A reference design for cost-effective visual-sensor-network nodes

Boštjan Murovec, Janez Perš, Rok Mandeljc, Vildana Sulić, Stanislav Kovačič

University of Ljubljana
Faculty of Electrical Engineering
http://vision.fe.uni-lj.si

Workshop on Architecture of Smart Camera, 5th-6th April 2012, Clermont-Ferrand, France

Team introduction

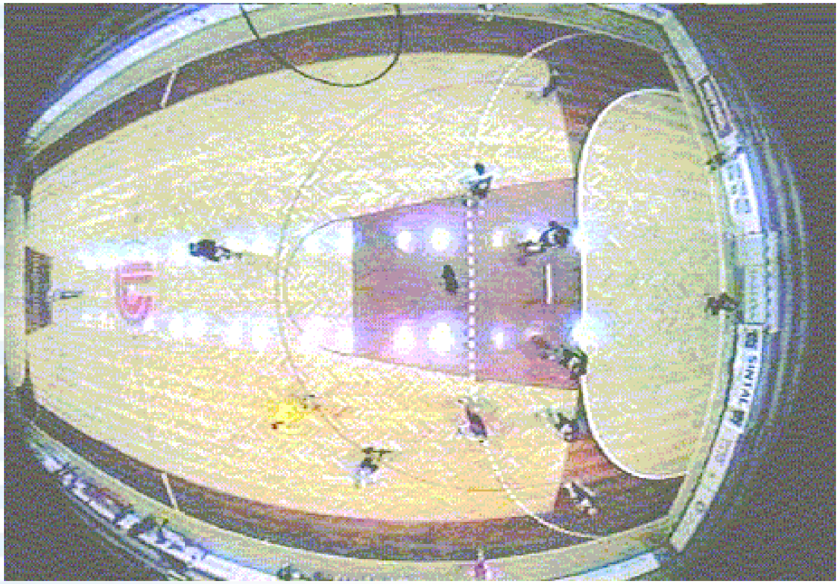
prof. dr. Stanislav Kovačič
head of the laboratory

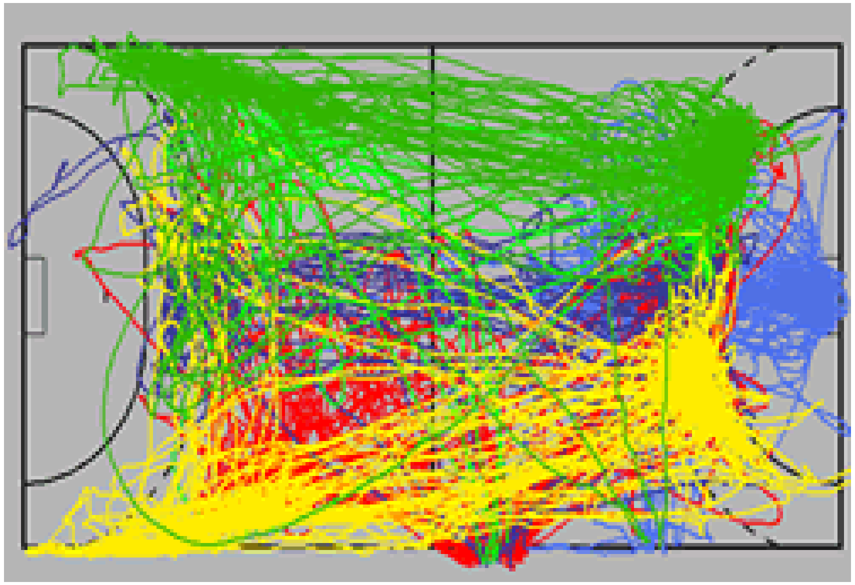
assistant professors
dr. Janez Perš
dr. Matej Kristan
dr. Boštjan Murovec

researcher dr. Vildana Sulić
junior researcher Rok Mandeljc

Tracking in sport

M. Kristan et.al. Sys., Man, and Cyber. December 2010.
M. Kristan et.al. Computer Vision and Image Understanding, 2009.
M. Kristan et.al. Pattern Recognition, 2009.
M. Perše et.al. Pattern Recognition, 2009.
M. Perše et.al. Computer Vision and Image Understanding, March 2009.
J. Perš et.al. Human Movement Science, July 2002.
G. Vučkovič et.al. European journal of sport science, March 2010.
G. Vučkovič et.al. Journal of Sports Sciences, June 2009.



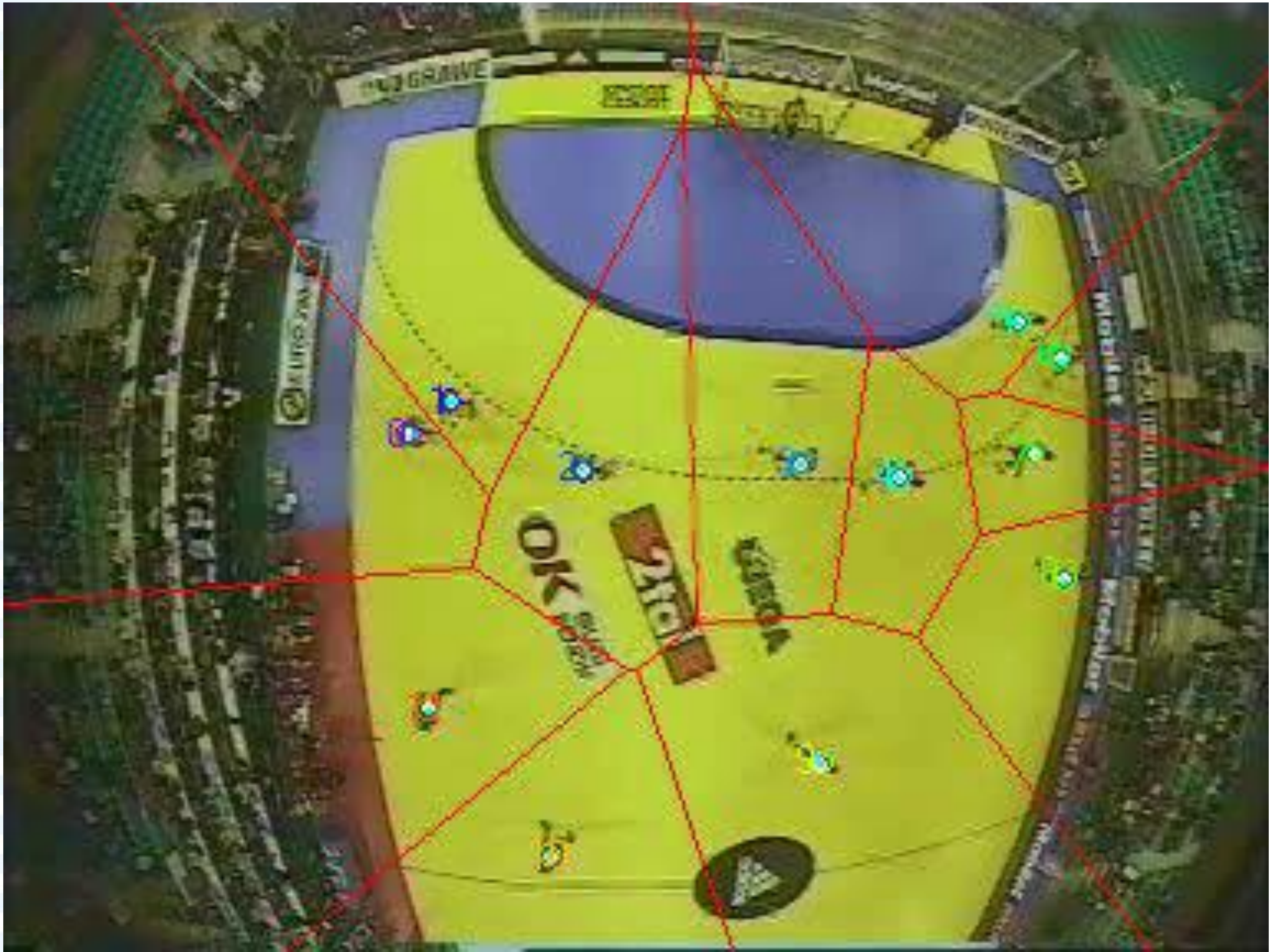


Tracking demos (1/3)

- **Basketball multi-object tracking (static cameras)**

Tracking demos (2/3)

- **Handball multi-object tracking**

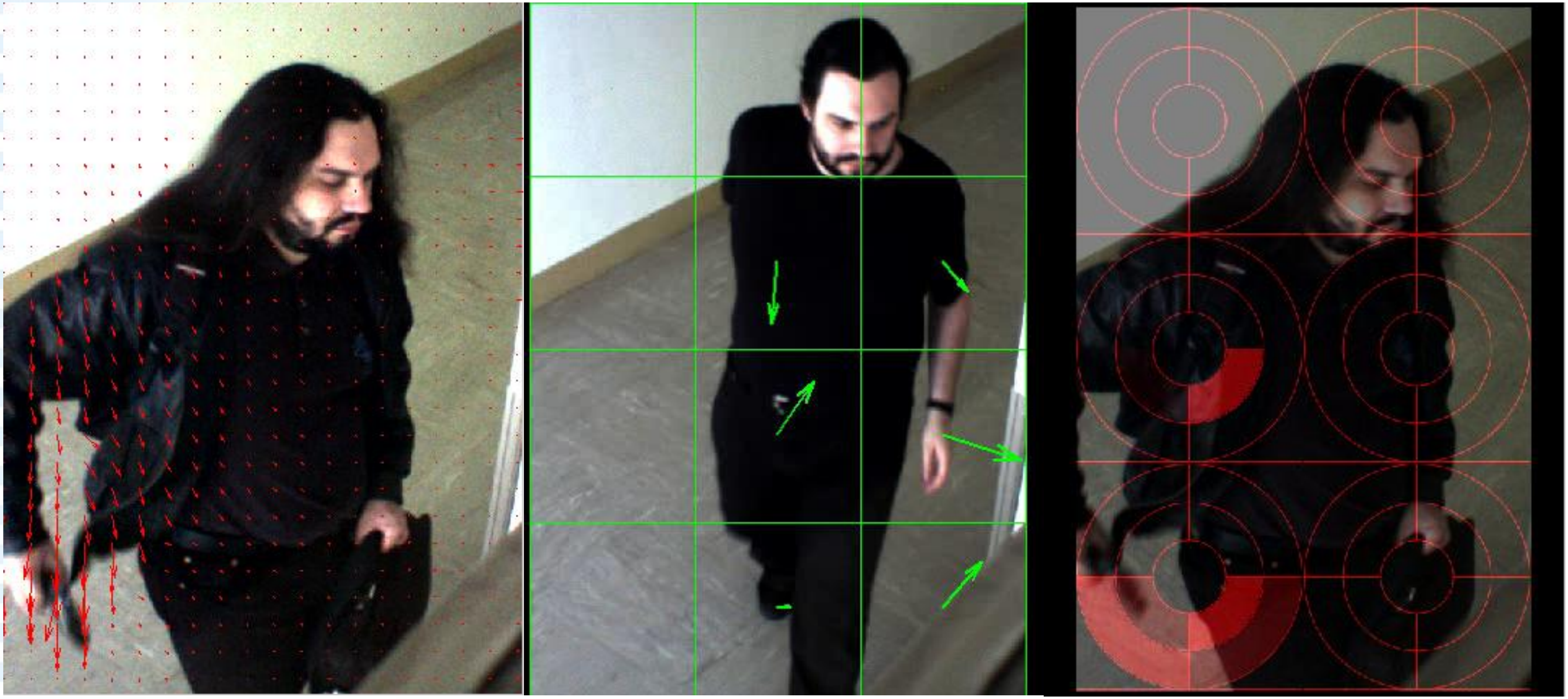


Tracking demos (3/3)

- Handball single-object tracking (sideways view)

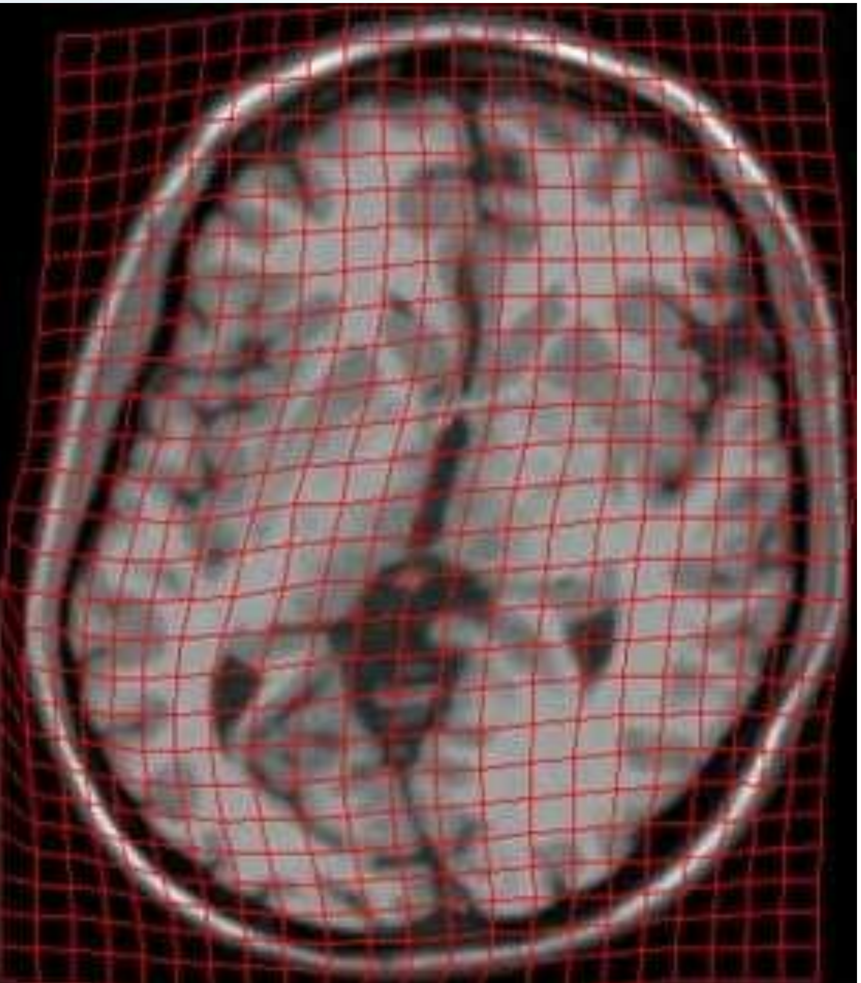
Human motion analysis

J. Perš et.al. Pattern Recognition Letters, 2010.



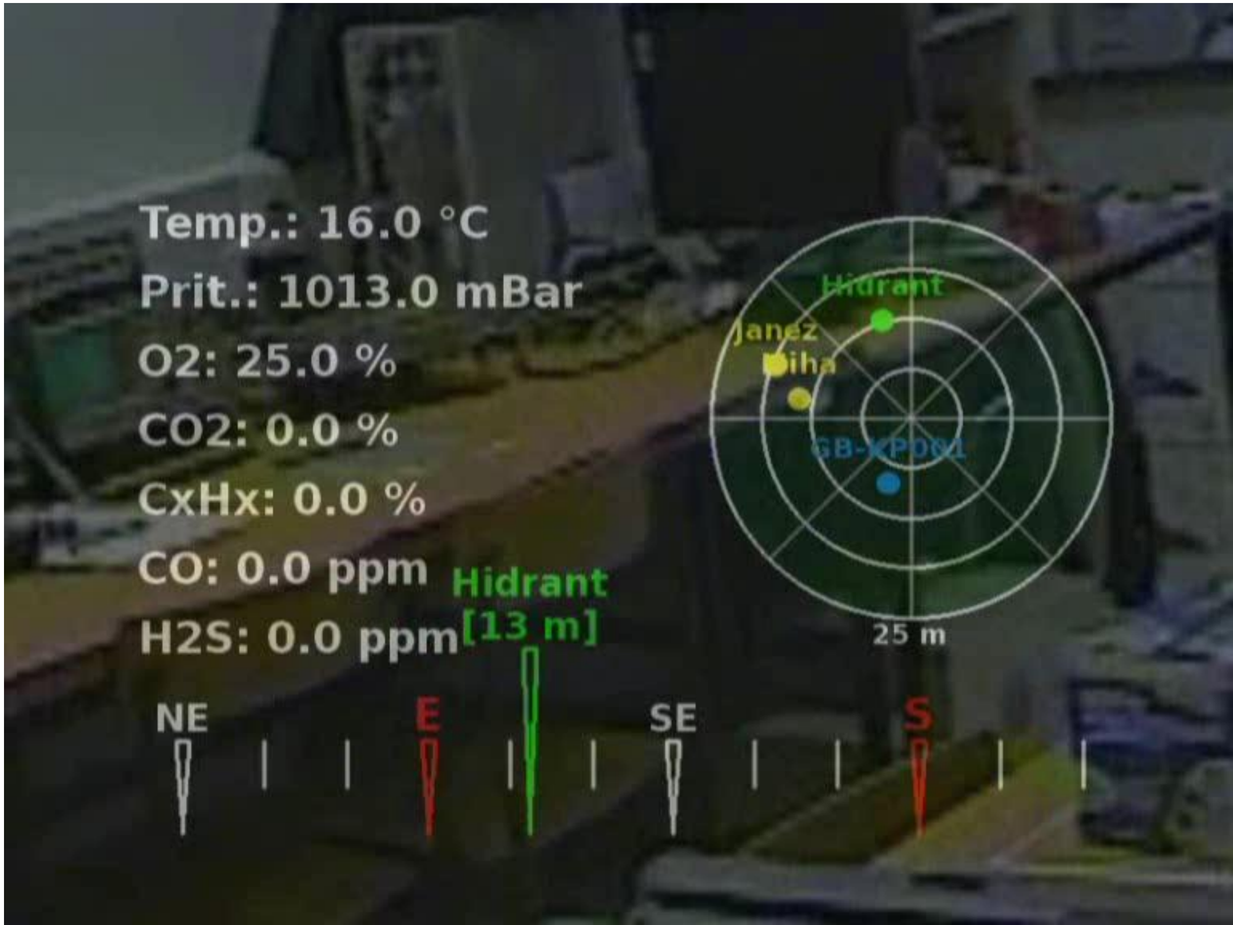
Medical image processing

A. Jarc et.al. Journal of Digital Imaging, 2010.
P. Rogelj et.al. Medical Image Analysis, 2006.
P. Rogelj et.al. Computer Vision and Image Understanding, 2003.



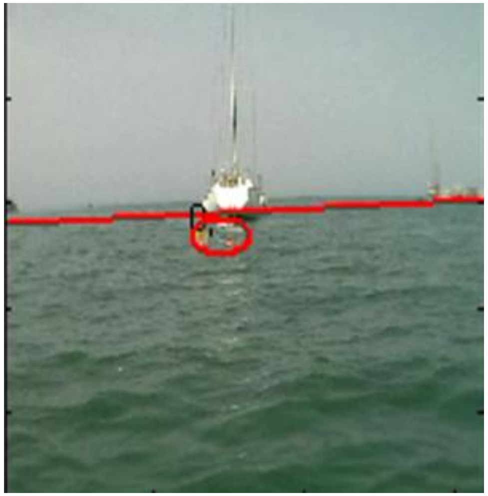
Firefighter support system

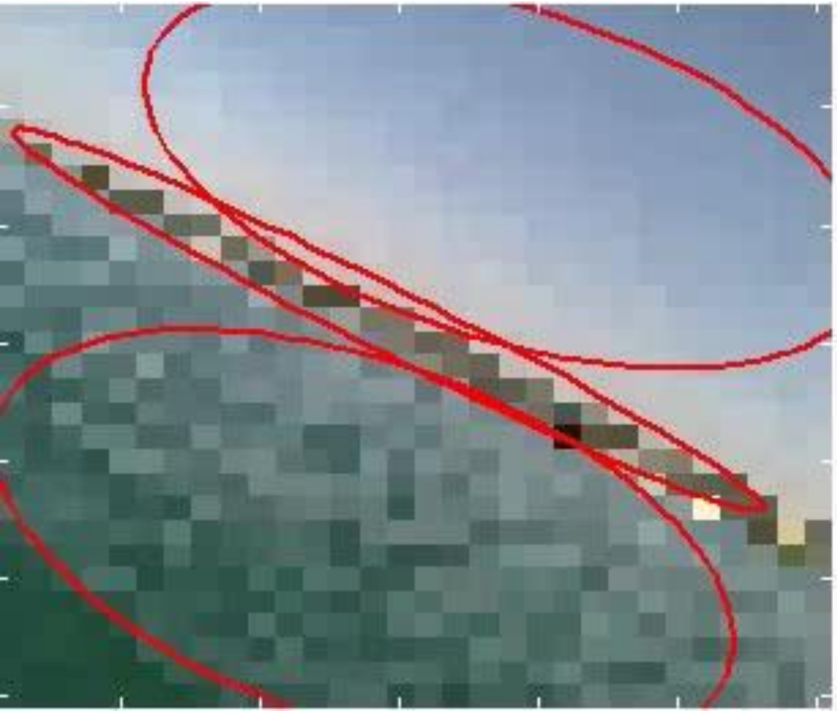
- thermal camera, see-through display, image processing
- environmental sensors, communications and telemetry



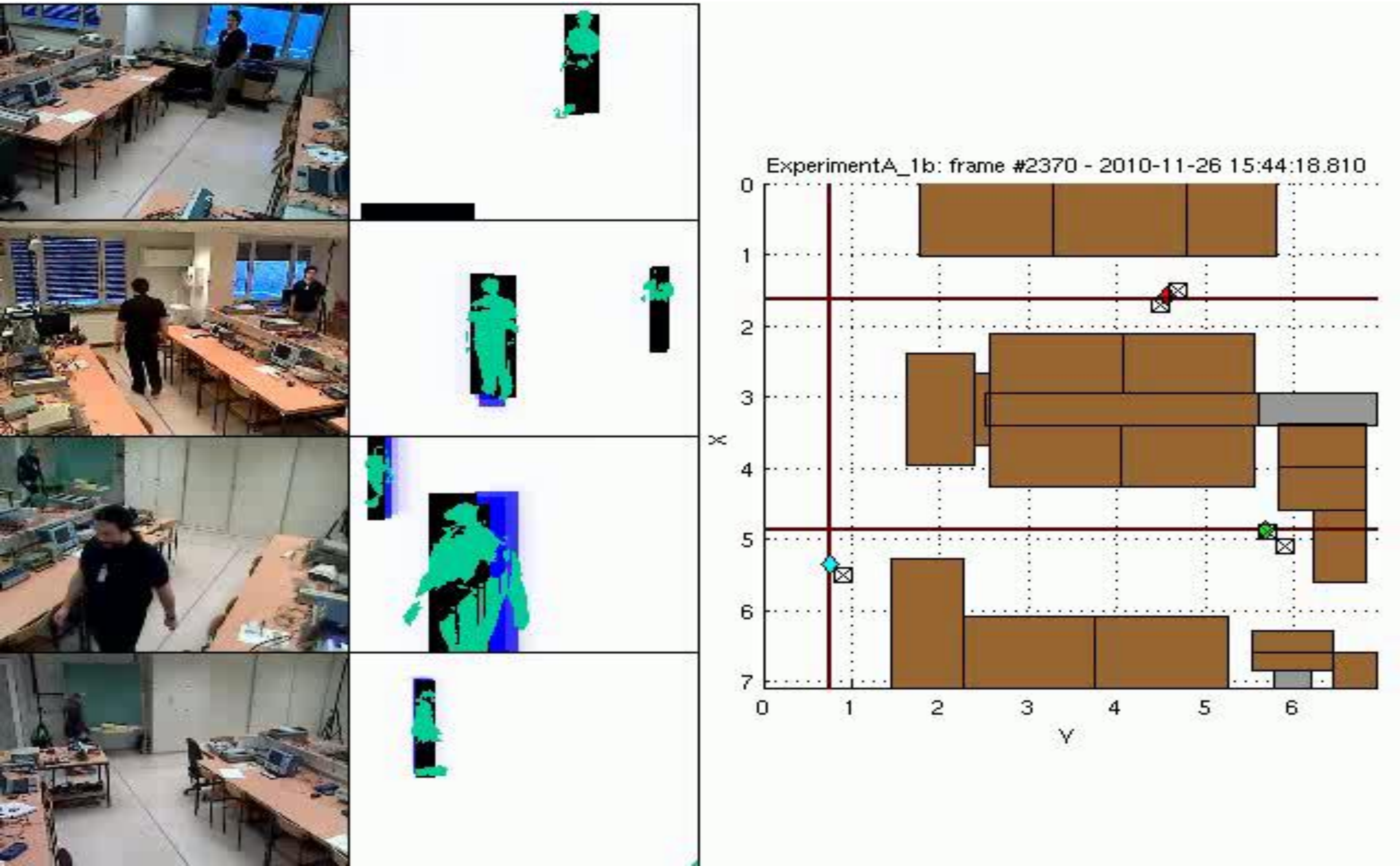
Autonomous vessel control

obstacle detection

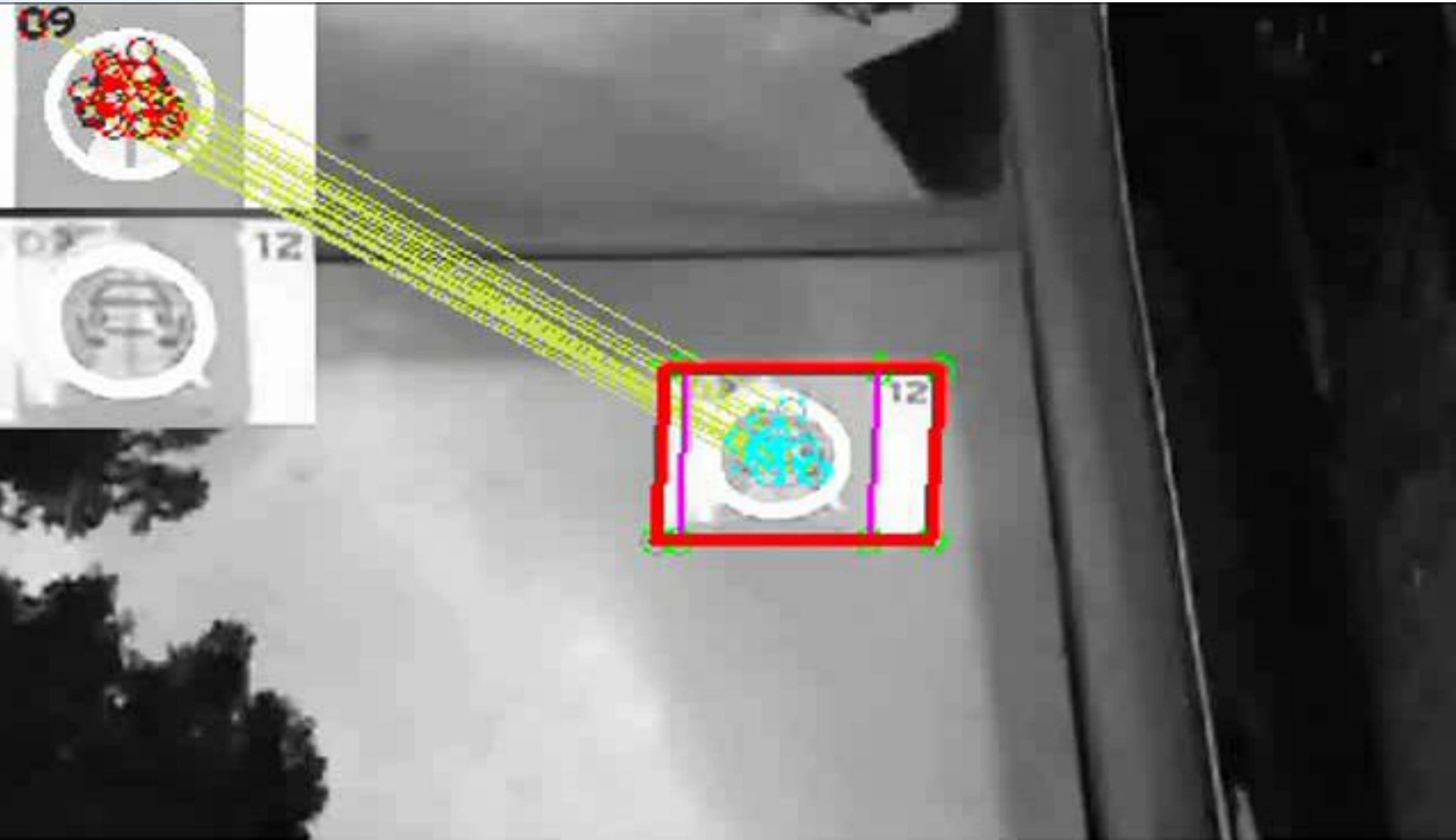




Sensor fusion – POM + UWB radio

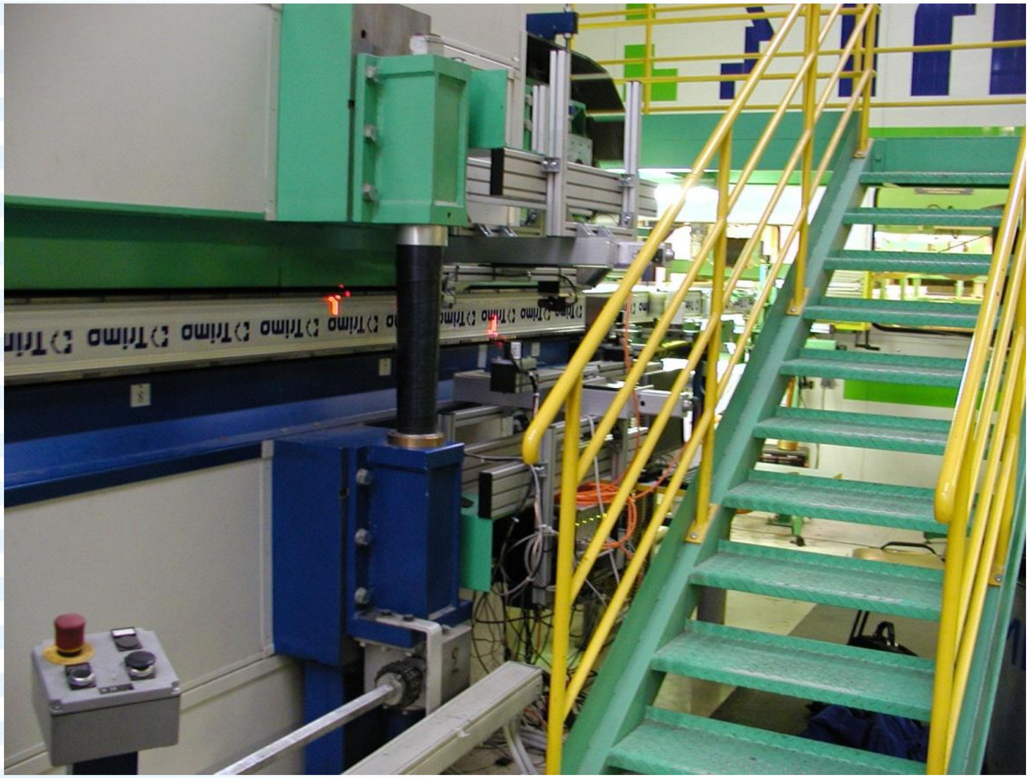


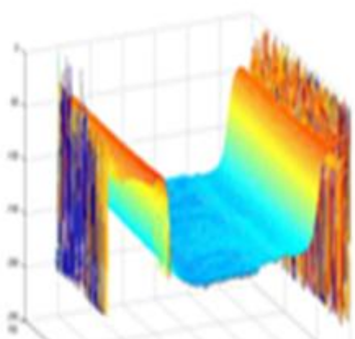
Highway licence sticker

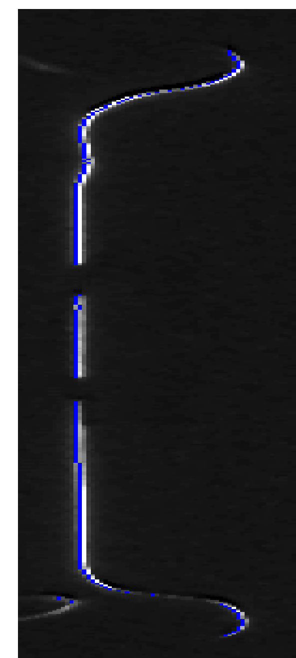


Industrial measurements

Profiles inspection

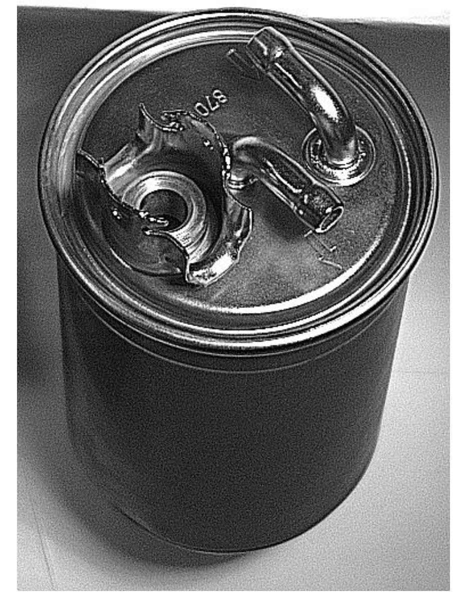


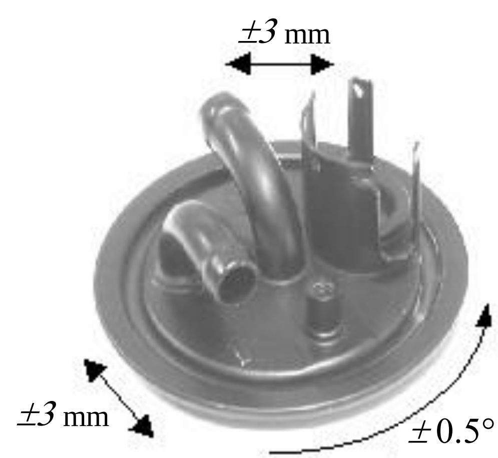




Oil filter inspection

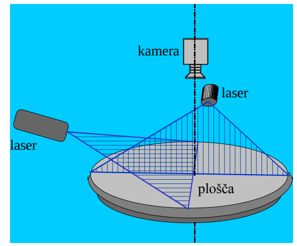
F. Lahajnar et.al. Int. j. adv. manufacturing technology, 2003.

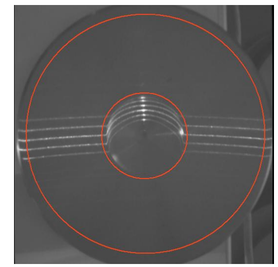


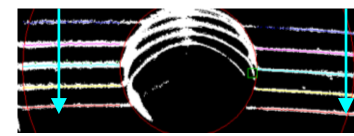


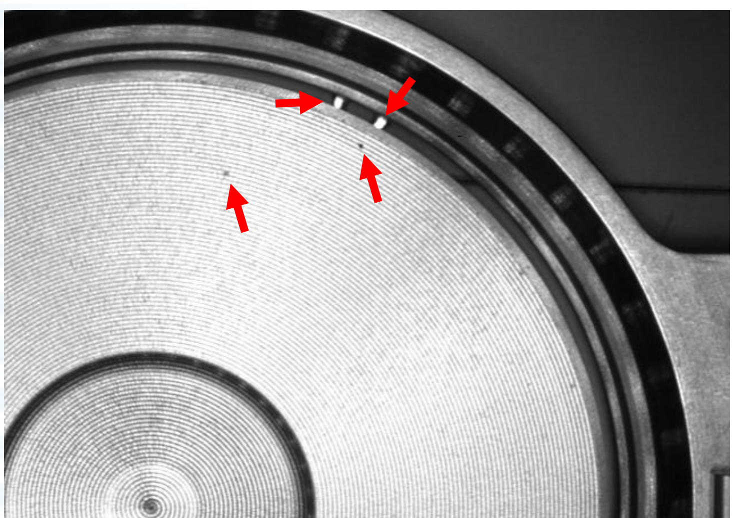
Cooking plates inspection

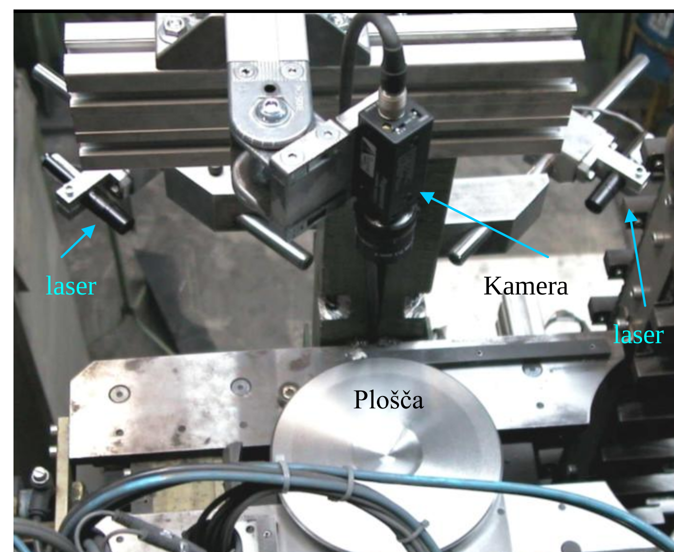
F. Lahajnar. Machine vis. sys. for inspection and metrology, 1998.









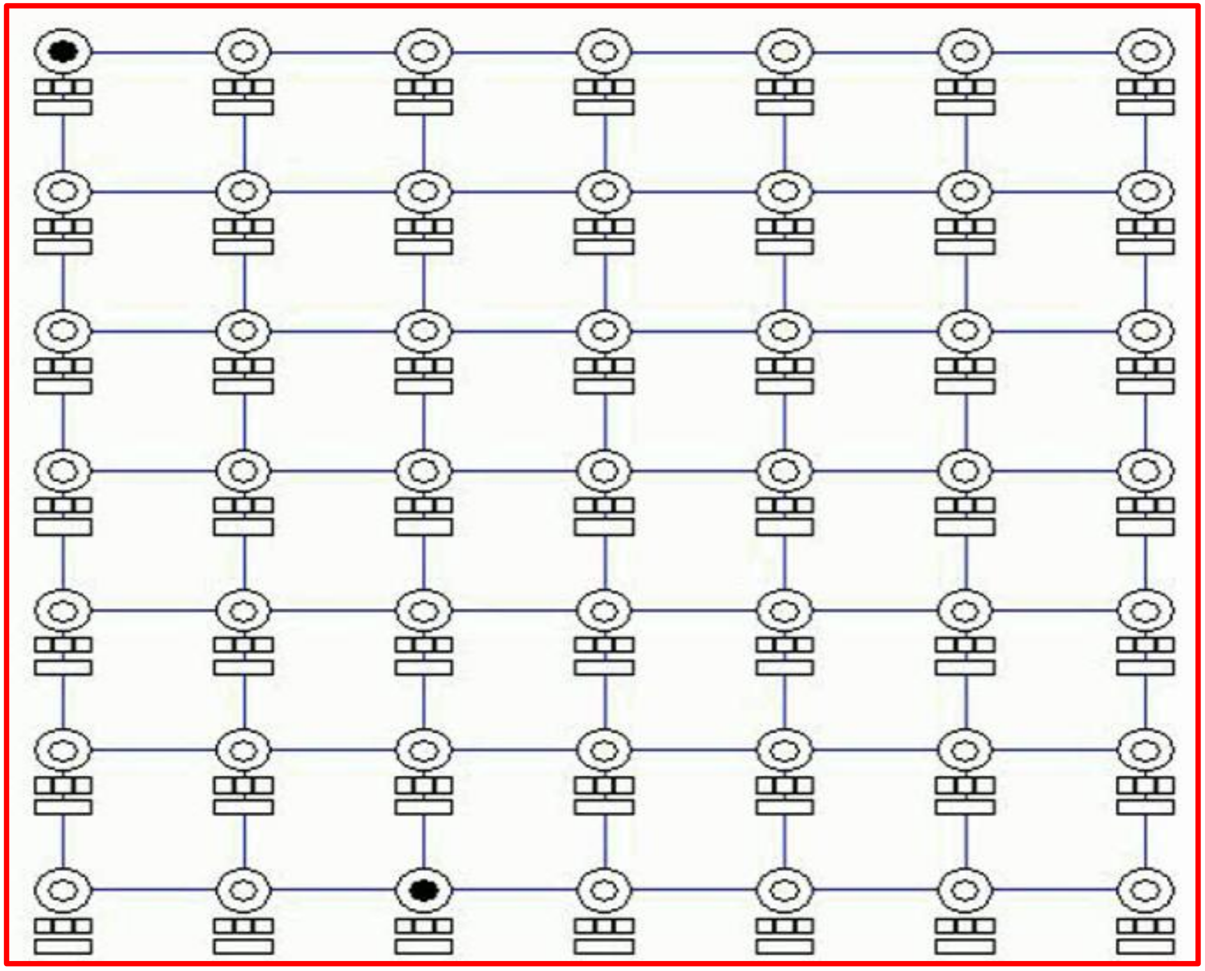


Visual-sensor networks

V. Sulić et.al. IEEE Trans. Circuits and Systems for Video Technology, 2011.

- optimal path for recognition queries in visual-sensor network
- based on hierarchically-structured features

verification on a simulator



A reference design for cost-effective visual-sensor-network nodes

MACHINE
VISION
LABORATORY

Boštjan Murovec, Janez Perš, Rok Mandeljc,
Vildana Sulić, Stanislav Kovačič

University of Ljubljana
Faculty of Electrical Engineering
http://vision.fe.uni-lj.si

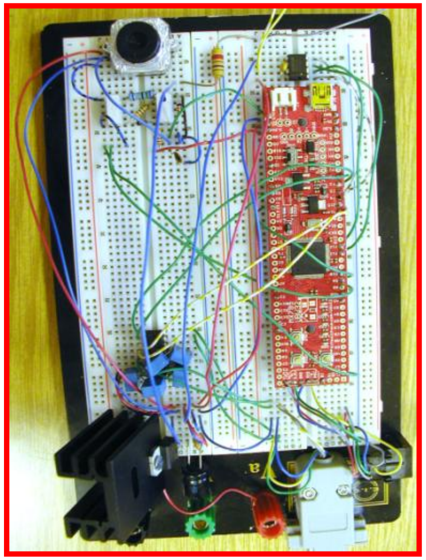
Workshop on Architecture of Smart Camera, 5th-6th April 2012, Clermont-Ferrand, France

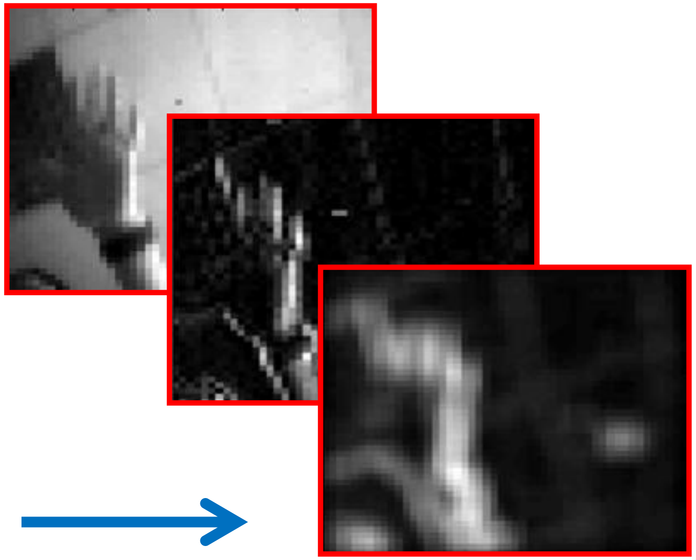
Motivation (1/2)

- Low-cost embedded smart camera **reference design**
  - commoditized technologies
  - low entry barrier
  - tailored toward CV developers

```
% covariance descriptor for the
% central (32x32) region of
% 50x50 pixel, 8 bit image.
function C = cov_descriptor (I)

% Copy, crop, convert
If = single(I(8:41,8:41));

% Convolution masks
f1 = [-1 0 1];
f2 = [-1 2 -1];

% Derivatives
Ix = conv2(f1, If);
Iy = conv2(f1,1,If);
Ixx = conv2(f2, If);
Iyy = conv2(f2,1,If);
```

GCC

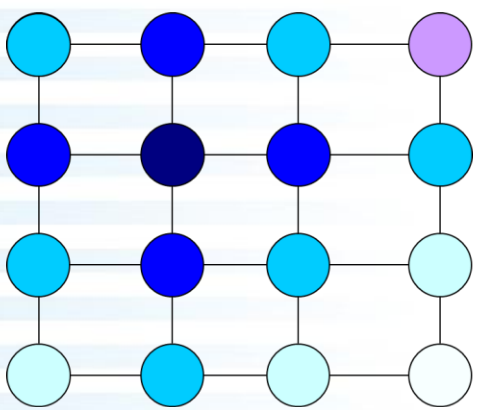


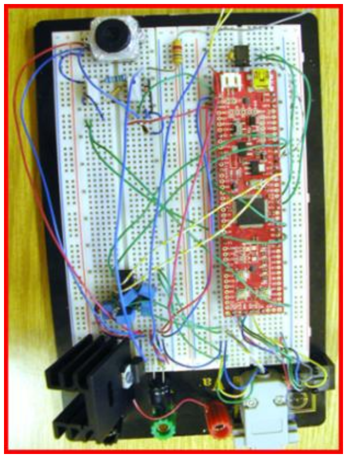


Motivation (2/2)

- Cost-effective Visual Sensor Network
  - 20 – 100 embedded cameras
  - low-cost implementation
  - powerful enough for a limited CV & PR

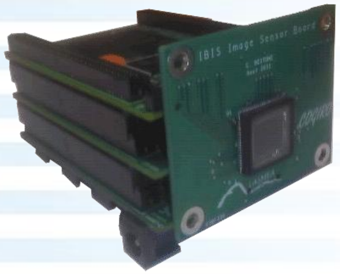
- General-purpose platform for visual sensors
  - not a camera in traditional sense (privacy concerns)





The current state of affairs

- (1/2) Examples (from WASC cover web page)



SeeMos (Dream)
Sensor: 640 x 480, logarithmic response
CPU: NIOS RISC + DSP + FPGA
SDRAM: 64 MB, SRAM: 5 x 2MB dedicated SRAM blocks

Citric Platform (Berkeley)
Sensor: 1280 x 1024 @ 15 fps (640 x 480 @ 30 fps)
CPU: Intel XScale PXA270, max 624 MHz, 32-bit
FLASH: 16 MB, RAM: 64 MB

The current state of affairs

- **(2/2) W. Wolf et. al. (2006)**

many CPU hungry applications for MP smart cameras

- MPEG compression, H.264, audio compression
- human-activity recognition

The bottom line…

- as powerful CPU & sensor as possible

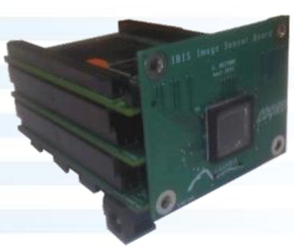
At the same time…

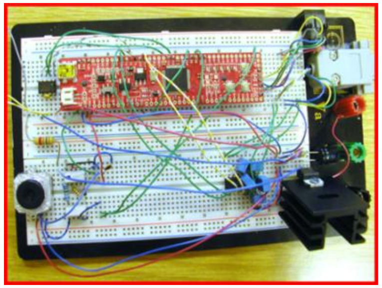
- usage of battery power
- preference to wireless connections

Is powerful architecture always needed?

- **Keyword: trade-offs**
  - battery power vs. long service intervals vs. powerful CPU
  - wireless network: battery uptime vs. bandwidth
  - illumination vs. battery power
  - specialized technologies vs. simplicity of coding
  - capabilities vs. costs

versus





How much less-powerful is realistic?

- An example from WASC cover web page

VITO Mouse Cam
Sensor: 30 x 30 pixel
CPU: Microchip dsPIC (80 MHz, 16-bit)
FLASH: 128 kB, RAM: 16 kB

Successful applications/processing
Viola-Jones face detection, background subtraction, motion estimation

Our doctrine: commoditization

- Commoditization as a driving force
  - IBM PC, Ethernet
- So far no influence on smart-camera development
  - nearly no low-cost SC (CMUCam: $190 w/o network)
  - redesigns due to parts discontinuity
  - no broadly available general-purpose VSN components
  - designs target specific applications & experts (FPGA, DSP)
  - bells & whistles not accessible for general CV community
- SC/VSN commoditization
  - reference designs that are flexible
  - commoditized parts with long term stability

Our paradigm (1/2)

- Wired power
  - CV is CPU intensive and power hungry
  - a need for illumination
  - changing batteries in large VSNs is a nuisance
- Wired network
  - higher bandwidth & longer distances than low-power wireless
- Drawbacks
  - cable cost, less suitable for retrofitting
  - multipath topologies and redundancy not available

  - battery power & wireless networks not ruled out

Wired power necessity

- **Power-consumption excerpts from references**

Model	Power [mW]	Endurance [days]
IC3D	100	3.75
Xetal	600	0.6
MeshEye	12	31
CmuCam3	650	0.57
Cyclops	23-65	5.7-16

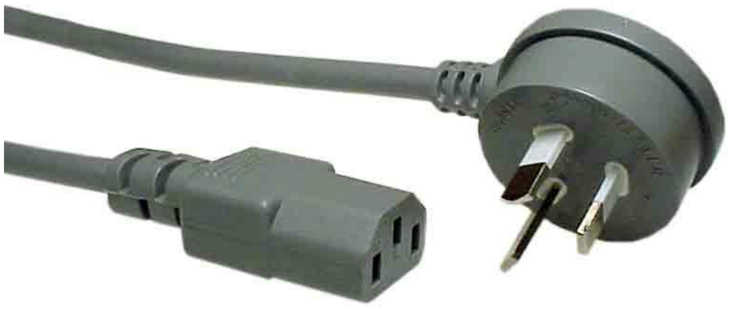
Peak Power Consumption
Average Power Consumption

1. Endurance is based on 9000 mWh capacity of 2 AA alkaline batteries.
2. Cameras do not necessarily work with such voltage.

- certain CV applications permit low-duty-cycle regime [MeshEye]
- we do not regard this as a low-power design!
- our doctrine: a camera is likely to be permanently fully operational

Wired network selection

- RS-485 physical layer
  - industry standard, robust, long-term stable specifications
  - bus topology is possible
  - connects to any UART (RS-232 software for two-point)
  - affordable (Max485E: 2.2 € in quantities of 25)
  - data rate 2.5 Mbps (Max308x for 10 Mbps)
- Observations from the field
  - 3.5 Mbps data throughput on a 125 m long 230V mains cord
  - tested with one transmitter and one receiver

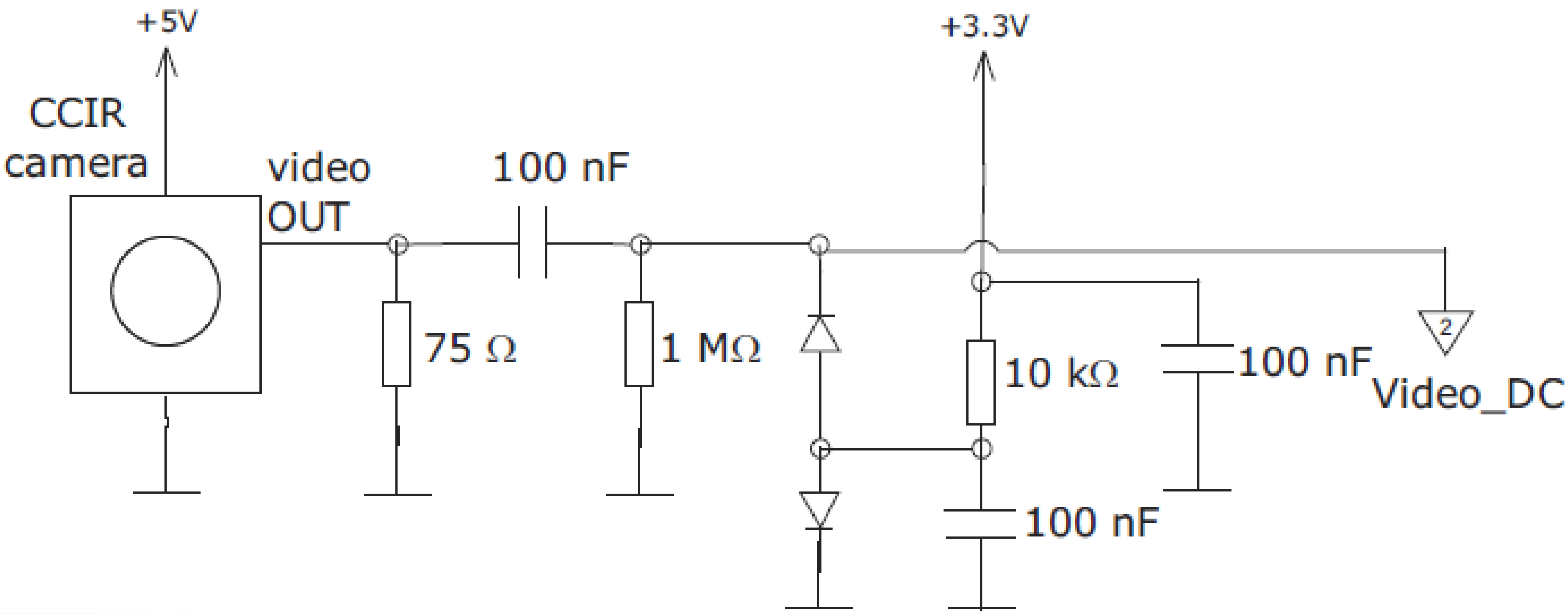


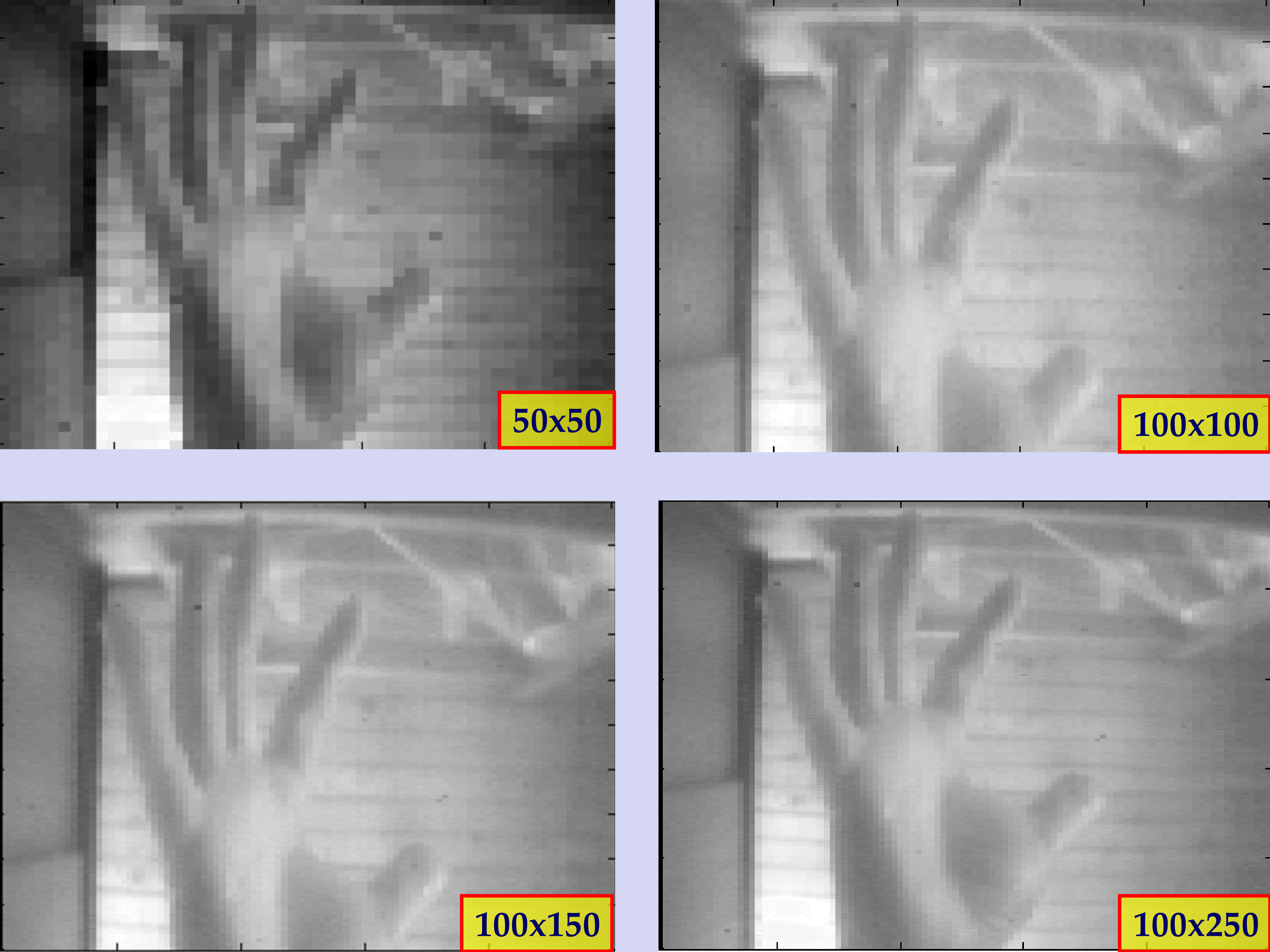
Our paradigm (2/2)

- Commoditized video sensor
  - black-&-white analog (CCIR) camera
  - long-term design stability
  - 4 € in small quantities

- Grabbing characteristics
  - grabbing with internal MCU periphery: Microchip PIC32
  - no external analog amplifiers & filters
  - typical resolutions: 50x50 … 50x250
  - combination of two interlaced images: 100x100 … 100x250

Excerpt from electrical scheme

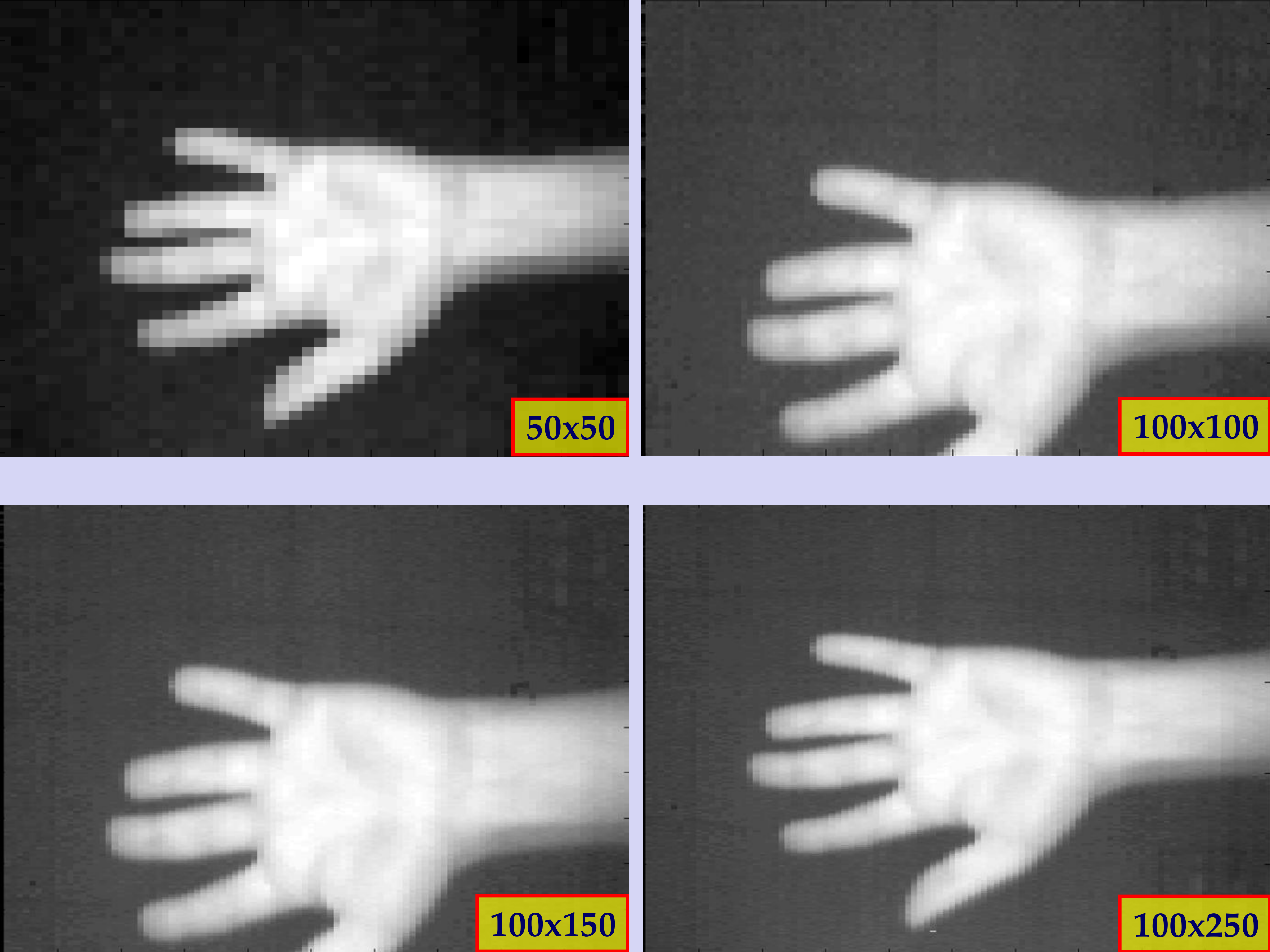


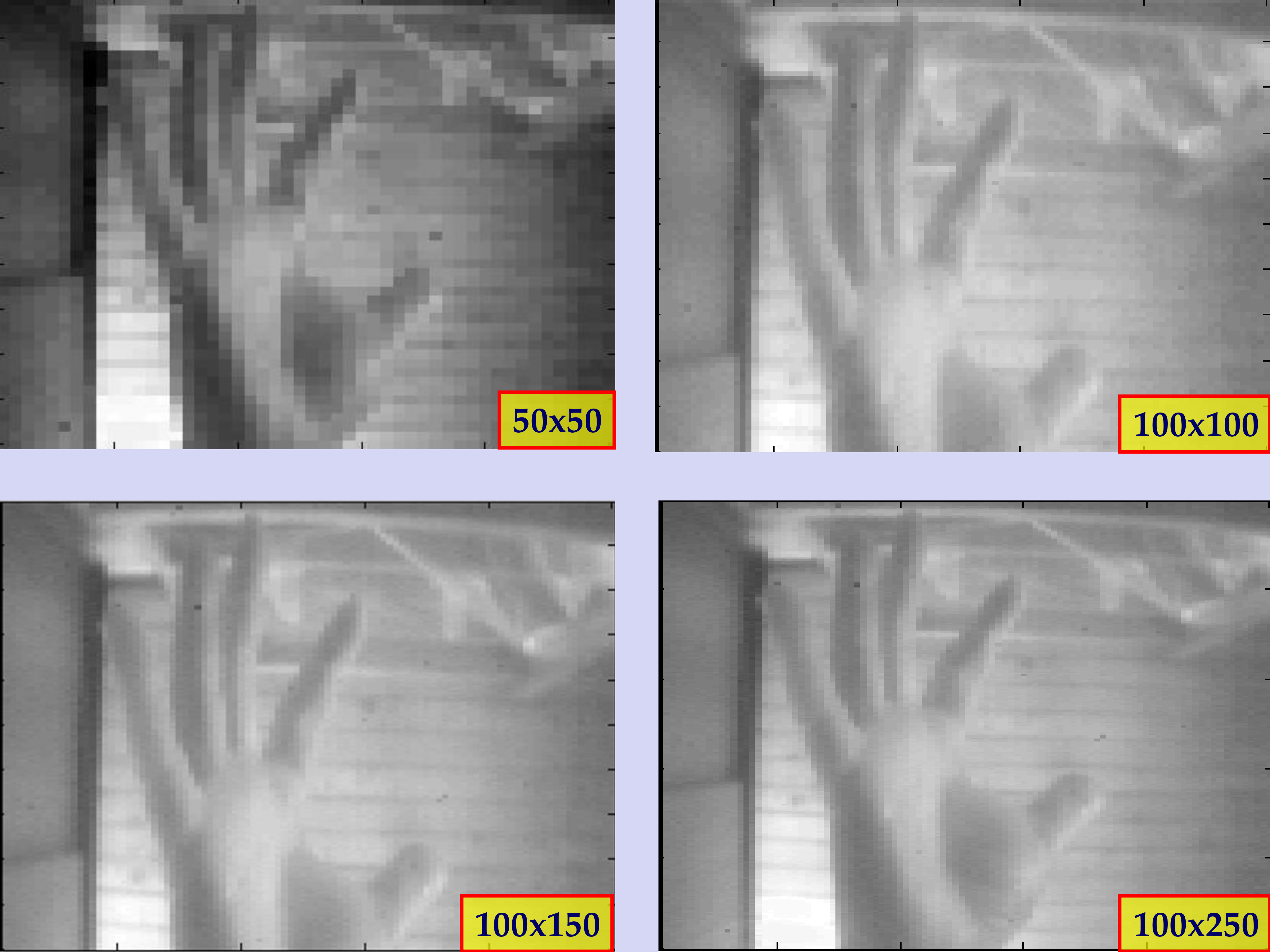

50x50
100x100
100x150
100x250

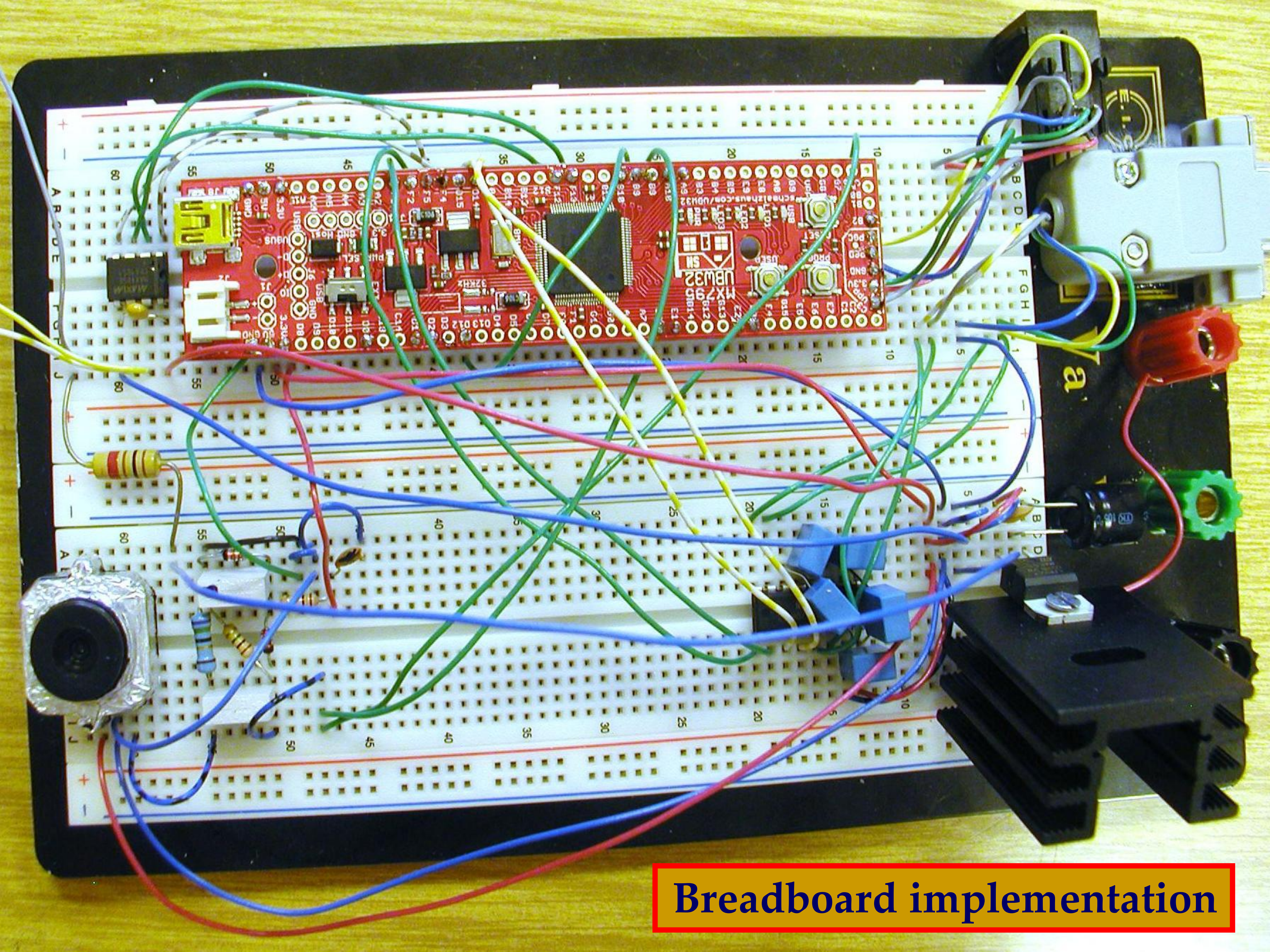
Optics and illumination

- Illumination
  - NIR LED illuminators (B&W images)
  - visible-light blocking NIR filter
  - Integration due wired power

- Standard interchangeable lens
  - may compensate low image resolution
  - standard for low-cost lens: M12, prices 3$ - 5$

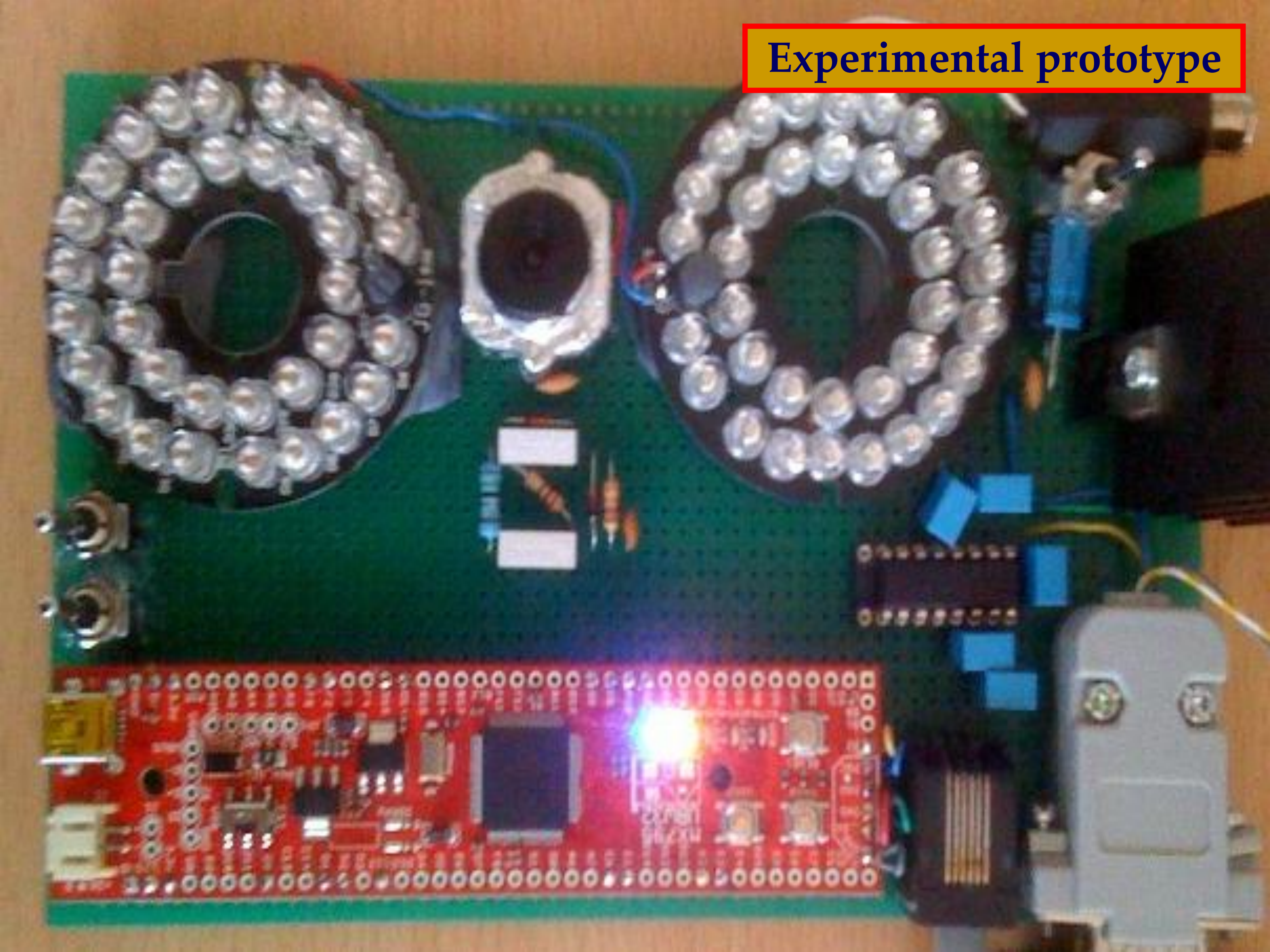

50x50
100x100
100x150
100x250


50x50
100x100
100x150
100x250

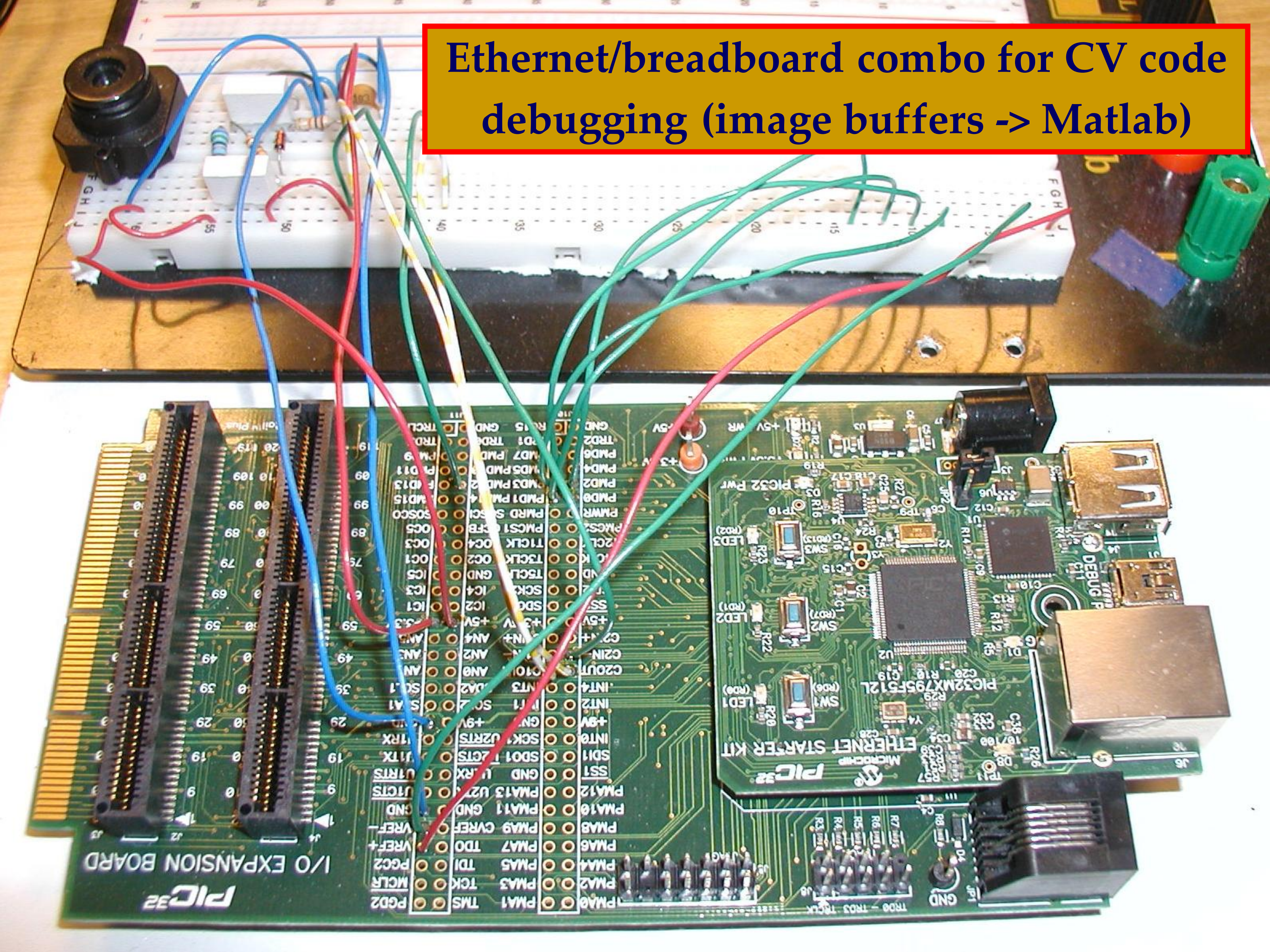


Breadboard implementation

Experimental prototype



Ethernet/breadboard combo for CV code debugging (image buffers -> Matlab)



Parts cost and power

Feature	Specification	Cost [$]
MCU	Microchip PIC32MX795F512L 128 kB RAM, 512 kB FLASH MIPS32 M4K CPU, 90 DMIPS at 60 MHz	6
Lens	M12 60^o (incl. with sensor) M12 180^o (option)	4
Illumination	NIR LED assembly Wratten #87 NIR filter	3 1
Sensor	1/4" analog CCIR camera	4
Communication	RS-232 (112 kbit/s) RS-485 (2.5 Mbit/s)	3 3
Discrete	CCIR signal path	1
Power	voltage stabilizer + capacitors	(?) 3
Total	**w/o PCB, housing**	**28**

Power: CPU + camera: 0.6 W

illumination: 2 x 7 W (2x 35-LED) NIR

Experiments (1/2)

- Motion detection
  - grab two images 50x50
  - compare the contents

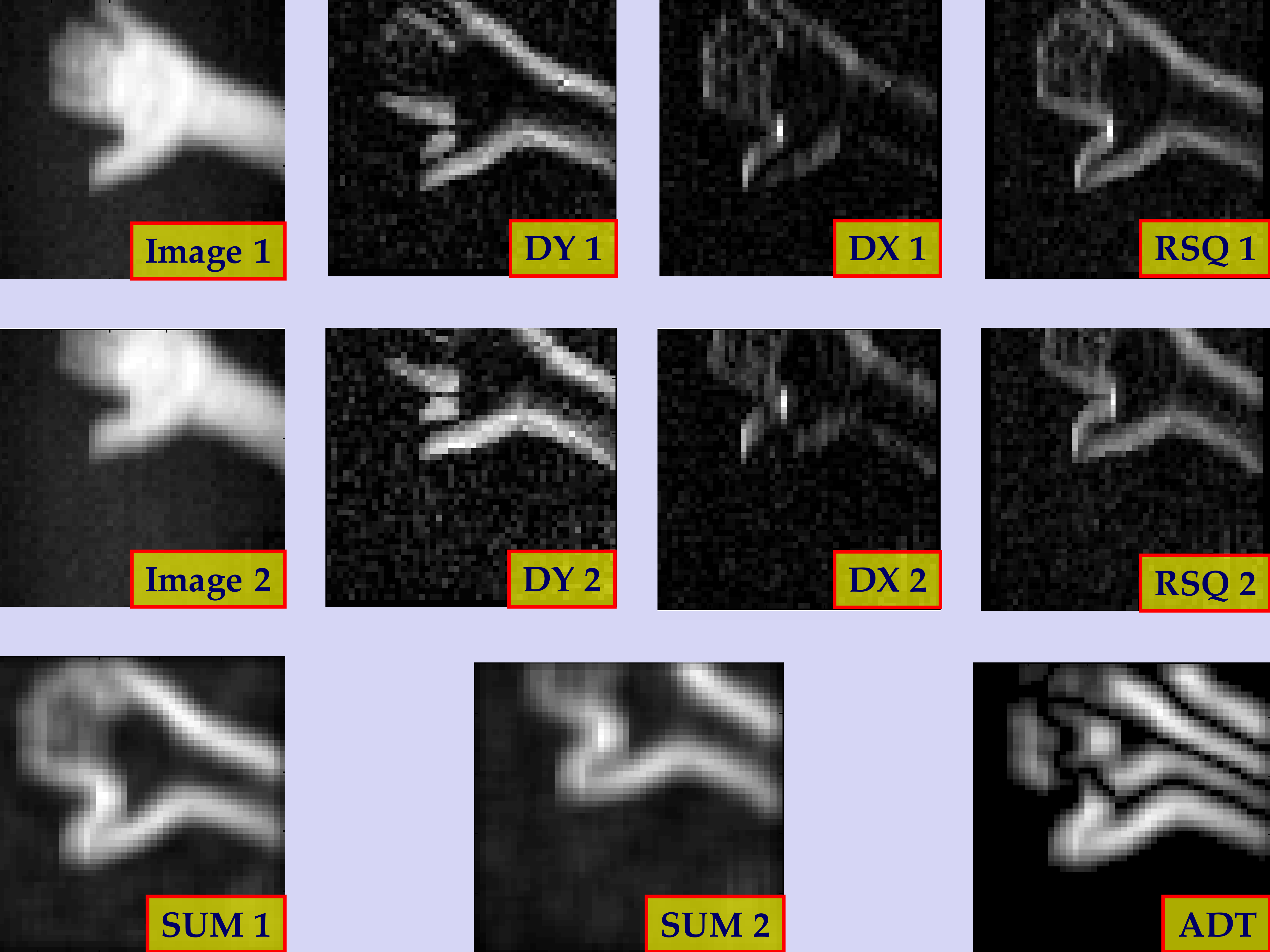


Image 1
DY 1
DX 1
RSQ 1
Image 2
DY 2
DX 2
RSQ 2
SUM 1
SUM 2
ADT

Experiments (1/2)

- **Motion detection**
  - **image processing: 8.08 ms**
  - **detection (three types)**
    - **sum of ADT: 0.10 ms**
    - **histogram of ADT + entropy: 0.45 ms**
    - **variance of ADT; floating point: 17.72 ms**
  - **all three detectors: full 25 fps**
  - **CPU utilization**
    - **without variance: 22%**
    - **with variance: 64%**

Experiments (2/2)

- Covariance descriptor (Matlab code)
  - O. Tuzel, F. Porikli, and P. Meer (ECCV 2006)
  - distance: generalized eigenvalues
  - 7.5 frames/sec
- HOG descriptor (C code)
  - Felzenschwalb's implementation
  - image 50x50, 6x6 blocks of 8x8 pixels
  - 9 unsigned grad., 18 signed grad., 2 texture features
  - Euclidean distance
  - 5.5 frames/sec
- Image-differencing tracker (C code, developed in Linux)
  - subtraction, filtering, region enumeration
  - Munkres assignment algorithm
  - 25 frames/sec

wide-angle lens

Application development

- **Vendor-provided ANSI C**
- **Direct use of Matlab code**
  - Matlab Coder toolbox, generates standard C code
- **Image-debugging capability**
  - special client/server regime
  - host PC can examine live image buffers on a chip during processing

```
% covariance descriptor for the
% central (32x32) region of
% 50x50 pixel, 8 bit image.
function C = cov_descriptor (I)

% Copy, crop, convert
If = single(I(8:41,8:41));

% Convolution masks
f1 = [-1 0 1];
f2 = [-1 2 -1];

% Derivatives
Ix = conv2(f1, If);
Iy = conv2(f1,1,If);
Ixx = conv2(f2, If);
Iyy = conv2(f2,1,If);

% Features
If = If(2:33,2:33);
If = If(:);
coordinates = 1:32;
X = repmat(coordinates,32,1);
Y = repmat(coordinates',1,32);
X = X(:);
Y = Y(:);
Ix = Ix(2:33,3:34);
Ix = Ix(:);
Iy = Iy(3:34,2:33);
Iy = Iy(:);
Ixx = Ixx(2:33,3:34);
Ixx = Ixx(:);
Iyy = Iyy(3:34,2:33);
Iyy = Iyy(:);
F = [If, X, Y, Ix, Iy, Ixx, Iyy];

% Descriptor
C = cov(F);

% Distance between the two
% covariance descriptors
function D = distance (C1, C2)
 D = sqrt(sum(log(eig(C1,C2)).^2));
```

Conclusion

- Proposal of commoditized SC/VSN design
  - low-end 32-bit CPU and low image resolution
  - wired power and networking (battery & wireless not ruled out)
  - integrated illumination (if needed)
  - supports properly designed Matlab code
  - image-debugging " interface"
  - a fairly low entry barrier (cost & complexity)
- Experiments confirm adequate capabilities to
  - perform certain image processing tasks at 25 fps
  - calculate HOG and covariance descriptors several times per second
  - track several objects at once