Challenges and Opportunities of Distributed Smart Cameras

Prof. Bernhard Rinner

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

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. Smart Cameras
   Integration of sensing & processing

2. Distributed Smart Cameras
   Distribution of sensing & processing

3. Applications & Case Studies

4. Challenges

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

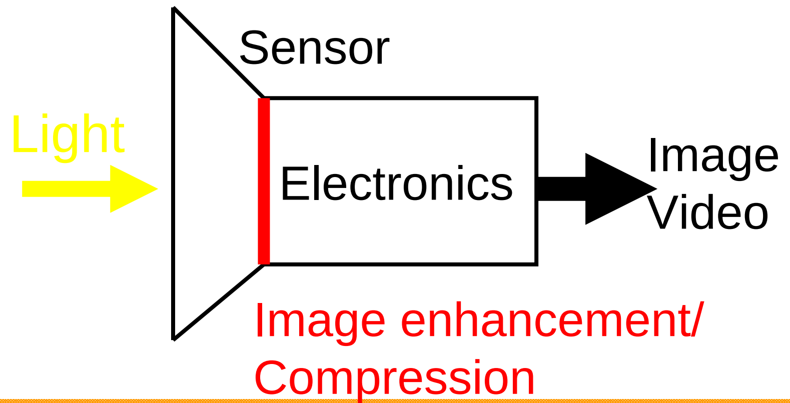
- Optics and sensor
- Electronics
- Interfaces

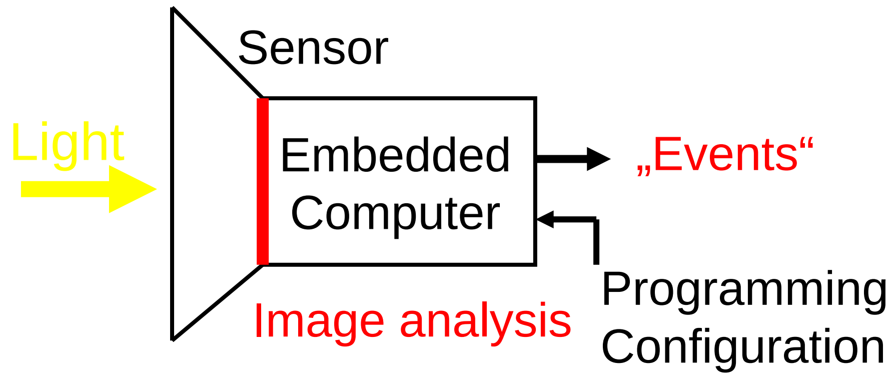
delivers data in form of (encoded) images and videos, respectively

Smart Camera

- Optics and sensor
- onboard computer
- Interfaces

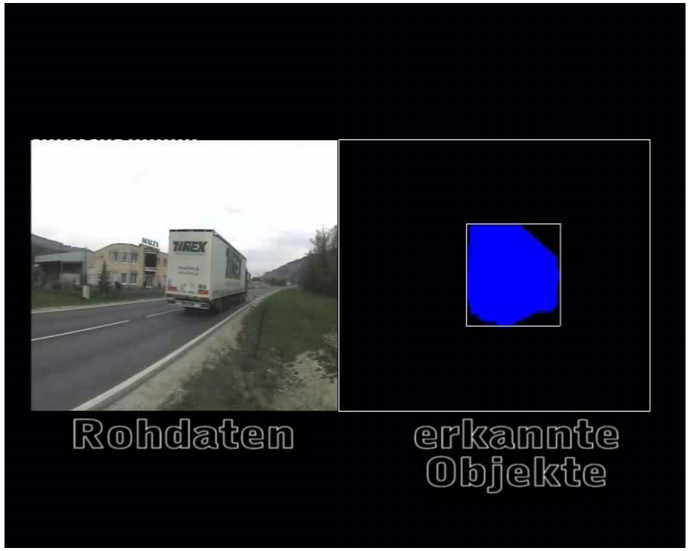
delivers abstracted image data is configurable and programmable

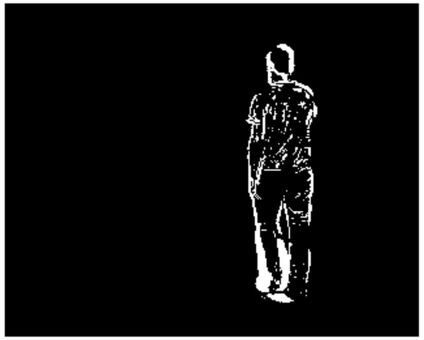




Smart Cameras look for important things

- 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

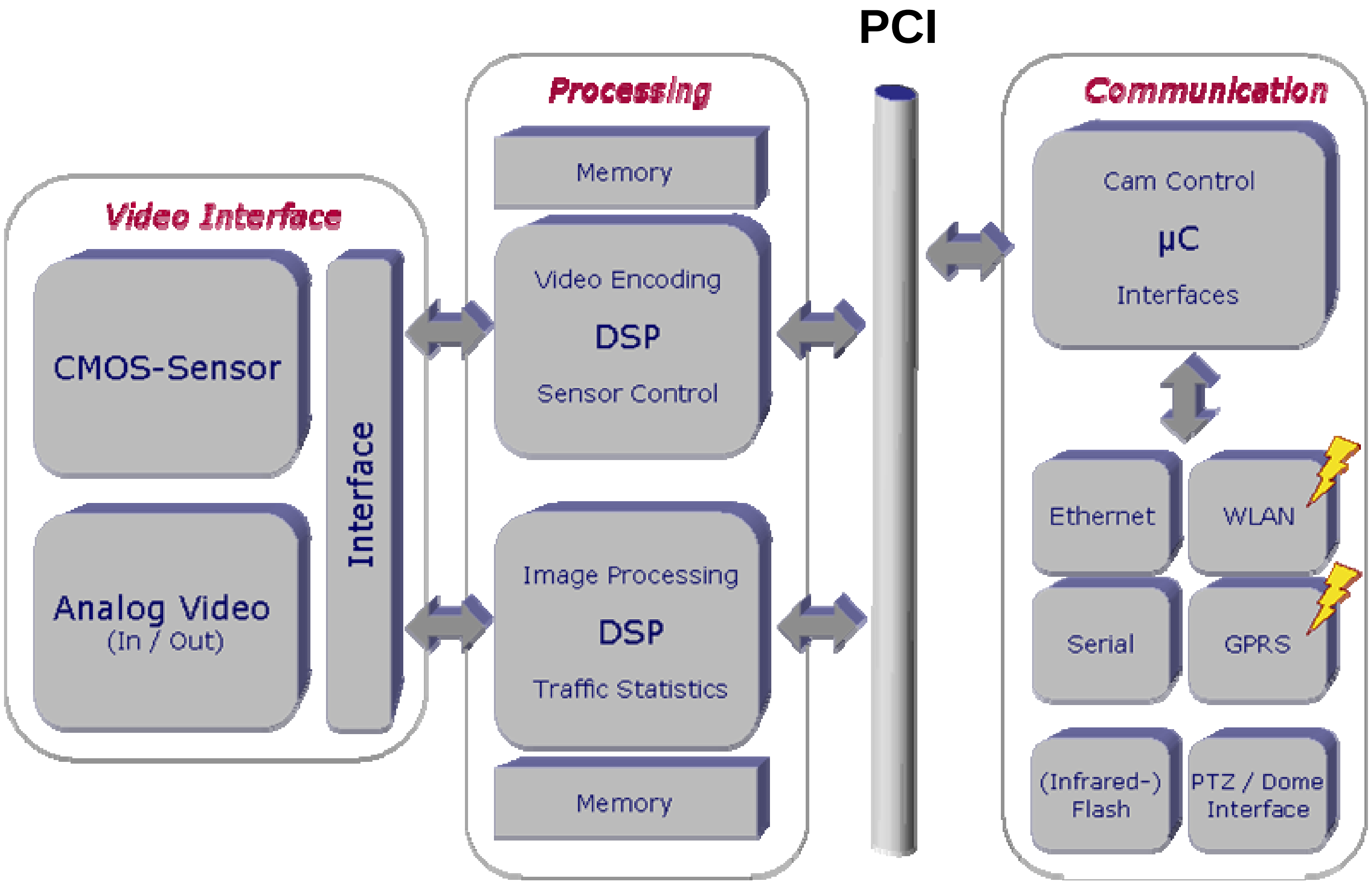
WiCa/NXP (Xetal SIMD)
50 GOPS @ 600mWatt

CMUcam3 (ARM7)
60 MIPS @ 650mW

CITRIC (PXA270)
660 MIPS @ 970mW

Scalable SmartCam Architecture

[Bramberger_Computer2006]



(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., PIEEE 2008]
[Rinner et al., ICDSC 2008]

Distributed Smart Cameras

Smart Cameras collaborate

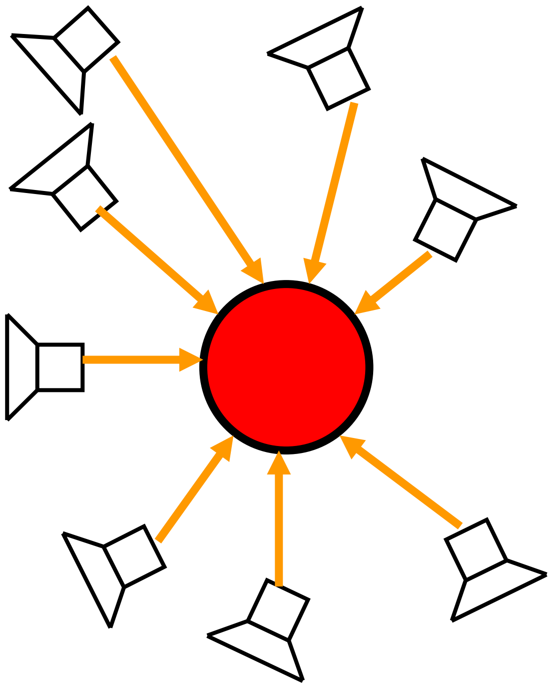
- Connecting autonomous cameras in a network
  - exploit smart cameras' capabilities (eg. avoid raw data transfer)
  - relax centralized/hierarchical structure of MC networks
  - introduce dynamic configuration (structure and functionality)
- Distributing sensing & processing introduce challenges
  - camera selection and placement
  - calibration & synchronization
  - distributed processing
  - data distribution and control, protocols and middleware
  - distributed computer vision (distributed signal processing)

(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
- …

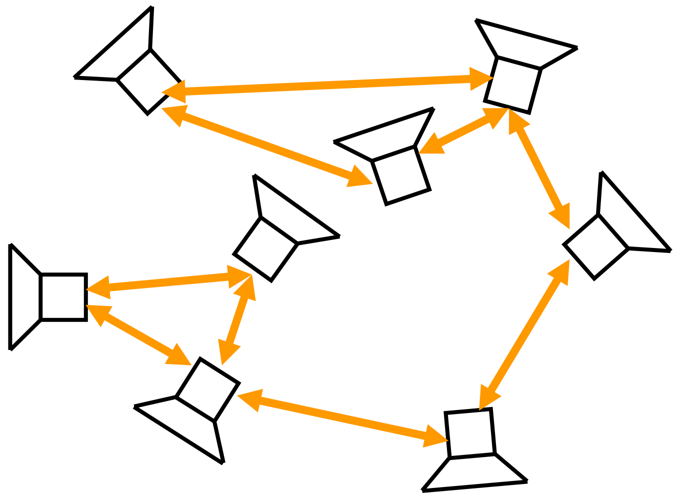
Networking

Traditional Camera Networks



Cameras stream images/ videos to „server“

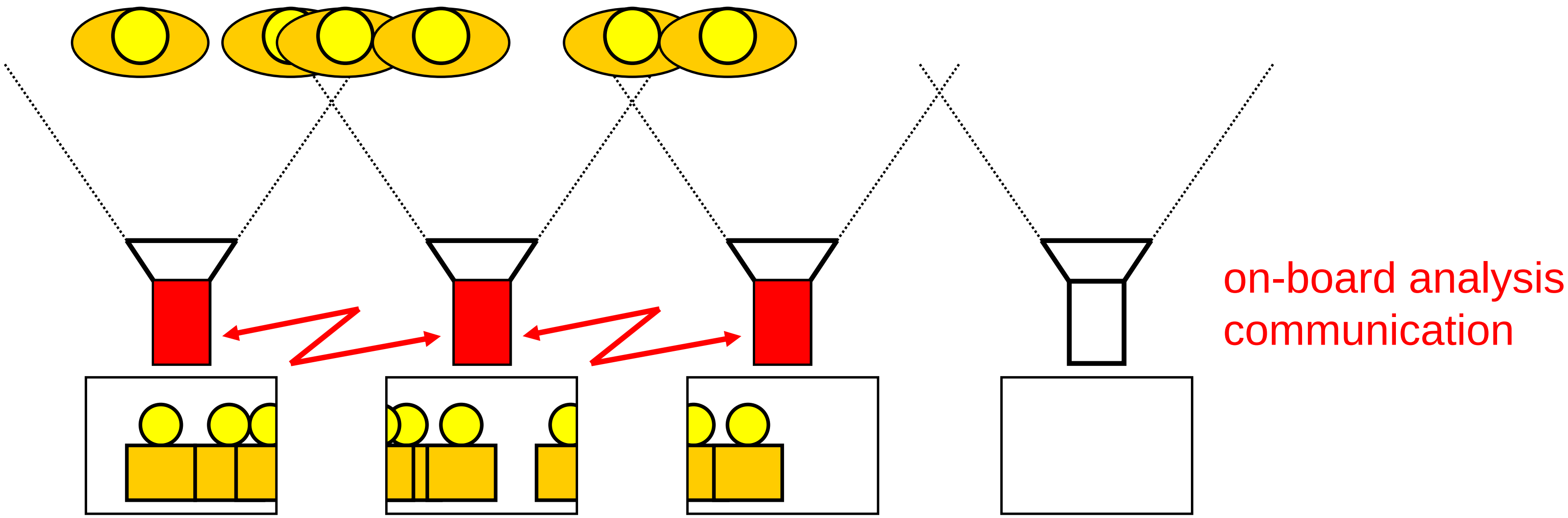
Smart Camera Networks



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

Distributed Processing in Network

- Example: autonomous tracking of mobile objects among multiple cameras



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

Applications & Case Studies

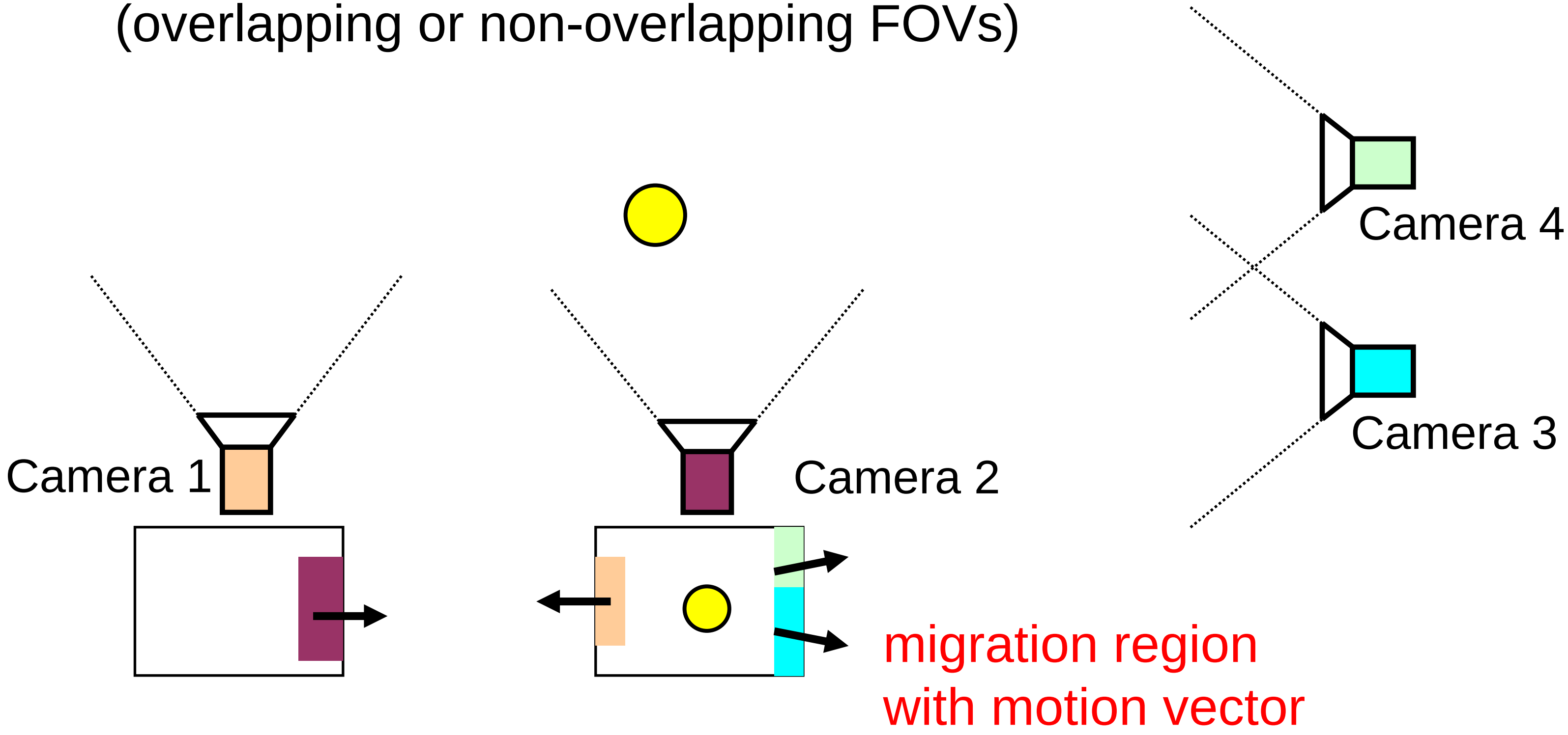
Autonomous Multi-Camera Tracking

[EURASIP JES 1/2007]

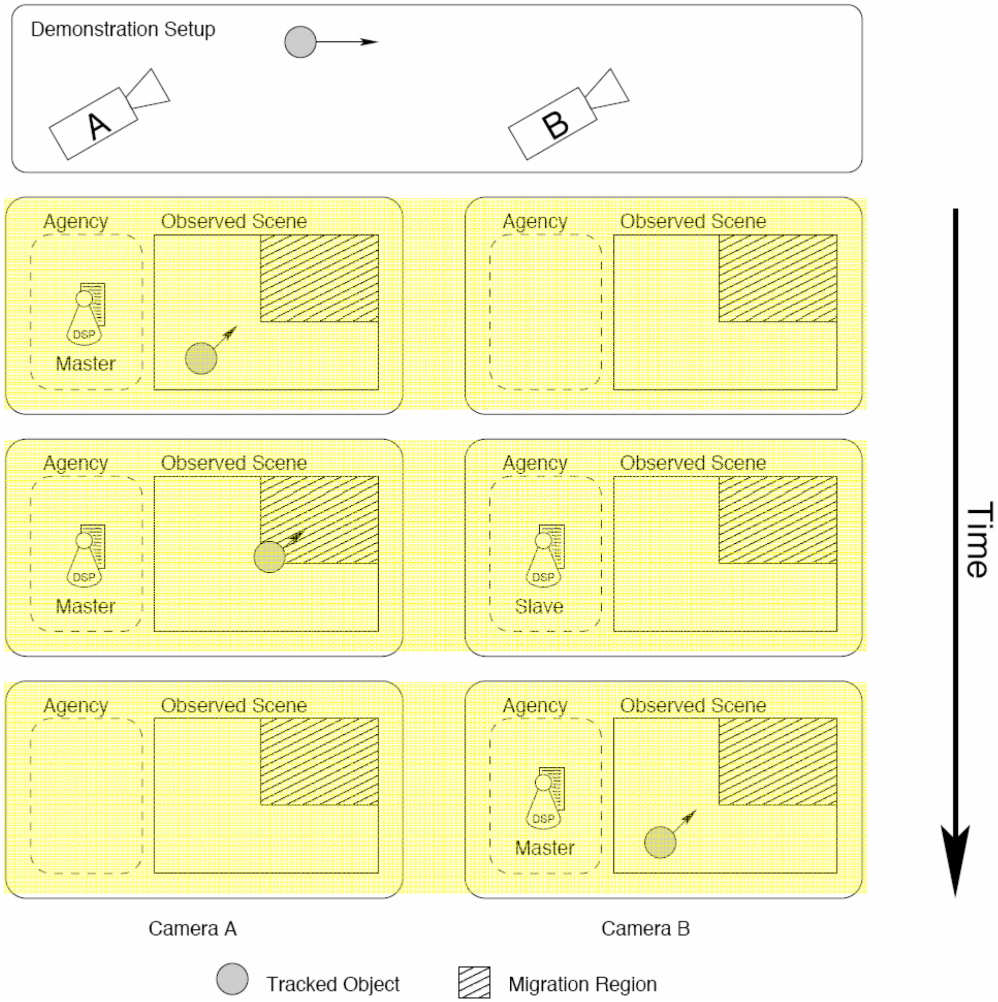
- Assumptions for multi-camera tracking
  - implement on distributed embedded smart cameras
  - avoid accurate camera calibration
  - do not rely on central coordination
- Important design questions
  - What (single-camera) tracking algorithm to use?
  - How to coordinate the cameras?
    i.e., distributed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

Spatial Relation among Cameras

- Static camera neighborhood relation
  - important for determining "next camera(s)"
  - based on pre-defined "migration region" in camera's FOV (overlapping or non-overlapping FOVs)



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

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

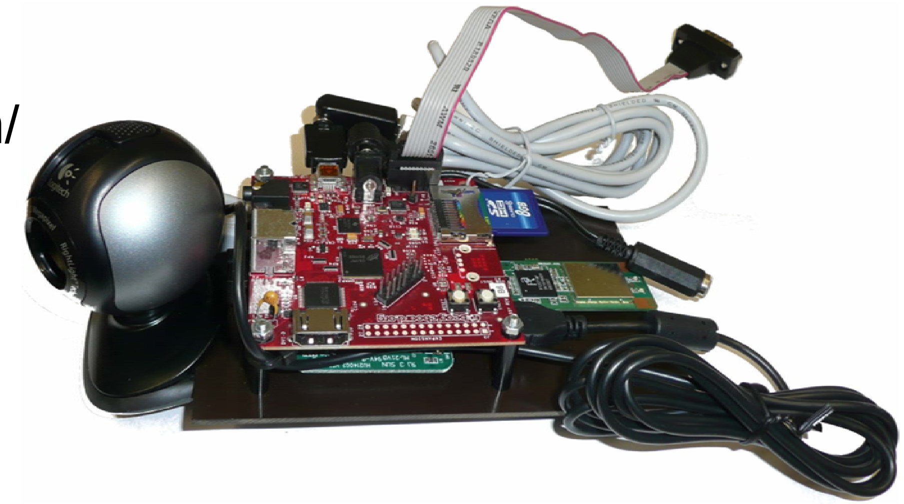
Dual Radio Communication in DSC

[Winkler_PerCom2009]

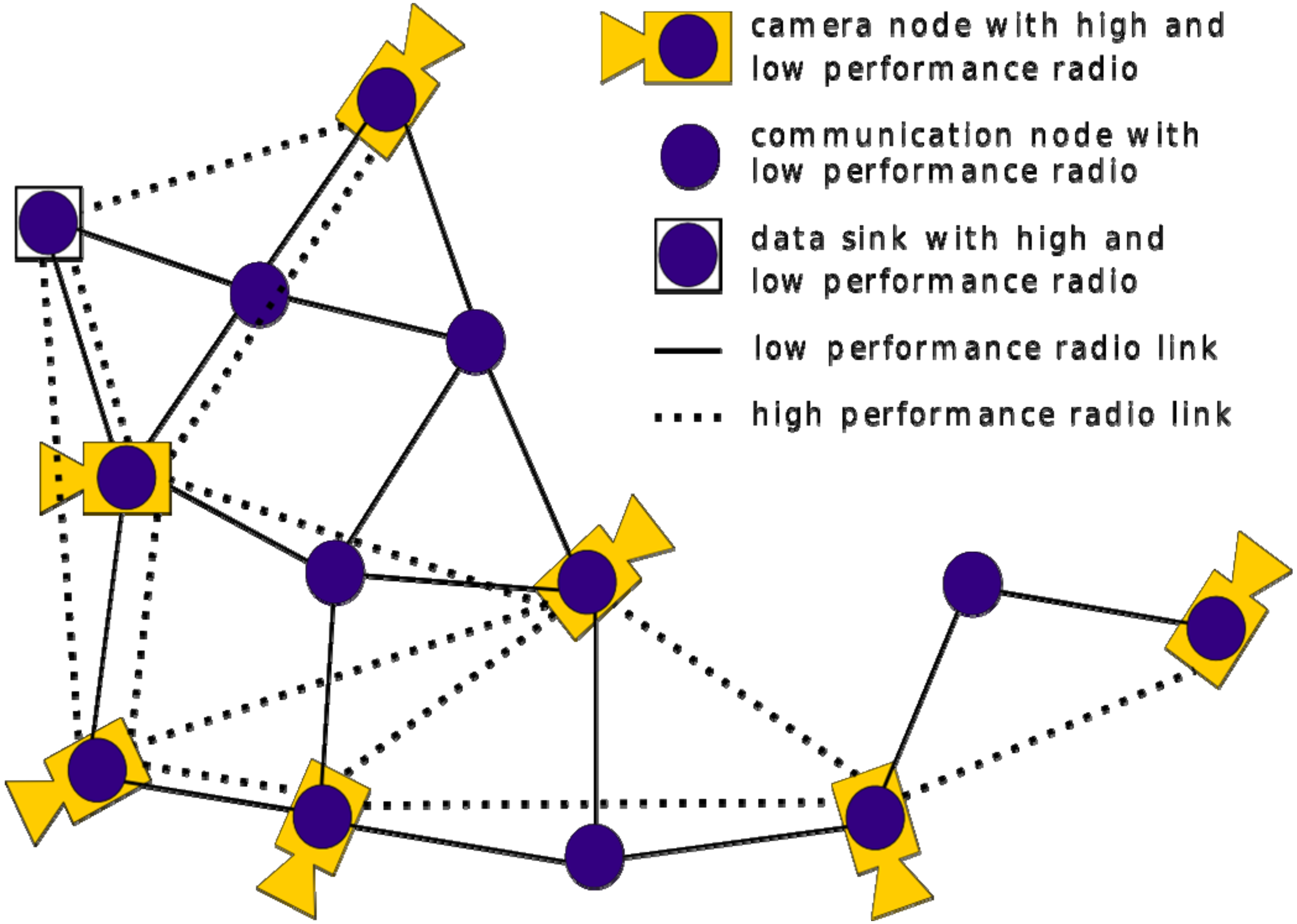
- Tradeoff among bandwidth, power consumption and streaming requirements in DSC networks
- One approach: dual radio networks
- Equip nodes with two radios: low-bandwidth & high-bandwidth
- Use low-bandwidth radio for normal operation
  - coordination, eventing,
  - transfer of low-resolution (still)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

Camera Node Prototype

- 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-in/



PSC Network Architecture



PSC Demo: Tracking

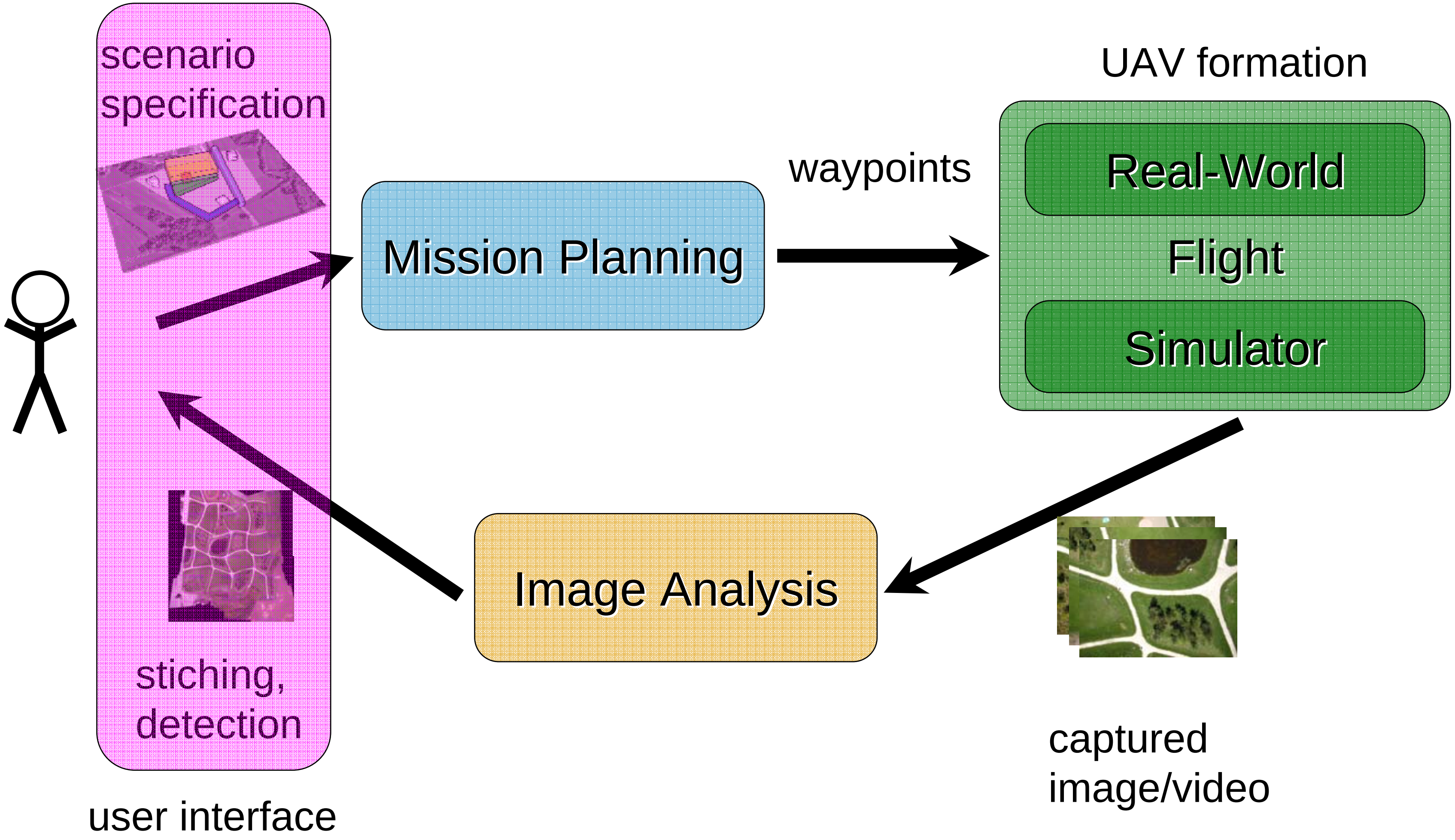
- Demonstrate tracking by using only low-bandwidth radio
  - initially transfer background image
  - perform tracking onboard
  - transfer tracking result (bounding box); 8 bytes/frame

Collaborative Microdrones

[Quaritsch_AUTONOMICS2008]

- 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

High-level “Processing Loop”



UAV Platform

[www.microdrones.com]

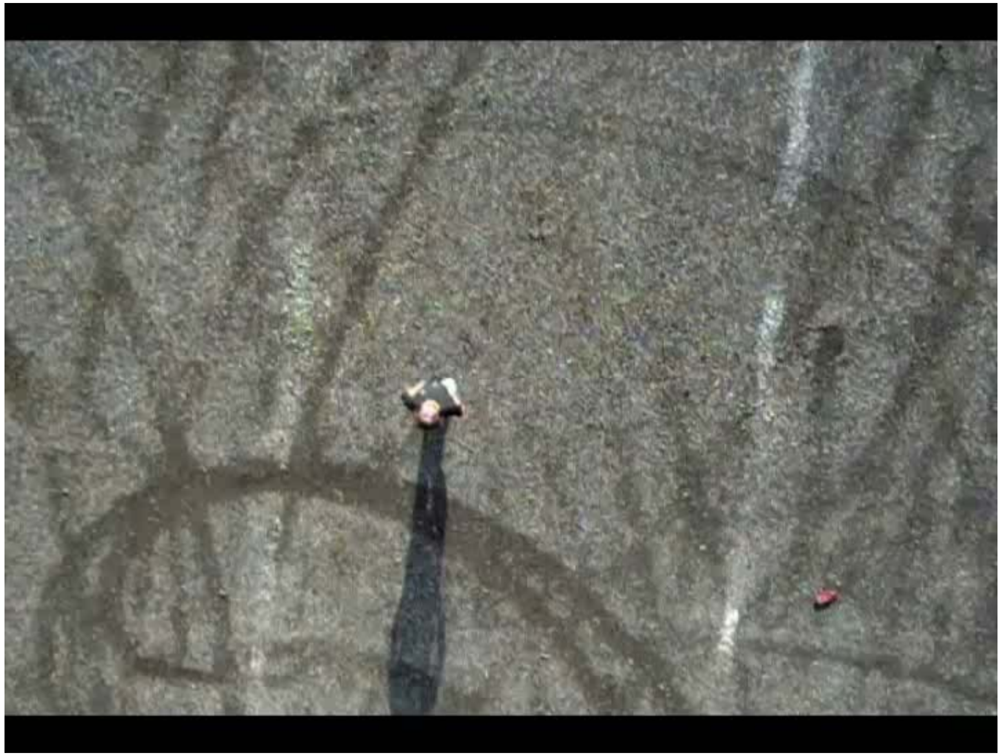
- Battery-powered quatrocopter
  - about 1 m size, 200g payload
  - 20 minutes operation time
  - onboard camera 10MPixel
- GPS-based waypoint navigation
- Communication
  - Uplink (RC channel): remote control;
  - Downlink (2.4 GHz channel): flight data, (low-resolution) images/video

Bird's Eye View

- 10 MPixel still images
- Video@25 fps
- Image quality
  - Ego motion

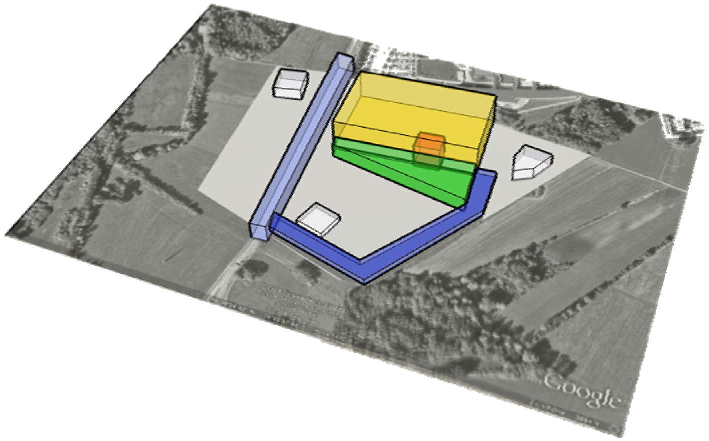
Bird’s Eye View

- Examples
  - Altitude: 20 – 60 m
  - Camera yaw: 45 – 90°
- Images/videos
  - Ego motion
  - With GPS/IMU data



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

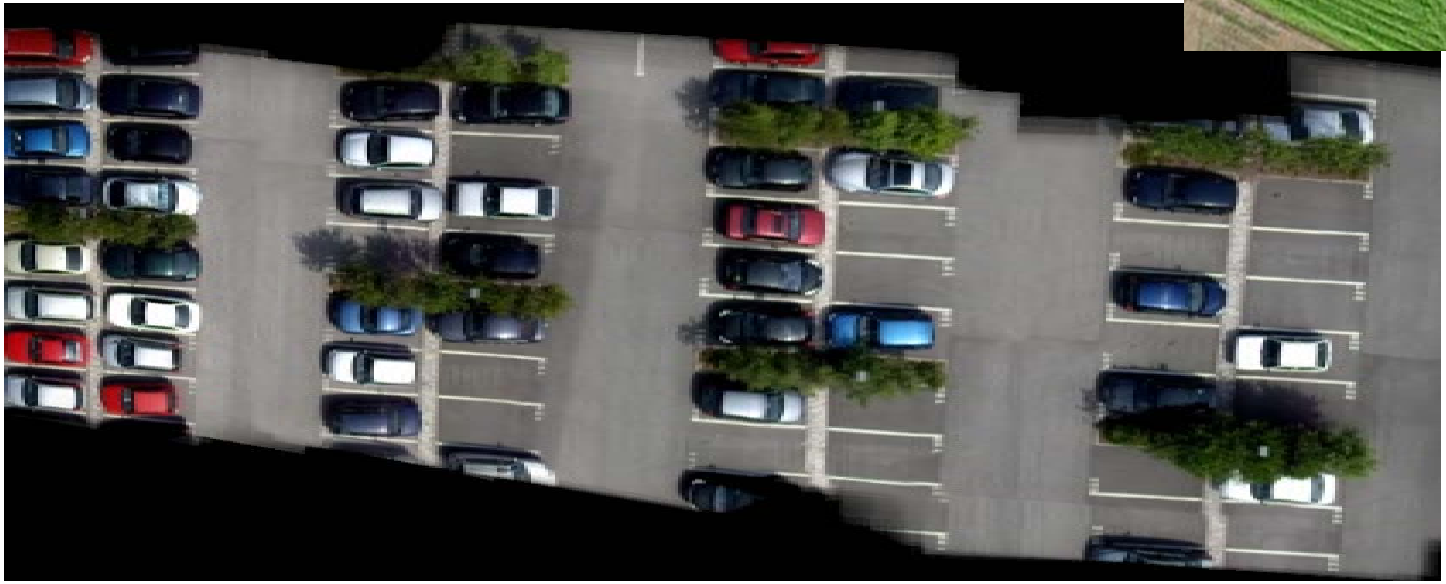


cDrones: UAV Formation

- 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



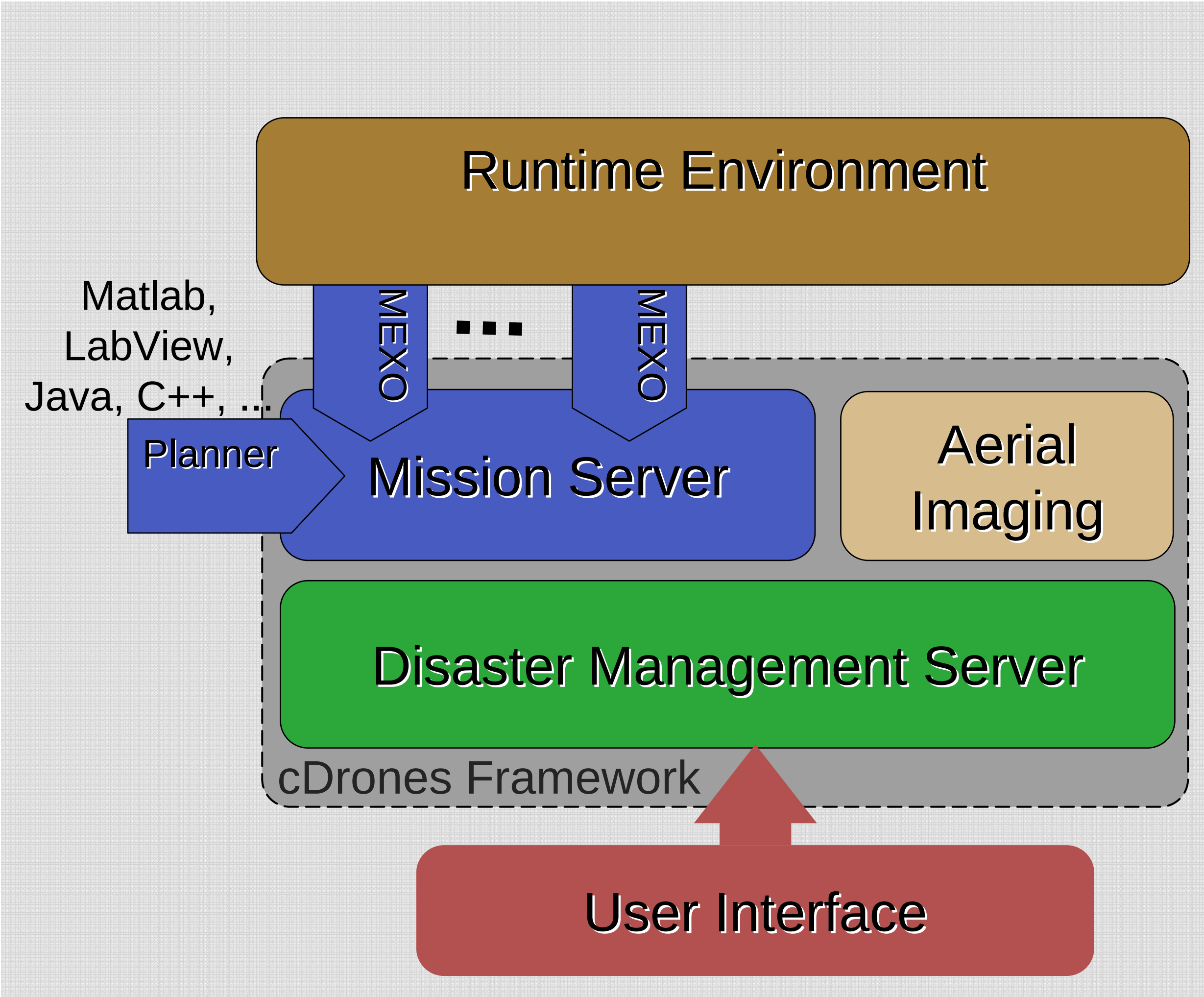
Research Topics

- UAV formation
  - Build, maintain, transfer into formations of 3-5 UAVs
  - Real-time processing of data over wirelessly connected UAVs
  - Networked control
- Mission planning
  - Find the optimal routes & formations (and sequence of actions) for the UAVs
  - Given a “scenario specification”
- Aerial imaging
  - Compose an overall (stiched) image
  - Detect objects of interest

Current Activities

- Integrate additional onboard computing platform
  - WLAN for up/downlink & interdrone communication
  - Online update of waypoints
  - Better camera control
- Software Framework
  - for integration of main components
  - transparent to simulated/real-world UAVs

Framework Architecture



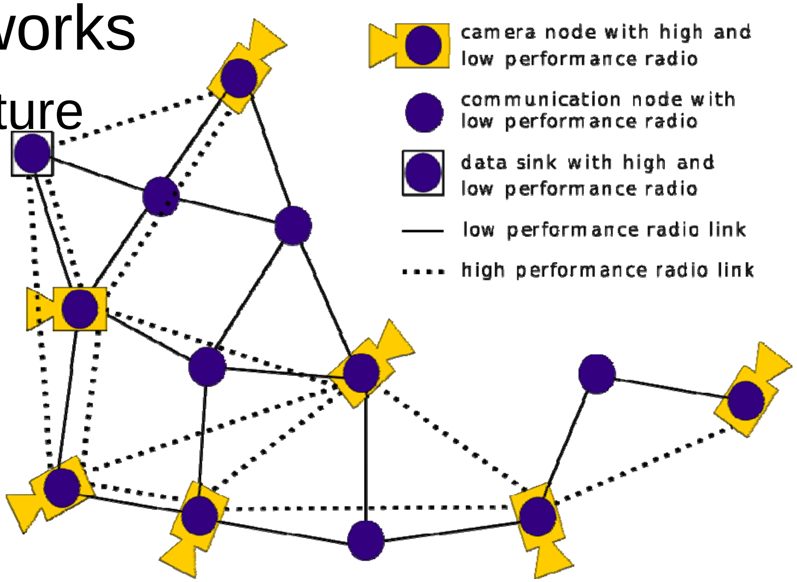
- ✓ Messages
- ✓ Communication Protocol
- ✓ MS, DMS
- ✓ Inter-module Communication
- ✓ Interface Planner
- ✓ Interface UI
- ✓ Unit tests
- ✓ Simple client

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

(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
- …

Trends and Challenges

- From static to dynamic and adaptive
  - Adaptation & learning (networking, functionality, scene,…)
- From small to large camera sets
  - E.g., more interest in statistics on behavior (instead of individuals)
- From vision-only to multi-sensor systems
  - Fusion of data from multiple (heterogeneous) sensors
- Development process of DSC
  - How to model, develop, deploy, operate, maintain applications
- Privacy & Security
  - Important cross-layer topic for user acceptance
- …

Conclusion

Smart Cameras

- combine
  - sensing,
  - processing and
  - communication

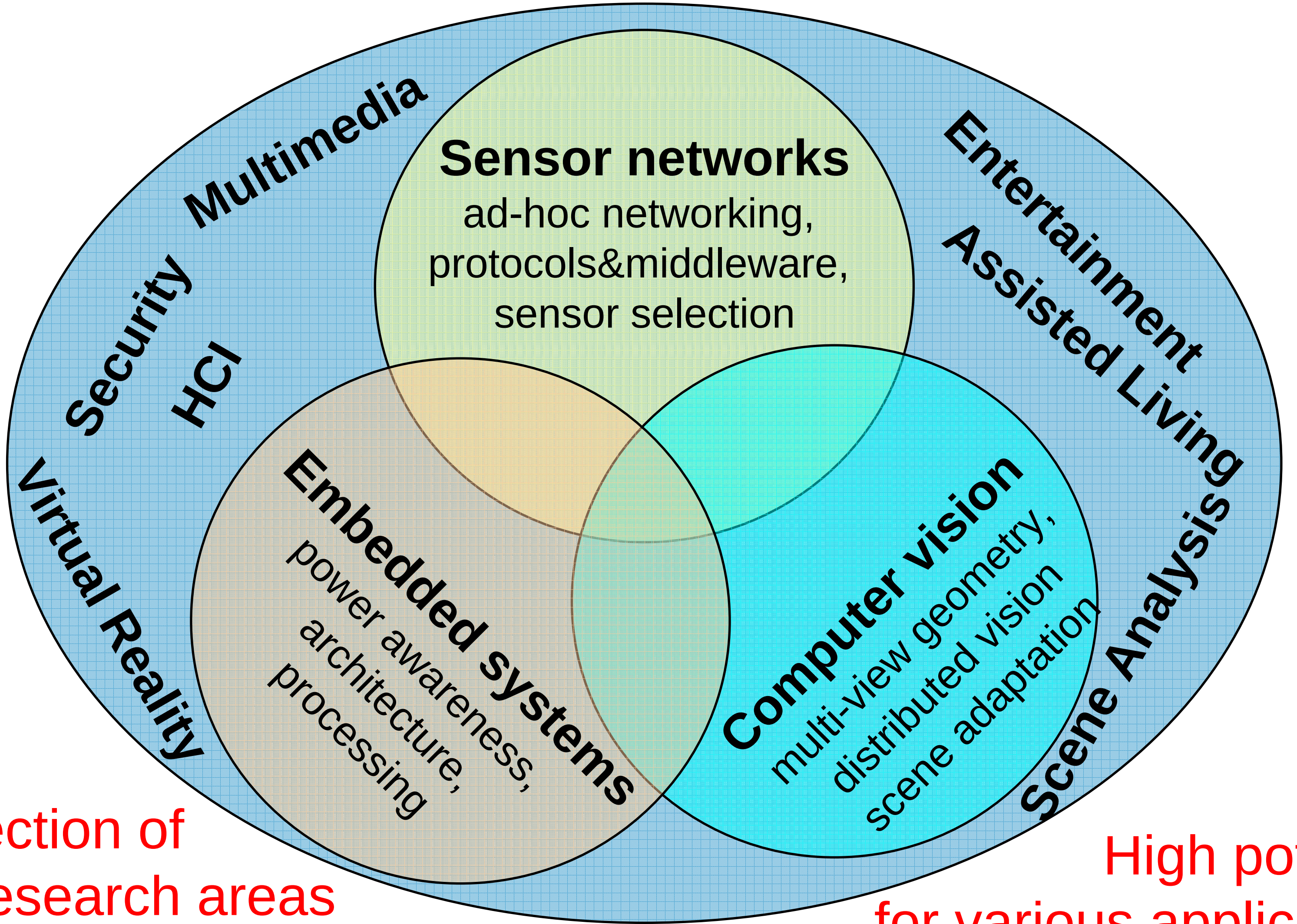
  in a single embedded device

- 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)

Smart Cameras as Key Technology

- For many applications including
  - Life Sciences
  - Security & Monitoring
  - Traffic
  - Entertainment
- Distributed cameras migrate to smart networks, which helps to overcome „hard problems“
  - occlusion
  - communication bandwidth
  - energy supply
  - reliability

DSC is Interdisciplinary Research



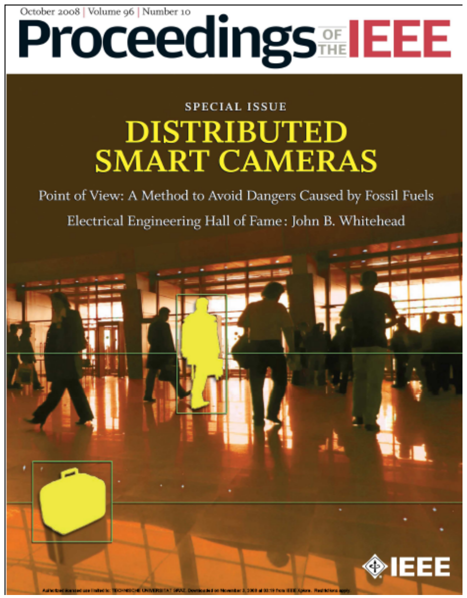
Intersection of "hot" research areas

High potential for various applications

To Probe Further

- **ACM/IEEE Int. Conf. on Distributed Smart Cameras**

Como, Italy (Aug30-Sep2, 2009) www.icdsc.org



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