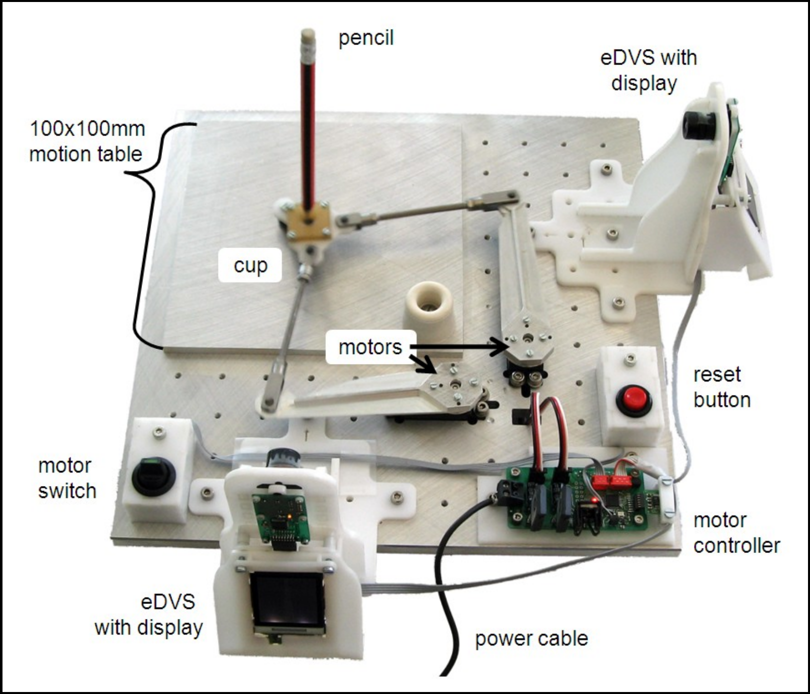

pencil
eDVS with display
100x100mm motion table
cup
motors
reset button
motor switch
motor controller
eDVS with display
power cable

Event-Driven Sensing

Embedded & Realtime Systems

Konstantinos Samaras-Tsakiris

June 1, 2016

Problem Statement

Requirements of many robotic applications:

Problem Statement

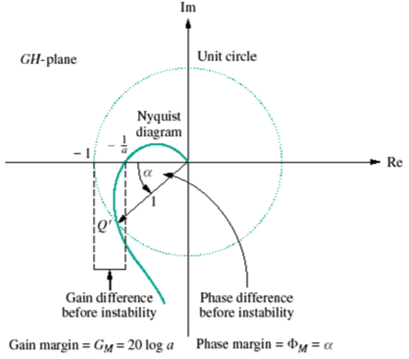
Requirements of many robotic applications:

- **Low latency control**

Problem Statement

Requirements of many robotic applications:

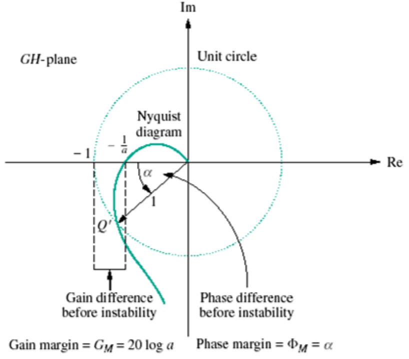
- **Low latency control**



Problem Statement

Requirements of many robotic applications:

- **Low latency control**
- Non-redundant information from sensors – saves on:
  - Processing resources
  - Power consumption
  - Communication bandwidth

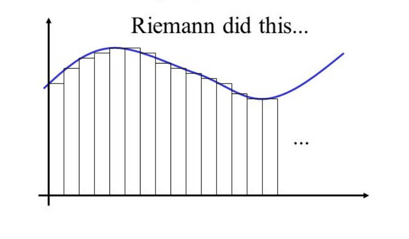


Event-driven Sampling

Lebesgue vs Riemann

Event-driven sampling

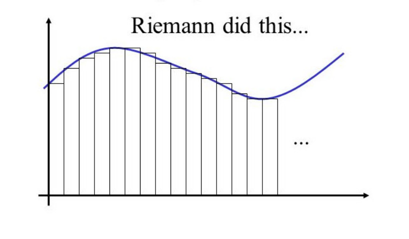
Also called "Lebesgue sampling"

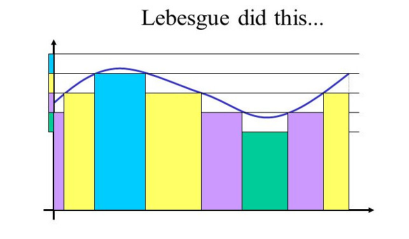


Lebesgue vs Riemann

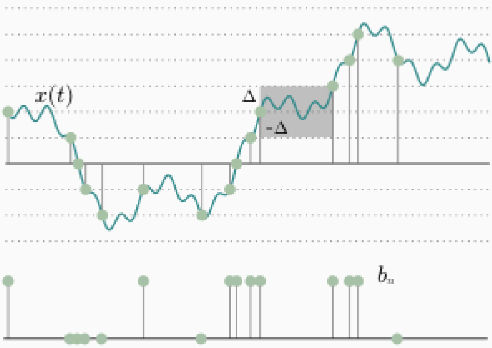
Event-driven sampling

Also called "Lebesgue sampling"

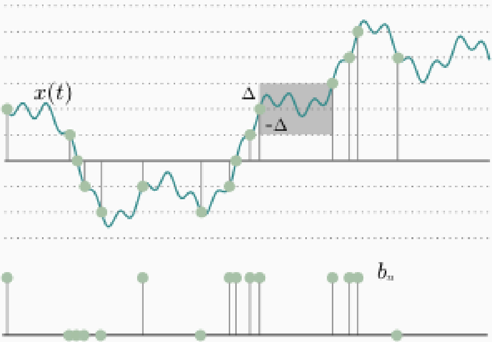




Lebesgue sampling → Events



Lebesgue sampling → Events



Lebesgue sampling $\rightarrow$ Events

Event Typically boolean "message" - a spike, or $\delta(t)$

Sensor output Asynchronous stream of events - a train of $\delta(t)$

Some event-driven systems

- Position encoders
  - Pulse when position has changed by specific amount

  $\rightarrow$ *Time encoding*

Some event-driven systems

- Position encoders
  - Pulse when position has changed by specific amount

  $\rightarrow$ *Time encoding*
- Pulse-frequency modulation
  - Light-load DC-DC conversion

  $\rightarrow$ *Rate encoding*

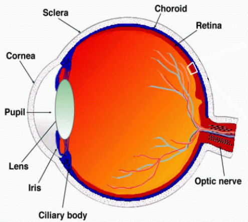
Some event-driven systems

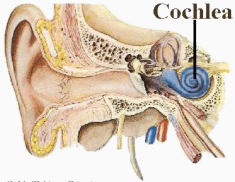
- Position encoders
  - Pulse when position has changed by specific amount

  $\rightarrow$ *Time encoding*
- Pulse-frequency modulation
  - Light-load DC-DC conversion

  $\rightarrow$ *Rate encoding*
- Biological sensory organs
  - Retina
  - Cochlea

  $\rightarrow$ *Encoding? Likely time and place*





Alec N. Salt, Washington University

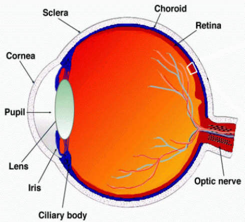
Some event-driven systems

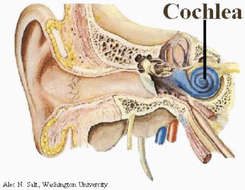
- Position encoders
  - Pulse when position has changed by specific amount

  $\rightarrow$ *Time encoding*
- Pulse-frequency modulation
  - Light-load DC-DC conversion

  $\rightarrow$ *Rate encoding*
- Biological sensory organs
  - Retina
  - Cochlea

  $\rightarrow$ *Encoding? Likely time and place*





Biological neurons!

Comparison of periodic & event-driven control

Definitions

The control attempts to keep the system state at the origin. Let the system to control be defined by:

$$dx = u\,dt + dv$$

where

x**:** System state

u**:** Control signal

v**:** Disturbance (Wiener process)

d**:** Lebesgue sampling interval

Comparison of periodic & event-driven control

$$dx = u\,dt + dv$$

Periodic sampling with period h

Using a minimum variance controller, the control law becomes:

$$u = -\frac{1}{h}\frac{3+\sqrt{3}}{2+\sqrt{3}}\,x$$

and the variance becomes:

$$V_R = \frac{3+\sqrt{3}}{6}\,h$$

Comparison of periodic & event-driven control

$$dx = u\,dt + dv$$

Event-driven sampling with mean period h_L

Need different control strategy: an impulse that instantly returns the system to the origin. T_d is the time it takes for $|x(t_k)| = d$ for the 1st time. Mean exit time and mean sampling period:

$$h_L := E[T_d] = d^2$$

And the steady state variance:

$$V_L = \frac{d^2}{6} = \frac{h_L}{6}$$

Comparison of periodic & event-driven control

Comparison

To compare, assume the mean sampling rates are equal: $h = h_L$.

Then:

$$\frac{V_R}{V_L} = 4.7$$

But each follows a different control strategy to allow simple analysis.

Comparison of periodic & event-driven control

Comparison

To compare, assume the mean sampling rates are equal: $h = h_L$.

Then:

$$\frac{V_R}{V_L} = 4.7$$

But each follows a different control strategy to allow simple analysis.

Periodic sampling with impulse control as well:

$$\frac{V_R'}{V_L} = 3$$

$\boxed{\text{Event-driven sampling offers } \textit{3 times less variance}}$

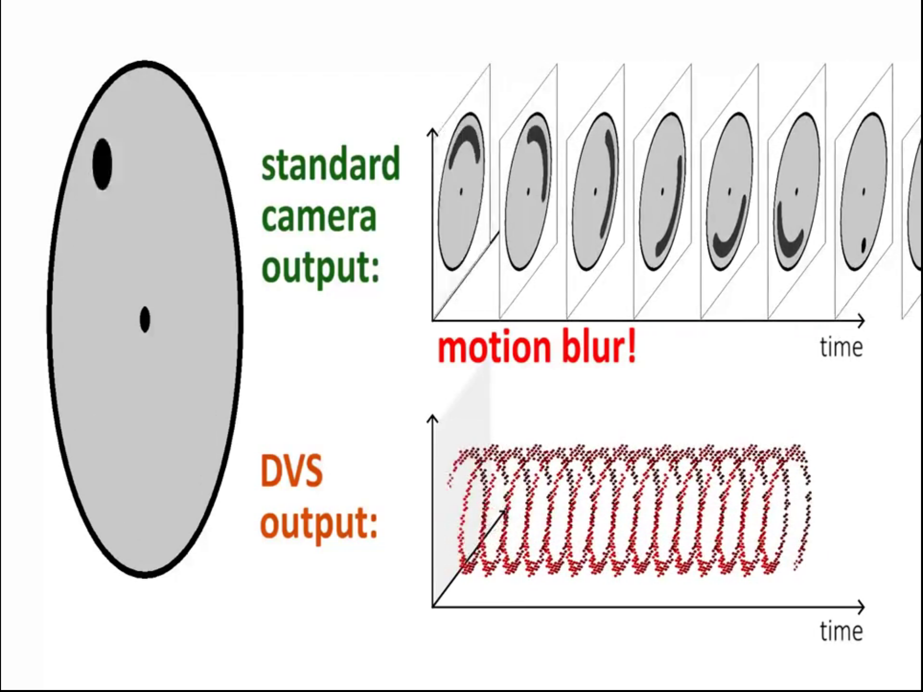
The difference is even larger for unstable systems

Consequences for control

Event signals are incompatible with traditional signal processing of continuous signals. But offer other benefits:

- Take control action only as necessary - not on steady state!
- Same control efficiency with lower sampling rate

Dynamic Vision Sensor


standard camera output:
motion blur!
time
DVS output:
time

From frames to events

Normal camera

From frames to events

Normal camera

- Rolling or global shutter
- Fixed frame rate
- Single pictures at constant rate

From frames to events

Normal camera

- Rolling or global shutter
- Fixed frame rate
- Single pictures at constant rate

Event-driven camera

From frames to events

Normal camera

- Rolling or global shutter
- Fixed frame rate
- Single pictures at constant rate

Event-driven camera

- Independent pixels
- Each detects brightness change in its receptive field
  - *Logarithmic* changes

From frames to events

Normal camera

- Rolling or global shutter
- Fixed frame rate
- Single pictures at constant rate

Event-driven camera

- Independent pixels
- Each detects brightness change in its receptive field
  - *Logarithmic* changes

Fechner's Law

From frames to events

Event-driven camera

- Independent pixels
- Each detects brightness change in its receptive field
  - *Logarithmic changes*
- change $>$ threshold $\rightarrow$ event (asynchronous)
  - event: $e_k = x_k, y_k, t_k, p_k$ – coordinates, timestamp & polarity

From frames to events

Event-driven camera

- Independent pixels
- Each detects brightness change in its receptive field
  - *Logarithmic changes*
- change $>$ threshold $\rightarrow$ event (asynchronous)
  - event: $e_k = x_k, y_k, t_k, p_k$ – coordinates, timestamp & polarity
- High temporal resolution (microseconds)
- Low spatial resolution ($128 \times 128px$)
- Dynamic range $120dB$ vs. $60dB$ for traditional image sensors
- Shared event bus
  - Statistical time division multiplexing

From frames to events

Event-driven camera

- Independent pixels
- Each detects brightness change in its receptive field
  - *Logarithmic changes*
- change $>$ threshold $\rightarrow$ event (asynchronous)
  - event: $e_k = x_k, y_k, t_k, p_k$ – coordinates, timestamp & polarity
- High temporal resolution (microseconds)
- Low spatial resolution ($128 \times 128px$)
- Dynamic range $120dB$ vs. $60dB$ for traditional image sensors
- Shared event bus
  - Statistical time division multiplexing

$\boxed{\text{Asynchronous event stream instead of frames}}$

Event Processing

Problem satisfaction

An event-driven sensor can address the issues stated at the beginning

- Low latency
- Little data redundancy

Event Processing

Problem satisfaction

An event-driven sensor can address the issues stated at the beginning

- Low latency
- Little data redundancy

Communication protocol

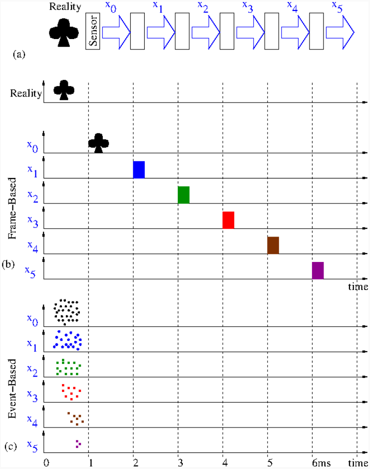
Address Event Representation

Requires different processing methods from standard sensors, based on spatio-temporal correllation

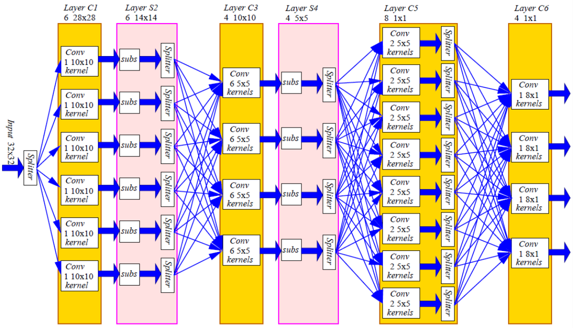
Pseudo-simultaneity

Pseudo-simultaneity of event processing

Each event adds only little information, but is processed very fast, and the response time is not throttled by the frame rate. The output event stream lags behind the input only by a few events.



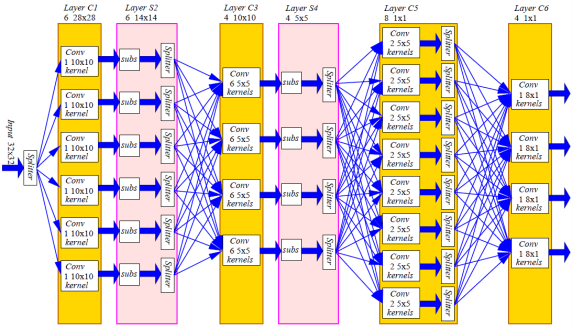
Example Visual Event Processor: Convolutional NN



Implementations:

1. GPU

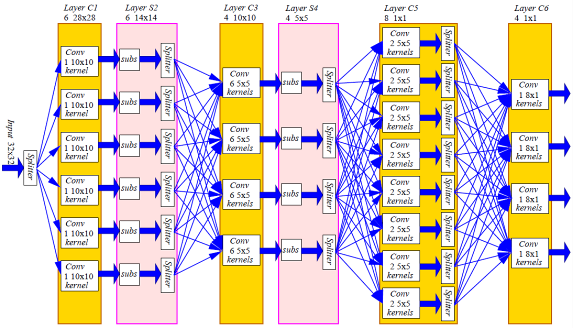
Example Visual Event Processor: Convolutional NN



Implementations:

1. GPU
2. FPGA

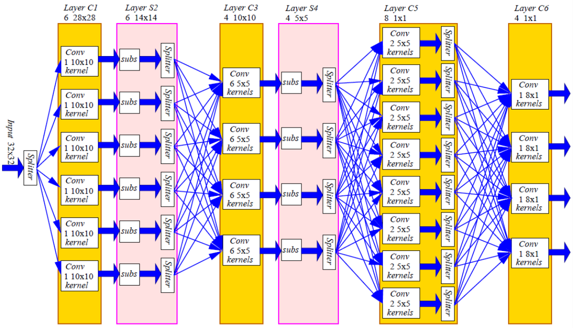
Example Visual Event Processor: Convolutional NN



Implementations:

1. GPU
2. FPGA
3. Custom digital hardware

Example Visual Event Processor: Convolutional NN



Implementations:

1. GPU
2. FPGA
3. Custom digital hardware
4. Custom analog hardware

Back to the Pencil Balancer

Pencil balancer description

Visual balancing robots

Typical
- Frame-based sensors
- Complex nonlinear control

Pencil balancer description

Visual balancing robots

Typical	- Frame-based sensors
	- Complex nonlinear control
This one	- Event-driven sensors
	- Simple PD control (thanks to *low latency*)

Pencil balancer description

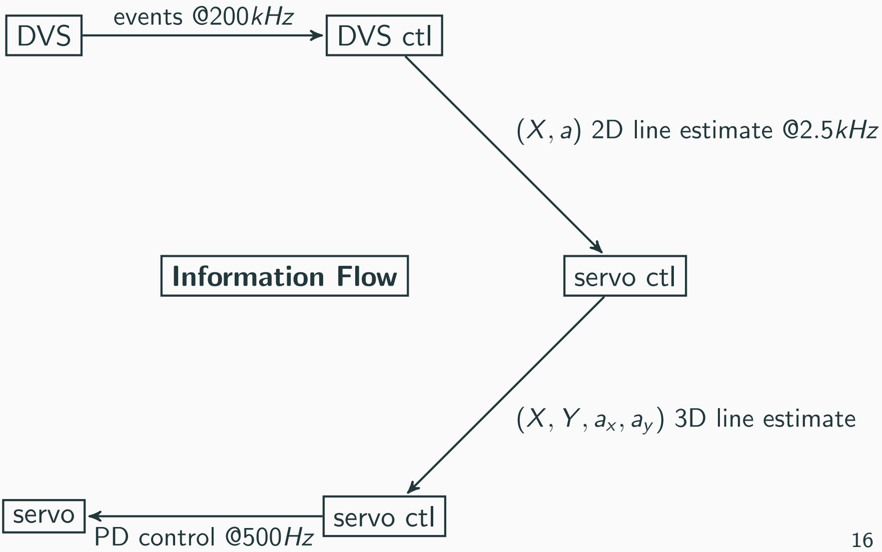
Visual balancing robots

Typical	- Frame-based sensors
	- Complex nonlinear control
This one	- Event-driven sensors
	- Simple PD control (thanks to *low latency*)

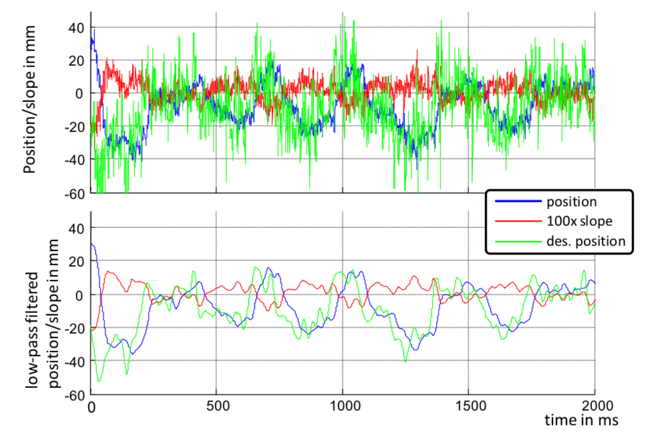
System elements:

1. Monitors object with 2 DVS
2. Actuates with 2 independent servos
3. Controls with linear PD algorithm

Wrapping up with pencil balancer



Wrapping up with pencil balancer



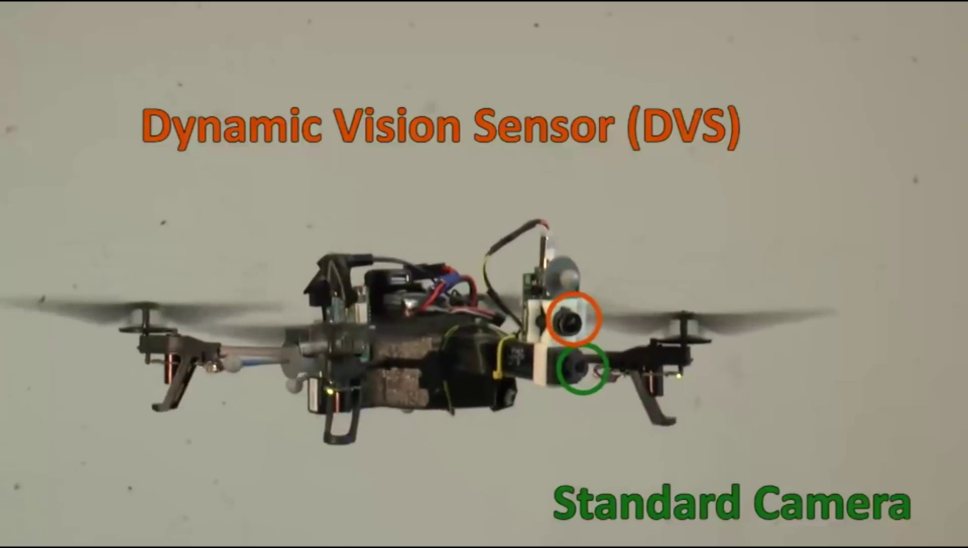
Thank you!

References I

1. Jonathan Tapson, Greg Cohen, Andre van Schaik, “ELM solutions for event-based systems”, Neurocomputing, 13 Jan 2014
2. K.J. Astrom, B.M. Bernhardsson, “Comparison of Riemann and Lebesgue sampling for first order stochastic systems“, Proc. 41st IEEE Conf. Decis. Control 2
3. Jorg Conradt, Raphael Berner, Matthew Cook, “An Embedded AER Dynamic Vision Sensor for Low-Latency Pole Balancing”, IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009

References II

4. L. A. Camunas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco, “Event-Driven Sensing and Processing for High-Speed Robotic Vision”
5. Shih-Chii Liu and Tobi Delbruck, “Neuromorphic sensory systems”, Current Opinion in Neurobiology, 2010


Dynamic Vision Sensor (DVS)
Standard Camera