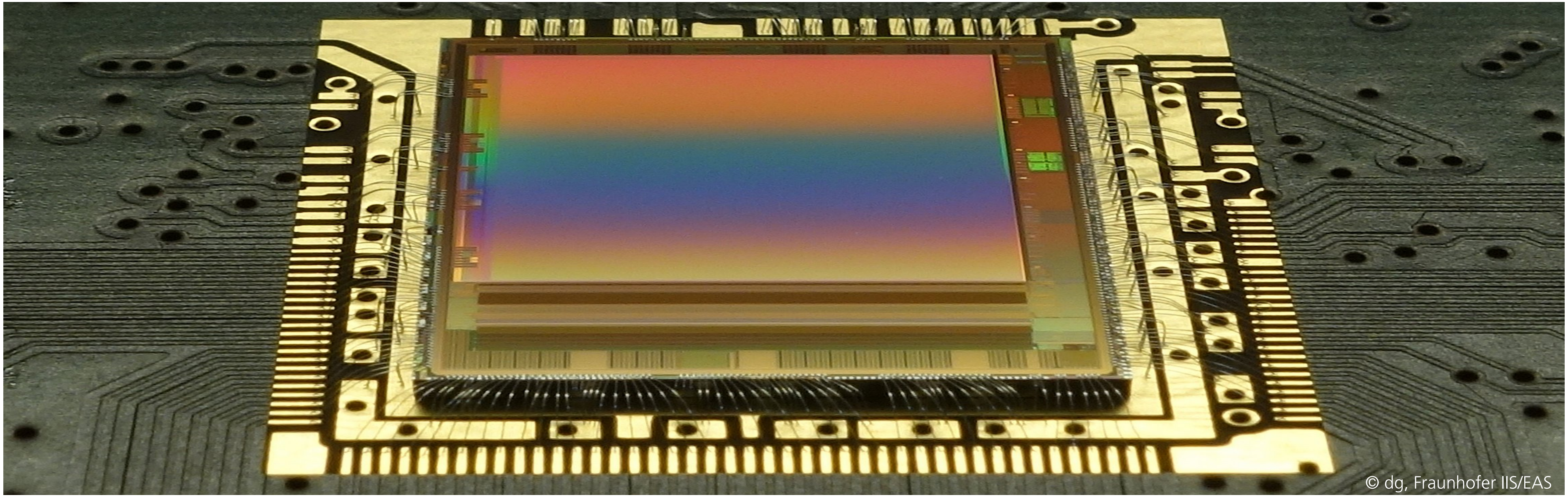
A NOVEL VISION-SYSTEM-ON-CHIP FOR EMBEDDED IMAGE ACQUISITION AND PROCESSING

Dr. Jens Döge, Head of Image Acquisition and Processing Group,

Fraunhofer Institute for Integrated Circuits IIS, Division Engineering of Adaptive Systems EAS



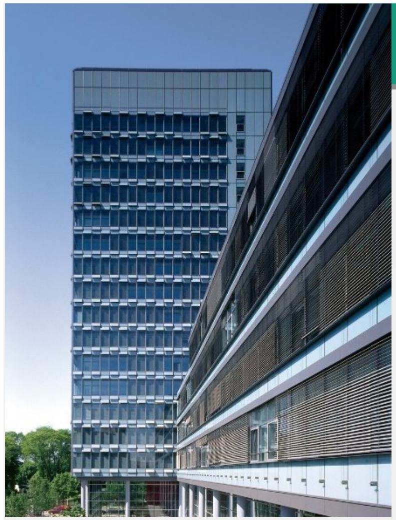
Outline

- Introduction
- Machine Vision Challenges
- Vision System-on-Chip
- Advanced Features
- Application Examples
- Conclusion

Fraunhofer IIS

Introduction

Facts and Figures



Fraunhofer-Gesellschaft

»Applied research of direct utility to private and public enterprises and of wide benefit to society«

- The largest organization dedicated to application-oriented research in Europe
- Important innovations, such as the music format mp3, white LEDs or high-resolution thermal imaging cameras

Fraunhofer Institute for Integrated Circuits

Founded	1985
Employees	more than 900
Budget	approx. 150 million €
Directors	Prof. Dr. Albert Heuberger Dr. Bernhard Grill

Division Engineering of Adaptive Systems

Founded	1992
Employees	approx. 110
Budget	approx. 13.1 million €
Director	Dr. Peter Schneider

Fraunhofer IIS

Introduction
Business Areas & Research Topics Division EAS

Challenges in the Development of Adaptive Systems

- complexity of the systems
- functional safety and reliability
- security and privacy
- variable application scenarios
- human-machine interaction

Design Methodology

- reliability and robustness of ICs
- functional safety

Efficient Electronics

- integrated sensor electronics
- system integration
- optical sensor technologies

Distributed Data Processing and Control

- industrial data analysis
- wireless-networked automation
- energy management

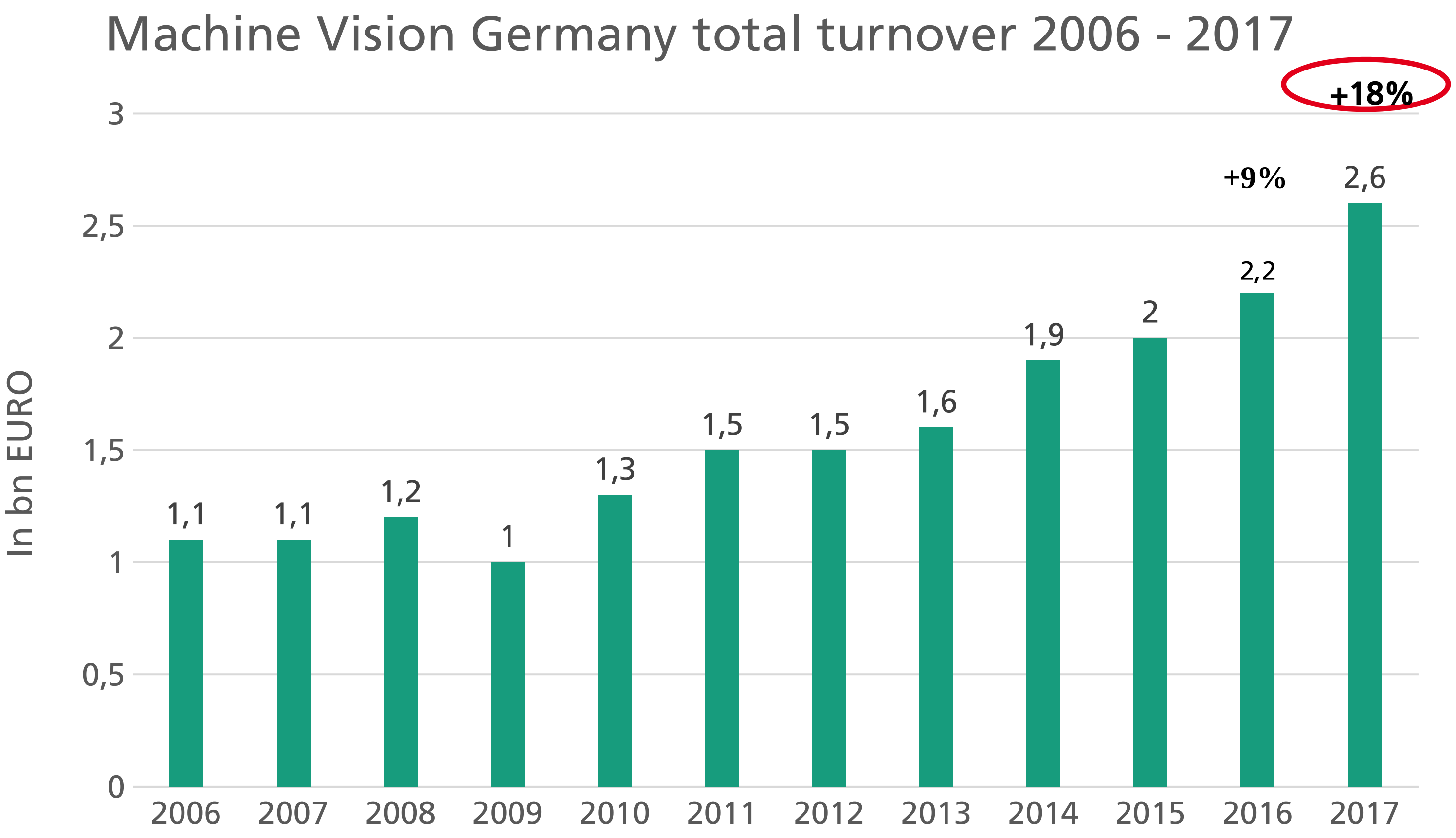
Fraunhofer IIS

Machine Vision

Relevance

2017 – VDMA growth prognosis *(VDMA Fachabteilung Industrielle Bildverarbeitung)*

- German robotic and automation:
  +7 → +11 %
- industrial image processing:
  +9 → +18 % → 2.6 Mrd €



Source: VDMA Robotics + Automation

Fraunhofer IIS

Machine Vision

Challenges and Drivers

- **continuous growth** of image processing industry **– on passing lane**
- **automation** will drive the development
- **automation** is essential to remain competitive
- hunger for **process know-how** and **quality data** for digital production

… and how to **integrate vision** to meet this **needs**? → possible solution: **Vision Systems-on-Chip!**

https://www.computer-automation.de/feldebene/bildverarbeitung/artikel/150487/

Fraunhofer IIS

Vision System-on-Chip
Key Factors for Success

Vision-System-on-Chip must do

image acquisition as good* as and **image processing much better****

than a **traditional image sensor setup!**

*** as good...**

- **image quality** equal to high-speed image sensors for equivalent **application** → **compare** measurement **results**

**** much better...**

- speed → **continuous frame rate** including processing and
- latency → for real-time **processing** of captured images
- at lower **total power consumption** for equivalent tasks **including processing** and
- higher **flexibility** at lower **total development costs**.

→ This concept leads to a multi-purpose »Software-Defined Smart Camera«

Fraunhofer IIS

Vision System-on-Chip
Application Principles (1)

First and foremost: **Don't generate data you don't need!**

- focal plane pre-processing
- A/D conversion at a viable resolution
  - 1 bit binarization for high speed
  - 10 bit only where really necessary (image data)
- software-defined A/D converter
  - column-wise flexible programmable thresholds
  - reprogrammable algorithms (single slope, multiple slope, SAR)

Second: **What do we gain, if we give up a certain »correctness / precision / beauty«?** → **modeling and system simulations**

- slope converter → (very simple) SAR
  - much higher speed,
  - but missing codes under some circumstances
- compare measurement results, not intermediate images

Fraunhofer IIS

Vision System-on-Chip

Application Principles (2) – Additional Considerations

- optimizations if possible in software
  - calibration
  - error correction

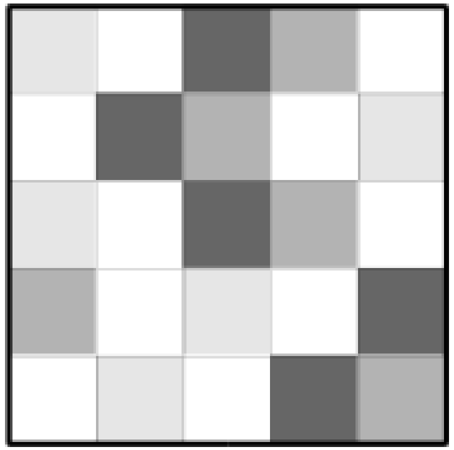
- VSoC-based solutions, if you definitely can't do them traditionally
  - speed
  - latency
  - flexibility

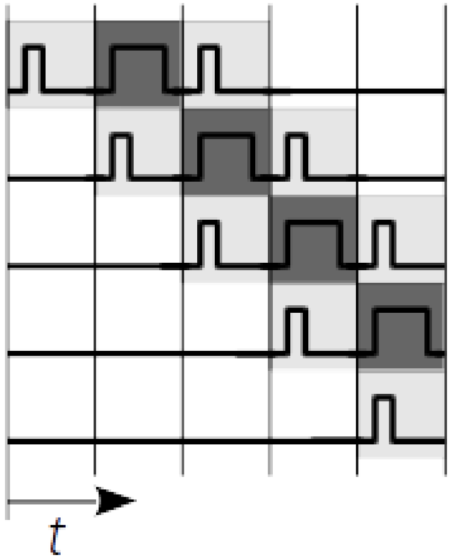
→ because **VSoC implementations** may be **challenging to implement**

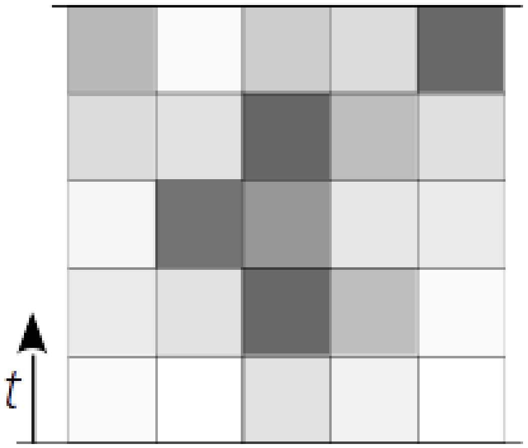
Vision System-on-Chip

Charge-Based Focal Plane Processing

- readout
  - line by line
  - windowing
- weighting
  - amplification
  - attenuation
  - local adaptation
- 1D convolution
  - low-pass filtering
  - contrast enhancement
  - edge detection





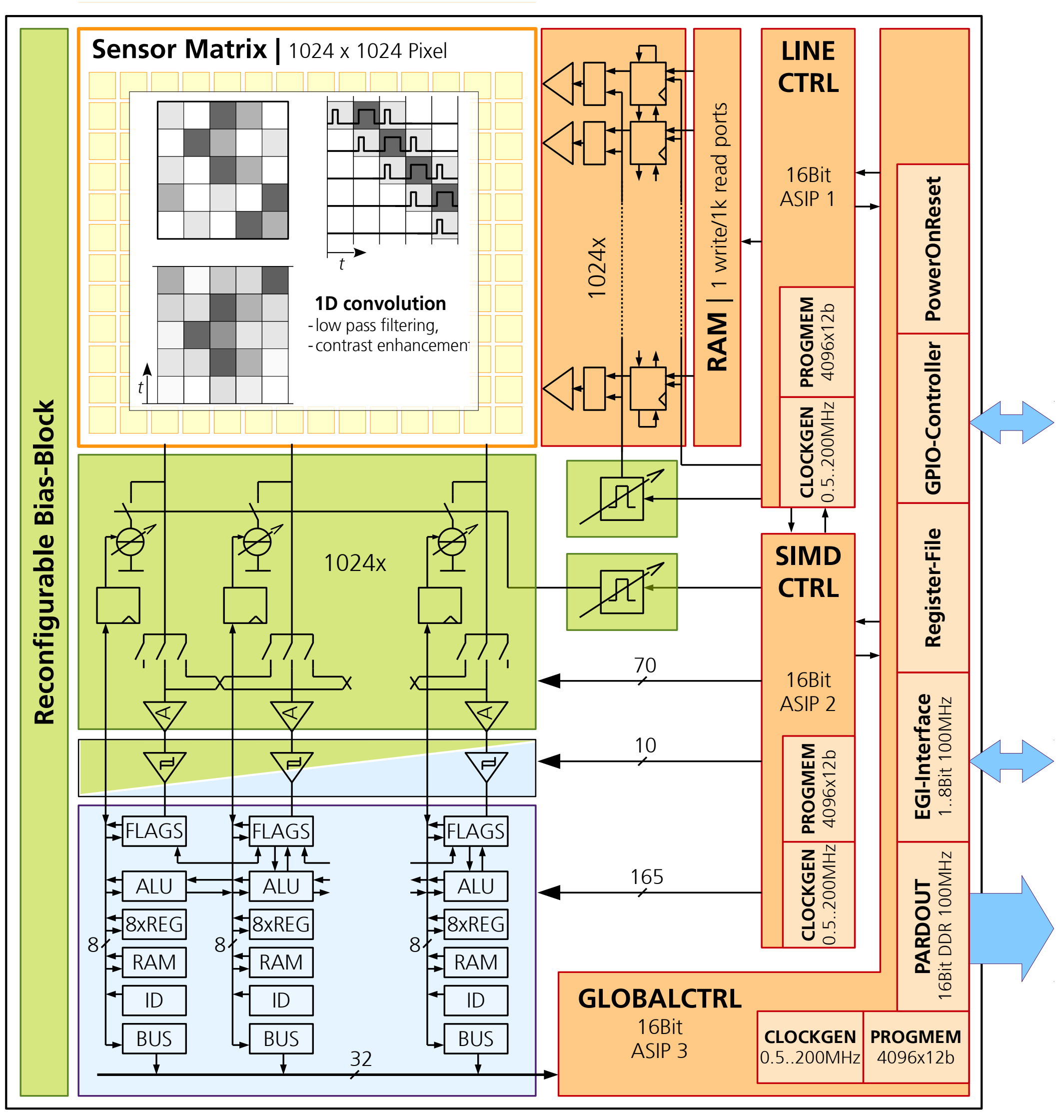


Fraunhofer IIS

Vision System-on-Chip

Basic Architecture and Features

- sensor matrix with 1024 x 1024 pixels
  - 8.75µm pixel pitch
  - linear / logarithmic (HDR) characteristics (with FET pixels)
- LINECTRL
  - pixel control
  - charge-based readout and processing
- SIMDCTRL
  - column-parallel analog frontend
  - column-parallel digital (SIMD) processing elements
- GLOBALCTRL
  - I/O – operation with various digital interfaces (Parallel, SPI and GP-I/O)
- infrastructure - reconfigurable bias block
- software and processing libraries
  - snapshot, rolling shutter, multi-RoI access
  - column-parallel analog filtering
  - column-parallel A/D-conversion (1...10 Bit)
  - digital feature extraction (corners, HoG, LBP)



Fraunhofer IIS

Vision System-on-Chip

Software Programming based on Python with Inline Assembler

```python
# Demo 4: Threshold Image
# apply a digital threshold value

# python setup
from chrispy.chrisp2.cores import simd
from chrispy.macro import CConstInt
from libcam2 import Cam21Platform as         \
Cam2Platform
from libch2dev import skeletons
from libch2dev.image_lpl8 import ImageLPL8
from libch2dev.vsoc_chrisp2 import           \
VSoCChRISP2

import cv2
# skeleton setup
skel = skeletons.SkelCam2GlobalShutterTime()
skel.add_roi(8, 32, 32, 960, 960)

# VSoC setup
platform =                                   \
Cam2Platfo                2.168.1.100")

vsoc = VSoCChRISP2(platform)

skel.configure_regfile(vsoc)

vsoc.set_clock("simd", 100)
vsoc.set_clock("lctrl", 100)
vsoc.set_clock("glb", 35)
```

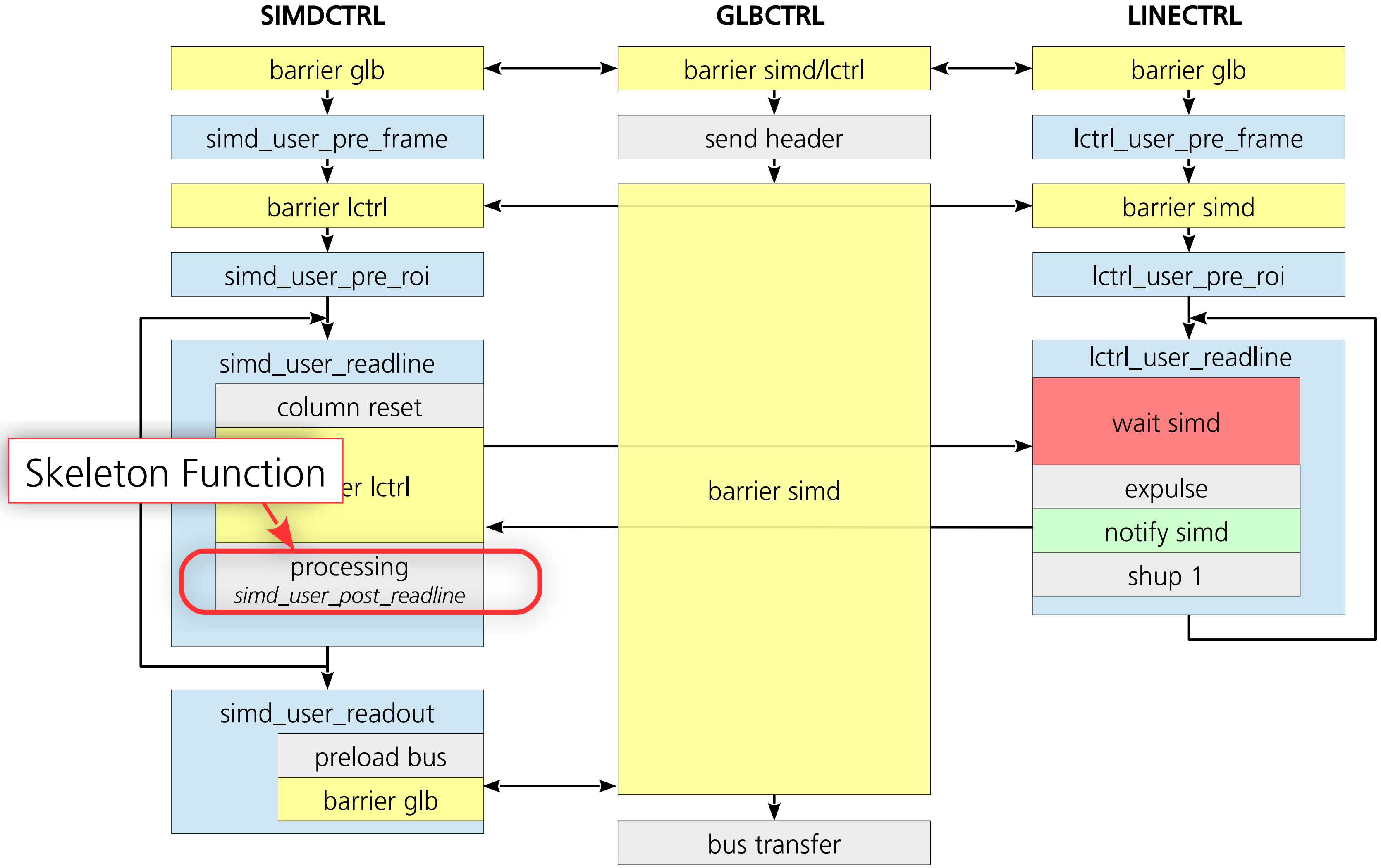
Skeleton

```python
# VSoC code: threshold algorithm
@simd.macro()
def simd_demo(roi_idx : CConstInt, roi_cnt : \
CConstInt):
    simd.api.inline_asm(["simd_mov r_0, r_1",
                         "ld 20",
                         "simd_cp_tos r_0",
                         "simd_movf flg_m0, flg_c",
                         "simd_xnor r_0, r_0",
                         "simd_selact ACT0",
                         "simd_xor r_0, r_0",
                         "simd_selact ACTIVE",
                         "simd_mov r_1, r_0"])
skel.reg_slot("simd_user_post_readline", simd_demo)

# program initialization
skel.compile_and_start(vsoc)

# host code: setup and main loop
platform.acq.mode.set_exposure_time(30000)
platform.acq.mode.set_frame_period(1e6/10)
while True:
    try:
        desc = platform.b_get_image()
        header, img = ImageLPL8(desc).get_roi(0)
        cv2.imshow("disp", img)
        cv2.waitKey(1)
    except:
        continue
```

Skeleton Function

Vision System-on-Chip

Vision Processing Library based on Skeletons

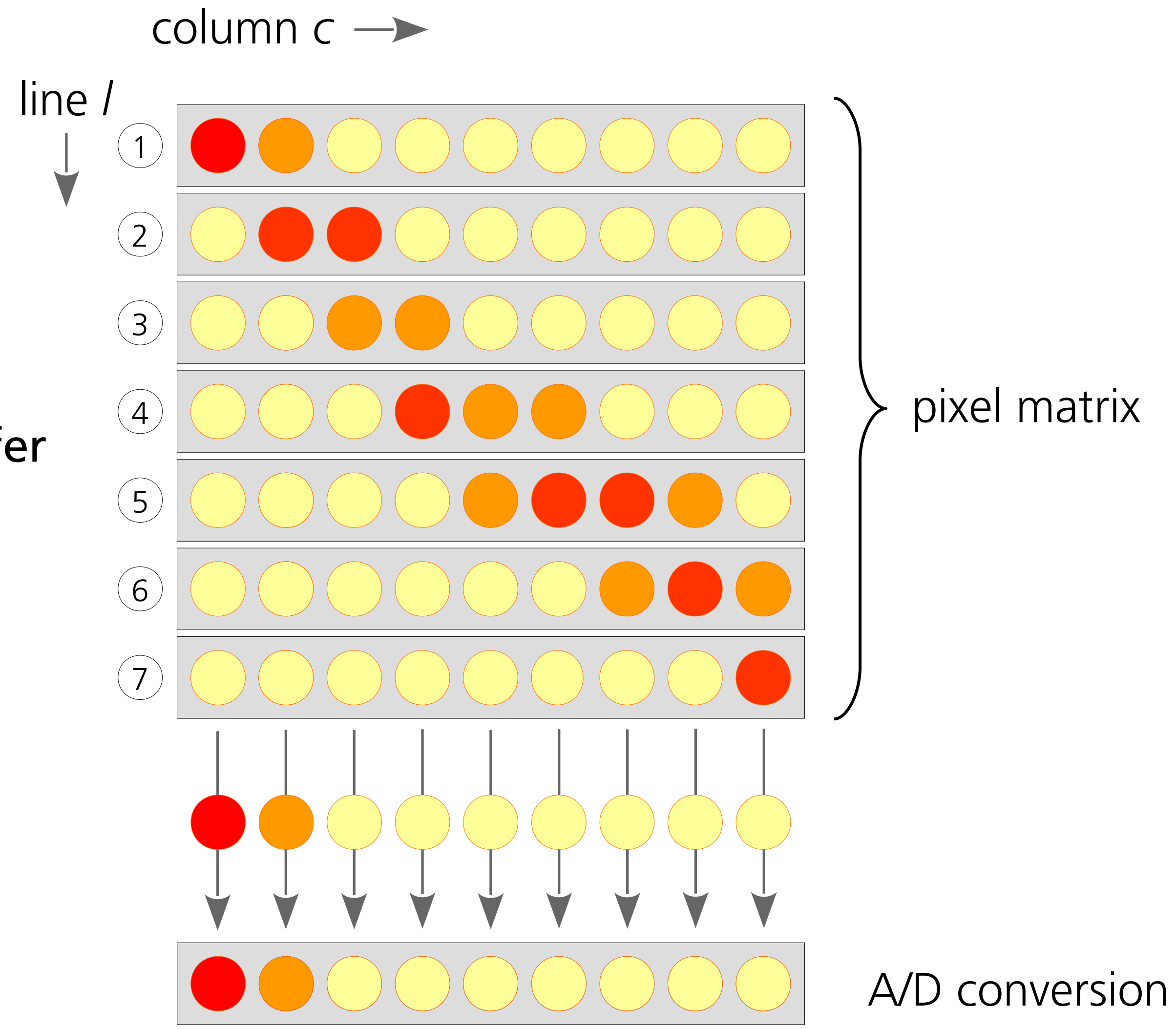


Fraunhofer IIS

Advanced Features

Scenario 1: Sparse Pixel Data

- pixel matrix with **sparse data in *L* lines**
  - pixel values above threshold
- example
  - laser line with orthogonal intensity profile
- sequential transfer of pixel values to readout buffer
  - transfer time: T_T
- sequential A/D conversion of each line
  - conversion time: T_C
- all operations for each pixel line
  - total readout time: $T_{tot} = L\ (T_T + T_C)$
  - total conversion power consumption für C columns: $P_{tot} = L\ P_C$

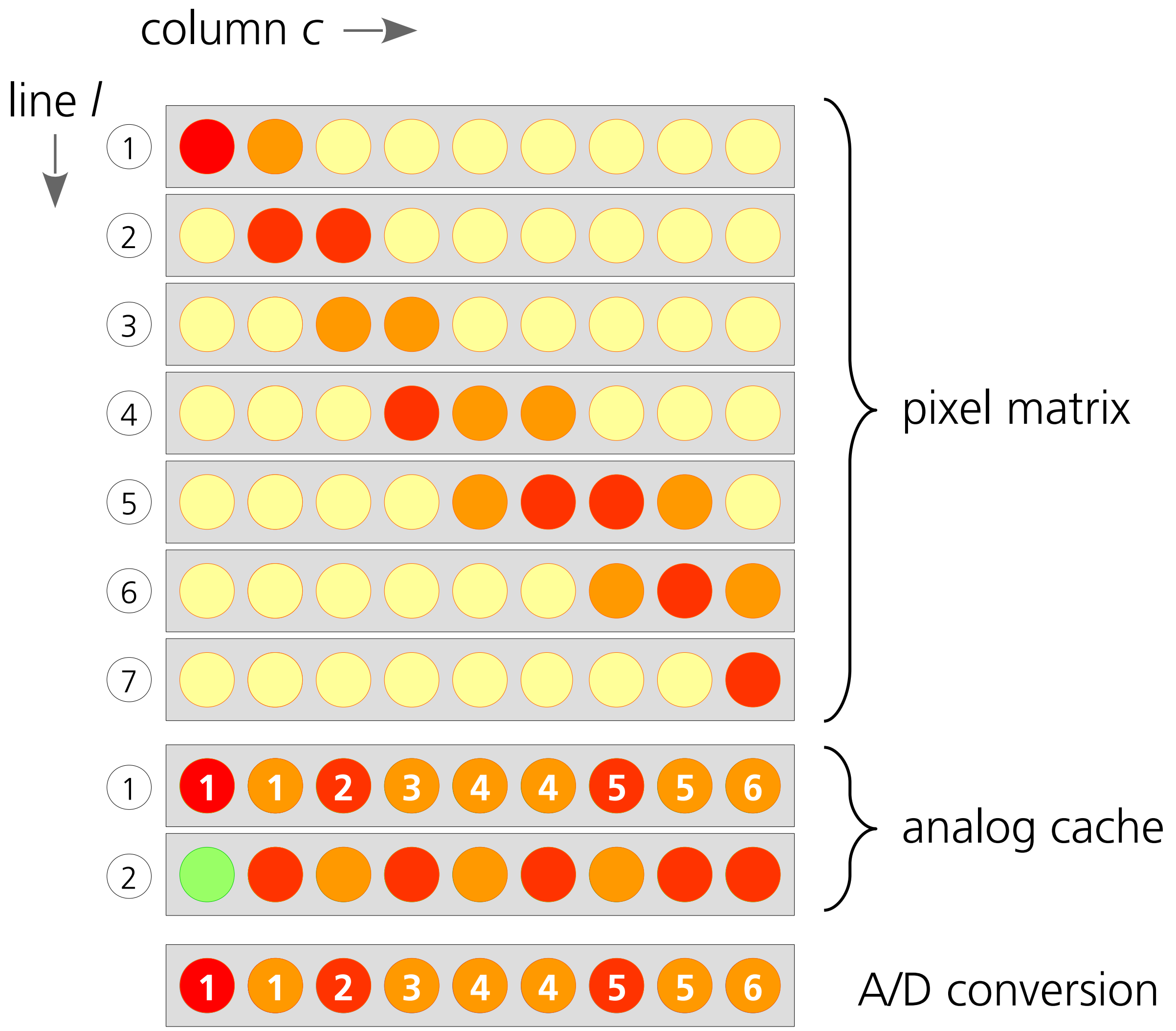


Fraunhofer IIS

Advanced Features

Compacting Pixel Readout

- column-specific **analog cache memory**
  - storage of *M* intermediate results
- fast pixel scan mode
  - readout of pixel values or convolution results
  - with hidden copy to cache memory
  - **high-speed data analysis** (e.g. compare)
  - **write access control based on results**
- high-precision readout mode
  - readout of cache memory
  - A/D conversion with 1…10 Bit
  - **high-precision data analysis**
- A/D conversion only for cache content
  - $T_{tot} = L\ T_T + M\ T_C$ & $P_{tot} = M\ P_C$

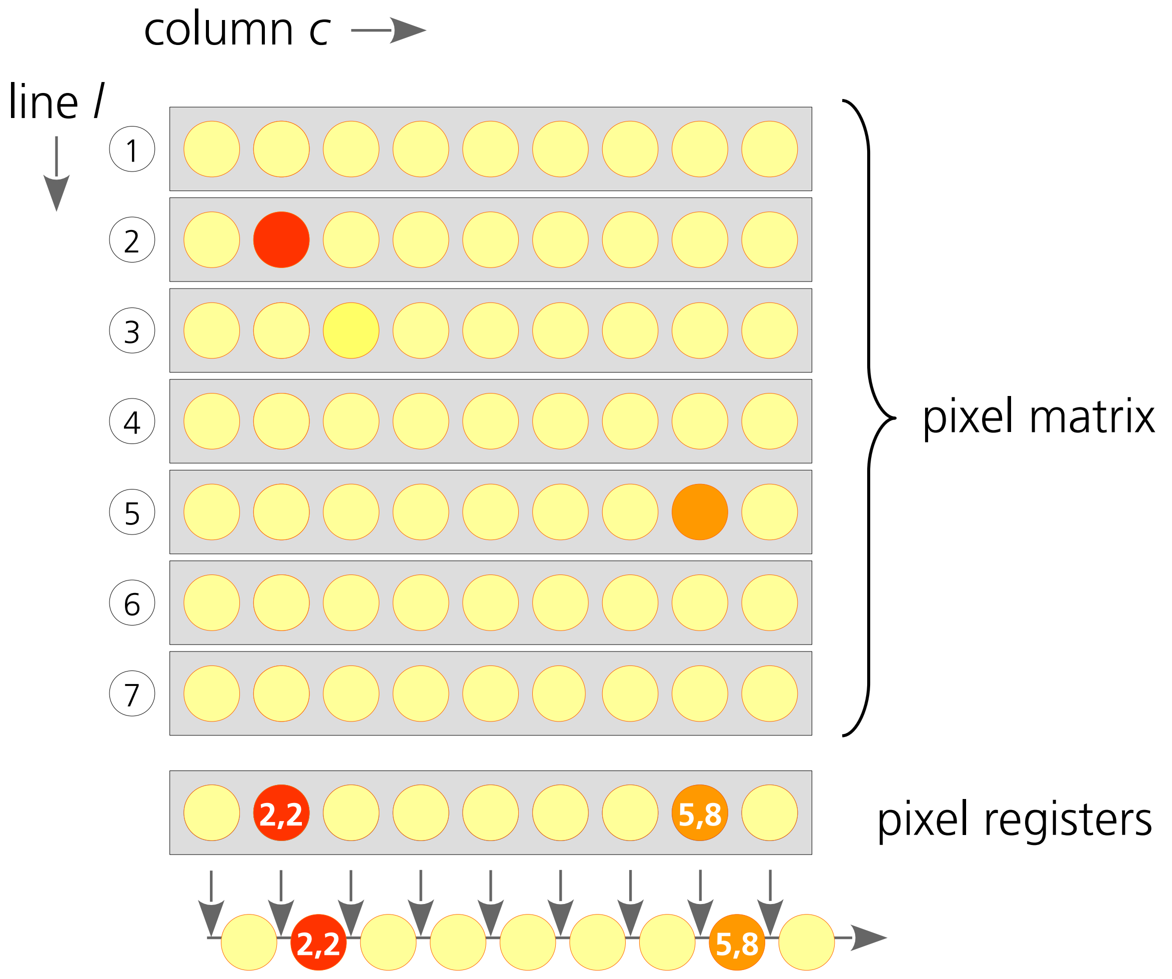


Fraunhofer IIS

Advanced Features

Scenario 2: Sparse Column Data

- pixel matrix with **sparse data in *N* pixels**
  - discrete »active« pixels
- Examples
  - interference modulation of image stack
  - pixel-based triangulation (e.g. vibrometry)
- sequential readout of pixel values
  - transfer time per data set: T_R
- total readout time: $T_{R,\,tot} = C\ T_R$

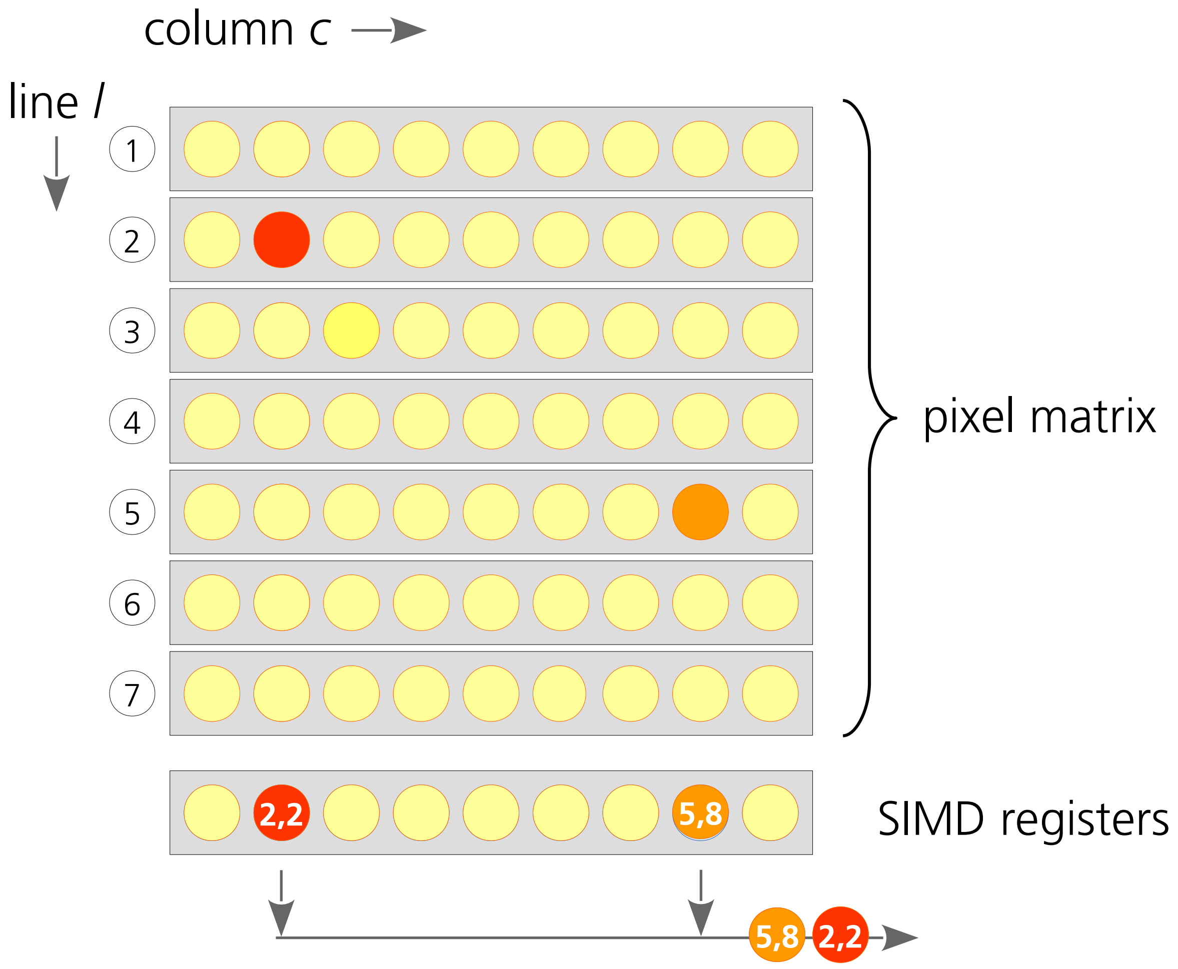


Fraunhofer IIS

Advanced Features

Compacting Asynchronous SIMD Readout

- **asynchronous readout of active elements**
  - activity controlled by each SIMD-element
  - compacting factor depending on number of active elements
- total readout time: $T_{R,\,tot} = C\ T_R$



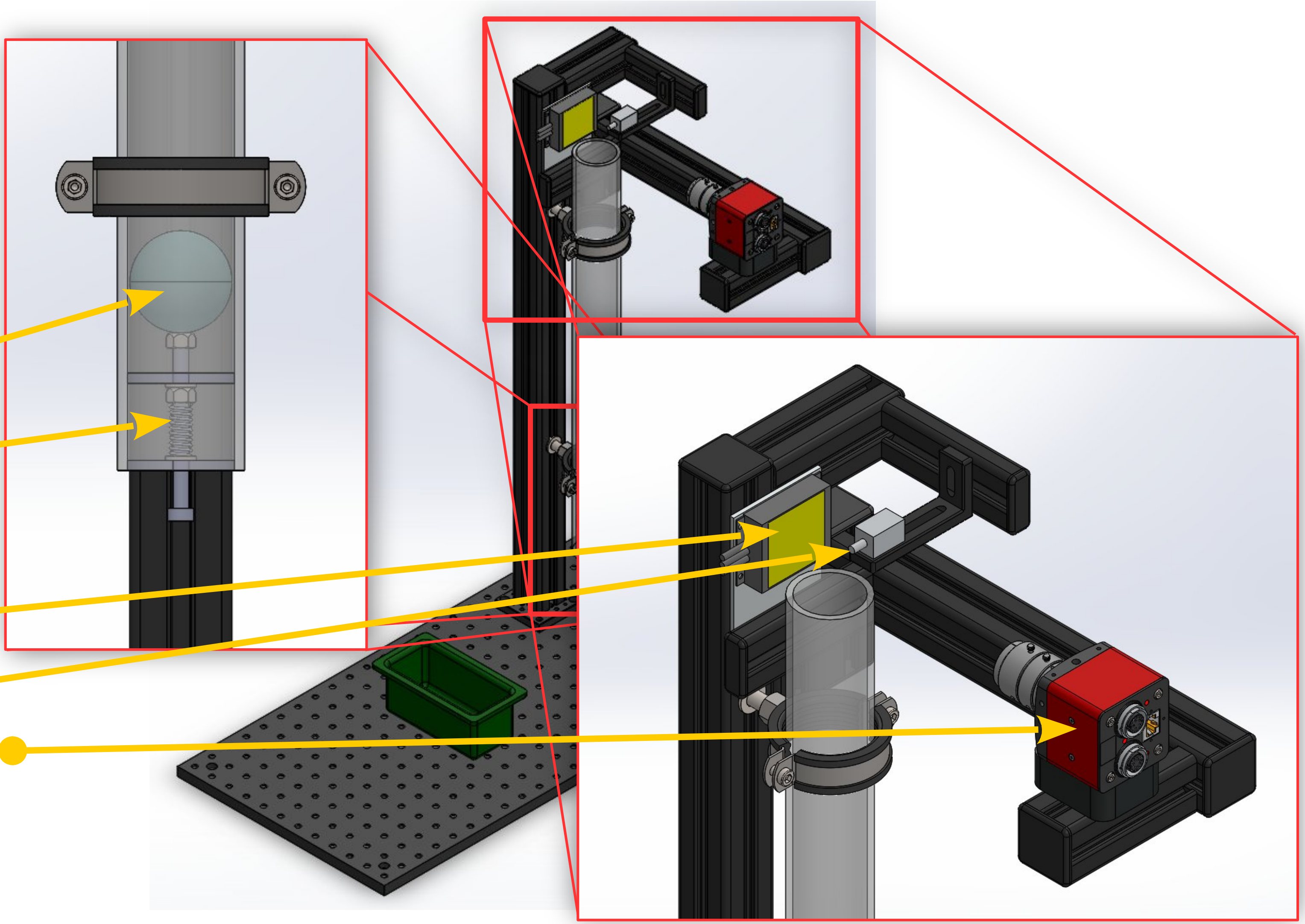
Fraunhofer IIS

Application Examples (1)

Low-Latency Processing Cycle

»Hit a freely **flying ball**
with the right **impulse**
at the right **moment**
so that it falls **into a vessel**«

- mechanical setup with
  - a ball in a pipe with
  - a spring-loaded plunger at the bottom
- sensor-actuator setup with
  - an area light,
  - a solenoid at the pipe exit and
  - a VSoC Camera for control

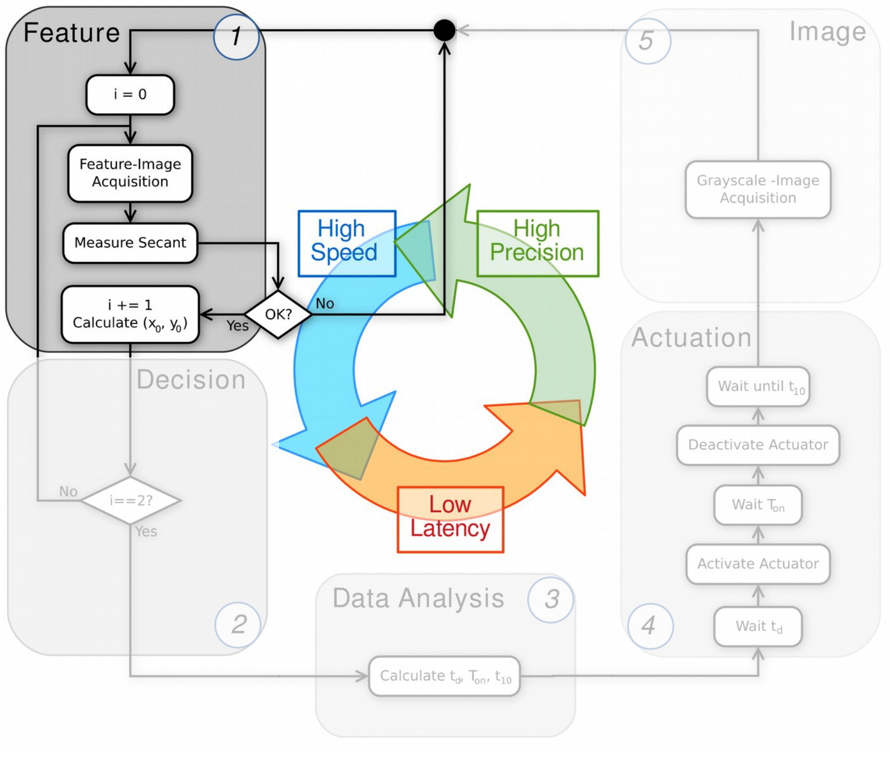


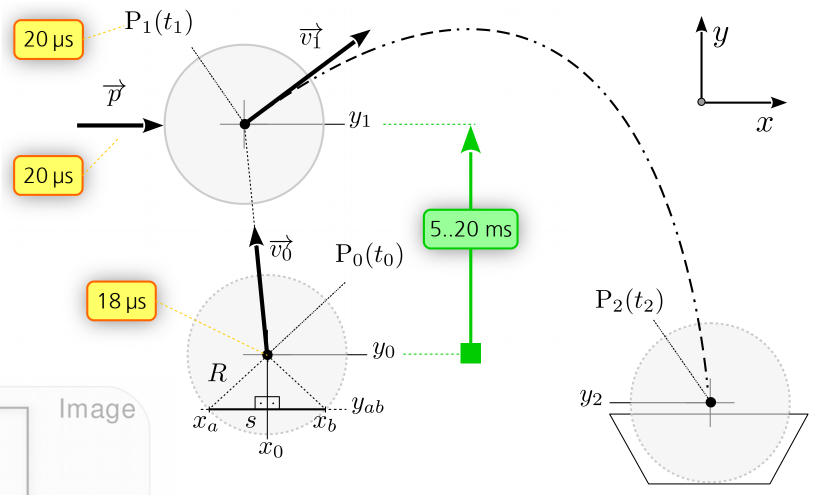
Fraunhofer IIS

Application Examples (1)

Generic Processing Cycle

- high-speed feature extraction (5 kHz) (1)
  - object edge detection
  - position calculation
- low-latency analysis and decision making
  - trigger on data validity (2)
  - object trajectory prediction (3)
  - actuator control (4)
- high-precision measurement data acquisition
  - image acquisition at flyby (5)
  - data acquisition for process analysis
  - object position & speed
  - actuator position and speed



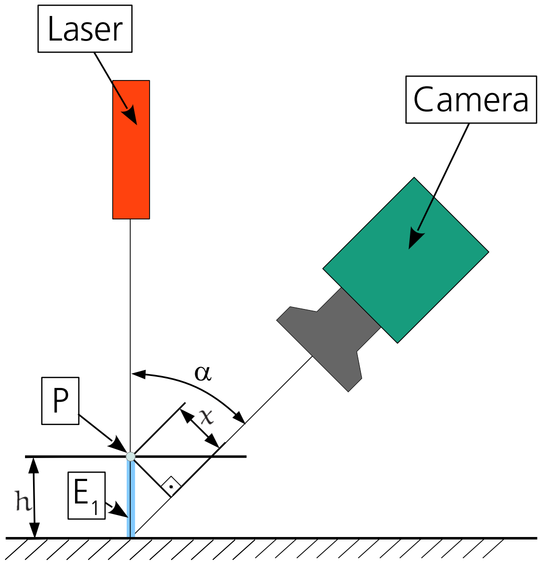


Fraunhofer IIS

Application Examples (2)
Sheet of Light

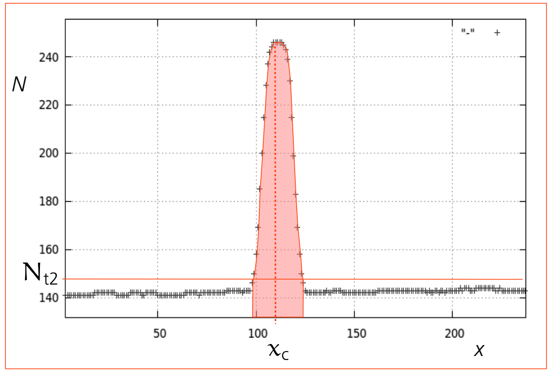
First and most important: **don't generate data you don't need**

- 1D filtering before A/D conversion
  - size of convolution kernel depending on laser line
  - ADC resolution depending on speed (in software)
  - only output laser line position



Second: **what do we gain, if we give up a certain »correctness / precision / beauty«?**

- no need to convert all speckles if we can filter beforehand
- much higher speed
- lin / log processing for non-cooperative surfaces → not a real CoG possible

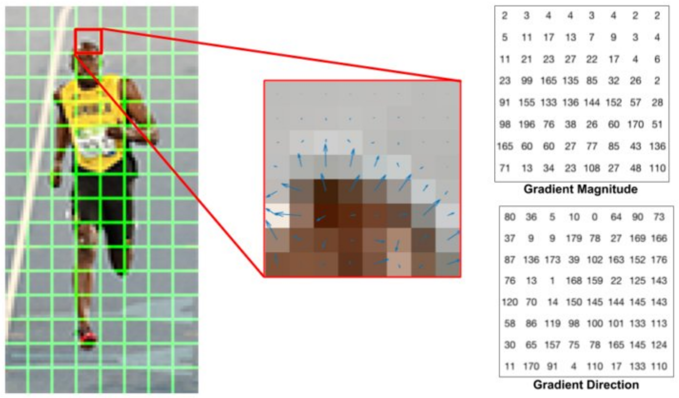


→ very high flexibility for profile post-processing / closed-loop control

Fraunhofer IIS

Application Examples (3)

Histogram of Oriented Gradients (HoG)



Source: *http://www.learnopencv.com/histogram-of-oriented-gradients/*

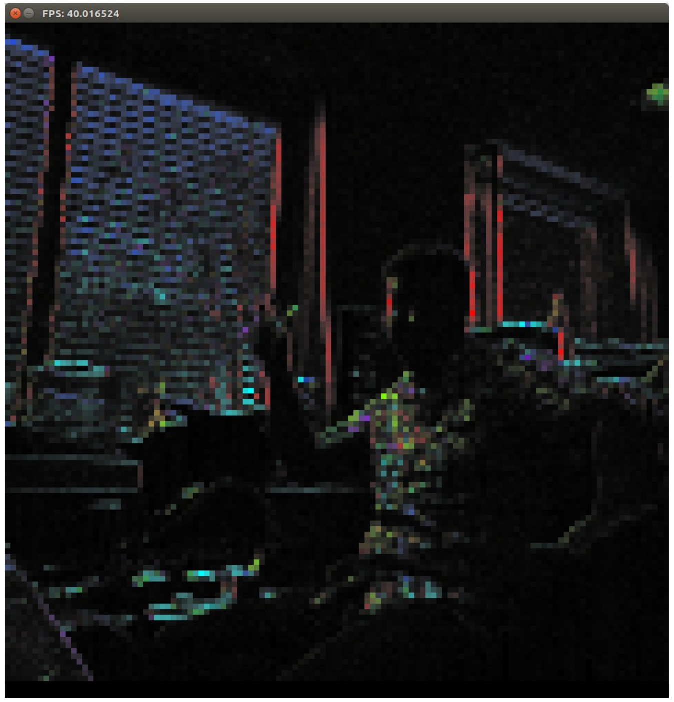
General Description

- paper by N. Dalal und P. Triggs (2005)
- gradient → absolute value + angle → histogram
- post-processing with SVM or other methods

Application Details

- example with 1 MPix / 8x8 pixel per region
  - OpenCV on Intel Core I7: ≈ *7 Hz / 150W*
  - skeleton on VSoC: approx. *40 Hz / 200mW*
- detection and localization of people

→ **highly flexible feature extraction / privacy by design**



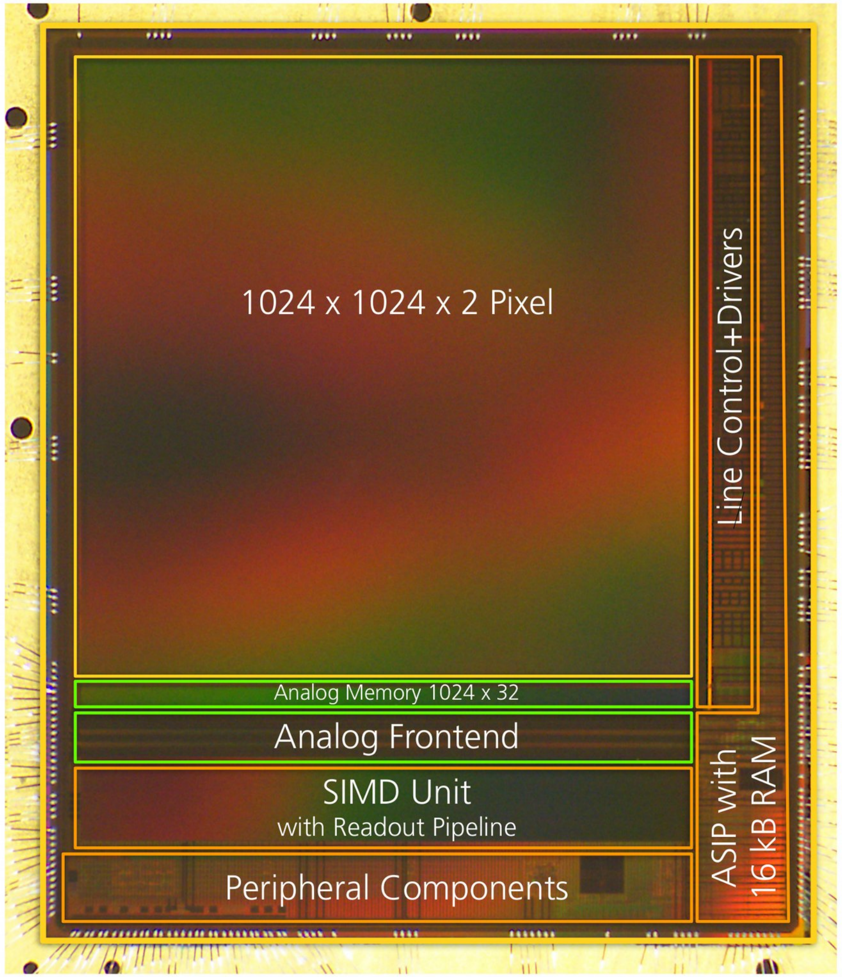
Conclusion

Our **Vision System-on-Chip** architecture

- has been developed based on essential customer needs,
- has unique features regarding low latency and dynamic range
- is extremely flexible for a large variety of special applications

now its time

- to **realize applications** that benefit from our VSoC approach and
- to **enable our customers** to bring this **leading edge technology** to **their markets.**



SPONSORED BY THE
Federal Ministry of Education and Research

The development was supported by the German Federal Ministry of Education and Research (BMBF) within the information and communication technology (IKT) research initiative for SME (KMU Innovativ), joint project “SmartFusionCam”, grant number 01IS15030B; the Innovation Initiative "Entrepreneurial Regions", project consortium 3Dsensation, project cSoC-3D, grant number 03ZZ0427E and others.

Fraunhofer IIS

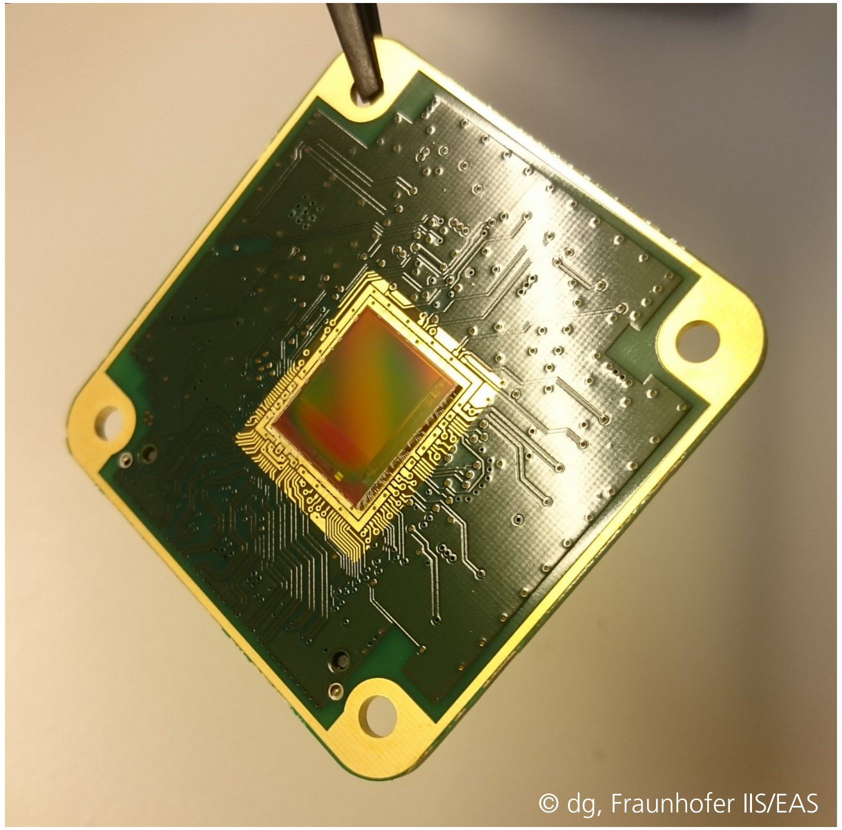
Your Benefits

- planning and realization of ambitious **optical measurement, testing** and **control tasks** for
  - **quality assurance** via **automated full testing** and
  - **quality improvement** through **optical process control**

- **customer-specific software-defined smart camera systems** for
  - the **flexible and efficient** implementation of
  - image acquisition and processing methods

- **sensor module** based on advanced **Vision-System-on-Chip**

- optimization of **total system costs** and **product life cycle**



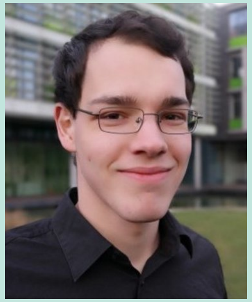
See you on Hall 1 / boot G42

Fraunhofer IIS

Contact

Dr. Jens Döge
Group Manager
Image Acquisition and Processing
Fraunhofer IIS/EAS

✉ Jens.Doege@eas.iis.fraunhofer.de
☎ +49 351 4640-831



Ludger Irsig
Product Management
Image Acquisition and Processing
Fraunhofer IIS/EAS

✉ Ludger.Irsig@eas.iis.fraunhofer.de
☎ +49 351 4640-848

Fraunhofer Institute for Integrated Circuits IIS
Engineering of Adaptive Systems EAS
Zeunerstraße 38
01069 Dresden, Germany

www.eas.iis.fraunhofer.de

Fraunhofer
IIS