Smart Cameras and Visual Sensor Networks

Prof. Bernhard Rinner

Pervasive Computing
Institut für Vernetzte und Eingebettete Systeme
Alpen-Adria-Universität Klagenfurt
http://pervasive.uni-klu.ac.at

Visual Information

- … is the primary sensory input for humans, but also for many technical systems such as
  - Machine vision
  - Robotics
  - Medical diagnostic
  - …
- … capturing and processing has become ubiquitous and “natural” for many applications
  - Digital still cameras, mobile phones, automotive
  - Facebook, YouTube etc.
- … has gained importance for pervasive computing applications

Revolution in Cameras

- Ongoing technological advances
    - lenses
    - image sensors
    - onboard processing
    - networking
    - …

  transform camera as box delivering images into spatially distributed that generate data and events

- Smart Cameras are one aspect of this revolution

Agenda

1. Single and Multi-Camera Systems
   Some fundamentals

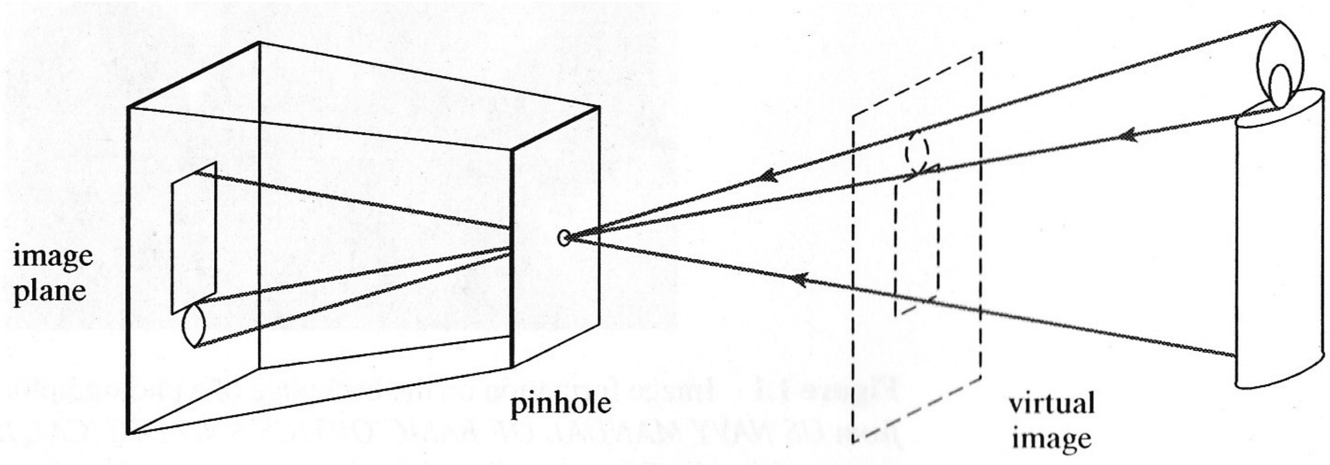
2. Distributed Smart Cameras
   Distribution of sensing & processing

3. Visual Sensor Networks
   Network services
   Applications & case studies

Camera Basics

Simple Camera

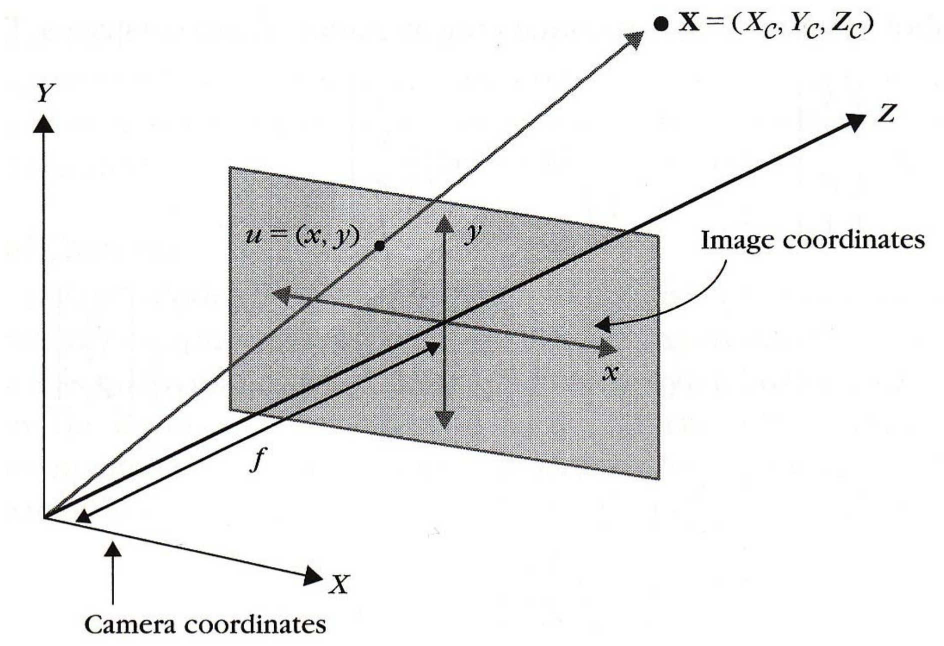
- Imaging device transforms light from 3D scene onto 2D image plane
- Pinhole cameras represent idealized model for such transformation



[] Forsyth, Ponce. Computer Vision – a modern approach. Prentice Hall. 2003

Perspective Projection

- Modeling of pinhole camera
  center of projection C, focal length f, orientation matrix R



Transformation from world to camera coordinate system

$$\mathbf{X} = \begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix} = R\left(\begin{bmatrix} Xo \\ Yo \\ Zo \end{bmatrix} - C\right)$$

Projection onto image plane

$$x = f\frac{Xc}{Zc} \qquad y = f\frac{Yc}{Zc}$$

[] Aghajan, Cavallaro. Multi-Camera Networks. Elsevier. 2009

Camera Matrices

- Homogeneous coordinates
    - Image point $\lambda(x,y,1)$
    - Scene point $\lambda(X,Y,Z,1)$
- Camera can be represented by 3x4 matrix Pc

$$Pc = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \left[R \middle| -RC \right] \qquad u \sim P\mathbf{X}$$

- Camera matrix P can be factored

$$P = KR\left[I \middle| -C \right] \qquad K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & s/m_x & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} fm_x & s & p_x m_x \\ & fm_y & p_y m_y \\ & & 1 \end{bmatrix}$$

Camera Parameters

- Extrinsic Parameters
  - From 3D world coordinates to 3D camera coordinates
  - Rotation and translation (6 degrees of freedom)
- Intrinsic Parameters
  - From 3D camera coordinates to 2D pixel coordinates
  - m_x, m_y pixel resolution
    (p_x, p_y) coordinates of principal point of the image
    s skew of the pixels
  - 5 degrees of freedom

Estimating the Camera Matrix

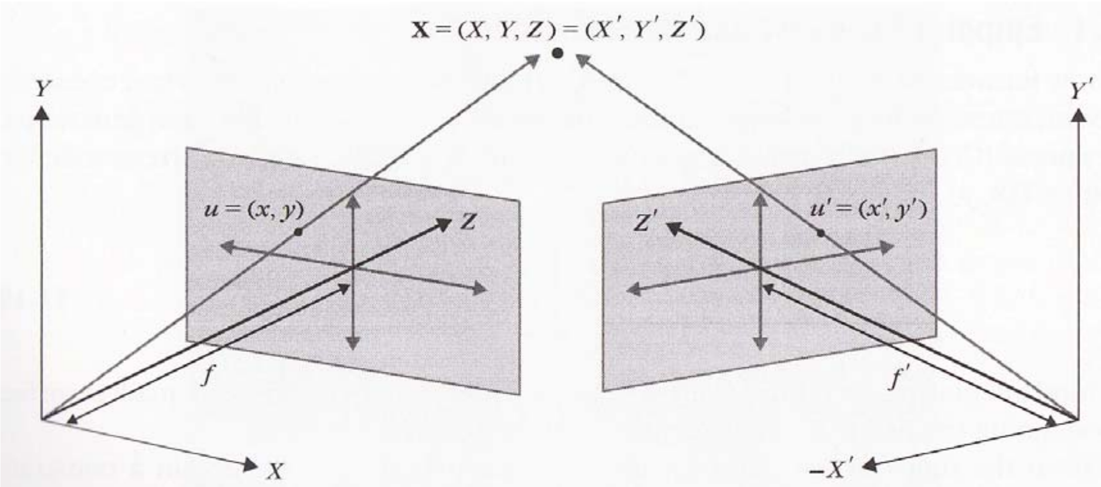
- Exploiting matched correspondences between
  - 2D Image points
  - 3D Scene points
- Minimizing the reprojection error of 3D points on calibration object
- Planar calibration grid method (checkerboard) very popular
  - Quite accurate but tedious method, Toolboxes available
  - http://www.vision.caltech.edu/bouguetj/calib_doc/

Camera Networks

Multi-Camera Geometry

- Image relationship from the same static scene
  - Two physically separated cameras C and C’
  - Single moving camera at different points in time

Rigid motion among C and C‘



[Aghajan 2009]

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \hat{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}$$

$$\hat{R} = R' R^{-1}$$

$$t = R'(C - C')$$

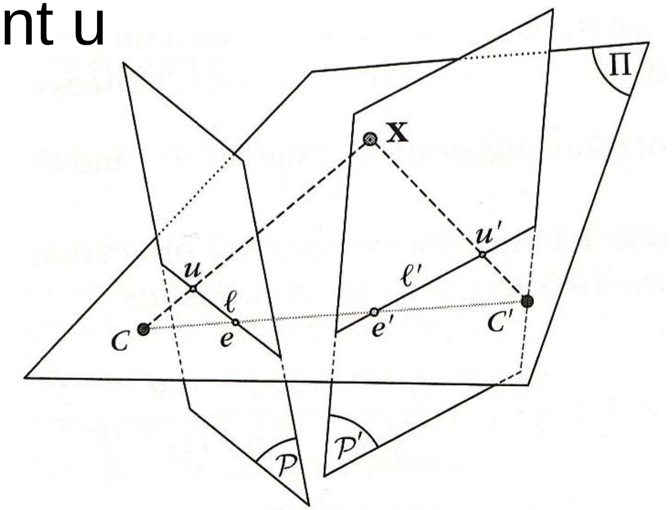
Epipolar Geometry

- Fundamental matrix F describes constraint on point correspondences between 2 images

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T F \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

- Epipolar line corresponding to point u

$$\ell_u = \left\{ u' = (x', y')^T \in P' \;\middle|\; \begin{bmatrix} u' \\ 1 \end{bmatrix} F \begin{bmatrix} u \\ 1 \end{bmatrix} = 0 \right\}$$



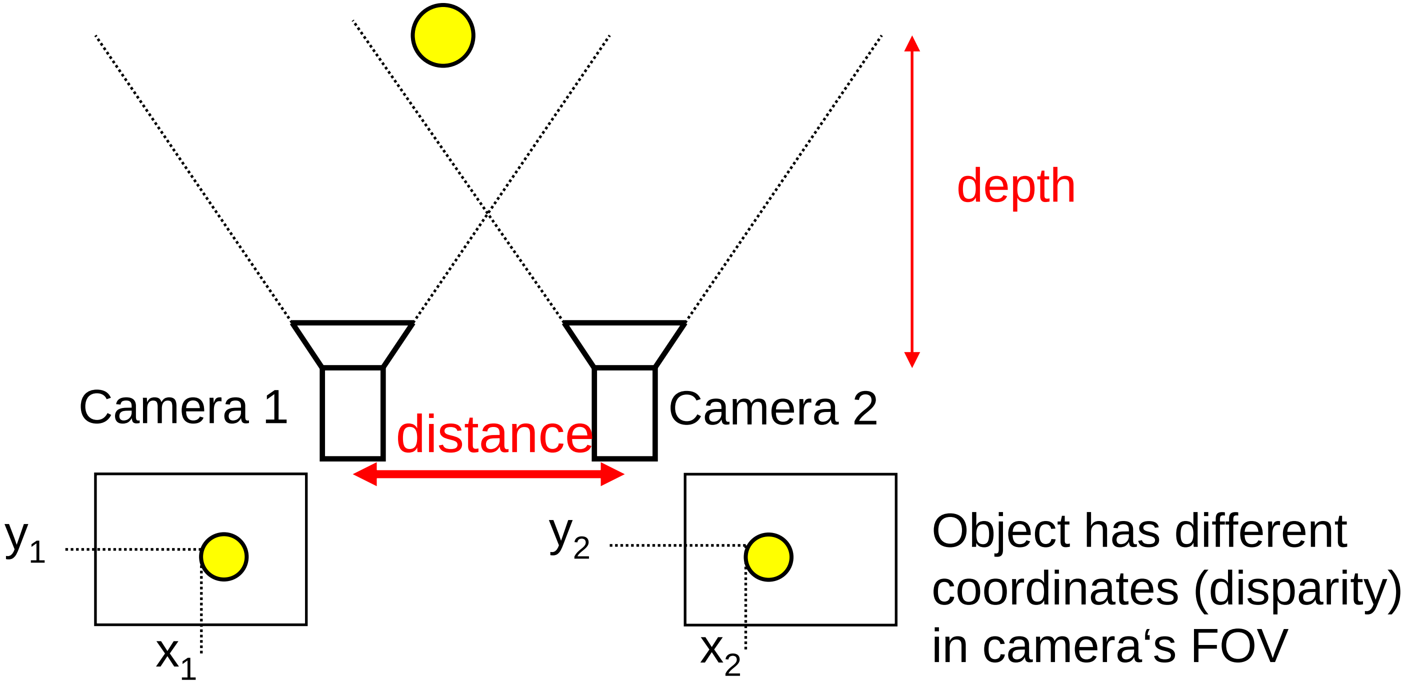
[Aghajan 2009]

- Epipoles e and e’ projections of camera centers to image plane

Advantages & Challenges

Advantage: 3D Information

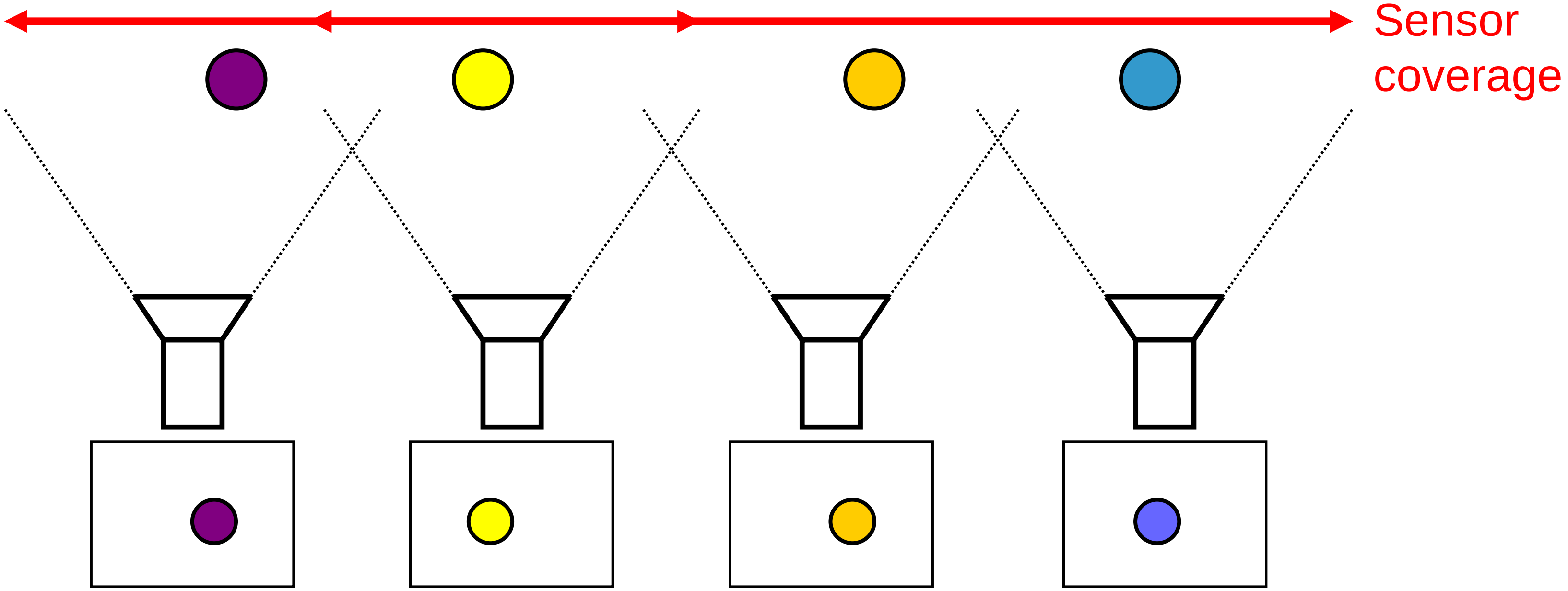
- When we know the camera geometry
  - compute depth information based on different perspectives
  - stereo camera setup



Object has different coordinates (disparity) in camera‘s FOV

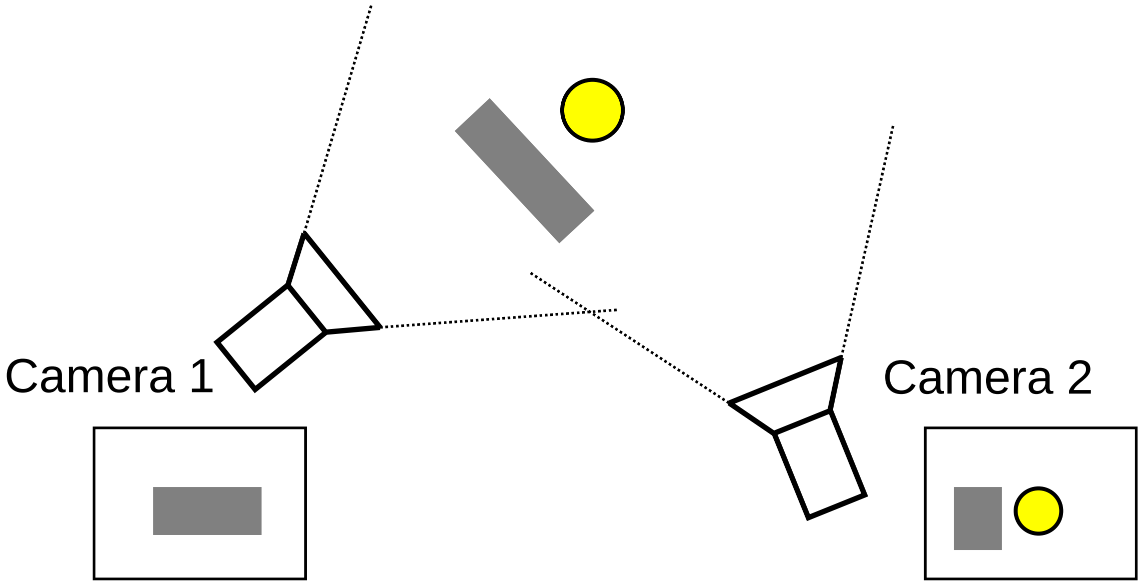
Advantage: Enlarged Field of View (FOW)

- Enlarge the sensor coverage
  - setup with overlapping or non-overlapping FOVs
  - at „constant“ resolution



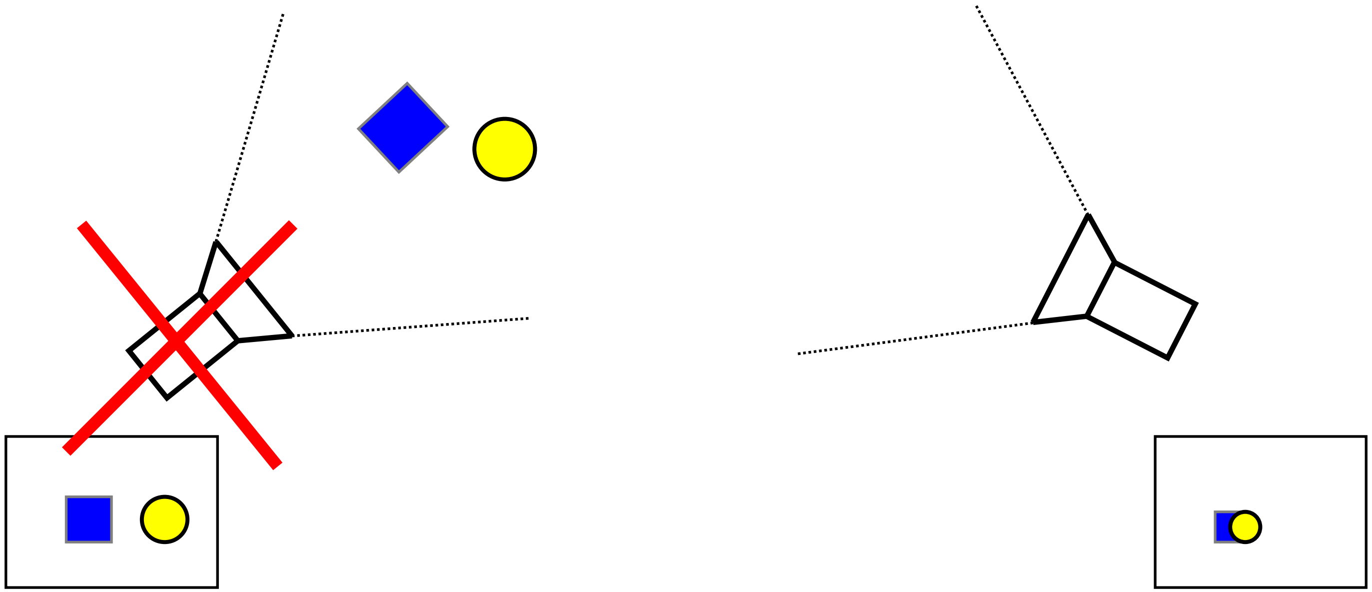
Advantage: Resolve Occlusions

- Alternate FOV may help to resolve occlusions
  - often in dynamic environments with moving objects



Advantage: Redundancy

- If a camera breaks down we may get useful information from another camera, typically with
  - different FOV
  - different resolution



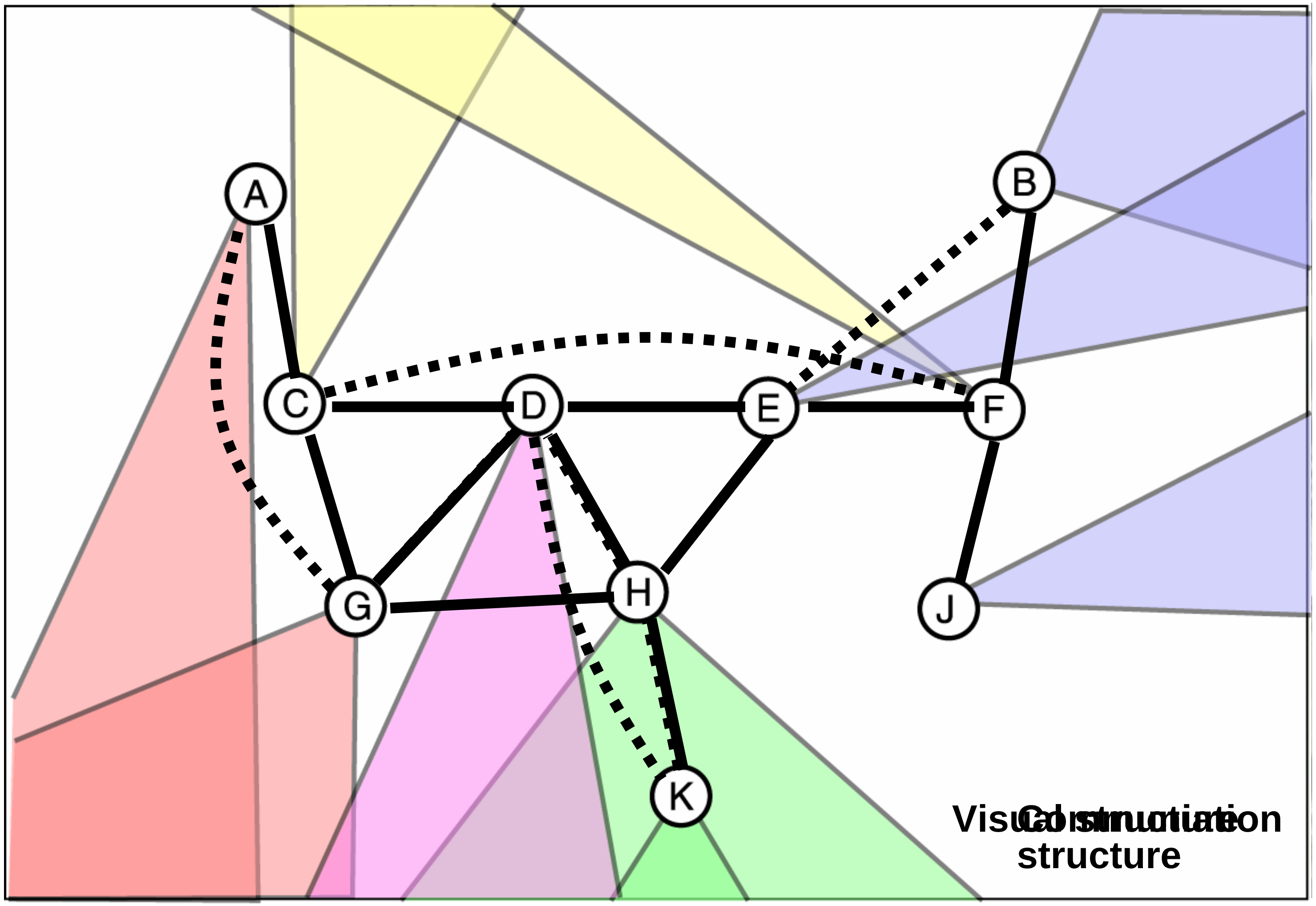
Challenge: Amount of Data

- A camera network produces a huge amount of data which has to be
    - transferred
    - stored
    - analyzed, processed, and „observed“, respectively
- Example: Subway in London with 40.000 cameras
    - single camera „generates“ approx. 260 Mbit/s (uncompressed)
    - requires extremely powerful network, storage and server!
- Video compression does not really help
    - compression rates in the range of 10 – 100
    - loss of image quality and large computational effort at camera

Challenge: Energy and Data Distribution

- Each camera requires energy and delivers data. Setting up the infrastructure for energy & data distribution is
  - tedious
  - expensive
  - and limits the applicability of multi-camera networks
- Reducing energy consumption and data transfer
  - battery-powered, energy harvesting
  - local processing, reduced bandwidth in wireless networks
- Dependency between energy consumption and data transfer
  - transferring data (much) more expensive than processing it

Challenge: Structure



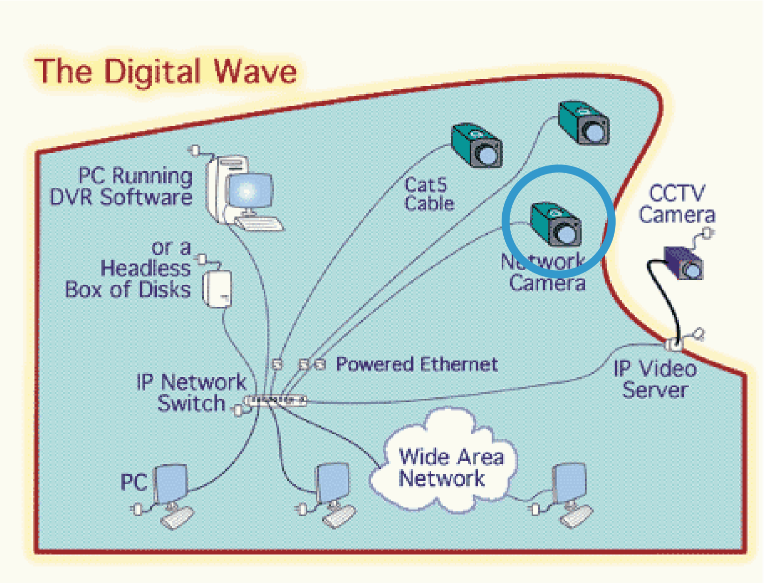
Challenge: Spatial & Temporal Calibration

- Images of (overlapping) cameras must be „calibrated“ in space and time
  - complex procedure – only required during initialization (stationary cameras)
  - at different accuracy

[Radke 2007]

Example: Video Surveillance Systems

- 1st and 2nd generation
  - primarily analog frontends
  - backend systems are digital
- 3rd generation
  - all-digital systems
- 3+ generation
  - smart cameras
  - surveillance tasks run on-site on smart cameras, e.g.,
    - video compression
    - accident detection
    - stationary vehicles (tunnels)
    - traffic statistics
    - wrong-way drivers
    - vehicle tracking



[] Regazzoni, Ramesh, Foresti. Special Issue on Video Communications, Processing and Understanding for Third Generation Surveillance Systems. Proceedings of the IEEE. October 2001

Smart Cameras

Basic Principle of Smart Cameras

- Smart cameras combine
  - sensing,
  - processing and
  - communication

  in a single embedded device

- perform image and video analysis in real-time closely located at the sensor and transfer only the results
- collaborate with other cameras in the network

Differences to traditional Cameras

Traditional Camera

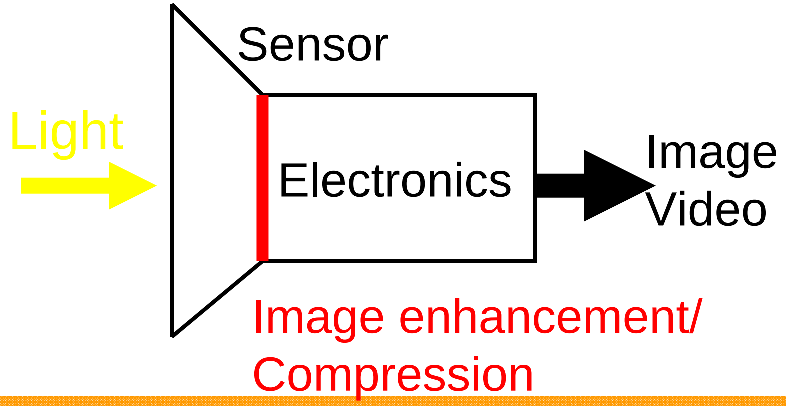
- Optics and sensor
- Electronics
- Interfaces

delivers data in form of (encoded) images and videos, respectively

Smart Camera

- Optics and sensor
- onboard computer
- Interfaces

delivers abstracted image data is configurable and programmable



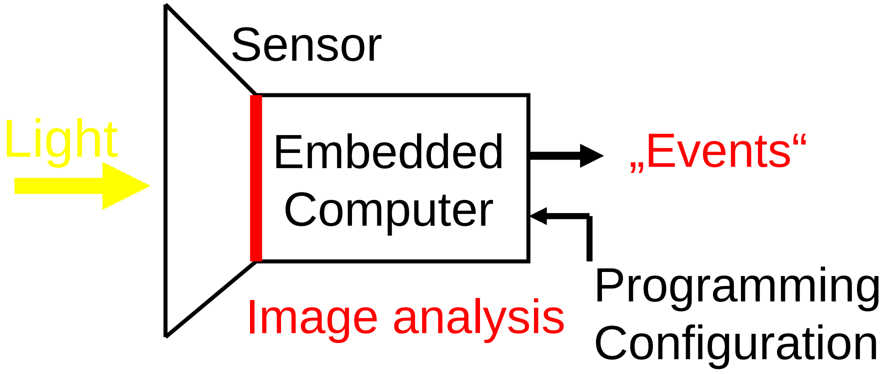
Light

Sensor

Electronics

Image Video

Image enhancement/ Compression



Light

Sensor

Embedded Computer

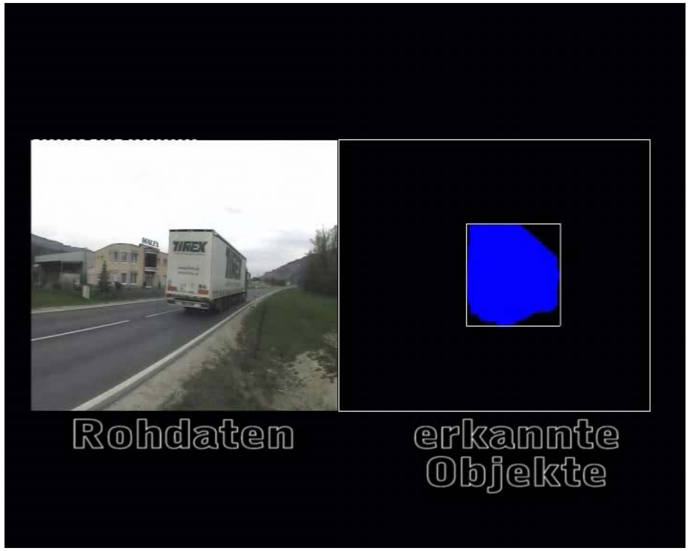
„Events“

Programming Configuration

Image analysis

Smart Cameras look for important things

- Examples for abstracted image data
  - compressed images and videos
  - features
  - detected events



Architectural Issues

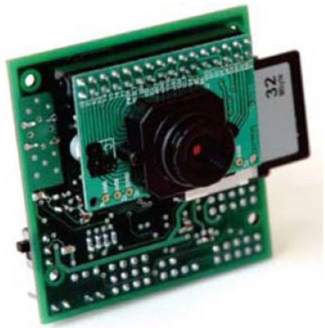
- Embedded processing of image pipeline
  - low-level operations (regular patterns on many pixels)
  - high-level analysis (irregular on few objects)
- Memory often bottleneck in streaming applications
  - capacity
  - bandwidth
  - standard techniques (caches etc.) may not be sufficient
- Processing platforms
  - FPGAs, DSPs, specialized processors (SIMD)
  - microcontroller, g-p processors
- Power consumption!

Various Prototypes

- Prototypes differ in various aspects
  - computing power, energy consumption
  - wired and wireless communication
  - optics and sensors

Rinner et al. (multi-DSP)
10 GOPS @ 10Watt

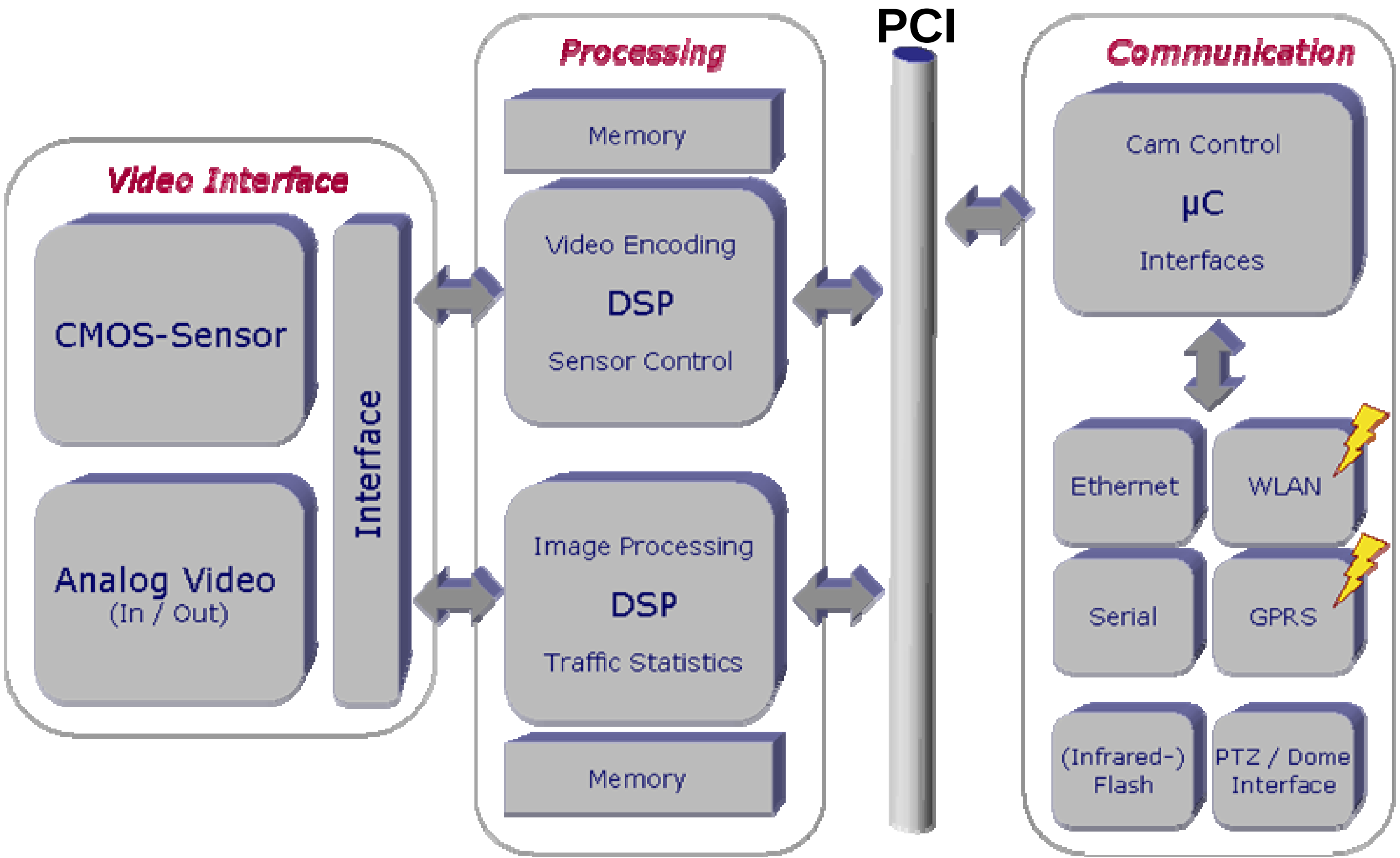
WiCa/NXP (Xetal SIMD)
50 GOPS @ 600mWatt



CMUcam3 (ARM7)
60 MIPS @ 650mW

CITRIC (PXA270)
660 MIPS @ 970mW

Scalable SmartCam Architecture



[] Bramberger et al. Distributed Embedded Smart Cameras for Surveillance Applications. IEEE Computer 2006

(Selected) Smart Camera Systems

System	**Year**	**Platform**	**Distribution/Proc.**	**Autonomy**
[Moorhead&Binni]	1999	ASIC	local	static
VISoc [Albani]	2002	SOC	local	static
[Wolf et al.]	2002	DPS (PC)	local	static
[Bramberger&Rinner]	2004	DSP	local	rem. conf.
[Dias&Berry]	2007	FPGA	local	active vis.
[Bauer]	2007	DSP	local	static
GestureCam [Shi]	2007	FPGA	local	static
[Bramberger et al.]	2006	multi-DSP	cooper. tracking	dyn. conf.
[Micheloni et al.]	2005	(PC)	MC-tracking	PTZ
[Fleck&Strasser]	2007	PowerPC	MC-tracking	static

(Selected) Smart Camera “Sensors”

System	**Year**	**Platform**	**Distribution**	**Radio**
Cyclops [Rahimi]	2005	ATmega128	coll. tracking	via Mica2
CMUcam 3 [Rowe]	2007	ARM7	local proc.	-
Meerkats [Margi]	2006	StrongARM	coll. tracking	ext. 802.11b
MeshEye [Hengstler]	2006	ARM7	local	via CC2420
WiCa [Kleihorst]	2006	Xetal (SIMD)	coll. gesture rec	via CC2420
CITRIC [Chen]	2008	PXA	tracking	via Tmote

More details

[] Akyildiz et al., Wireless Multimedia Sensor Networks: Applications and Testbeds. PIEEE Oct. 2008
[] Rinner et al., The Evolution from Single to Pervasive Smart Cameras. In Proc. ICDSC 2008

Distributed Smart Cameras

Smart Cameras collaborate

- Connect autonomous cameras in a network
  - exploit smart cameras' capabilities (eg. avoid raw data transfer)
  - relax centralized/hierarchical structure of MC networks
  - introduce dynamic configuration (structure and functionality)

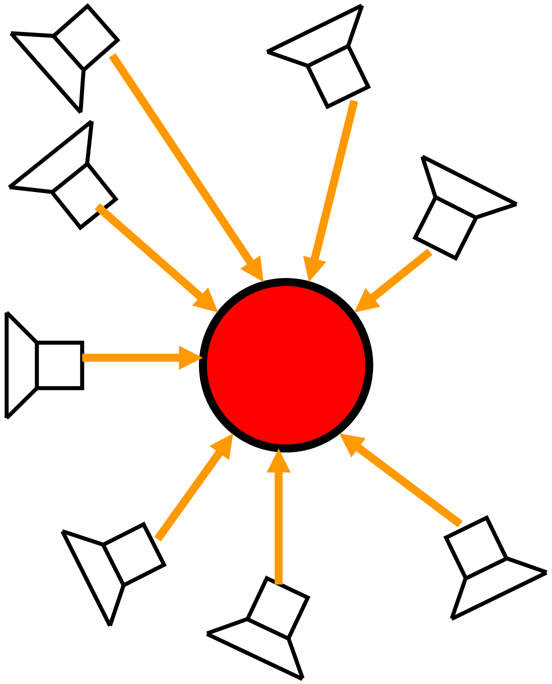
- Challenges for distributing sensing & processing
  - camera selection and placement
  - calibration & synchronization
  - distributed processing
  - data distribution and control, protocols and middleware
  - distributed computer vision (distributed signal processing)
  - real-time, energy-awareness, …

(Potential) Advantages of DSC

- Scalability
  - no central server as bottleneck
- Real-time capabilities
  - Short round-trip times; “active vision”
- Reliability
  - High degree of redundancy
- Energy and Data distribution
  - Reduced requirements for infrastructure; easier deployment?
- Sensor coverage
  - Many (cheap) sensors closer at “target”; improved SNR
- …

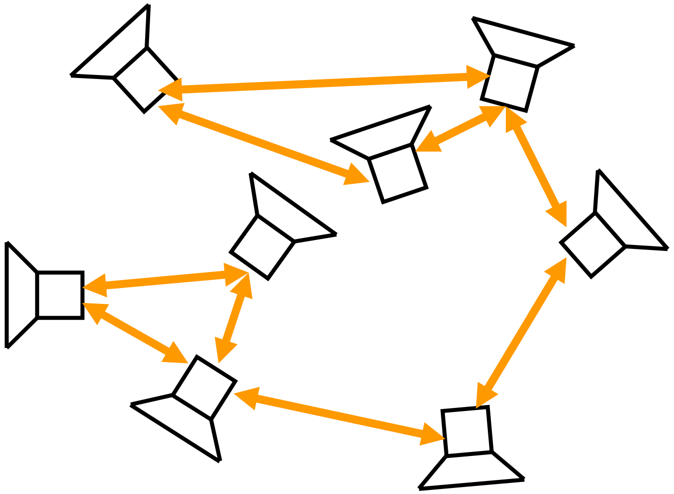
Networking

Traditional Camera Networks



Cameras stream images/ videos to „server“

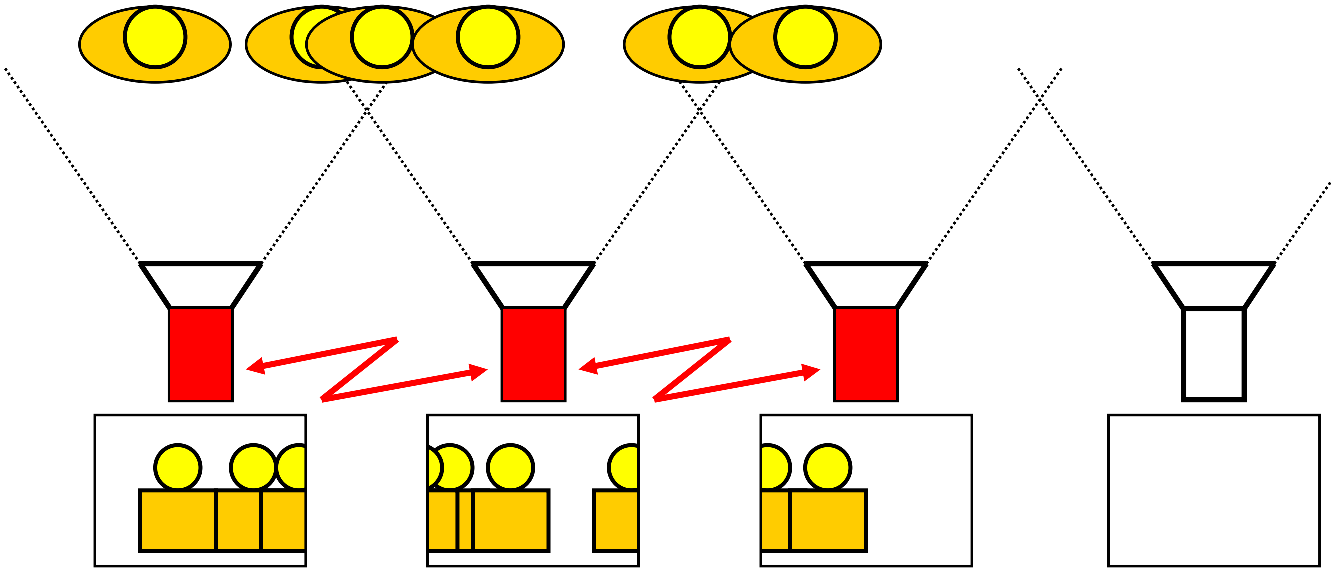
Smart Camera Networks



Cameras collaborate directly (spontaneous, p2p, ad-hoc)

Distributed Processing in Network

- Example: autonomous tracking of mobile objects among multiple cameras



on-board analysis
communication

- Computation follows (physical) object
  - requires spontaneous communication; distributed control & data

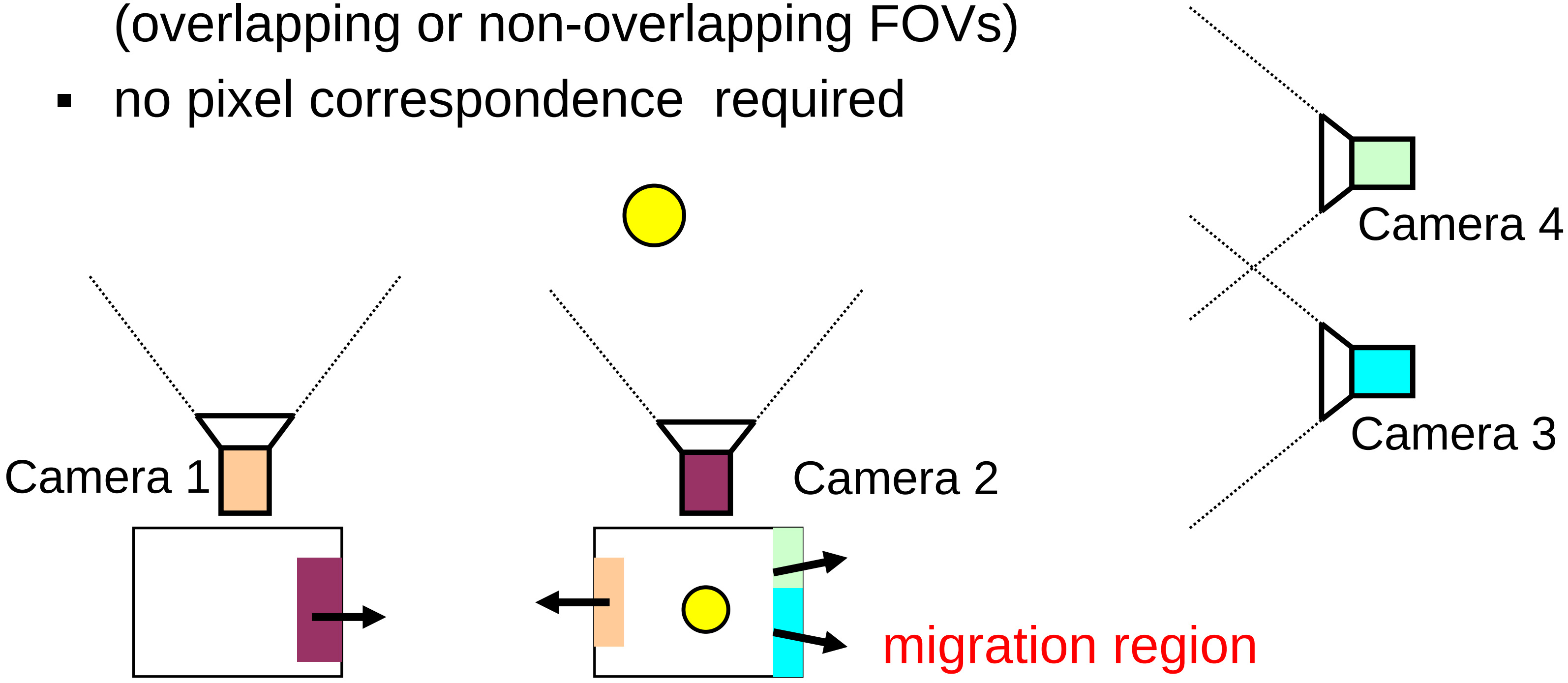
Autonomous Multi-Camera Tracking

[] Quaritsch et al., Autonomous Multicamera Tracking on Embedded Smart Cameras EURASIP JES 1/2007]

- Assumptions for multi-camera tracking
  - implement on distributed embedded smart cameras
  - avoid accurate camera calibration
  - do not rely on central coordination
- Important design questions
  - What (single-camera) tracking algorithm to use?
  - How to coordinate the cameras?
    i.e., distributed control, exploit locality
  - How to hand over tracking from one camera to next?
- Treat questions independently
  - standard ("color-based") CamShift tracker
  - focus on hand over strategy

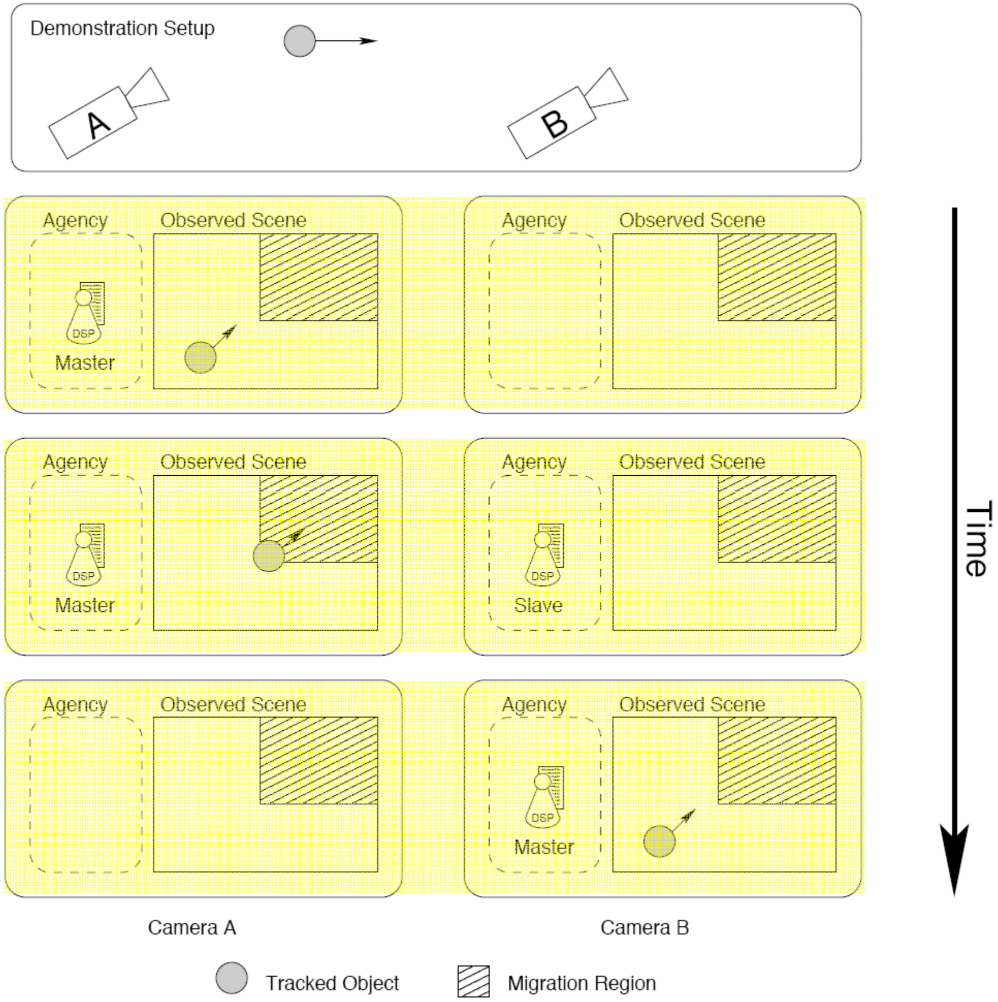
Spatial Relation among Cameras

- Camera neighborhood relation
  - important for determining “next camera(s)”
  - based on pre-defined “migration region” in camera’s FOV (overlapping or non-overlapping FOVs)
  - no pixel correspondence required



migration region
with motion vector

Multi-Camera Handover Protocol



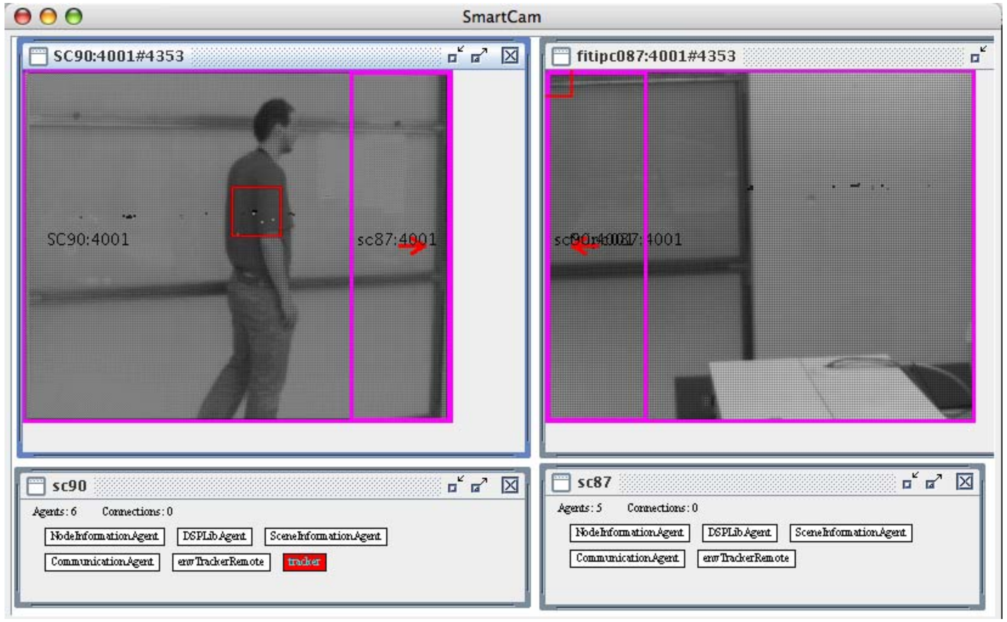
Master/Slave handover

1. camera A tracks object
2. whenever object enters migration region tracking agent is cloned on “next” camera (slave)
3. slave starts tracking when slave identifies object master gets terminated

Tracker initialization

- color histogram a initialization data

Implementation & Results



Visualization

- migration region (magenta)
- tracked object (red rectangle)

- tracking agent (red box)

Code size	15 kB
Memory requirement	300 kB
Internal state	256 B
Init color histogram	< 10 ms
Identify object	< 1ms

CamShift (single camera)

Loading dynamic executable	8 ms
Initializing tracking algorithm	250 ms
Creating slave on next camera	18 ms
Reinitializing tracker on slave	2 ms
Total	**278 ms**

Multi-camera performance

Toward Visual Sensor Networks

Characteristics of VSN

- In-network image sensing & processing
- Data streaming as well as eventing
- Resource limitations (power, processing, bandwidth …)
- Autonomy & service-orientation
- Ease of deployment

Multi-view Calibration

- Standard calibration methods are tedious
  - performed offline
  - require physical appearance of reference objects
  - limited scalability in large networks
- Automatic methods are necessary in visual sensor networks
  - Limited knowledge about initial position and orientation of cameras
  - Mobility of camera nodes
  - No human/expert available
- Estimation methods
  - Vision Graph
  - Calibration of neighboring cameras

Estimating the Vision Graph

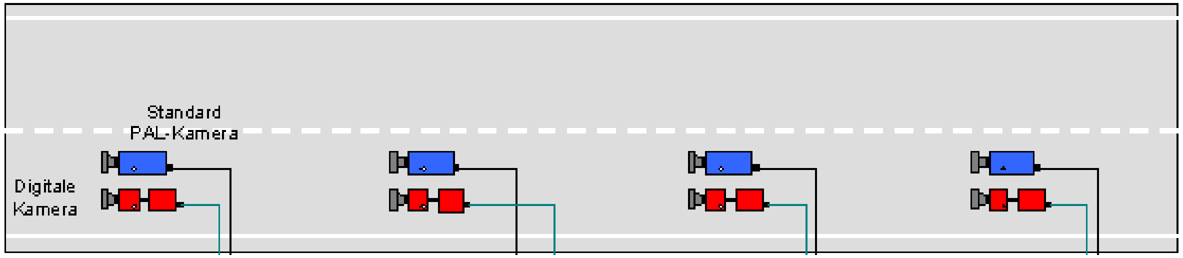
- Identify cameras with overlapping FOV
  - Also referred to as topology of the network
  - Exploit spatiotemporal tracks of moving objects
  - Often assume common ground plan
- Determine the “area” of overlap
  - Compute offline (if cameras are fixed)
  - Model camera projection (if parameters are known)

Multi-Camera Calibration

- Focus on calibration only among neighboring cameras
  - Determine reliable corresponding points
  - Estimate parameters of neighboring cameras
- Distributed calibration algorithms
  - Avoid transferring images
- Exploit information about position and orientation of cameras
  - Often available in sensor networks
  - Calibration not exclusively based on captured images

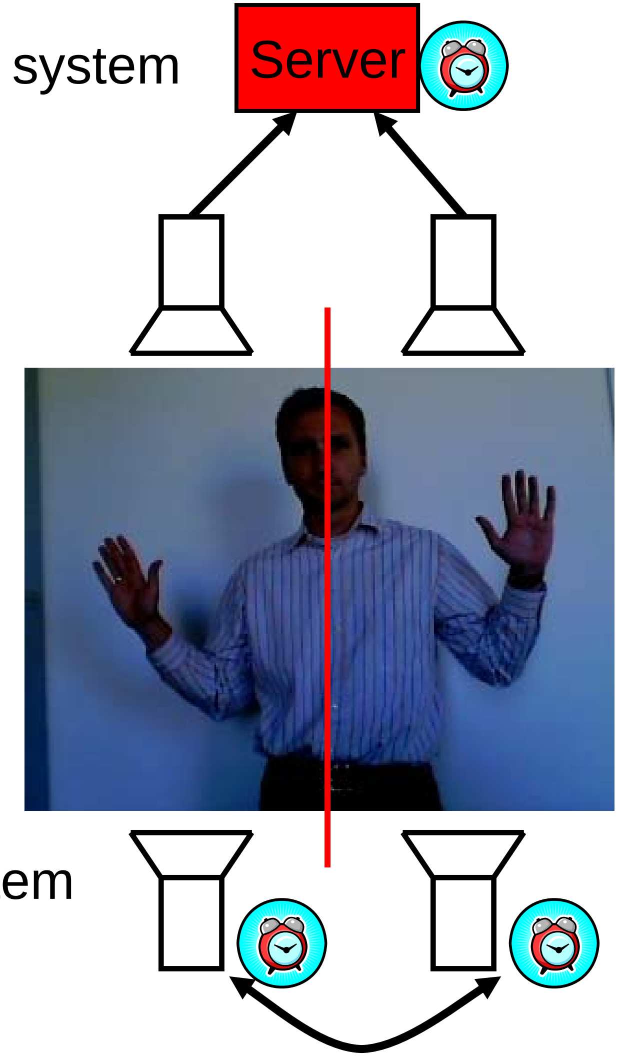
Multi-Camera Calibration (2)

- Relaxing calibration requirements
  - What to do when there is no overlap (cp. epipolar geometry)?
  - Accurate calibration not required for some applications
- Example: Camera Hand-off in MC-Tracking
- Camera network topology
  - Applications pose strong constraints (traffic, buildings etc.)



Temporal Calibration / Synchronization

- Cameras need to be synchronized for distributed analysis. Problems
  - No global clock
  - Communication delays (unknown, jittering)
- Example
  - Fusing individual views from two cameras



Centralized system
global clock
no delays

Distributed system
local clocks
delays

Synchronization

- Synchronization accuracy
    - Depends on application and on level of local processing
    - Often “frame-accurate” synchronization sufficient
- Apply methods from sensor networks
    - Tutorial of K. Römer

Resource-Awareness

- Visual sensor nodes have limited resources
  - Embedded platform
- Critical resources
  - Sensing
  - Computing and memory capacity
  - Communication
  - Power
- Manage resources effectively
  - Switch off unused components: dynamic power management
  - Trade performance, quality, time etc: reconfiguration

Quality of Service

- Cameras and VSN provide different quality levels
- Low-level QoS
  - Image resolution
  - Communication bandwidth, delay
- More abstract QoS
  - Different detection performances

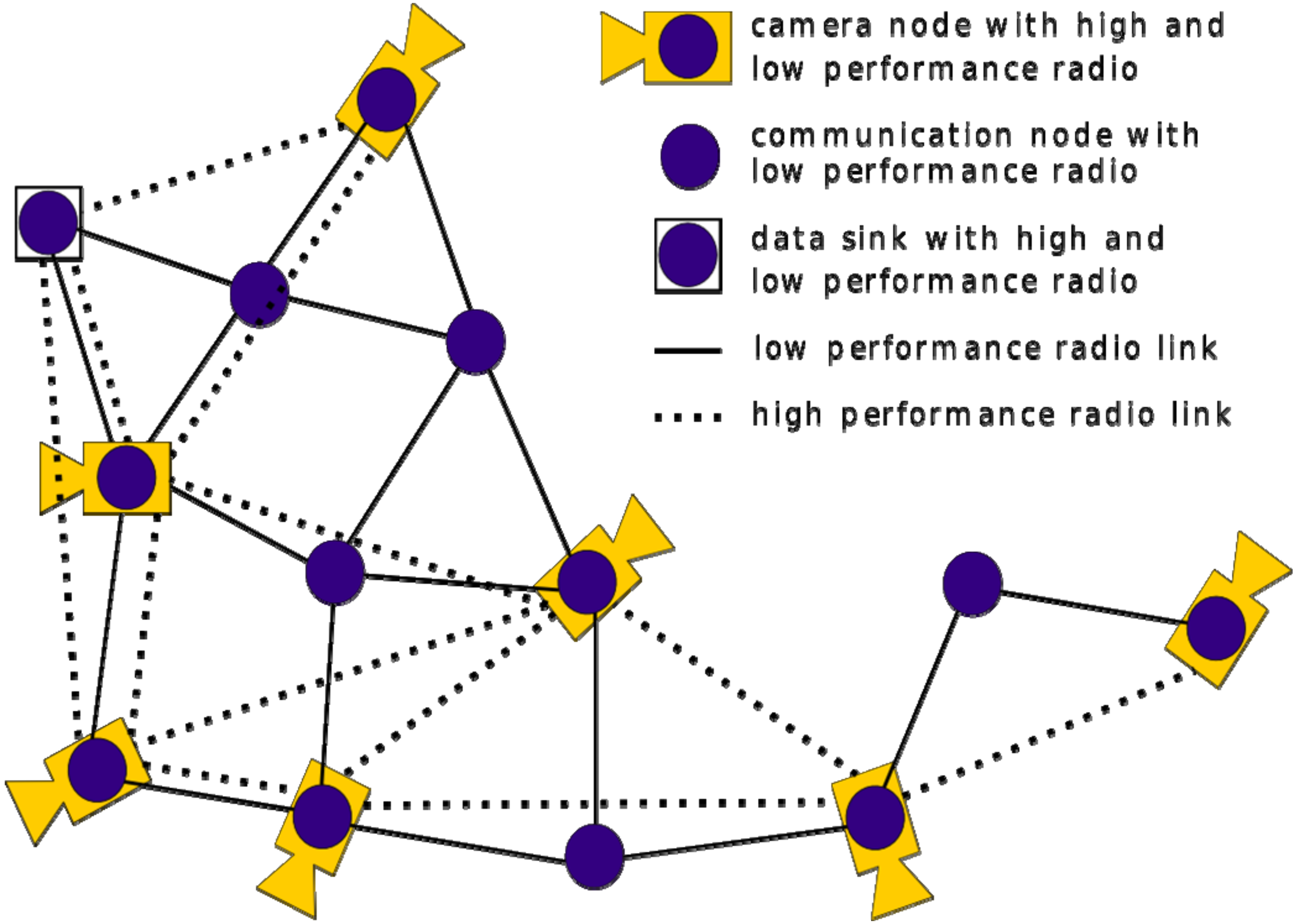
Applications & Case Studies

Pervasive Smart Camera Network

- Tradeoff among bandwidth, power consumption and streaming requirements in VSN
- One approach: dual radio networks
- Equip (some) nodes with two radios: low-bandwidth & high-bandwidth
- Use low-bandwidth radio for normal operation
  - coordination, eventing,
  - transfer of low-resolution (still) images
- Use high-bandwidth radio for streaming

[] Winkler, Rinner. Pervasive Smart Camera Networks exploiting heterogeneous wireless Channels. In Proc. PerCom 2009

PSC Network Architecture



PSC Camera Network

- Visual Sensor Network Platform
- Sensor Nodes
  - Embedded board with USB connected peripherals
  - TI OMAP3530 processor: ARM Cortex A8 @ 600MHz, TI C64x DSP @430MHz
  - 128MB RAM, 256MB Flash
  - SD-Card, USB, DVI, audio-i

PSC Demo: Tracking

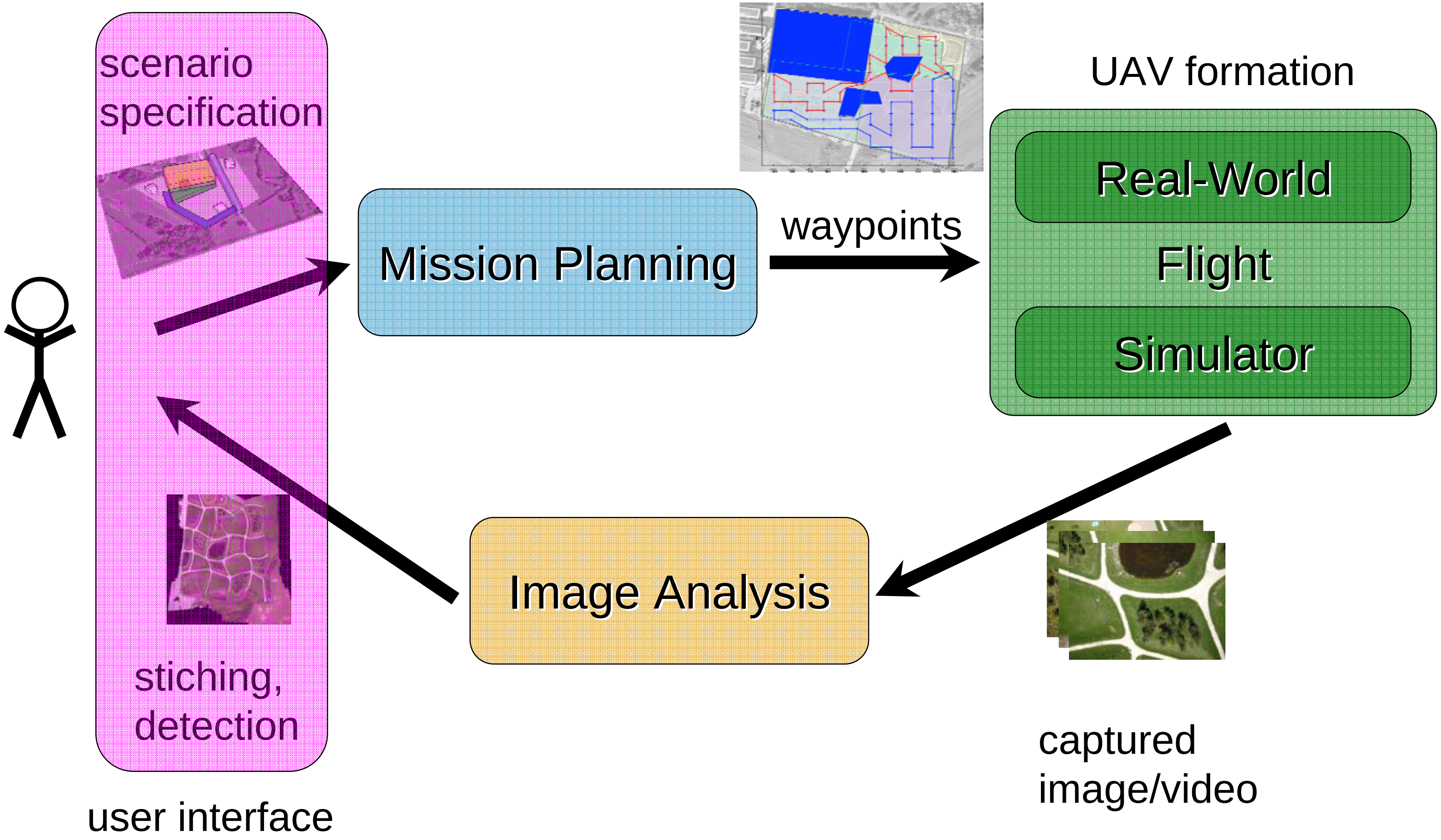
- Demonstrate tracking by using only low-bandwidth radio
  - initially transfer background image
  - perform tracking onboard
  - transfer tracking result (bounding box); 8 bytes/frame

Collaborative Microdrones

- UAVs for disaster management
  - deploy a group of small UAVs for disaster management applications
  - fly over the area of interest in structured way (formations)
  - sense the environment
  - analyze the sensor data (image stiching, object detection etc.)
- Provide “bird’s eye view” to special task forces in real-time
- Support high autonomy and an intuitive user interface

[] Quaritsch et al., Collaborative Microdrones: Applications and Research Challenges. In Proc. Autonomics 2008

High-level “Processing Loop”



scenario specification

Mission Planning

waypoints

UAV formation

Real-World

Flight

Simulator

captured image/video

Image Analysis

stiching, detection

user interface

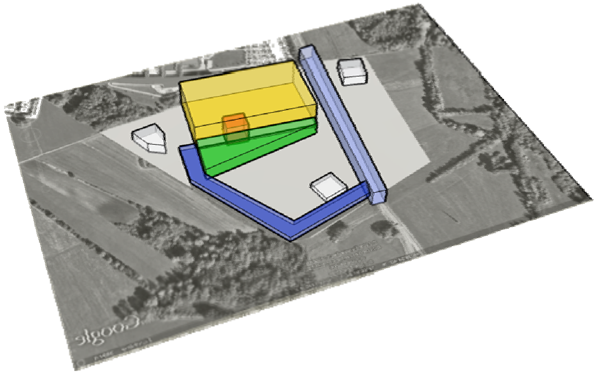
UAV Platform

[www.microdrones.com]

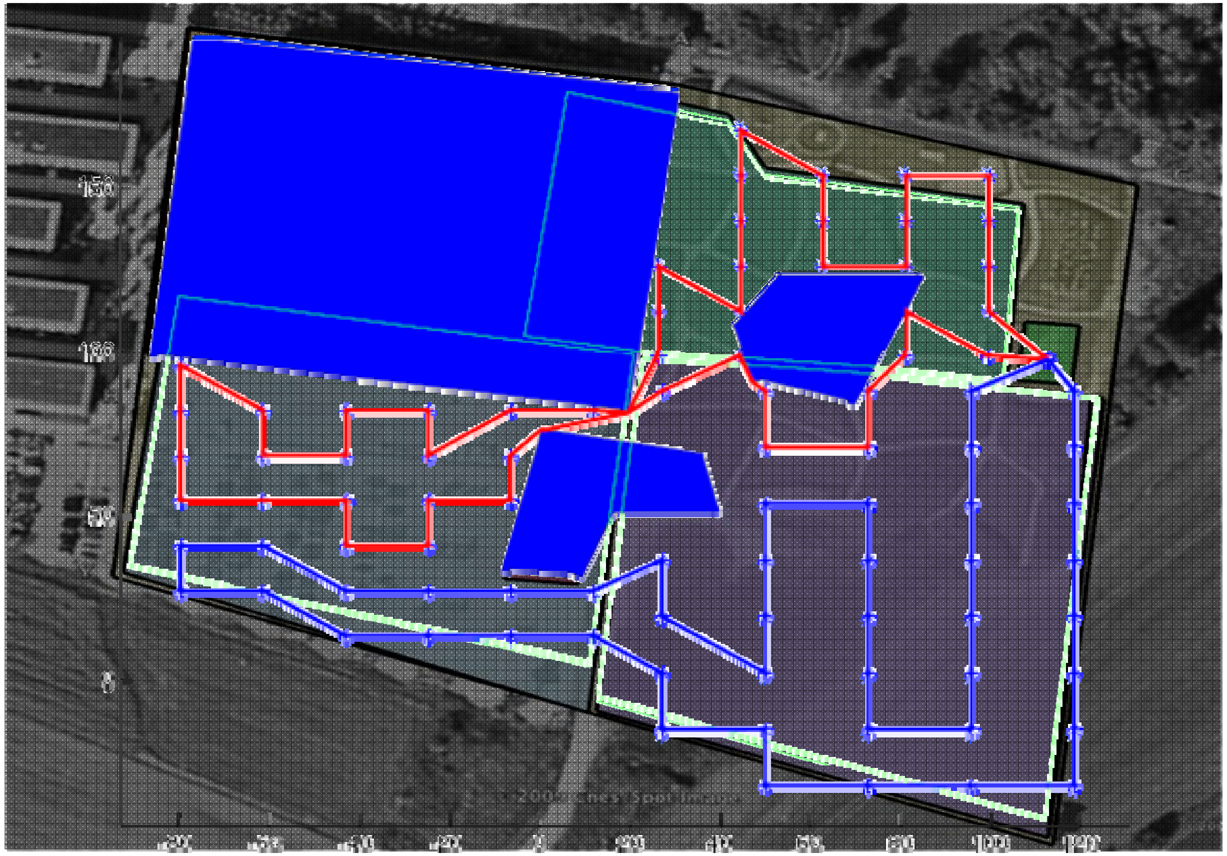
- Battery-powered quatrocopter
    - about 1 m size, 200g payload
    - 20 minutes operation time
    - onboard camera 10MPixel
- GPS-based waypoint navigation
- Communication
    - Uplink (RC channel): remote control;
    - Downlink (2.4 GHz channel): flight data, (low-resolution) images/video

cDrones: Mission Planning

- Find the optimal routes & formation for a small group of UAVs
    - Sequence of waypoints & actions
- Given the scenario description
    - Simplified 3D representation
    - Areas of interest, no-fly zones
- Considering various constraints
    - Power, flight time
    - Target resolution, update rate etc.
- Current approach
    - CSP-based planning



cDrones: Mission Planning (2)



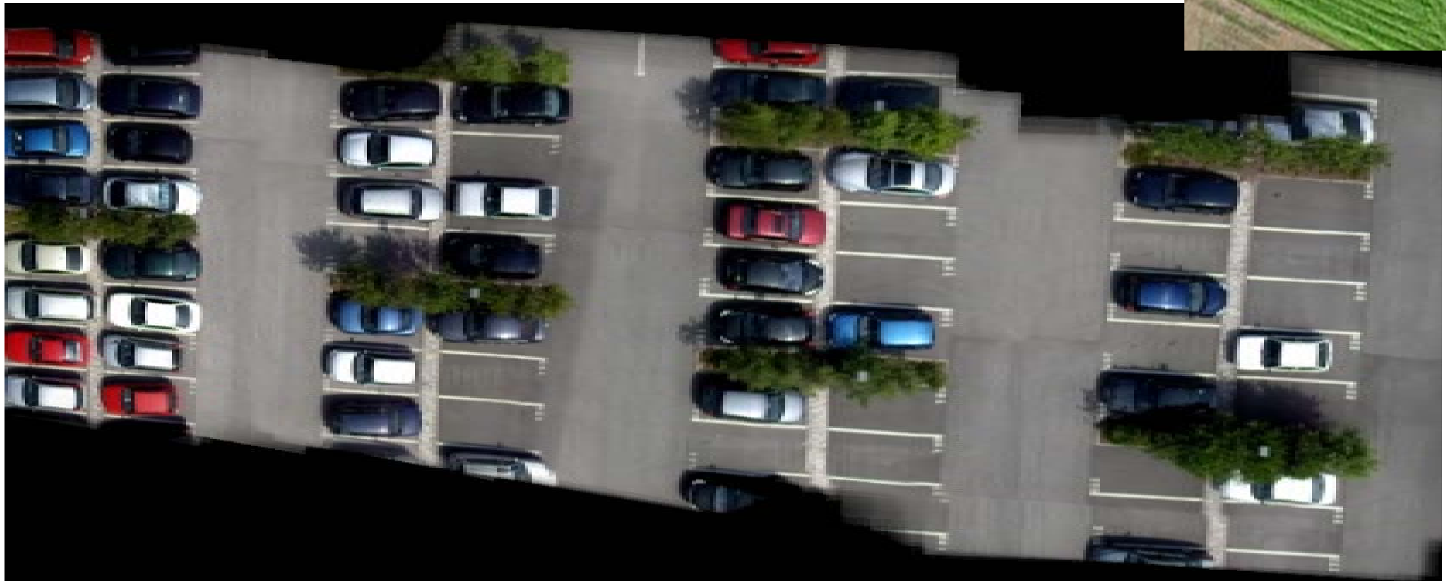
Input: Scenario definition
Picture point generation
Route generation (optimize coverage)

cDrones: UAV Formation

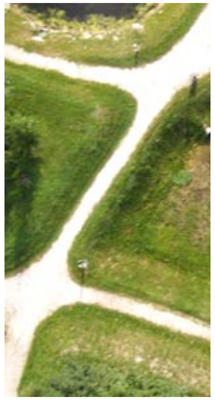
- Build and maintain a formation
  - e.g. “parallel”, “triangle” (of 3-5 UAVs)
  - Follow the waypoint routes given by mission planning
- Exploit GPS and IMU data of UAVs
  - Guarantee safe flight routes for individual UAVs
  - No online obstacle detection
- Provide real and simulation environment
  - Simplify testing
  - Modeling the UAV dynamics

cDrones: Aerial Imaging

- UAVs connected via wireless network (eg 802.11)
- Preliminary imaging
  - Mosaicing using COCOA



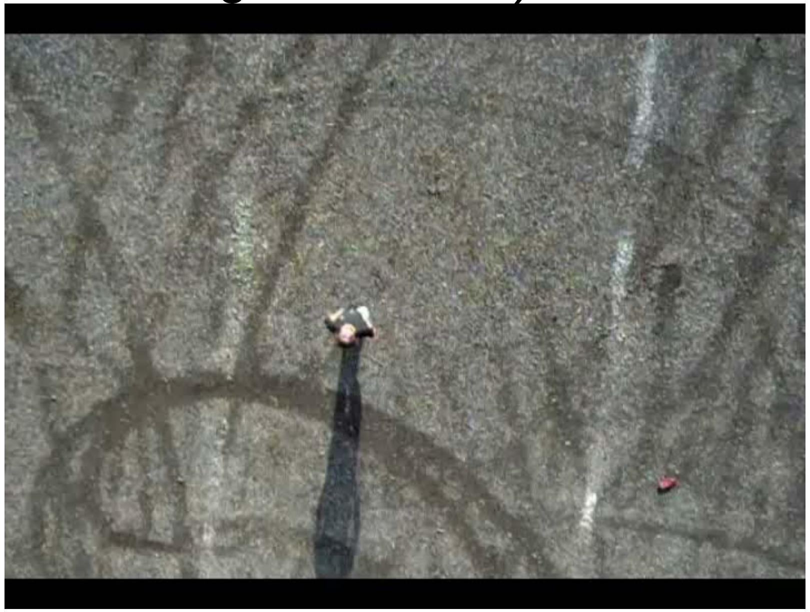
cDrones: Aerial Imaging



Mosaicking individual images to „overview image“

cDrones: Aerial Imaging (2)

- Video analysis
  - Alignment of frames (ego motion compensation)
  - Object detection & tracking (relative movement within aligned frames)



raw video

analysis

(Potential) further Applications

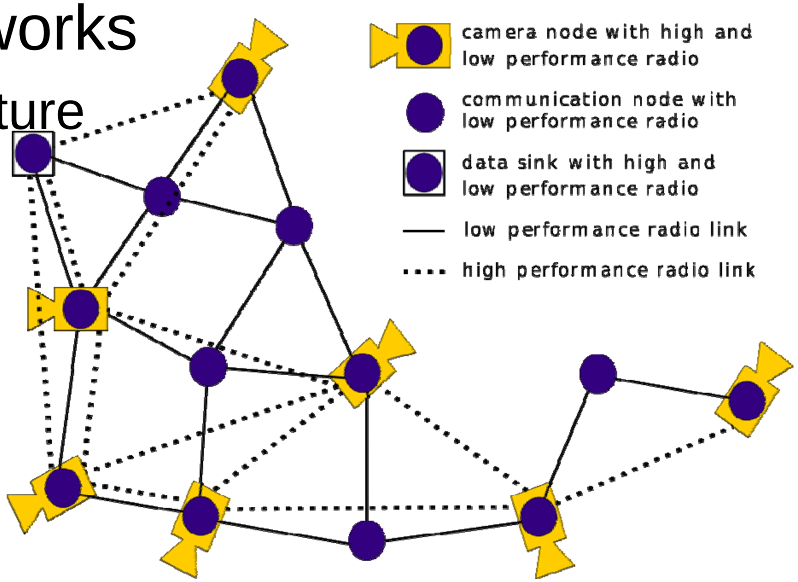
- Entertainment (computer games)
  - in 3D environments
- „Smart Rooms / Smart Environments
  - detection gestures, sign language, room occupancy …
- Environmental monitoring
  - sensor fusion, habitat monitoring
- Security
  - Safety enhancement (trains, cars), access control, surveillance
- „Virtual Reality“
  - augment real world with digital information
- …

Challenges

#1: Architecture

How to design resource-aware nodes and networks

- Low-power (high performance) camera nodes
  - Dedicated platforms: vision processors, PCBs, systems
  - Many examples: CITRIC, NXP
- Visual/Multimedia Sensor Networks
  - Topology and (multi-tier) architecture
  - Multi-radio communication



#2: Networking

How to process and transfer data in the network

- Ad hoc, p2p communication over wireless channels
  - Providing RT and QoS
  - Eventing and/or streaming
- Dynamic resource management
  - (local) computation, compression, communication, etc.
  - Degree of autonomy: dynamic, adaptive, self-organizing
  - Fault tolerance, scalability
  - Network-level software, middleware

[Doblander_ACMTECS2009], [Rinner_ICASSP2007] , [Shin_2007]

#3: Distributed Sensing & Processing

Where to place sensors and analyze the data

- Sensor placement, calibration & selection
  - Optimization problem
  - Distributed approaches eg., consensus, game theory

    [Soto_CVPR2009], [Devarajan_PIEEE2008]
- Collaborative data analysis
  - Multi-view, multi-temporal, multi-modal
  - Sensor fusion

    [Kushwaha_ICCCN2008], [Cevher_TransMM2007]

#4: Mobility

How to exploit networks of mobile cameras

- Ubiquitous mobile cameras
  - PTZ, vehicles, robotics etc.
  - Mobile phones
- Advanced vision algorithms
  - Ego motion, online calibration
  - Closed-loop control, active vision

#5: Usability

How to provide useful services to people

- Ease of deployment, maintenance
  - Self-* functionality
- Privacy and Security

  [Serpanos_PIEEE2008]
- Killer application

Conclusion

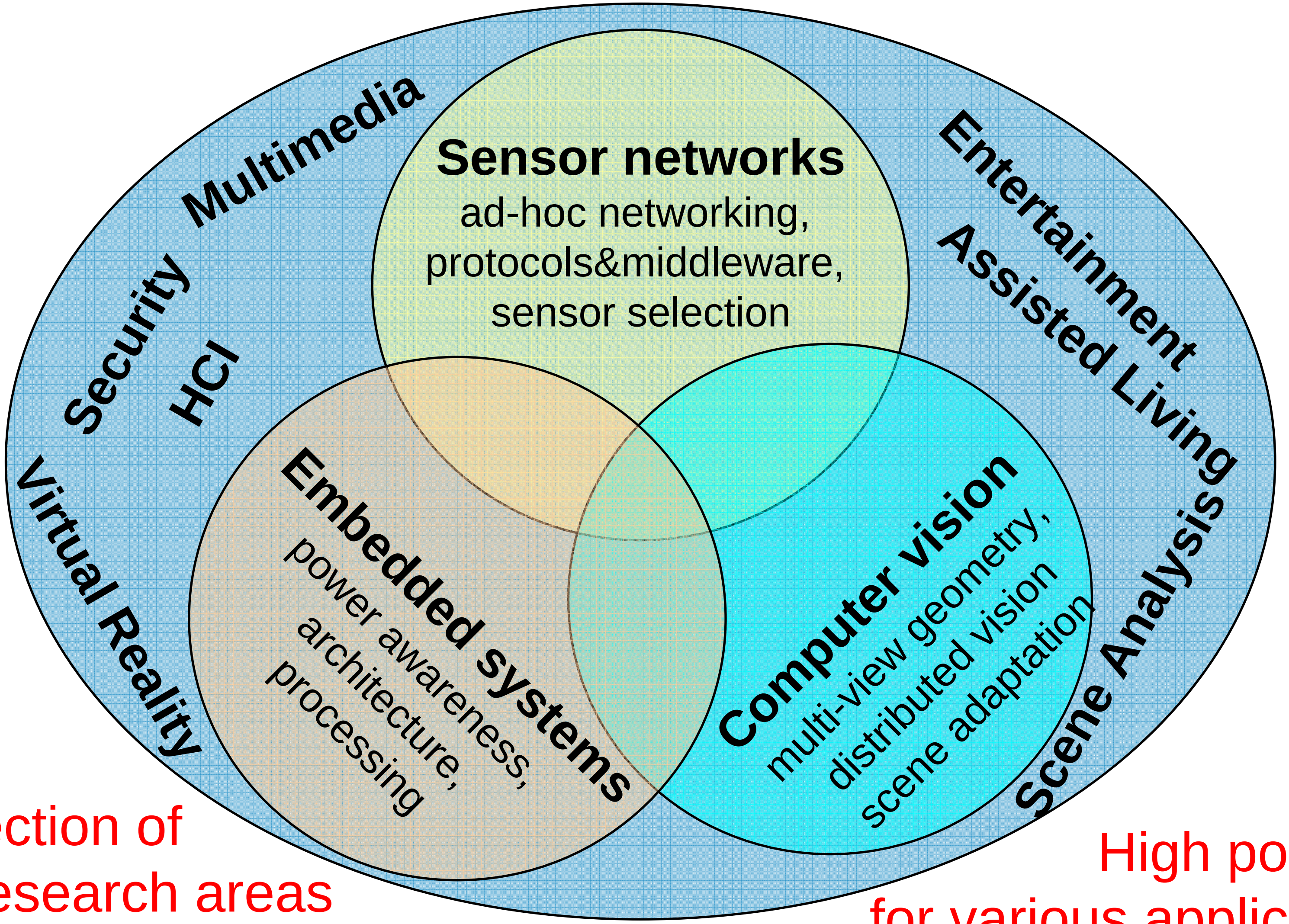
Smart Cameras

- combine
  - sensing,
  - processing and
  - communication

  in a single embedded device

- perform image and video analysis in real-time closely located at the sensor and transfer only the results
- collaborate with other cameras in the network (multi-camera system)

DSC is Interdisciplinary Research



Sensor networks
ad-hoc networking,
protocols&middleware,
sensor selection

Embedded systems
power awareness,
architecture,
processing

Computer vision
multi-view geometry,
distributed vision
scene adaptation

Intersection of
“hot” research areas

High potential
for various applications

Further Information

- Mail

  Pervasive Computing
  Lakeside B02b
  9020 Klagenfurt
- P: +43 463 2700-3670
- F: +43 463 2700-3679
- W: pervasive.uni-klu.ac.at

“Interactive & Cognitive Environments”

- Erasmus Mundus Joint Doctorate
  - University of Genua (Coordinator)
  - Klagenfurt University
  - UPC Barcelona
  - TU Eindhoven
  - Queen Mary University of London

Education and Culture

Erasmus Mundus

- 15 scholarships per year for EU and non-EU PhD students
- Starting in Fall 2010 (until 2017)
- More info coming soon
  - check http://pervasive.uni-klu.ac.at