Event Sensors for Embedded Edge AI Vision Applications

Christoph Posch, Co-founder and CTO, PROPHESEE
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
**Image Sensing and Computer Vision**

**Image Sensor**
- Integrate light intensity in a time interval (exposure time)
- read out *average* exposure
- at *regular* intervals (frame rate)
- for all pixels

⇒ Image = snapshot in time (static)

CV: Motion acquisition based on series of static images

Problems with frames in CV:
1. Temporal details lost (under-sampling)
2. Displacement, blur (under-sampling)
3. Redundant data (over-sampling)
**Introduction Event-based Vision**

Different dynamics in different parts of the scene
- Each pixel sees a different signal

Not one sampling rate for all pixels (=frame rate)
- but many (= as many as pixels), and
- sampling rates can *vary*, on the fly, and pixel-individually

How? Each pixel individually controls its own sampling based on the input signal
- Change sampling domain → from time to *amplitude*
- Encode information in "events"

From images to stream of pixel-individual data

**Event sensor characteristics**
- Captures fine temporal detail of motion → *fast*
- Inherent data compression by eliminating redundant information → *sparse* (focus on relevant dynamic data)
- Pixel-individual log-domain operation (relative change) → *high dynamic range*
Introduction > Event Sensor Acquisition Principle

UNIZH, Robotics and Perception Group, https://rpg.ifi.uzh.ch/
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Event Sensors for Edge-AI Applications

Challenges for event sensors usage and integration

- Diverse fields of application for event sensors: Industrial, surveillance, IoT, AR/VR, mobile, automotive, ...
- Unconventional format of the event data
- Unfamiliar encoding of dynamic visual information as events
- Non-constant data rates
- Non-standard interfaces and data formats, non-standardized interface protocols

Prophesee has developed the first of a new generation of event sensor designed with the explicit goal to improve integrability and usability of event sensing technology in embedded at-the-edge vision system.
Design Targets

Integrability and usability in embedded edge-AI and IoT vision systems

- **Interfacing and system integration**
  - Flexible/programmable event data pre-processing, filtering and formatting
  - Industry standard interface compatibility (MIPI, DCMI)
  - Low-latency connectivity for low-power uCs, neuromorphic processor architectures, SNN accelerators, ...

- **Power optimization**
  - Hierarchy of power-modes for adaptative application-specific power optimization
  - Ultra-low power/always-on operation with sensor and system wake-up features
  - On-chip power management
  - Embedded microcontroller core for improved sensor flexibility and useability at-the-edge (data statistics, meta data, microcode execution)
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Sensor Chip Overview

- **320x320 pixels, 1/5" optical, 13mm² die**
  - Cu-Cu Stacked BSI, (CIS 65nm, CMOS 40nm)
  - Pixel pitch: 6.3um

- **Sensor features:**
  - Global Contrast Detector (GCD)
  - Pixel-level Flexible Region-of-Interest (ROI)
  - Temperature sensor
  - Ambient light sensor
  - ULP ultra-low power modes

- **Processing features:**
  - Digital ESP (Event Signal Processing) pipeline with
  - Noise/Flicker Filters, Edge enhancement, ERC, EDF
  - Embedded RISC-V CPU
  - Data interfaces: MIPI (D-PHY), CMOS Parallel (DCMI, AER)
  - Configuration interfaces: SPI/I²C
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Sensor Features > Global Contrast Detector

- Global Contrast Detector (GCD) → circuit block for contrast detection from multiple photodiode signals
- 9 GCDs, each connected to 108x108 photodiodes, forming a 3x3 matrix of “pixel-clusters”
- GCD circuit features:
  - Adjustable low/high-pass filter
  - Gain, contrast sensitivity controls
  - Threshold comparators
- Enables smart motion detection at ultra-low power consumption
Sensor Features > Global Contrast Detector

- GCD detect moving object entering/exiting pixel-cluster’s FoV
- Enables ultra-low power / always-on smart scene activity detection
- Create internal and external (chip pin) wake-up signals in Passive Ultra Low-Power mode.
Sensor Features > Pixel-level Flexible ROI

• All pixels embed in-pixel latch for activation/deactivation
• Each latch is individually controllable
• Real-time operation

• Usages:
  – Hot pixel removal
  – Fully customizable ROI shapes
  – Dynamic saliency
  – Limit power consumption outside ROI

• Windows mode ROI controller:
  – Ease user programming of multiple rectangular pixel array regions
  – Speed-up programming for dynamic usage
  – Manages automatically up to 18 ROI windows
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Processing features > ESP-Event Signal Processing

Event signal Processing(ESP) pipeline:
- Pixel array & Read Out
- Noise Filter (NFL)
- Anti-Flicker (AFK)
- Spatio-Temporal Contrast (STC)
- Event Histograms Computer (EHC)
- Event-Rate Control (ERC)
- Event formatting (EDF)

Output Interfaces:
- MIPI
- CMOS Parallel Interface (CPI)
Processing features > Noise Filter (NFL)

• NFL removes events according to instantaneous event rates during a sliding time window.

• Programmable thresholds are used to determined if the NFL is dropping or passing incoming events.

• Thresholds are programmable separately for the two event polarities.
Processing features > Anti-Flicker Filter (AFK)

Natural scenes often contain light sources flickering at constant frequencies: Fluorescent lights, LEDs, screens, ...

AFK requirements:
• Digital band stop (notch) filter
• Programmable center/width
• Frequency Range 50-500Hz
• Invertible (band pass)
**Processing features > AFK Measurements**

**Band stop (notch)**
- Center frequencies: 50Hz, 100Hz, 200Hz
- Bandwidth ±10% of center
- Typical attenuation ~50dB

**Band stop**
- Center frequency: 100Hz
- Bandwidth 100Hz
- Attenuation >50dB

**Inverted (band pass)**
- Center frequencies: 50Hz, 100Hz, 200Hz
- Bandwidths ±2Hz of the center
Processing features > Embedded CPU (RISC-V)

RISC-V RV32IMC ISA

- Interrupt controller, 16 IRQ lines
- Peripherals:
  - 3 x 64-bits Timers; Real Time Clock (1MHz)
  - Mailbox system for host communication
  - JTAG interface, HW breakpoint

- Usage:
  - Boot Initialization (Time to First Data)
  - Power manager (entering/exiting power modes)
  - MIPI meta data insertion (e.g. frame number, data statistics, etc...)
  - (No algo-based event data processing)
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Event Sensor ➔ Compute Platform

**SENSOR**

- Pixel Array
- Event Signal Processor
- I/O

**EVENT**

- Digital crop
- Anti-Flicker
- Spatio-Temporal Contrast Filtering
- Event-rate Control
- Event compression

**COMPUTE**

- **Application Processor**
  - Continuous EVT Stream
  - Embedded vision applications

- **Microcontroller**
  - Direct 2D event frames / histograms with controllable rate and activity thresholds,
  - Smart wake-up
  - e.g. CNN-based edge-AI applications

- **Neuromorphic processor**
  - Direct AER input (spiking NN)
  - AI Algos for object detection and tracking, motion analysis, optical flow, attention tracking, surveillance, gesture recognition, ...
Event streaming

- **EVT2.0**: legacy, low event rate, 32-bit per pixel, uncompressed, timestamp
- **EVT3.0**: compressed vector 16-bit encoding, high event rate, removes redundant event data, best encoding on average
- **EVT2.1**: vectorized, high event rate, 64-bit based
- **AER**: Address Event Representation format, legacy format used by the first event-based sensors. Low latency, no timestamps

<table>
<thead>
<tr>
<th>Data Formats</th>
<th>CPI @50MHz</th>
<th>MIPI @1.5Ghz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 bits Mevt/s</td>
<td>8 bits Mevt/s</td>
</tr>
<tr>
<td>EVT2.1</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>EVT2.0</td>
<td>6.25</td>
<td>12.5</td>
</tr>
<tr>
<td>EVT3.0</td>
<td>133.33</td>
<td>266.67</td>
</tr>
<tr>
<td>AER</td>
<td>10</td>
<td>16.67</td>
</tr>
</tbody>
</table>

Event accumulation

- **Histograms**: Accumulating events into a 2D grid structure
  - fixed time window
  - fixed number of events
- **Accumulation methods**:  
  - **Histo3D**: 2 separate containers per pixel (positive, negative events)  
  - **Diff3D**: 1 container per pixel (add positive, subtract negative events)
Output Data Interfaces

CPI PARALLEL INTERFACE:
- 4/8-bit CMOS parallel
- Up to 400Mbps @50Mhz (267Mevt/s)
- Low-power (PLL not required, clock can be adjusted for different bandwidth reqs)
- Low end-to-end latency (very little buffering)
- Different transmission modes:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Pin config</th>
<th>Output Data format</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSEE-IF</td>
<td>4/8bits</td>
<td>AER, EVTx, Histogram</td>
</tr>
<tr>
<td>DCMI-JPEG</td>
<td>8bits</td>
<td>X, X</td>
</tr>
<tr>
<td>DCMI-Mono</td>
<td>8bits</td>
<td>X, X</td>
</tr>
<tr>
<td>AER-IF</td>
<td>4/8bits</td>
<td>X</td>
</tr>
</tbody>
</table>

MIPI INTERFACE:
- 1 lane, up to 1.5Gbps @1.5Ghz (1Gevt/s)
- Compliant with MIPI D-PHY V1.2
- Supported formats:
  - Compressed/uncompressed vector formats (EVTX.X)
  - Event histograms
- Frame size, frame rate configurable
- Optionally fixed frame size/fixed frame rate with data padding to comply with standard configuration for FB cameras.

Low-power / Low-latency

High Bandwidth
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
**Power Modes - Overview**

- **PM0**: Digital Core is OFF, activity detection based on Analog GCD
- **PM1**: Digital CPU-only is ON, activity detection with advanced CPU processing on GCD data
- **PM2**: Digital Core is partially ON, activity detection with advanced CPU processing on pixel data, no streaming
- **PM3C**: All ON (except PLL), full streaming on CPI interface
- **PM3M**: All ON, full streaming on MIPI interface

---

### Functional Mode

<table>
<thead>
<tr>
<th>Mode</th>
<th>Ultra Low Power passive</th>
<th>Ultra Low Power active</th>
<th>Low Power PM2</th>
<th>CPI streaming PM3C</th>
<th>MIPI streaming PM3M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-system</td>
<td>PM0</td>
<td>PM1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pixel array</td>
<td>3x3 GCD resolution</td>
<td>3x3 GCD resolution</td>
<td>Full resolution</td>
<td>Full resolution</td>
<td>Full resolution</td>
</tr>
<tr>
<td>Digital Registers+CPU</td>
<td>Power off</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
</tr>
<tr>
<td>Digital readout</td>
<td>Power off</td>
<td>powered, clock gated</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
</tr>
<tr>
<td>Digital ESP + Output IF</td>
<td>Power off</td>
<td>powered, clock gated</td>
<td>powered, clock gated</td>
<td>powered, clocked</td>
<td>powered, clocked</td>
</tr>
</tbody>
</table>

### Analog
- PM0: 35µW
- PM1: 35µW
- PM2: 1.2mW
- PM3C: 1.2mW
- PM3M: 1.2mW

### Digital
- PM0: 0 µW
- PM1: 915µW
- PM2: 1.3mW
- PM3C: 2.3mW
- PM3M: 5mW
- MIPI+PLL: 10mW

### Total
- PM0: 35µW
- PM1: 950µW
- PM2: 2.5mW
- PM3C: 3.5mW
- PM3M: 16.2mW
Power Modes and Wake-Up Features

Wake-up on events from 3x3 clusters
- Internal: Power-mode cycling
- External: system wake-up signal on pin

PM2: Activity monitoring in 4x4 programmable zones
- Tuneable thresholds
- CPU statistical processing for wake-up filtering (false positive)
Power Modes > Streaming Modes: MIPI vs CPI

Power consumption - MIPI vs CPI

- MIPI+PLL (Low activity): 16.2 mW (Data Interface), 6.2 mW (Sensor)
- MIPI+PLL (10 Mevt/s): 22.5 mW (Data Interface), 9.3 mW (Sensor)
- CPI (Low activity): 13.2 mW (Data Interface), 5.6 mW (Sensor)
- CPI (10 Mevt/s): 10.1 mW (Data Interface), 5.48 mW (Sensor)

Legend:
- Red: Data Interface
- Blue: Sensor
Outline

• Introduction event-based vision
• Event sensors for edge-AI applications - requirements and design targets
• Design and implementation:
  – Sensor features
  – Processing features
  – Data formats and interfaces
  – Power modes
• Summary / Conclusion
Summary / Conclusion

• Specifically designed for embedded vision at-the-edge
• Cu-Cu Stacked BSI CIS on CMOS (65nm/40nm)
• Features multiple data pre-processing, filtering and formatting functions to adapt to variety of use cases and processing platforms
• Two data interface covering low latency applications (CPI) and high bandwidth applications (MIPI)
• Multiple transmission options in both interfaces ease integration and compliance to industry standards
• Adaptative hierarchical power modes facilitate operability in power-sensitive edge vision applications
• On-chip power management and an embedded uC core further improve sensor flexibility and useability at-the-edge
Thank you for your attention
Copyright Notice

This presentation in this publication was presented as a tinyML® EMEA Innovation Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org