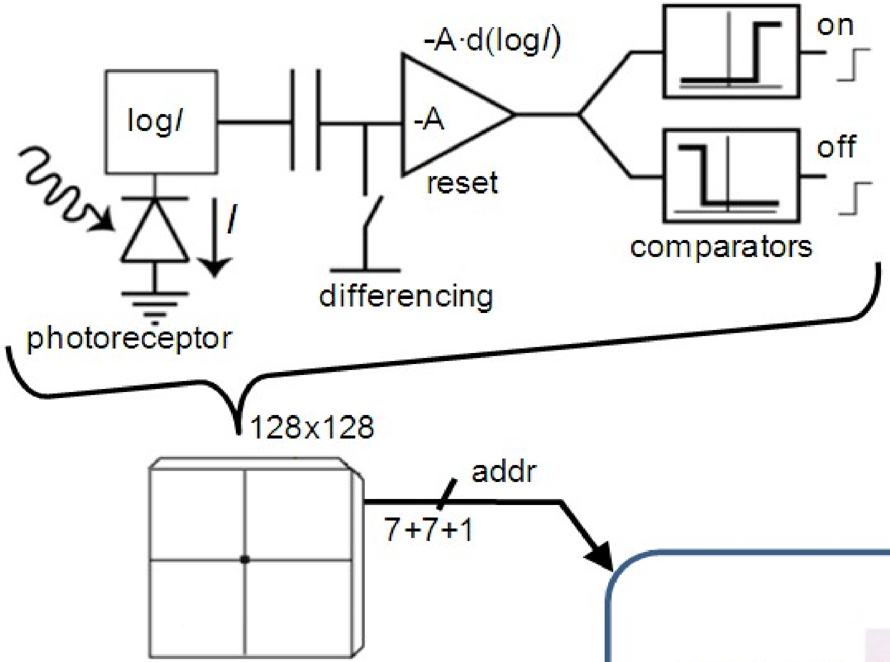
- Sparse data stream: ~0.1 MB/s bandwidth
- ~10us response time
- Local contrast adjustment per pixel
- Automatic preprocessing

Challenges

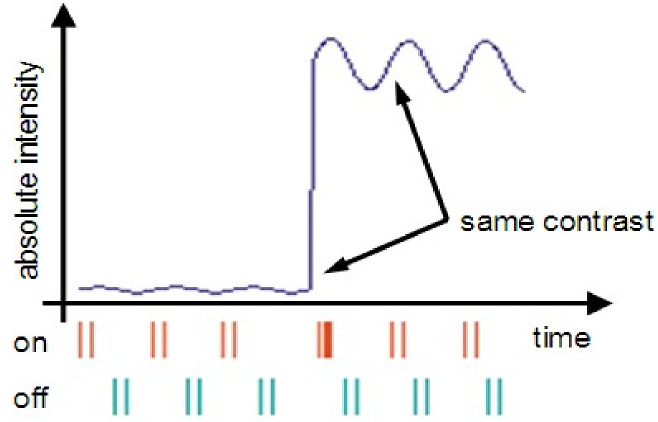
- Static scenes
- New vision algorithms
  for tracking, localization, ...



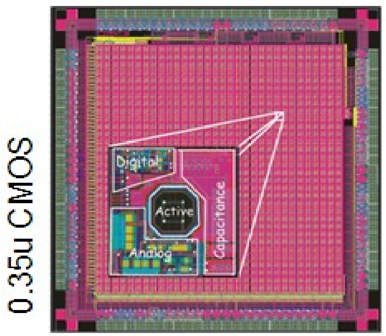
An Asynchronous Temporal Difference Vision Sensor



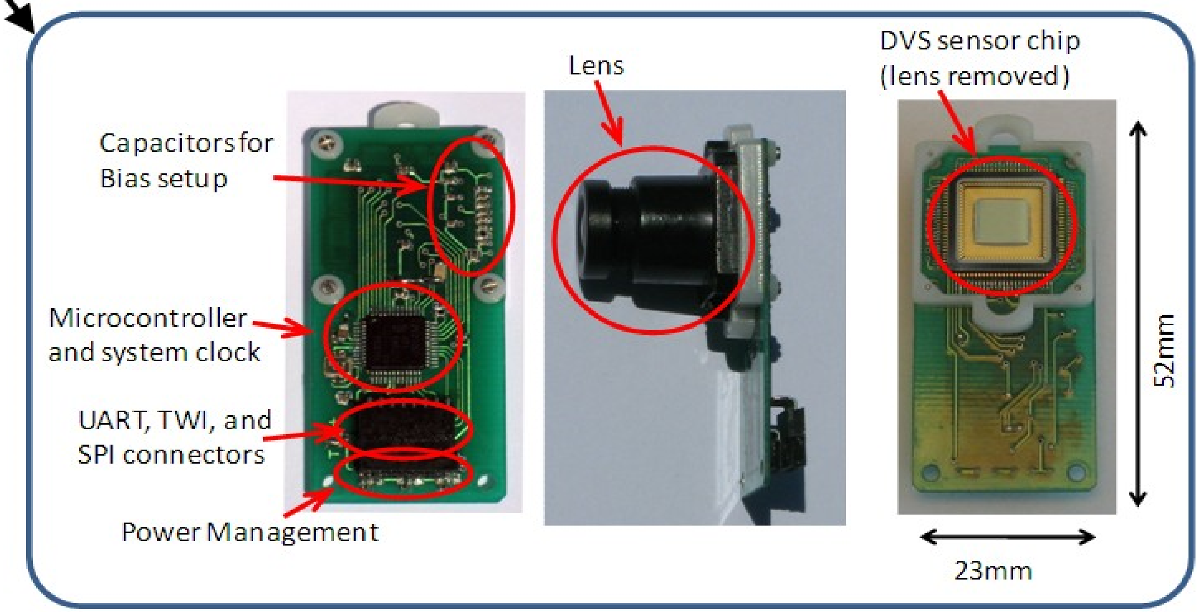
DVS pixel and camera architecture (simplified pixel schematics)



A relative change in illumination causes an event

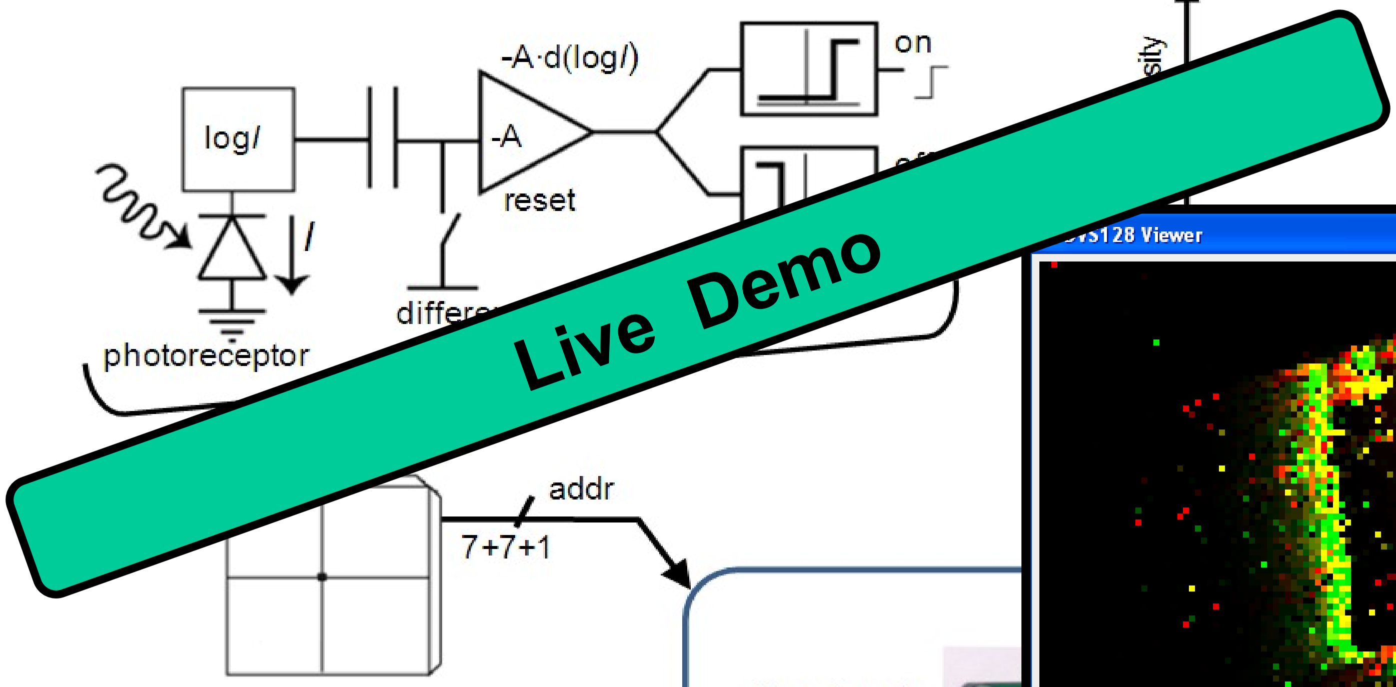


Photograph of retina chip

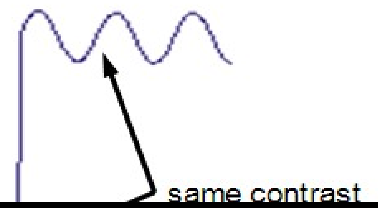


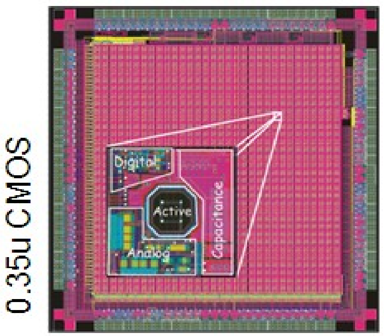
Embedded Microcontroller and DVS on small PCB

An Asynchronous Temporal Difference Vision Sensor

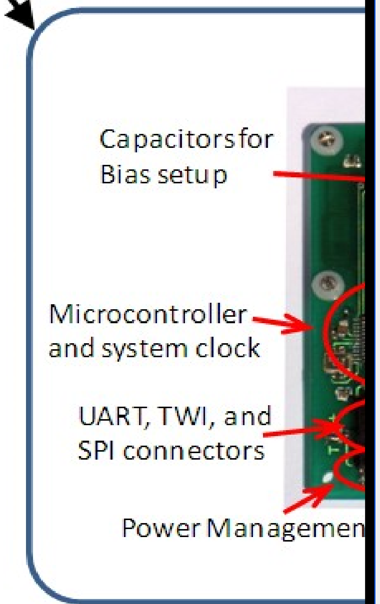


DVS pixel and camera architecture
(simplified pixel schematics)

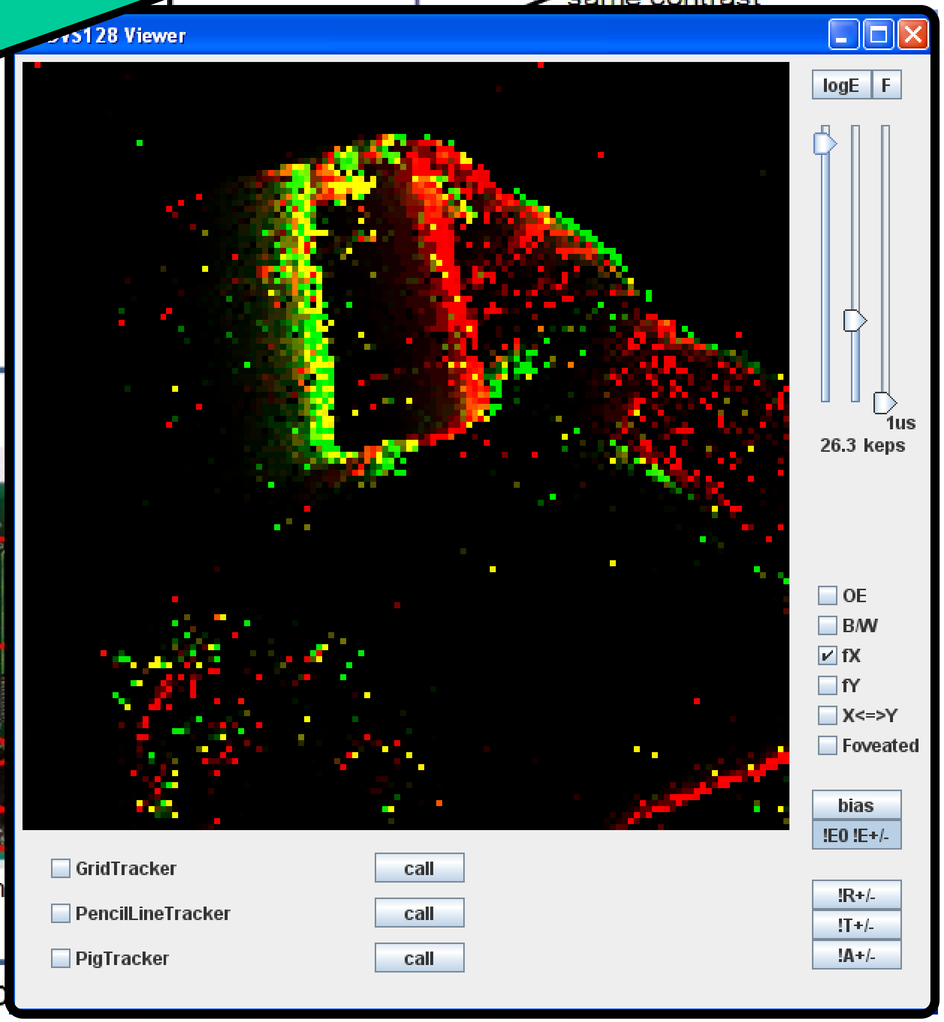




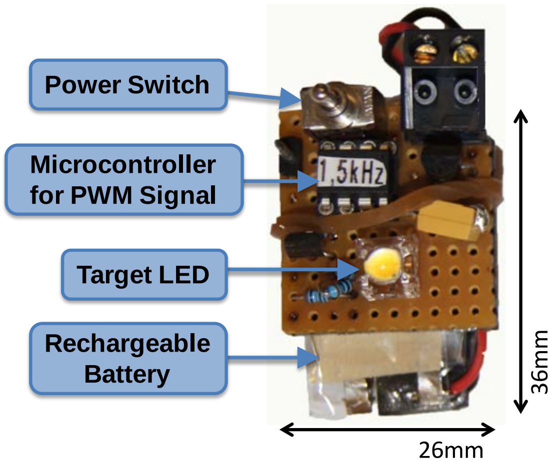
Photograph of retina chip

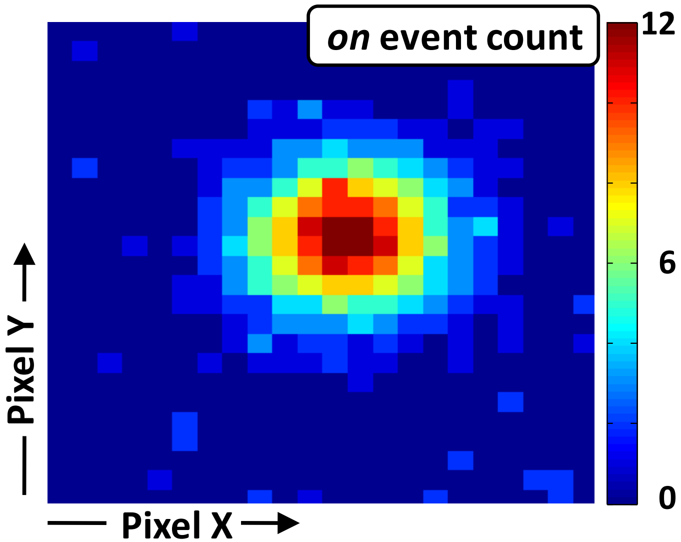


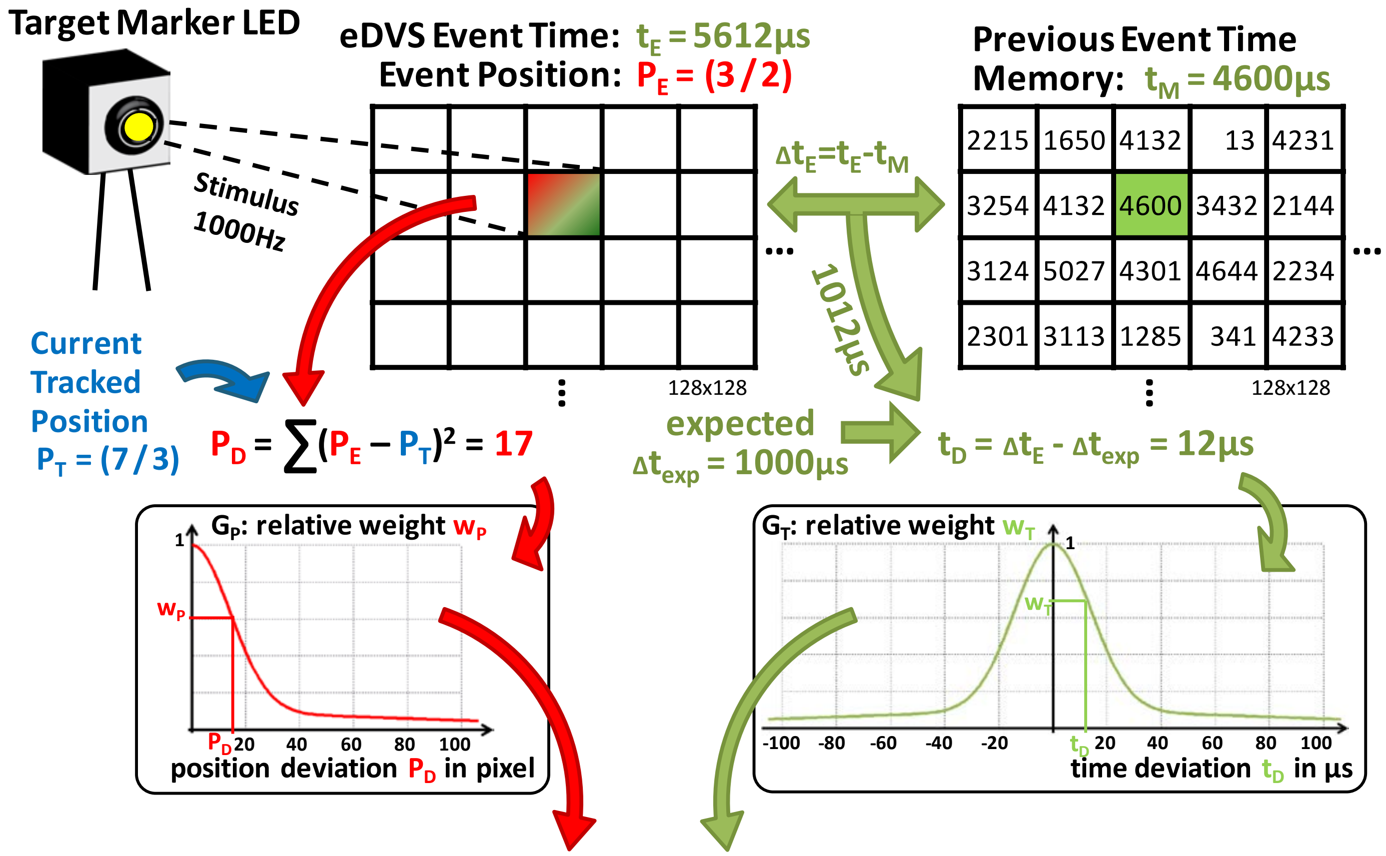
Embedded Microc



Application Example: Tracking an Active Object



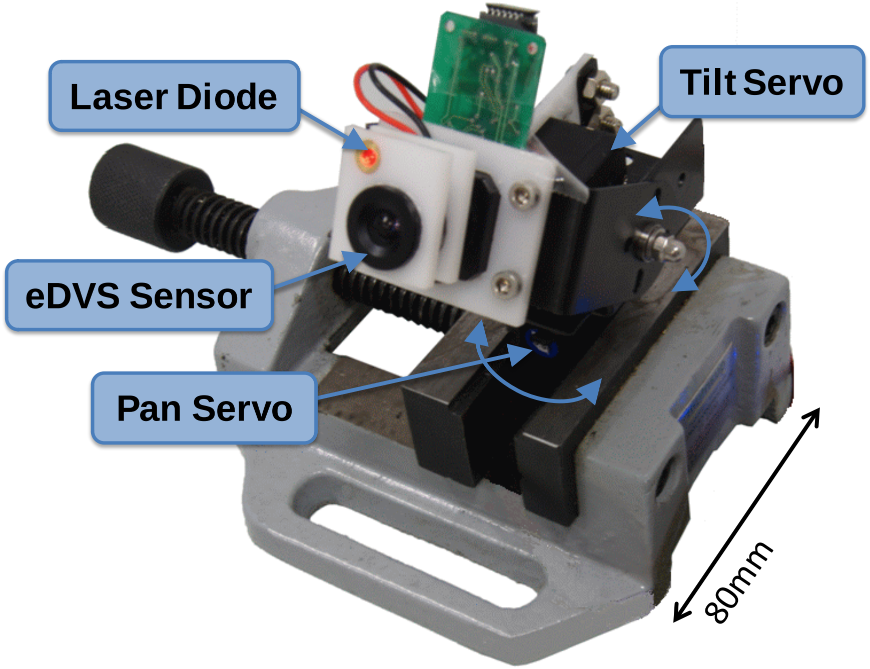


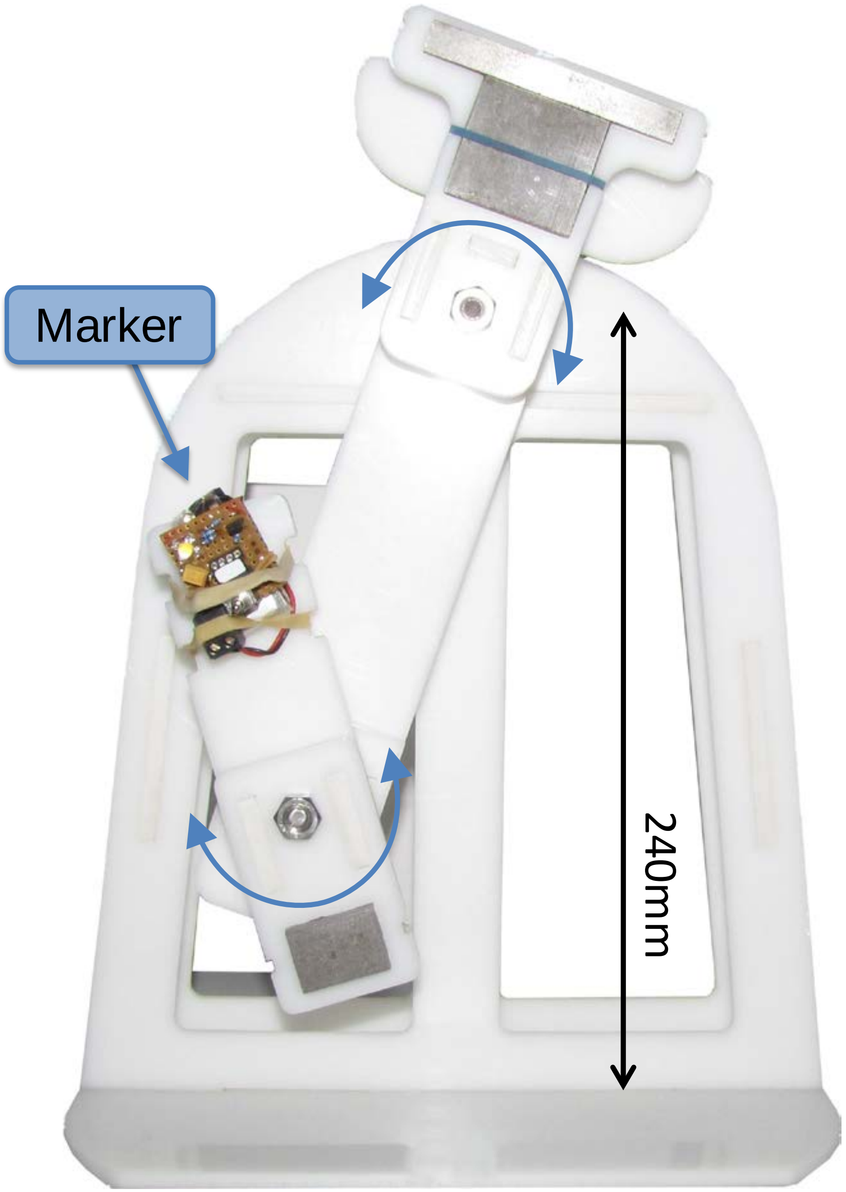


Position Update: $P_{T+1} = (1 - \eta(w_T + w_P)) \cdot P_T + \eta(w_T + w_P) \cdot P_E$

Müller et al, ROBIO 2011

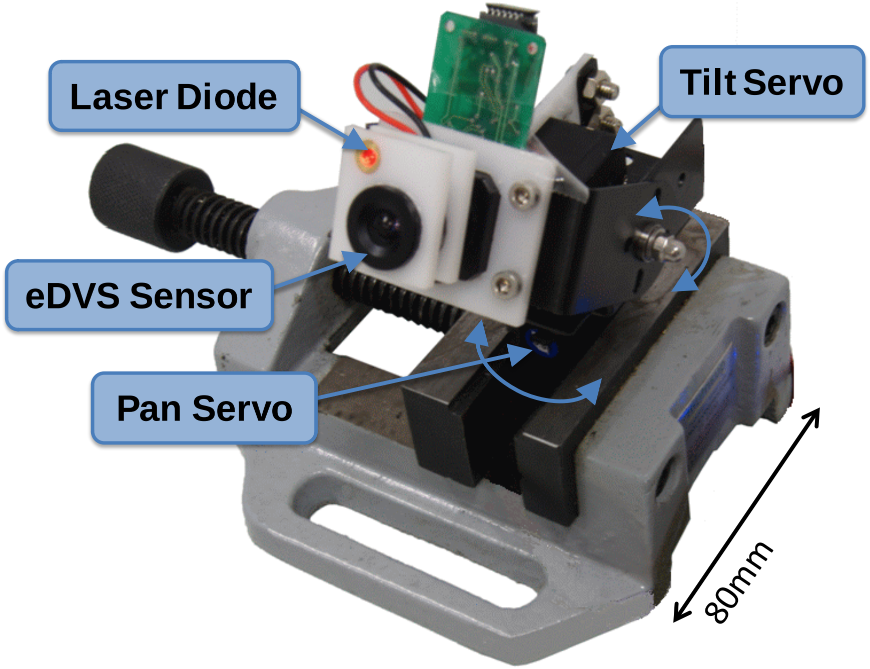
Application Example: Tracking an Active Object

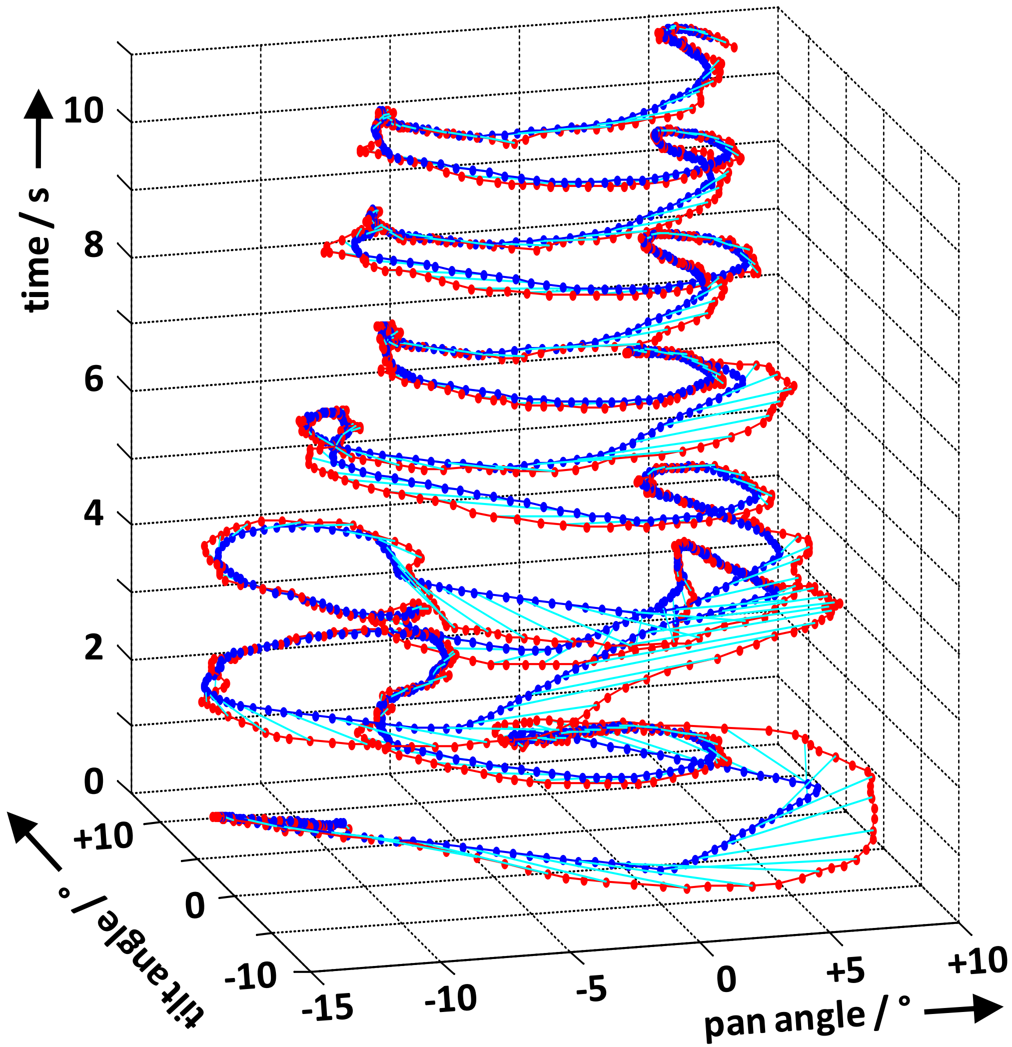




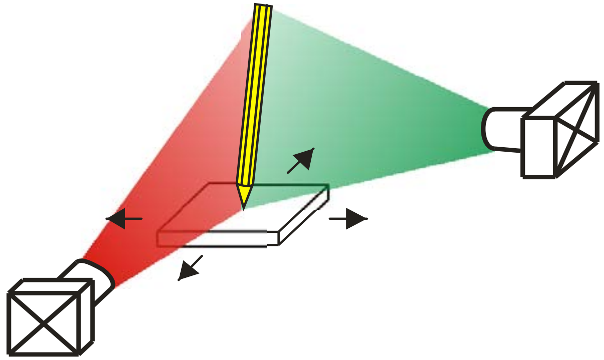
Application Example: Tracking an Active Object

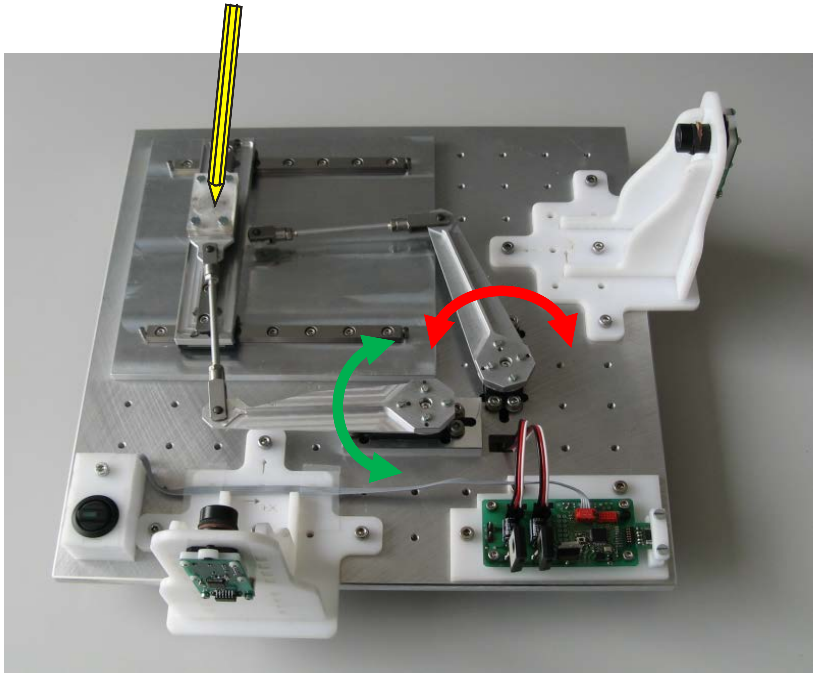
Georg Müller





(Fun) Application Example: High Speed Balancing of Short Poles

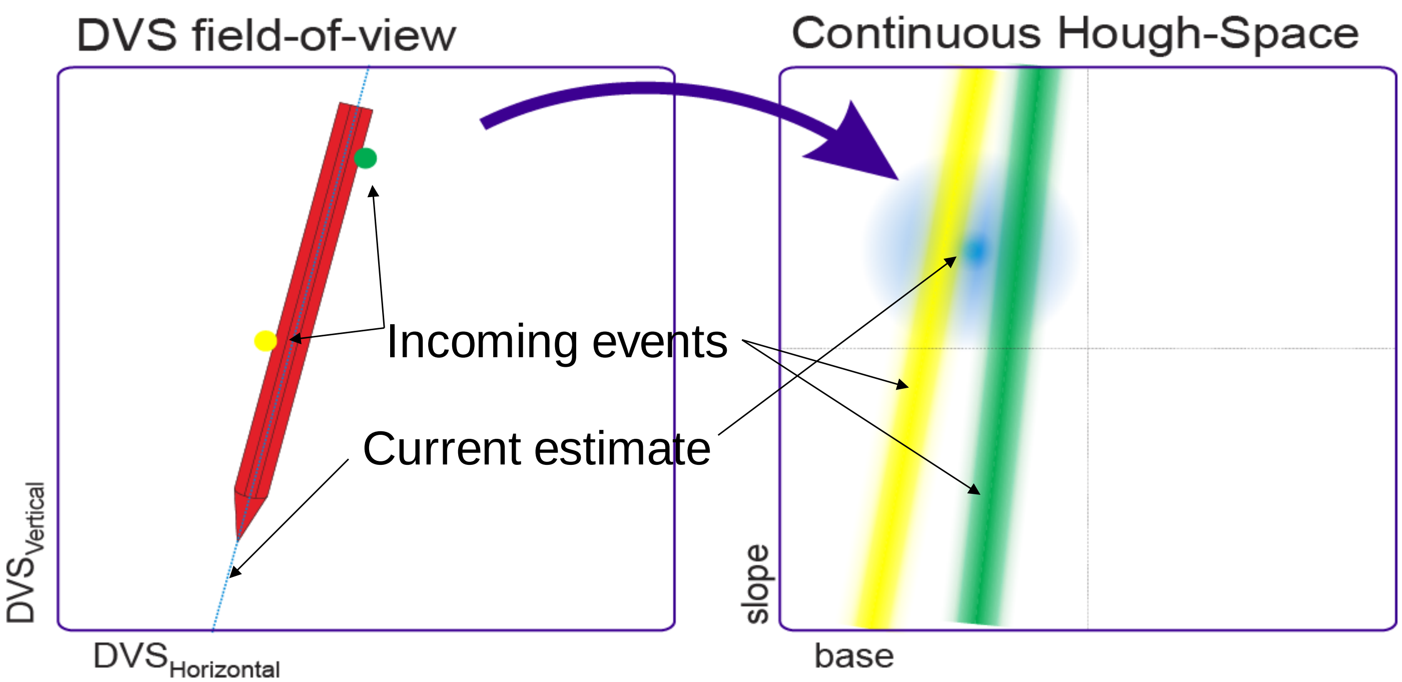


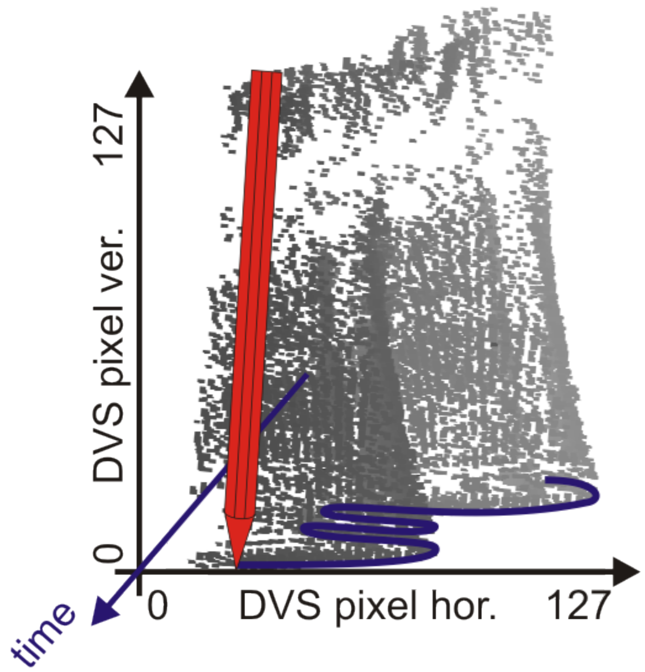


Conradt et al, ISCAS 2009

(Fun) Application Example: High Speed Balancing of Short Poles

Converting Events into Meaning



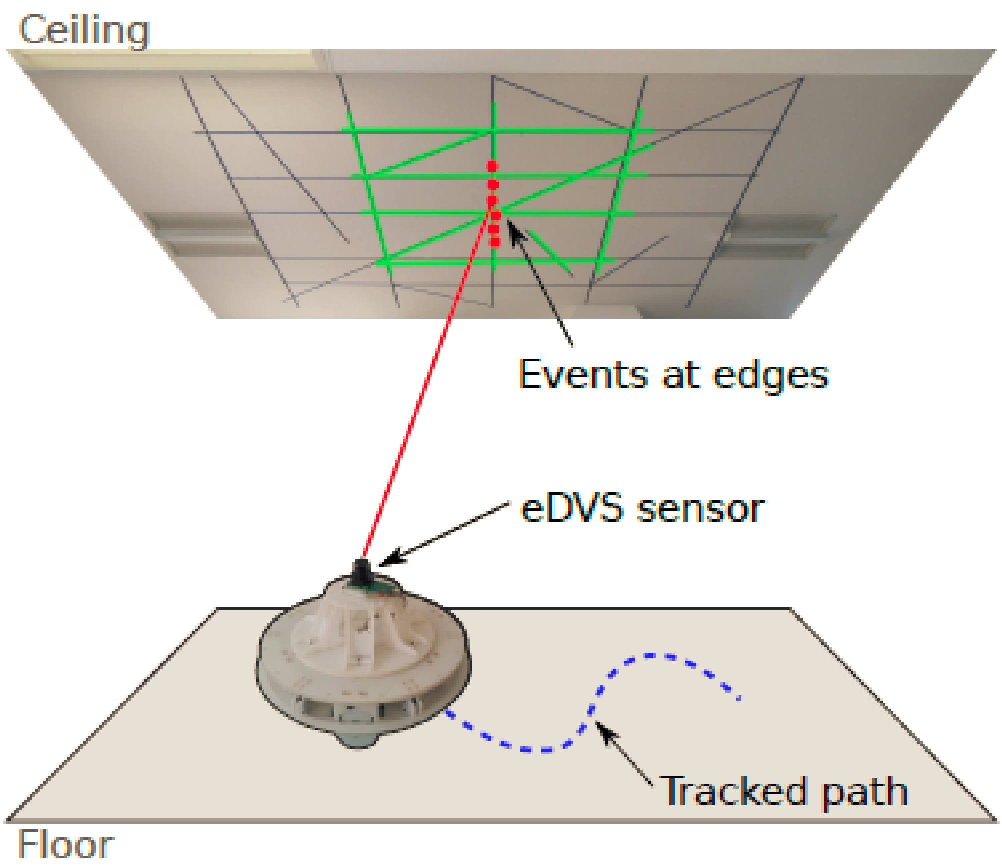


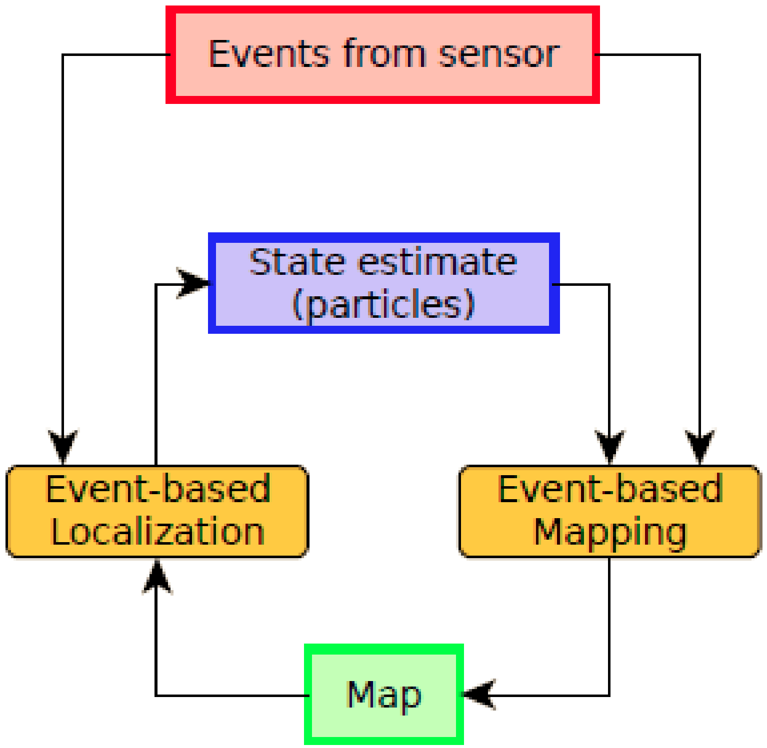
Conradt et al, ECV 2009

Extended Application Example: High Speed Tracking

Application Example: SLAM for Event-Based Vision Systems

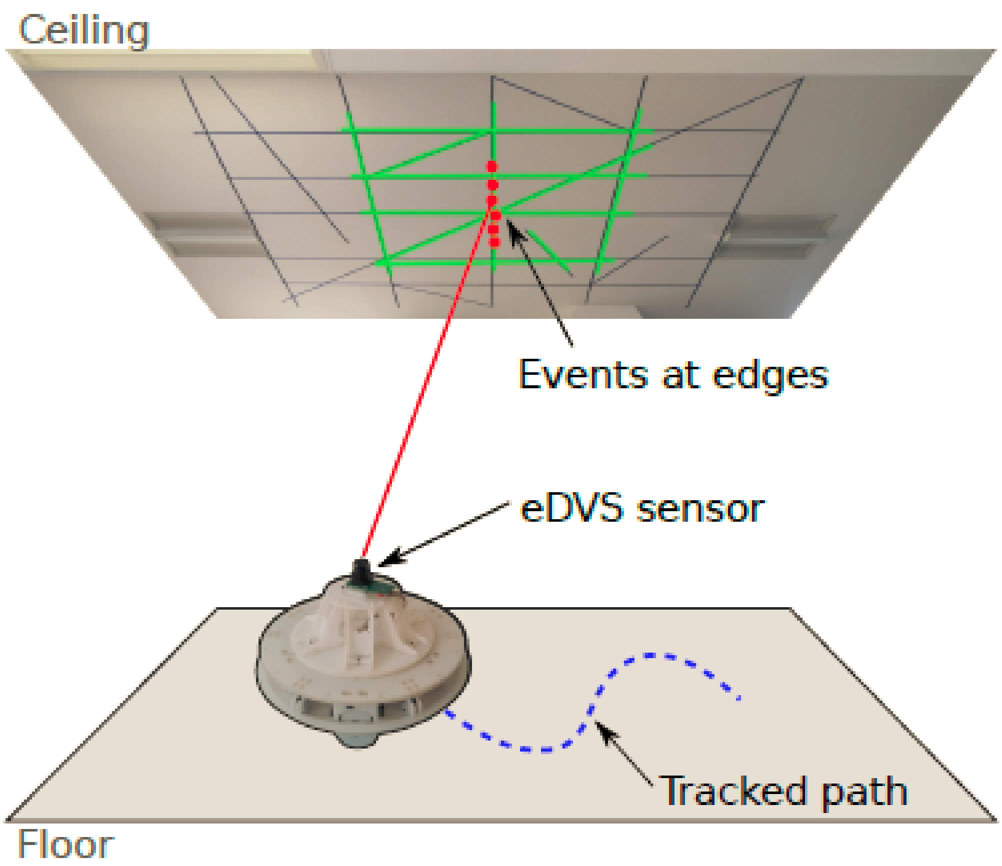
Framework: Particle Filter

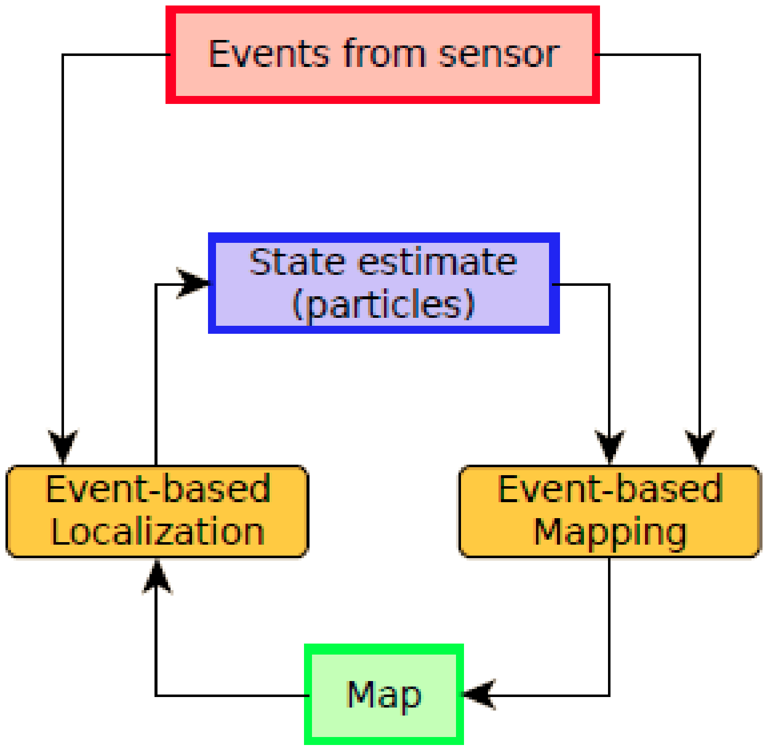




Application Example: SLAM for Event-Based Vision Systems

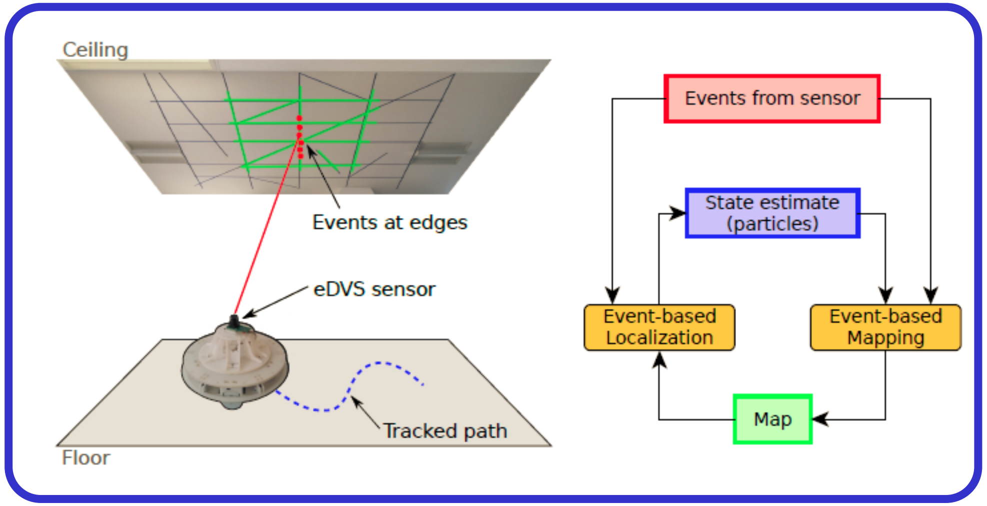
Event Based Localization





Application Example: SLAM for Event-Based Vision Systems

Event Based Mapping



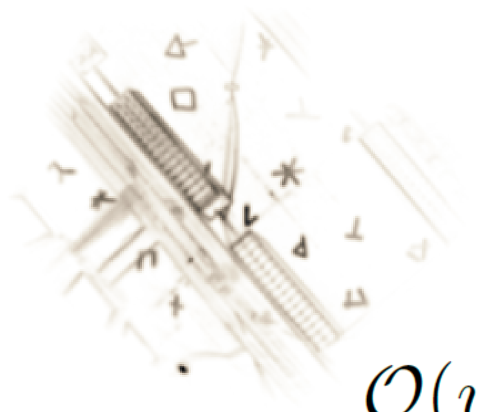
$$\mathcal{M}(u) = \frac{\#\text{ of occurred events for } u}{\#\text{ of possible obervations for } u} =: \frac{\mathcal{O}(u)}{\mathcal{Z}(u)}$$

$$\mathcal{O}^{(k)}(u) = \mathcal{O}^{(k-1)}(u) + \sum_{i=1}^{n} s_i^{(k)} \mathcal{N}\left(u \,\middle|\, \mu^{-1}(e^{(k)}|x_i^{(k)}), \sigma\right), \; \mathcal{O}^{(0)} = 0$$

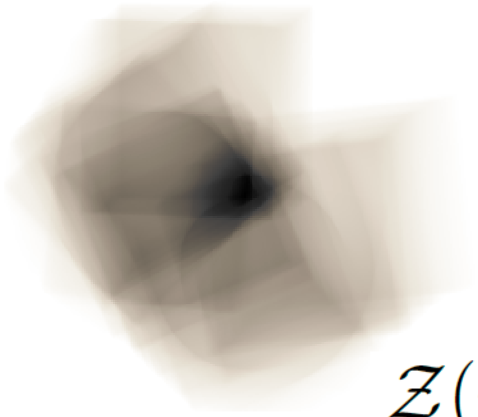
$$\mathcal{Z}^{(k)}(u) = \mathcal{Z}^{(k-1)}(u) + \|\mu(u|x_*^{(k)}) - \mu(u|x_*^{(k-1)})\|, \; \mathcal{Z}^{(0)} = 0$$

Application Example: SLAM for Event-Based Vision Systems

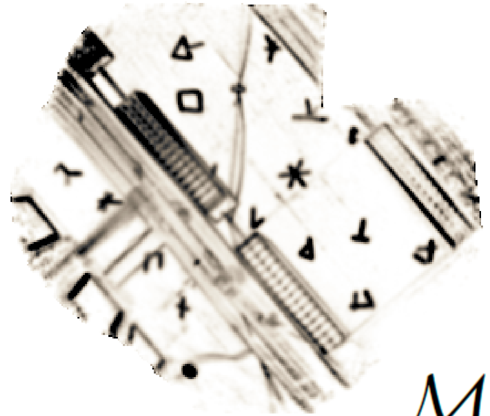
Event Based Mapping



$\mathcal{O}(u)$



$\mathcal{Z}(u)$



$\mathcal{M}(u)$

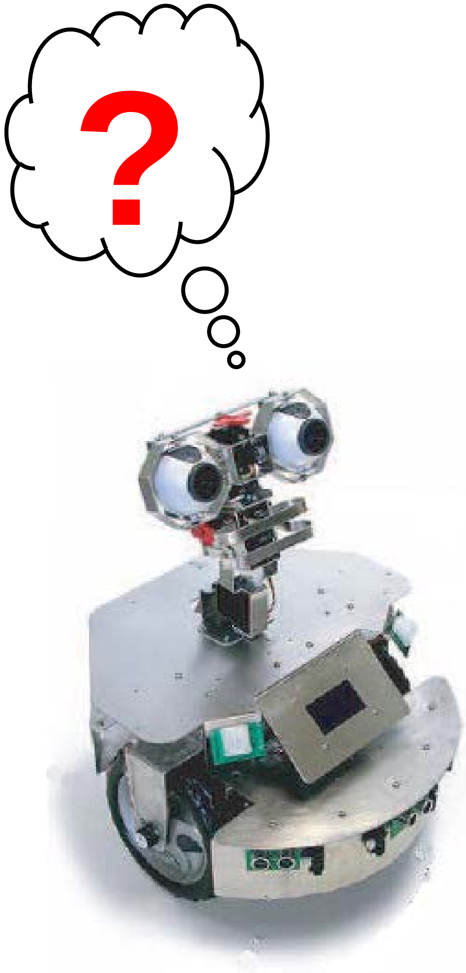
$$\mathcal{M}(u) = \frac{\#\text{ of occurred events for } u}{\#\text{ of possible obervations for } u} =: \frac{\mathcal{O}(u)}{\mathcal{Z}(u)}$$

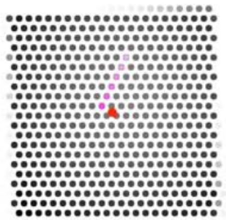
$$\mathcal{O}^{(k)}(u) = \mathcal{O}^{(k-1)}(u) + \sum_{i=1}^{n} s_i^{(k)} \mathcal{N}\left(u \,\middle|\, \mu^{-1}(e^{(k)}|x_i^{(k)}), \sigma\right), \; \mathcal{O}^{(0)} = 0$$

$$\mathcal{Z}^{(k)}(u) = \mathcal{Z}^{(k-1)}(u) + \|\mu(u|x_*^{(k)}) - \mu(u|x_*^{(k-1)})\|, \; \mathcal{Z}^{(0)} = 0$$

Weikersdorfer et al, ICVS 2013

Application Example: SLAM for Event-Based Vision Systems





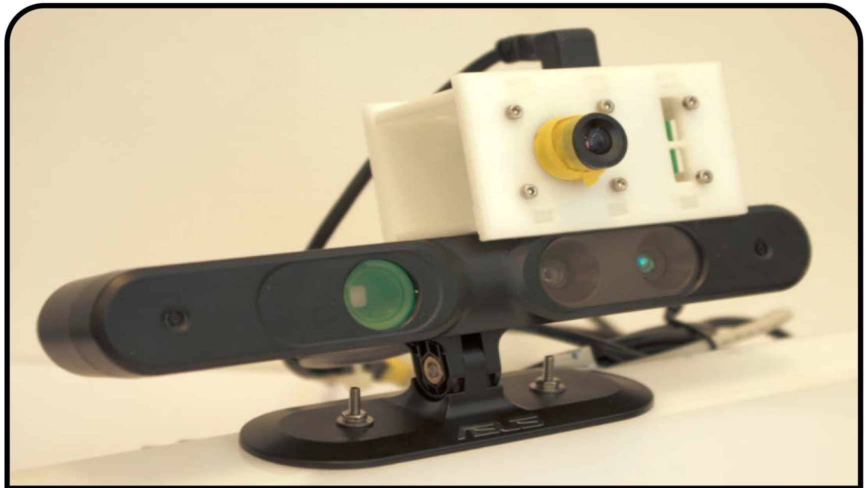
5x real time

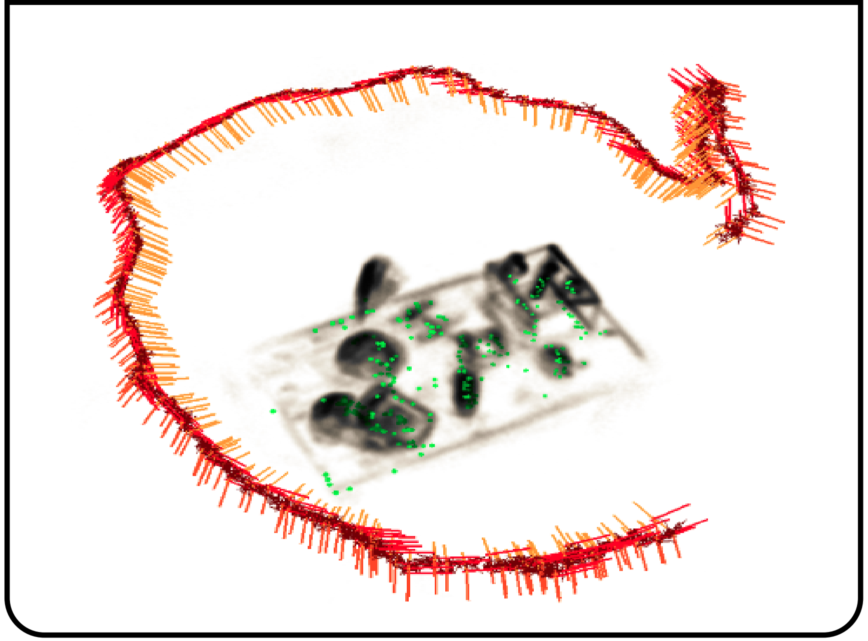
Event-based 3D SLAM
with a depth-augmented Dynamic Vision Sensor

David Weikersdorfer, David B. Adrian,
Daniel Cremers, Jörg Conradt

Technische Universität München, Germany

- Sensor combination: event-based dynamic vision and PrimeSense object distance
- Sparse sensory data allows real-time processing and integration
- Localization update rates of > 100 Hz on standard PC
- Suitable for small autonomous mobile robots in high-speed application scenarios



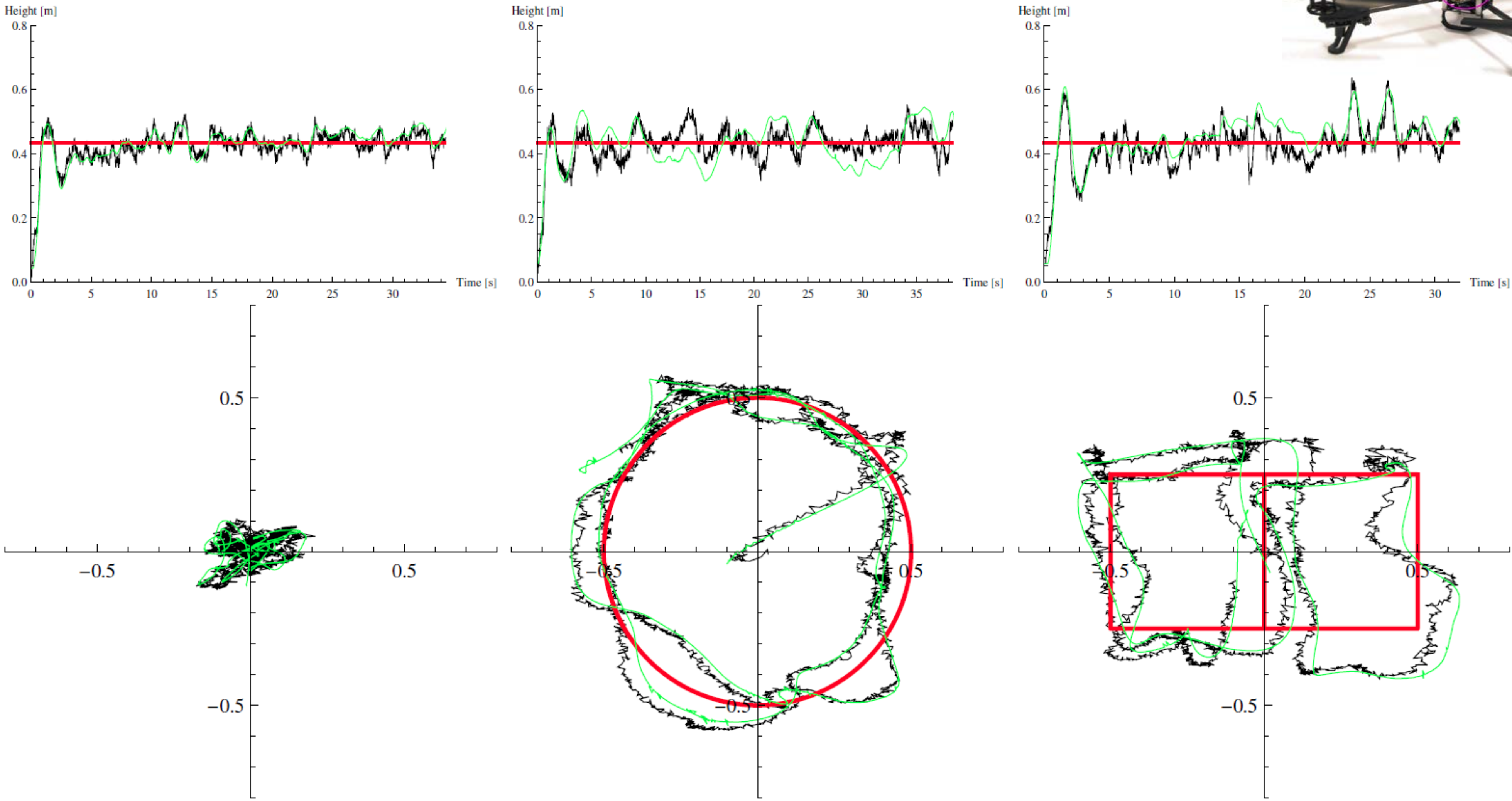


An Autonomous Indoor Quadrocopter (eb3D SLAM)

Parrot AR.Drone
diam. ca. 80 cm

Selected hardware components : eDVS (**red**), Mini-PC (**blue**),
Primesense IR projector (yellow) and IR camera (**green**), and drone control board (**magenta**).

An Autonomous Indoor Quadrocopter (eb3D SLAM)

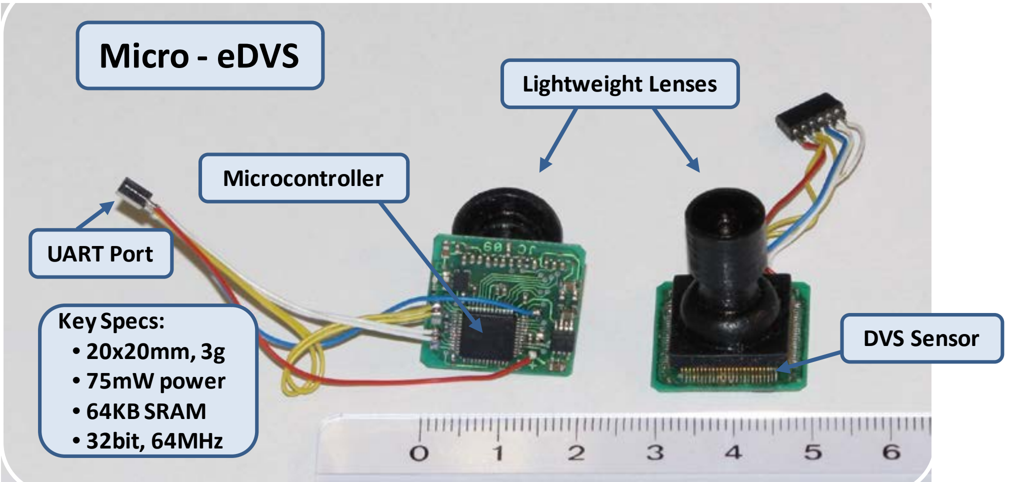


Evaluation against ground truth: **eb-3DSLAM**

desired trajectory **ext. tracker (OptiTrack Flex13)**

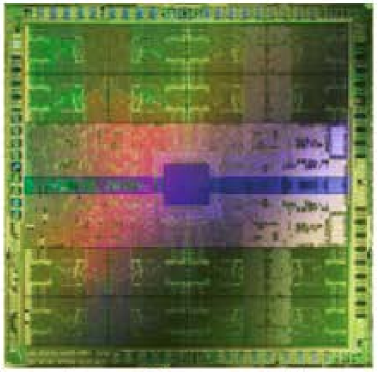
Outlook: An Autonomous Indoor Micro Quadrocopter (3D SLAM)

Key Technical Specs:
- Diameter 16cm
- Weight 38g
- Autonomy 14min



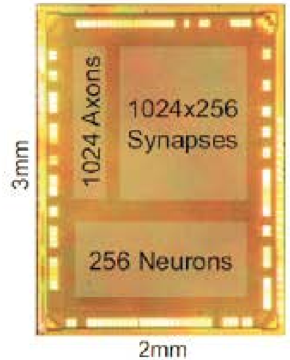
Neuromorphic Computing Landscape

A neuro-computing renaissance

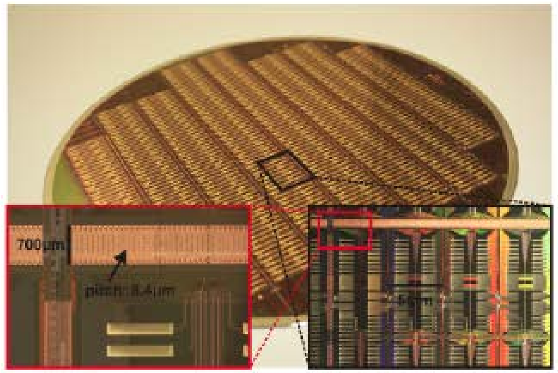


SW simulated neural networks (CPUs, GPUs).

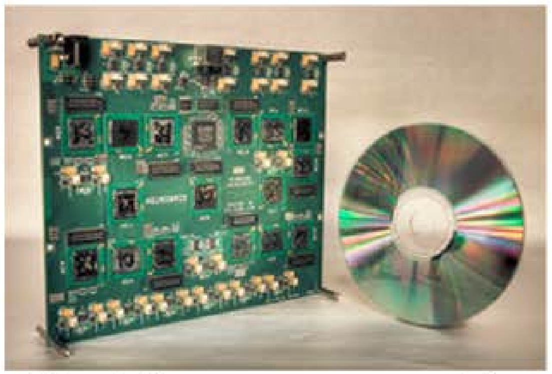
Real-time ARM-based neural network simulator (SpiNNaker).



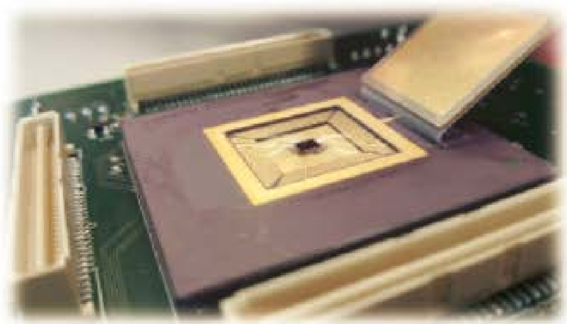
Fully digital *cognitive computing* chips (IBM).



Wafer-scale analog neural accelerators (BrainScaleS).



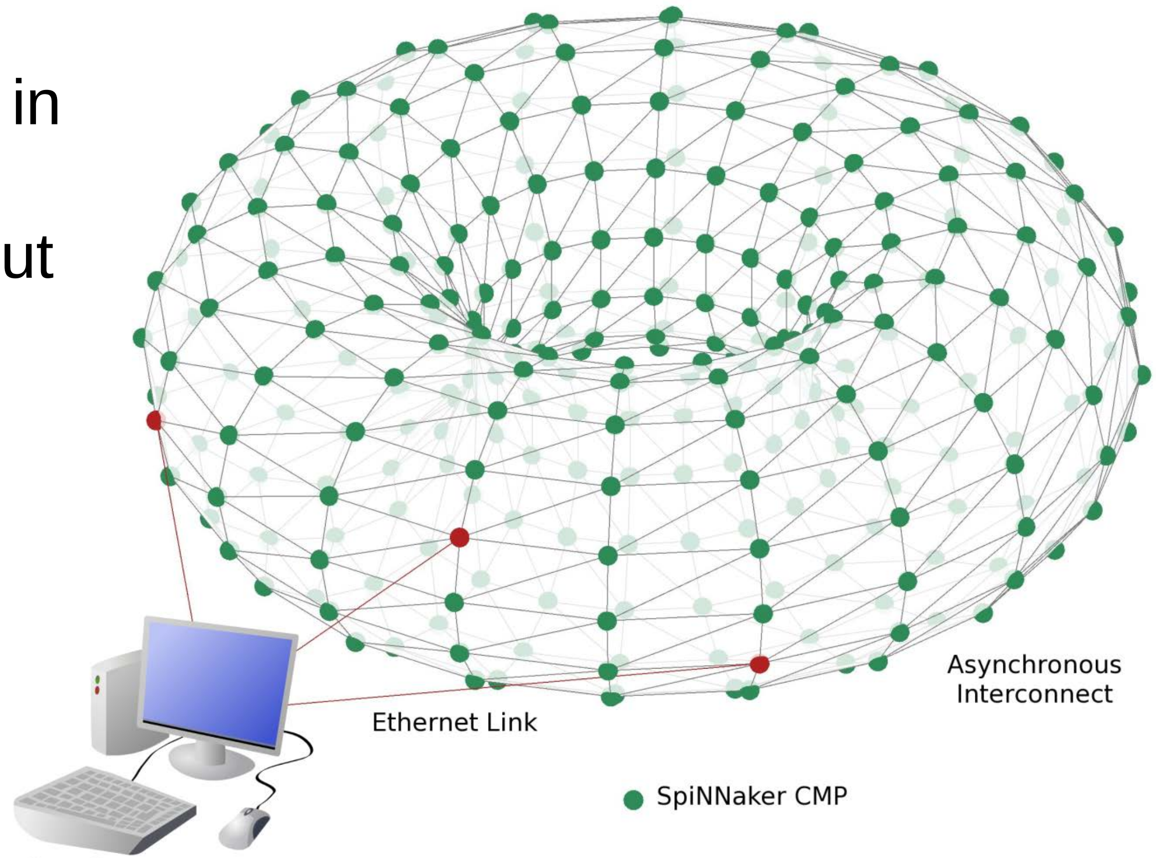
Real-time neuromorphic multi-chip emulator (NeuroGrid).



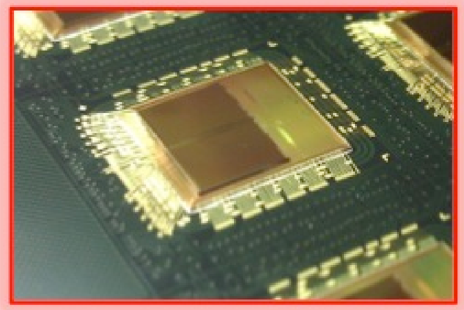
Real-time multi-neuron chip with plastic synapse neuron circuits (neuroP).

SpiNNaker Project

- A million mobile phone processors in one computer
- Able to model about 1% of the human brain…
- …or 10 mice!

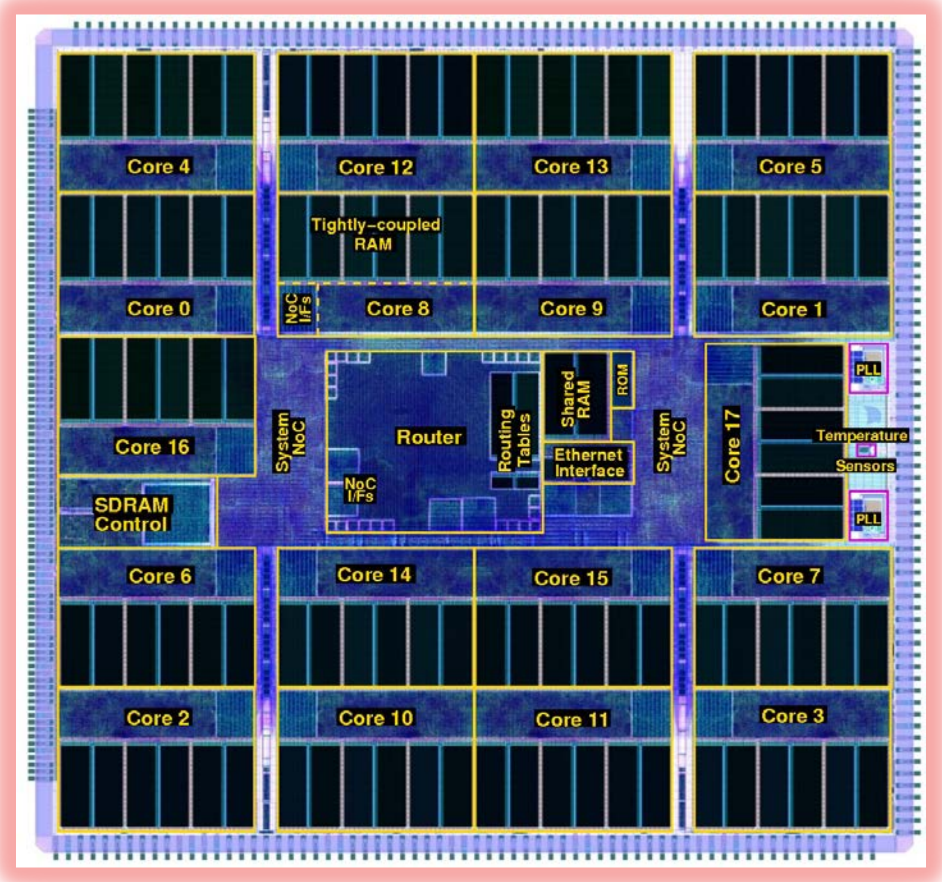


SpiNNaker Chip



Mobile
DDR
SDRAM
interface

Multi-chip
packaging by
UNISEM Europe



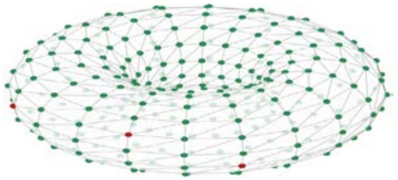
SpiNNaker

Biologically
Inspired
Massively
Parallel
Architectures

MANCHESTER 1824

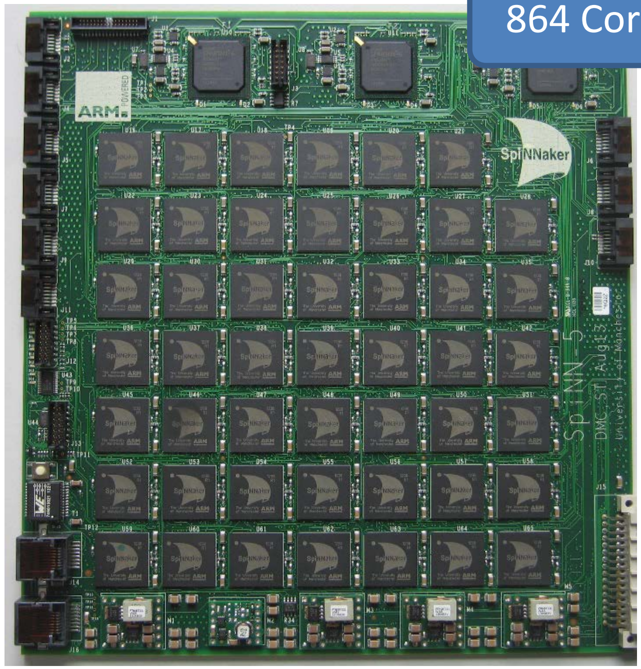
Distributed Neuronal Computation "on chip"

Asynchronous Spiking Neural Computation Hardware for low-power real-time operation in Closed-Loop Systems



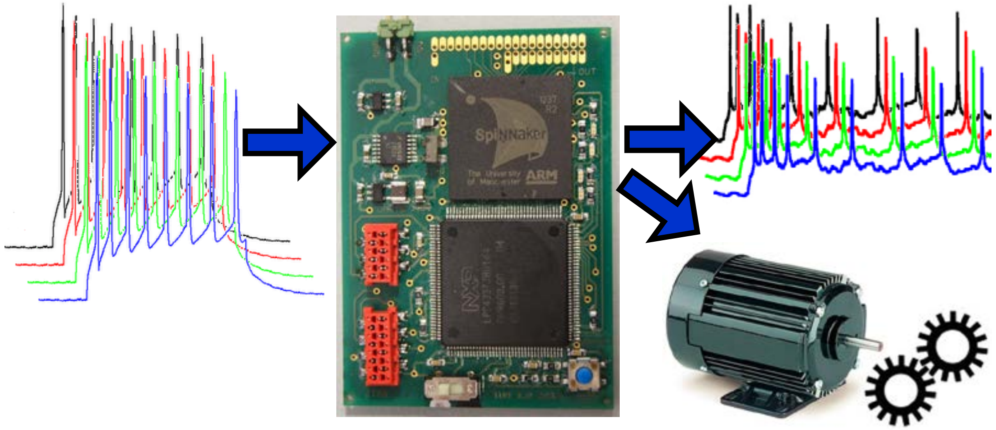
... to simulate 1 Billion
Spiking Neurons
in real-time

864 Core "Rack Version"



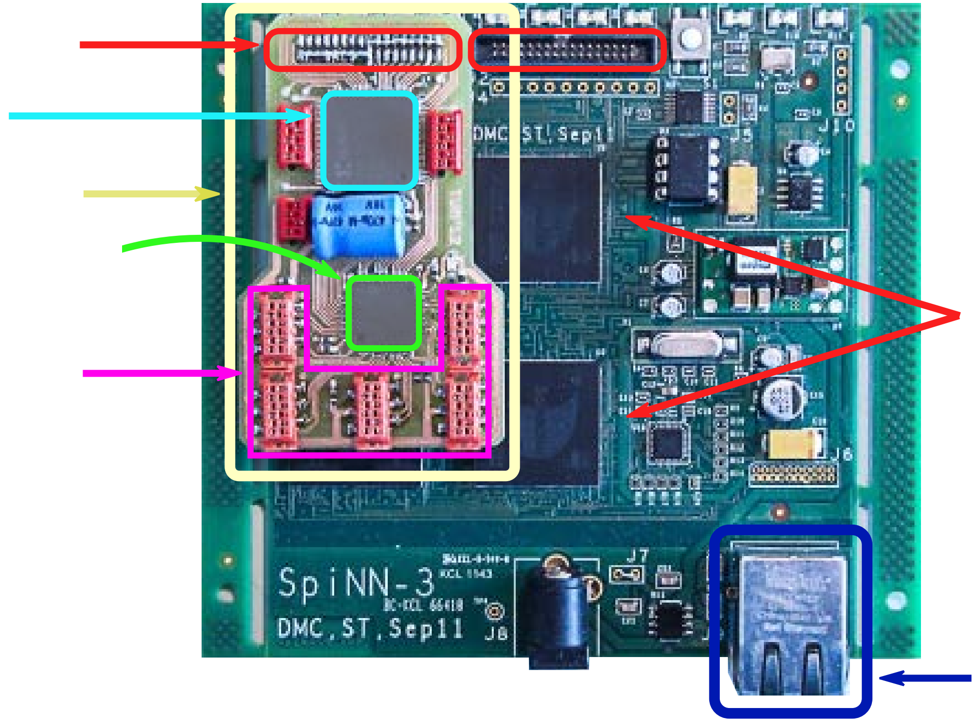
72 Core Evaluation

18 Core Stand-Alone Prototype

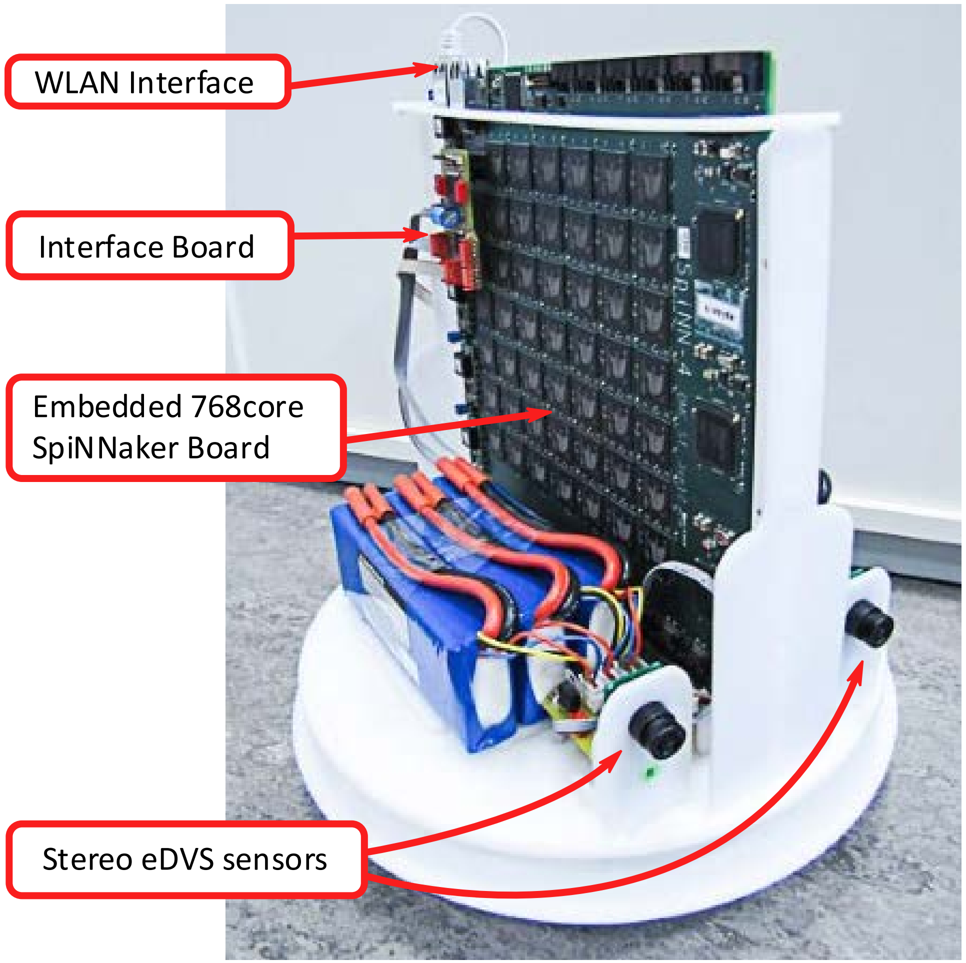


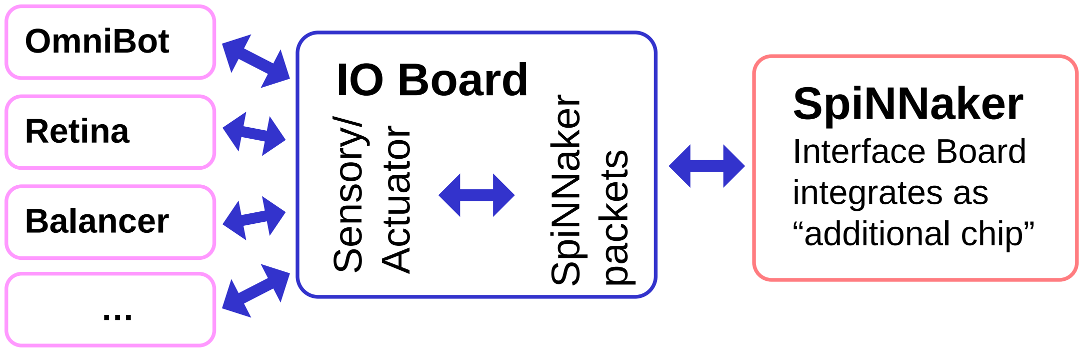
- Multi-channel spiking input and output
- Stand-alone spiking computing system
- Simulates ~20.000 neurons in real time
- Small (~20x20mm); low power (~600mW)
- Flexibly configurable, extendable, stackable

The SpiNNaker – Robot Interface Board



NST SpOmniBot





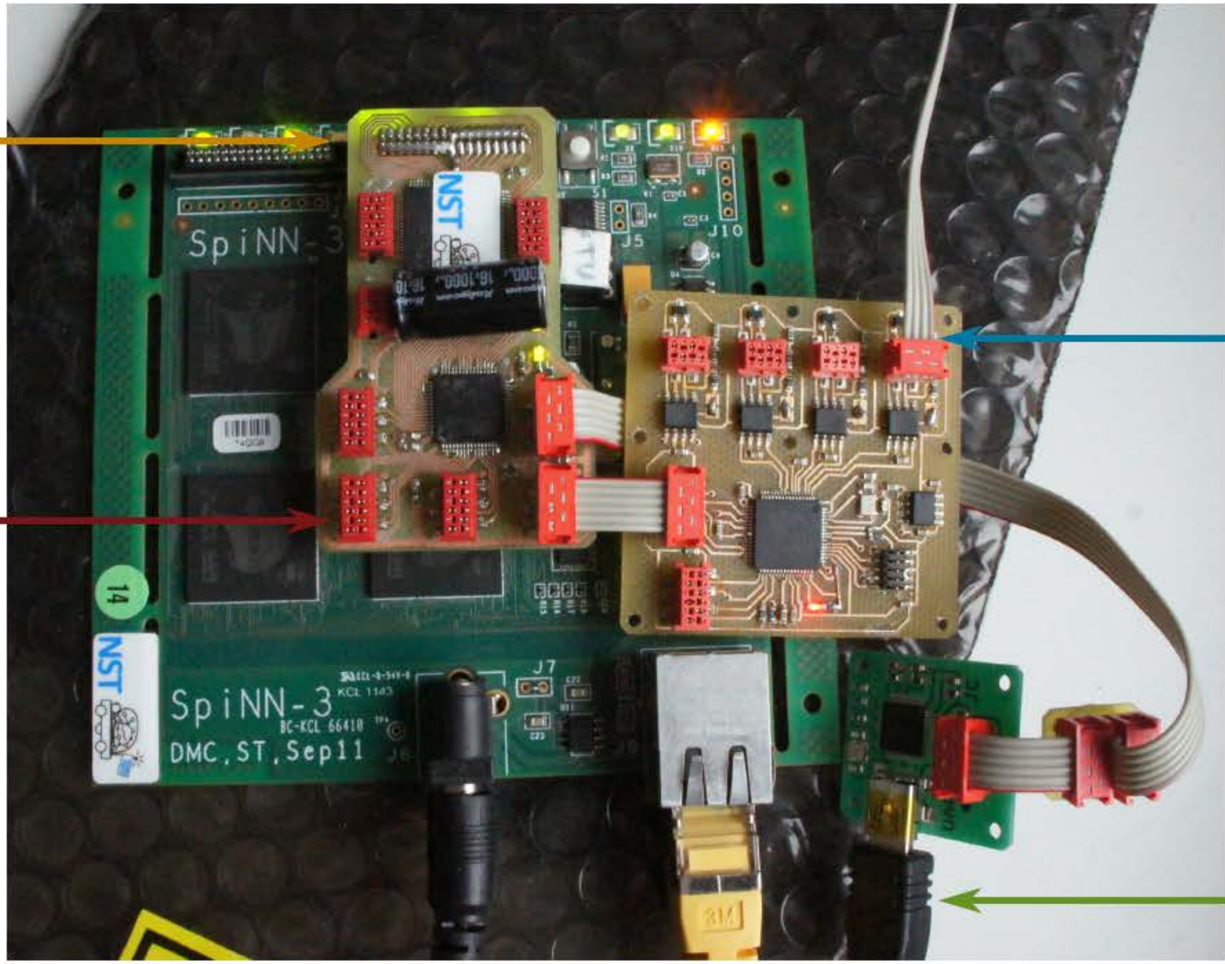
Interface to Compliant Robotic Arm

SpiNN-Link:

IO-board interacts with SpiNNaker on its native BUS and protocol

UART:

IO-board UART ports to connect to peripherals



CAN:

IO-board's own 4x CAN controller

USB/UART:

interact with IO-board, live interaction with running simulation

Power:

5 V, <1 A

Ethernet:

control SpiNNaker, set up simulation

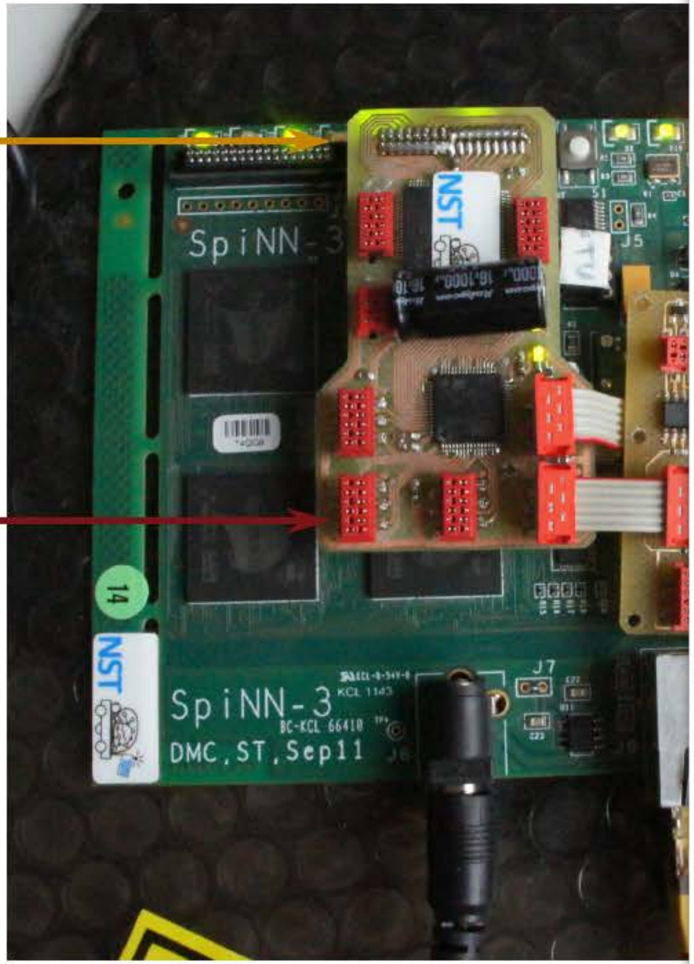
Interface to Compliant Robotic Arm

SpiNN-Link:

IO-board interacts with SpiNNaker on its native BUS and protocol

UART:

IO-board UART ports to connect to peripherals



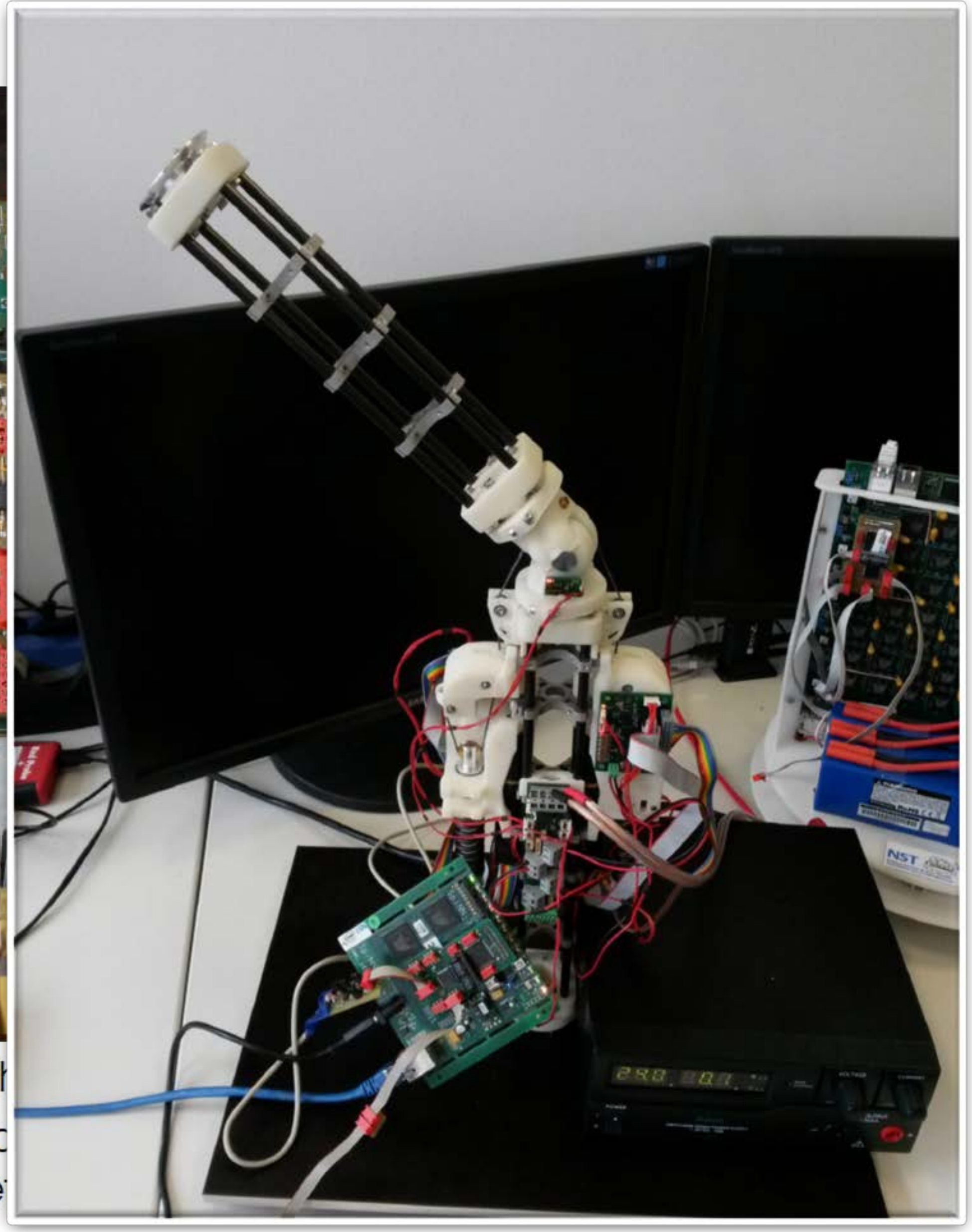
Power:

5 V, <1 A

Eth

co

se



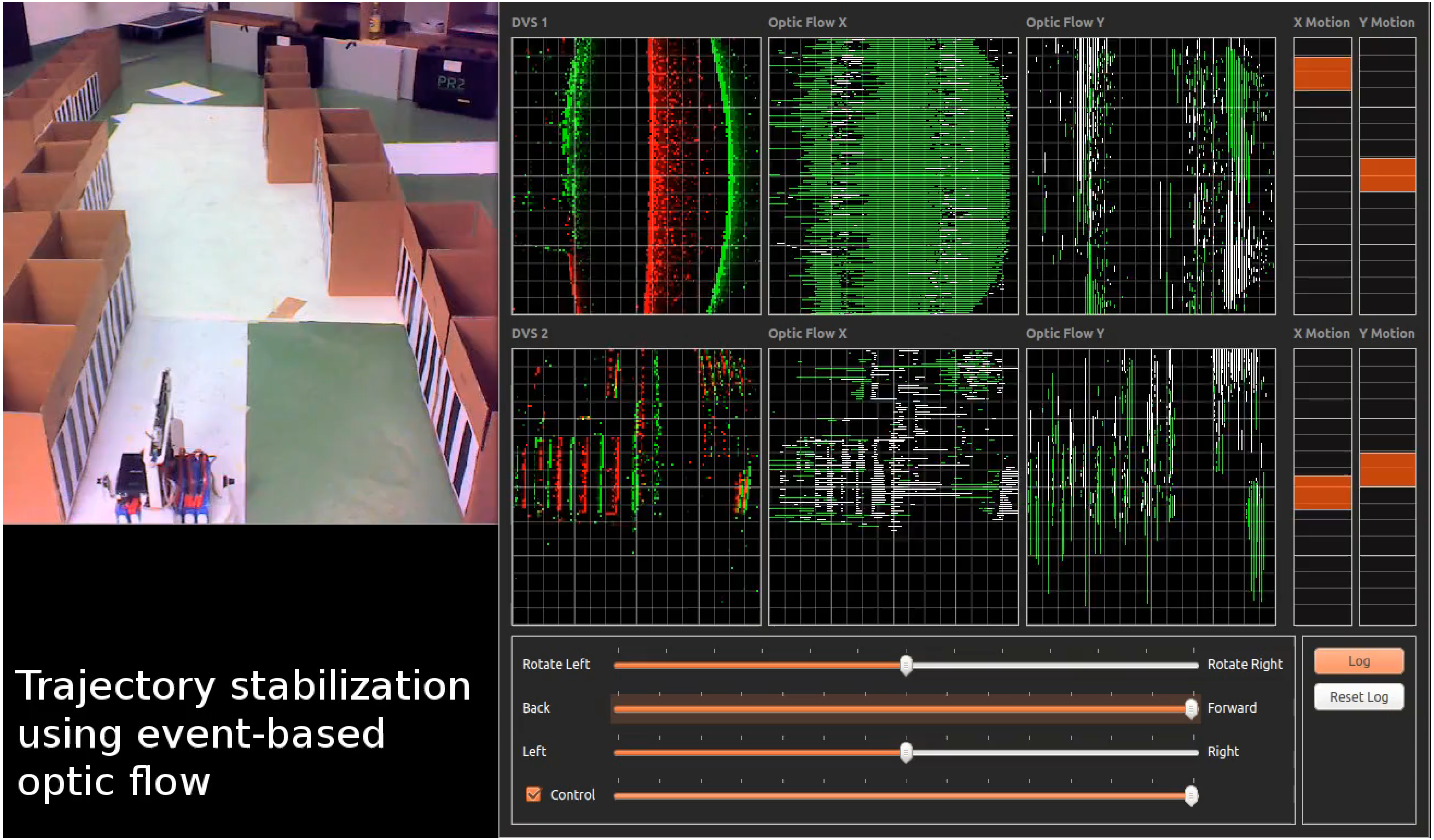
Example Projects: SpOmniBee

- 2 laterally facing retinas
- Retina events feed into SpiNNaker
- Distributed computation of local optical flow
- Global combination of optic flow
- Computation of Motor commands on SpiNNaker, send to robot

Goal:

keep SpOmnibot centered in hallway with marking, similar to Srinivasan’s Bee experiments

Application Example: Event-Based Optic Flow for Robot Control



Example Projects: Ball Balancer

- One Retina to observe the platform in top-down view
- Two motors to control platform's pan and tilt angles
- SpiNNaker Interface Board is connected to retina and motors

Goal: Keep ball (one or more!) on a certain trajectory

Neural Principles for System Control