Distributed Smart Cameras

Prof. Bernhard Rinner

Pervasive Computing
Institut für Vernetzte und Eingebettete Systeme
Alpen-Adria Universität Klagenfurt
http://pervasive.uni-klu.ac.at
Cameras are ubiquitous …
... and are important for many Applications

- Entertainment
- Security
- Production
- Medical application
- Environment
- Automation
- Robotics
- Multimedia
- Biometric
- ...

B.Rinner
Agenda

1. Traditional Camera Networks
   Advantages & Challenges

2. Smart Cameras
   Principle & Architecture

3. Distributed Smart Cameras
   Research Challenges
   Distributed Computing
   Applications
Traditional Camera Networks
Camera Networks

Advantages and challenges of multiple cameras
Advantage: 3D Information

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

Object has different coordinates 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
Herausforderung: Registrierung (2)

G. Yang, C.V. Stewart, M. Sofka, C. Tsai, PAMI 2007; http://www.vision.cs.rpi.edu/gdbicp/
Example: Surveillance Systems

- Large, complex system
  - many (wide spread) sensors/cameras
  - visualization at central monitoring station
  - 24/7 operation

- Characteristics
  - provides varied degrees of assistance to humans/operators, main focus: display & record
  - centralized system: computation takes place at monitoring station
  - static configuration: physical sensors & functionality

- Challenges for “intelligent” surveillance systems
  - increased functionality & flexibility; autonomous operation
  - standards, sensor integration, open systems
  - many non-technical issues, eg, privacy
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
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

Sensor
Electronics
Image enhancement/
Compression

Light

Sensor
Embedded
Computer

"Events"
Programming
Configuration

Image analysis

Image
Video

Light
Smart Cameras look for important things

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

© CMU
Scalable SmartCam Architecture

[IEEE Computer 2/2006]
SmartCam Prototypes

1. generation (single DSP)
   - COTS (NVDK, Ateme)

2. generation (multi-DSP & processor)
   - COTS (Intel baseboard, NVDKs)
   - 3 variations (different host processors)
     XScale PXA, XScale IXP, P4M

3. generation
   - PCB (10 x 25 cm), IXP+C6415+C6455
   - Spartan II for sensor interface&preprocessing
   - currently under development
Prototypes with different Performance

- 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
## (Selected) Smart Camera Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Year</th>
<th>Platform</th>
<th>Distribution/Proc.</th>
<th>Autonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Moorhead&amp;Binni]</td>
<td>1999</td>
<td>ASIC</td>
<td>local</td>
<td>static</td>
</tr>
<tr>
<td>VISoc [Albani]</td>
<td>2002</td>
<td>SOC</td>
<td>local</td>
<td>static</td>
</tr>
<tr>
<td>[Wolf et al.]</td>
<td>2002</td>
<td>DPS (PC)</td>
<td>local</td>
<td>static</td>
</tr>
<tr>
<td>[Dias&amp;Berry]</td>
<td>2007</td>
<td>FPGA</td>
<td>local</td>
<td>active vis.</td>
</tr>
<tr>
<td>[Bauer]</td>
<td>2007</td>
<td>DSP</td>
<td>local</td>
<td>static</td>
</tr>
<tr>
<td>GestureCam [Shi]</td>
<td>2007</td>
<td>FPGA</td>
<td>local</td>
<td>static</td>
</tr>
<tr>
<td>[Micheloni et al.]</td>
<td>2005</td>
<td>(PC)</td>
<td>MC-tracking</td>
<td>PTZ</td>
</tr>
<tr>
<td>[Fleck&amp;Strasser]</td>
<td>2007</td>
<td>PowerPC</td>
<td>MC-tracking</td>
<td>static</td>
</tr>
</tbody>
</table>
## (Selected) Smart Camera “Sensors”

<table>
<thead>
<tr>
<th>System</th>
<th>Year</th>
<th>Platform</th>
<th>Distribution</th>
<th>Autonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclops [Rahimi]</td>
<td>2005</td>
<td>ATmega128</td>
<td>coll. tracking</td>
<td>static</td>
</tr>
<tr>
<td>CMUcam 3 [Rowe]</td>
<td>2007</td>
<td>ARM7</td>
<td>local proc.</td>
<td>static</td>
</tr>
<tr>
<td>Meerkats [Margi]</td>
<td>2006</td>
<td>StrongARM</td>
<td>coll. tracking</td>
<td>static</td>
</tr>
<tr>
<td>MeshEye [Hengstler]</td>
<td>2006</td>
<td>ARM7</td>
<td>local</td>
<td>rem. conf.</td>
</tr>
<tr>
<td>WiCa [Kleihorst]</td>
<td>2006</td>
<td>Xetal (SIMD)</td>
<td>coll. gesture rec</td>
<td>static</td>
</tr>
</tbody>
</table>
Distributed Smart Cameras
Smart Cameras collaborate

- Autonomous cameras connected in a network
  - no central server
  - collaborative analysis among multiple cameras
  - dynamic configuration (structure and functionality)

- Challenges for such collaborative DSC
  - 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

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

on-board analysis communication
Need for Dynamic Reconfiguration

- Dynamic communication among cameras
  - cameras may be included to or removed from network
  - communication pattern depends on observed scene

- Modification of functionality
  - adaptation/configuration of on-board image processing
  - „load” new algorithms

- Changes in available resources
  - hardware failures
  - different QoS requirements
Dynamic Reconfiguration

- What is a “configuration”?  
  - Executing services $s$ at various QoS-levels $q_s$ on different resources $r$  
  - Configuration of single camera $C_i = (s \times q_s \times r)$  
  - Configuration of network: $C = (C_1 \times \ldots \times C_i \times \ldots \times C_N)$  

- Find optimal configuration of the network at runtime  

- Various optimization parameters  
  - QoS, power consumption, reliability,…  
  - multi-criterion optimization  
  - requires a “system model”
Dynamic Reconfiguration Loop

Analysis of Contextual Information → Computation of (new) optimized Cluster Configurations

Sensing of Contextual Information ← Dynamic Reconfiguration

Objective Functions:
- Max. QoS
- Min. power

Distributed SmartCam-Cluster

Initial Cluster Configuration

User Desires

Energy Constraints

Application Specific Events
Dynamic Reconfiguration Applications

- Combined **power and QoS optimization** [IEEE AINA 2006]
  - exploiting dynamic power management
  - switching hardware components to different power levels
  - implemented on single- and multi-processor SmartCam

- Improving **fault tolerance** and service availability
  - requires onboard monitoring & diagnosis
  - in case of a detected fault, start a reconfiguration

- Application-specific reconfiguration
  - “download” services to cameras on demand
  - may overcome resource limitations on camera
Autonomous Multi-Camera Tracking

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

![Diagram showing camera neighborhood relation with overlapping FOVs and migration region.

Camera 1
Camera 2
Camera 3
Camera 4

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)

<table>
<thead>
<tr>
<th>Code size</th>
<th>15 kB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory requirement</td>
<td>300 kB</td>
</tr>
<tr>
<td>Internal state</td>
<td>256 B</td>
</tr>
<tr>
<td>Init color histogram</td>
<td>&lt; 10 ms</td>
</tr>
<tr>
<td>Identify object</td>
<td>&lt; 1 ms</td>
</tr>
</tbody>
</table>

CamShift (single camera)

<table>
<thead>
<tr>
<th>Loading dynamic executable</th>
<th>8 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initializing tracking algorithm</td>
<td>250 ms</td>
</tr>
<tr>
<td>Creating slave on next camera</td>
<td>18 ms</td>
</tr>
<tr>
<td>Reinitializing tracker on slave</td>
<td>2 ms</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>278 ms</strong></td>
</tr>
</tbody>
</table>

Multi-camera performance

B. Rinner
Application: Traffic Monitoring

- **Online traffic data**
  - classification/counting
  - lane utilization
  - fusion of audio & video

  [IEEE ICDSC-07]

- **Vehicle tracking**
  - speed estimation
  - traffic jam detection

[Leistner et.al CVPRW 2007]
Application: Privacy Protection

- Security with privacy protection
  - automatic head detection and tracking
  - encryption of head area

Boult, Univ. Colorado
Application: Assisted Living

- Assist Living for elderly people
  - detect and report dangerous situations
  - example: Detect fallen person
  - challenge: Privacy

Aghajan, Stanford
(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

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

Embedded systems
power awareness, architecture, processing

Computer vision
multi-view geometry, distributed vision, scene adaptation

Scene Analysis

Entertainment-Assisted Living

Virtual Reality

HCI

Security

Multimedia

Intersection of “hot” research areas

High potential for various applications
DSC-related Activities

- ACM/IEEE Int. Conf. on Distributed Smart Cameras
  - Stanford (Sep. 7-11) [www.icdsc.org](http://www.icdsc.org)
    - Workshops, Tutorials, PhD-Forum, ...

- Special Issue on Distributed Smart Cameras
  (Oct 2008) [Proceedings of the IEEE](http://proceedings.ieee.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