Austrian Computer Science Day, June 2014
On Privacy-Protecting and Self-Organizing Cameras

Bernhard Rinner
http://bernhardrinner.com
Ubiquitous Cameras

• We are surrounded by billions of cameras in public, private and business spaces
• Various well-known domains
  – Transportation
  – Security
  – Entertainment
  – Mobile
• Cameras serve a purpose and provide some utility
  – Providing documentation/archiving
  – Increasing security
  – Enabling automation
  – Fostering social interaction
Paradigma Shifts in Video Processing

- Towards online/onboard processing
- Towards distributed, in-network analysis
- Towards ad-hoc deployment and mobile and open platforms
- Towards user-centric applications

Emergence of Smart Camera Networks!
Smart Cameras as Enabling Technology

- Smart cameras combine
  - sensing,
  - processing and
  - communication
  in a single embedded device

TrustEYE.M4 prototype on top of RaspberryPi

- 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

Agenda

1. Onboard privacy protection in (single) camera
   - Explore tradeoff among utility/protection/resources
   - Embed protection mechanisms close to sensor
2. Autonomous in-network analysis
   - Self-organize tracking in camera networks
   - Learn advantageous strategies of cameras
Onboard Privacy Protection
Privacy Protection in Images

Utility and Privacy-Protection Tradeoff

Utility

Multi-dimensional design space

Privacy-Protection

no  selected  global  full
Observations and Key Challenges

• Most techniques focus on protecting sensitive regions from unauthorized access
  – Global filters protect entire frame
  – Object-based filters protect ROIs (depend on detection performance)

• No single best privacy protection method, but a large design space along protection/utility/resource dimensions

• Privacy protection goes hand-in-hand with security to provide
  – Non-repudiation
  – Confidentiality

Approach: Trustworthy Sensing (TrustEYE)

- **Objective:**
  - Protect access to sensor via a trusted component “TrustEYE”
  - Make security and privacy protection an inherent feature of the image sensor
  - Provide resource-efficient and adaptable privacy protection filters

- **Benefits:**
  - Sensor delivers protected and pre-filtered data
  - Strong separation btw. trusted and untrusted domains
  - Camera software does no longer have to be trustworthy
  - Security can not be bypassed by application developers
  - TrustEYE is anchor for secure inter-camera collaboration

http://trusteye.aau.at/
TrustEYE Overview

Camera Host System

Application-Specific Computer Vision
and
General Purpose OS and Software Framework

Network Interface

Camera Data

Non-Sensitive Image

Abtracted Sensitive Data
(e.g., Edge Image, Histogram, ...)

Protected Sensitive Data
(e.g., Faces), Image Signatures, Timestamps,...

Secure Element

Raw Data

Image Sensor
Privacy Protection by Cartooning

- Abstract parts or entire image by **blurring and color filtering**
- Cartooning pipeline

**ROI-based cartooning**

- **Embed cartooning** as privacy feature into smart cameras
ROI-based Cartooning

(c) MediaEval Dataset

- Privacy protection depends on performance of region detectors (faces, persons etc.)
- Adapting the filter characteristic beneficial

Adjustable Global Cartooning

original

cartooning (small)

cartooning (std)

cartooning (strong)

(c) MediaEval Dataset
Evaluating Privacy/Utility Tradeoff

• Establish an **objective evaluation framework** among key dimensions, i.e.,
  - Privacy protection
  - Utility
  - Appearance
  - Resource consumption
  - Identification of objects of interest
  - Detection/tracking of objects
  - Structural similarity with unprotected frame
  - Achievable frame rate

• Measure the performance using standard CV algorithms with protected videos (and use labeled test data as ground truth)
  - Independently for each frame
  - Measure protection among object’s traces

Comparison of Global Filter Approaches

- Performance of standard CV algorithms compared to unprotected video or other protection filters

**Cartooning**

**Blurring**

**Pixelation**

**Protection:** object re-identification performance

**Utility:** object detection performance

**Appearance:** structural similarity index
TrustEYE.M4 Architecture

Bottom Side (not visible):
2MB SRAM, TPM Security IC, Power Management IC (LiPo Charger), Micro USB Connector, Reset Button
TrustEYE.M4 Prototypes

• Processing board (50x50 mm)
  – ARM Cortex M4 @ 168MHz
  – 4 MB SRAM
  – TPM IC: ST33TPM12SPI via SPI
  – Keil RTX RTOS

• WiFi extension board (50x50 mm)
  – Redpine Signals RS9110-N-11-02
  – 802.11 b/g/n
  – Encryption: WPA2-PSK, WEP
  – Interconnect: SPI bus on 15pin ext. header

• RaspberryPI mounting option
  – Interconnect: SPI bus via dedicated RPI
  – Daterate: 32 Mbit/s

B. Rinner
TrustEYE in Action
Autonomous In-Networking Analysis
Self-organizing Camera Network

• Perform autonomous, decentralized and resource-aware network-wide analysis

• Demonstrate **autonomous multi-object tracking** in camera network
  – Exploit single camera object detector & tracker
  – Perform camera handover
  – Learn camera topology

• **Key decisions** for each camera
  – When to track an object within its FOV
  – When to initiate a handover
  – Whom to handover
Virtual Market-based Handover

- **Initialize auctions** for exchanging tracking responsibilities
  - Cameras act as self-interested agents, i.e., maximize their own utility
  - Selling camera (where object is leaving FOV) **opens the auction**
  - Other cameras **return bids** with price corresponding to “tracking” confidence
  - Camera with highest bid continues tracking; trading based on **Vickrey auction**

Fully distributed approach
no a-priori topology knowledge required
Camera Control

- Each camera acts as an agent maximizing its utility function
  \[ U_i(O_i) = \sum_{j \in O_i} [c_j \cdot v_j \cdot \Phi_i(j)] - p + r \]

- Local decisions
  - When to initiate an auction (at regular intervals or specific events)
  - Whom to invite (all vs. neighboring cameras)
  - When to trade (depends on valuation of objects in FOV)

- Learn neighborhood relations with trading behavior (“pheromones”)
  - Strengthen links to buying cameras
  - Weaken links over time
Learn Neighborhood Relationships

• Gaining knowledge about the network topology (vision graph) by exploiting the trading activities
• Temporal evolution of the vision graph
Six Camera Strategies

• **Auction initiation**
  – “Active”: at regular intervals (at each frame)
  – “Passive”: only when object is about to leave the FOV

• **Auction invitation**
  – “Broadcast”: to all cameras
  – “Smooth”: probabilistic proportional to link strength
  – “Step”: to cameras with link strengths above threshold (and rest with low probability)

• **Selected strategy influences network performance (utility) and communication effort**
Tracking Performance

- Tradeoff between utility and communication effort

Scenario 1 (5 cameras, few objects)  Scenario 2 (15 cameras, many objects)

- Emerging Pareto front

[Esterle et al. Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks. ACM Trans. Sensor Networks. 10(2), 2014]
Assigning Strategies to Cameras

• Identical strategy for all cameras may not achieve best result

Homogeneous strategies (3 cameras)  Heterogeneous strategies (3 cameras)

• Strategy depends on various parameters (FOV, neighbors, scene ...)
  – Let cameras learn their best strategy
Decentralized Multi-Agent Learning

- Exploit **bandit solver** framework to maximize global performance
  - Co-dependency among agents’ performance
  - Complex relationship between local reward global performance

Multi-camera Experiment

- Indoor demonstrator with 6 cameras tracking 6 persons
- Each camera performs
  - Color-based tracking
  - Fixed or adaptive handover strategies (bandit solvers)
  - Exchange of color histograms for person re-identification
Conclusion

• Smart cameras process video data onboard and collaborate autonomously within the network

• Our cartooning approach protects image data “at the sensor” but stills provides reasonable utility with low resource usage

• We apply socio-economic techniques to learn control strategies for autonomous multi-camera tracking
  – Global configurations emerge from local decision using local metrics
  – Adaptive strategies extend Pareto front of best static configurations

• Techniques applicable to various decentralized networked systems (e.g., Internet of Things)
Acknowledgements & Further Information

Pervasive Computing group
Institute of Networked and Embedded Systems
http://nes.aau.at
http://bernhardrinner.com

Funding support

• KWF/FWF “Trustworthy Sensing and Cooperation in Visual Sensor Networks”
• FP7 FET “Engineering Proprioception in Computing Systems”