Pervasive Smart Cameras

Bernhard Rinner
The Digital Universe

• Forecasts from a recent IDC report [1]
  – „The amount of digital information in the world will grow to almost 35 trillion GByte by 2020.“
  – “The amount of digital information created already exceeds the available storage.” By 2020 this storage gap grows to more than 60 %
  – „Cameras play a significant part for data creation“


➢ Storing, analyzing, searching, protecting, etc. these huge amount of data becomes a real challenge
Ubiquitous Cameras

- We are surrounded by billions of cameras in public, private and business spaces
- Various well-known examples
  - Transportation
  - Security
  - Entertainment
  - …
- How to explore all the captured data?

➤ Different view on camera(s) required, applies especially for pervasive computing
Revolution in Cameras

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

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

- Huge amount of visual information is processed in a network of resource-limited embedded nodes in dynamic environment

- Make cameras smart, autonomous and collaborative
Agenda

- **Smart Cameras**
  - Introduction
  - Trends
- **Selected Applications**
  - Tracking
  - Configuration
  - Security & privacy
- **Challenges**
  - Research question
  - Conclusion
Smart Cameras
Traditional Camera Networks

- Cameras capture images/videos
- Raw or compressed data is streamed to central server
- Image data is displayed/archived/analyzed at central point
- Data and energy is transferred over wired infrastructure

➢ Centralized and static architecture, heavy infrastructure required

[Regazzoni et al. Special Issue on Video Communications, Processing and Understanding for Third Generation Surveillance Systems. Proc. IEEE. October 2001]
Making Cameras smarter

• Smart cameras integrate **sensing, processing and communication** on single embedded device

Traditional Camera

- Optics and sensor
- Electronics
- Interfaces

delivers data in form of (encoded) images or videos.

Smart Camera

- Optics and sensor
- Onboard computer
- Interfaces

delivers **abstracted image data** and is configurable and programmable

![Diagram showing traditional and smart camera systems](image.png)
Smart Camera Architecture

• Main components

Process data where it is captured

- Perform **image and video analysis in real-time** closely located at the sensor
- Deliver (only) abstracted events
- Reduce data transfer
  - From raw data to features or events
  - Example: tracking
- “Smart cameras look for important things”
Collaborate spontaneously

Traditional Camera Networks

Cameras stream images/videos to „server“

Smart Camera Networks

Cameras collaborate directly (spontaneous, p2p, ad-hoc)
Perform advanced in-network analysis

• From data collection and streaming to **dynamic collaboration**
  – More demanding processing possible (eg., online learning)
  – Analysis may change depending on network state and environment
• Exploit heterogeneous sensors
  – Different cameras (static, PTZ, RGB/IR …) but also audio, laser etc.
  – Perform intra and/or inter node fusion
  – Synchronization and calibration necessary
• Deliver multimedia data at required QoS level
• Support **autonomous** operation at **network level**
  – Self-* methods

Be aware of scarce Resources

• Major resource limitations
  – Processing power
  – Communication bandwidth
  – Onboard memory
  – Energy

• Various Prototypes

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

[Rinner et al. The Evolution from Single to Pervasive Smart Cameras. Proc. ICDSC 2008]
Why Networks of Smart Cameras?

- **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

➤ **Compare with sensor networks**
Selected Applications
Quest for Novel Pervasive Applications

• Some requirements
  – Easy deployment
  – Adaptive and scalable
  – Reactive/interactive
  – Secure- and privacy-aware

• Application domains
  – Distributed surveillance and security
  – Smart homes / smart buildings
  – Ambient intelligence
  – Human-computer interfaces
  – Mobile and robotic networks
  – Virtual reality systems
  – ...
Example 1: Multi-camera Tracking

- Track mobile objects autonomously among multiple cameras
- Computation follows (physical) object
  - requires spontaneous communication; distributed control & data
Autonomous Migration of Processing

• **Camera Handoff**
  
  – Initialize object tracker on “neighboring” camera(s)
  
  – Similarity function for object re-detection
  
  – Various approaches for neighbor selection, e.g.,
    a priori definition, learning, virtual markets

[Quaritsch, Kreuzthaler, Rinner, Bischof, Strobl. Autonomous Multicamera Tracking on Embedded Smart Cameras. EURASIP Journal on Embedded Systems. 2007]
Example 2: Mobile Camera Configuration

- Pan-Tilt-Zoom (PTZ) cameras allow to change their FOV
- Adapt coverage dynamically, eg., to
  - modify area of interest
  - follow targets
- **Active** visual sensor networks have to react in real-time
  - Estimate the current state (based on image analysis)
  - Compute the PTZ configuration (based on accurate modeling of 3D coverage)
  - Cooperate among cameras may be required for state estimation
- Comparison of different approaches

Different forms of cooperation
Example 3: Security and Privacy

- System level approach addressing the following security issues in cameras:
  - **Integrity**: detect manipulation of image and video data
  - **Authenticity**: provide evidence about the origin of image and videos
  - **Confidentiality**: make sure that privacy sensitive image data cannot be accessed by an unauthorized party
  - **Multi-level Access Control**: support different abstraction levels and enforce access control for confidential data

- Considered attack types: only software attacks

Our Approach: TrustCAM

- We integrate Trusted Computing into camera prototype
- Trusted Computing (TC) is a hardware security solution based on microchip called Trusted Platform Module (TPM)
- Reasons for using TPMs:
  - Implement a well defined set of security functions
  - Public and well reviewed specification
  - Cheap and readily available
  - Hardware provides higher security guarantees than software
  - Using established technology is better than re-inventing the wheel (especially when doing security)
- Main challenge: TPMs are relatively slow
- Careful integration into camera is required
TrustCAM Prototype

- TI OMAP 3530 CPU:
  - ARM @ 480MHz and
  - DSP @ 430MHz
- 256MB RAM,
  - SD-Card as mass storage
- VGA color image sensor
- wireless: 802.11b/g WiFi
  - and 802.15.4 (XBee)
- LAN via USB
  - (primarily used for debugging)
- Atmel hardware TPM
  - on I2C bus
### Hardware/Software Stack

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Application 2</th>
<th>Application N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TrustCAM Software Framework</strong></td>
<td><strong>System Libraries</strong> (libjpeg, zlib, libexif, OpenSSL, IVT...)</td>
<td><strong>TrouSerS TSS with I2C TDDL</strong></td>
</tr>
<tr>
<td><strong>Linux Kernel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OMAP 3530</strong> (ARM Cortex A8 and TMS320C64x+ DSP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram of hardware stack]

- Embedded linux system (Angstrom based)
- Custom kernel with TPM integration
- Customized TrouSerS software stack for TPM access
- Component based application development framework
Architecture Overview

- Each Camera is equipped with a TPM called $TPM_C$

- Cameras are controlled from central back-office
Multi-level security and privacy

Perform cryptographic operations onboard

**Signing**: integrity and authenticity

**Encryption**: confidentiality and multi-level access control

B. Rinner
Control Station

- Video viewer prototype
- Abstracted regions of interest
- Frame groups signatures embedded as custom EXIF data
- History: circular buffer with last 64 frames
  - Unverified frames: orange
  - Verified frames: dark green
  - Last frame of group: light green
Example 4: Authentic User Feedback

- **How to certify** what a camera is doing?
- An authentic communication channel between user & camera
  - Wireless channel is problematic
- Alternative: **visual communication** for device pairing
  - direct line of sight - attackers are easy to spot
  - intuitive way to select the intended camera
- Camera returns a list of hash sums of executed applications
- TrustCenter helps to translate hash sums into properties
User Feedback- Camera Selection

- User is equipped with a trusted handheld device
- 2D barcodes displayed on the user's handheld
- Barcode encodes attestation request and a challenge
Feedback on Mobile Device

- Provide detailed results available to users
- Show running vision processing blocks and their interactions
- Present description and check sums of blocks
Example 5: Collaborative Aerial Cameras

- Develop **autonomous multi-UAV system** for aerial reconnaissance
- Up-to-date aerial overview images are helpful in many situations: “Google Earth with up-to-date images in high resolution”

- **Quadcopter** platform with onboard sensors and computation
- GPS receiver for autonomous **waypoint flights**
- Limitations on payloads, flight time, weather conditions
Autonomous UAV Operation

Mission Planning

Image Analysis

Real-World Flight Simulator

Single/multiple UAV

waypoints

captured image/video

stitching, detection

scenario specification

user interface
User Interface

Define high-level tasks, i.e., observation area

Real-time overview image and execution status

http://pervasive.uni-klu.ac.at/cDrones

[Videos]
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

- Dynamic Power Management
  - For sensing, processing and 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
#3: Deployment, Operation, Maintenance

Consider the entire life cycle of the camera network

- Development support for applications
  - Model/simulate the application (function, resources, QoS)
  - Reuse/exchange of software/libraries
  - Software updates, debugging etc.

- Autonomous calibration and scene adaption
  - Avoid manual procedures
  - Adapt to different scenes and settings

- Network configuration
#4: Distributed Sensing & Processing

Where to place sensors and analyze the data

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

- Compressive Sensing

- Collaborative data analysis
  - Multi-view, multi-temporal, multi-modal
  - Sensor fusion

- Online/real-time processing
  - Can not effort to store large amounts of data
#5: Mobility

How to exploit networks of mobile cameras

- Mobile cameras are ubiquitous
  - PTZ, vehicles, robotics etc.
  - Mobile phones
- Advanced vision algorithms
  - Ego motion, online calibration
  - Closed-loop control, active vision
#6: Usability

How to provide useful services to people

- Ease of deployment, maintenance
  - Self-* functionality
  - “Smart cameras for dumb people”

- Privacy and Security
  - Trust of the user
  - Control the privacy setting

- Interaction with the camera network
#7: Applications

What applications can (only) be solved by DSC

• Demonstrations
  – Large scale networks eg., for surveillance
  – Small scale networks eg., for entertainment, home environments
  – Only single camera application?

• Market opportunities

• Killer Application
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
- **Computer vision**: multi-view geometry, distributed vision, scene adaptation
- **Embedded systems**: power awareness, architecture, processing
- **Multimedia**: Multi-view geometry, distributed vision, scene adaptation
- **Virtual Reality**: Distributed vision, scene adaptation
- **Security**: Distributed vision, scene adaptation
- **HCI**: Distributed vision, scene adaptation
- **Entertainment**: Distributed vision, scene adaptation
- **Assisted Living**: Distributed vision, scene adaptation
- **Scene Analysis**: Distributed vision, scene adaptation

Intersection of “hot” research areas

High potential for various applications

B. Rinner
Further Information

Web site: http://pervasive.uni-klu.ac.at

To probe further:

www.icdsc.org  www.icephd.org