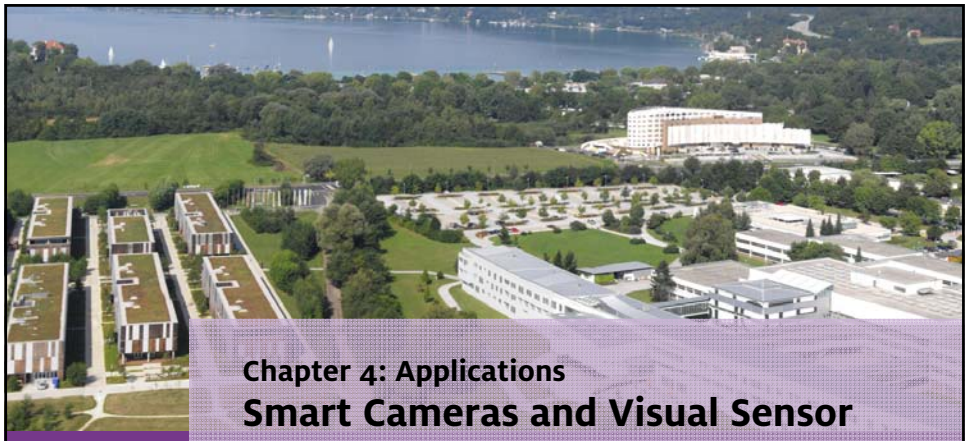


Chapter 4: Applications

Smart Cameras and Visual Sensor Networks

ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT | WIEN GRAZ

Bernhard Rinner

FAKULTÄT FÜR TECHNISCHE WISSENSCHAFTEN
Institut für Vernetzte und Eingebettete Systeme

Agenda

Chapter 4: Applications

- TrustCAM
  - Privacy and Security in Smart Camera Networks
- Collaborative Microdrones
  - Airborne visual sensor network

TrustCAM:
Security- and Privacy-awareness

Motivation for Privacy & Security

- Video Cameras are part of life
  - video surveillance in cities, on airports, train stations,…
  - traffic monitoring
  - elderly care and assisted living
  - ….
- Smart camera security considerations
  - amount of software on cameras is growing
  - cameras are equipped with wireless interfaces
  - cameras and delivered data become attractive target for attackers
  - operators need assurance that video is authentic and unmodified
  - privacy of monitored persons is a hot topic
- System security is often considered as afterthought

Goals and Assumptions

- We present a system level approach that addresses the following security issues:
  - 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

[Winkler, Rinner. Securing Embedded Smart Cameras with Trusted Computing. EURASIP Journal on Wireless Communications and Networking, 2011]

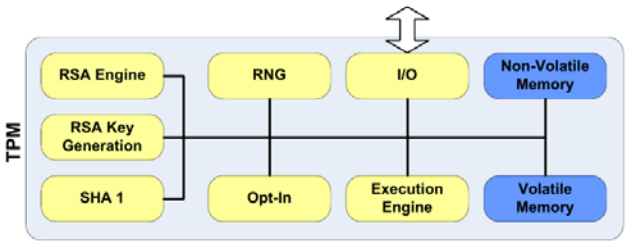
Approach

- 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

Trusted Computing Overview

TPM Architecture

- Execution engine: command parsing, verification, execution
- random number generator
- SHA-1 hash function (160 bits)
- RSA engine and key generation (2048 bits)
  - keys never leave the TPM unprotected
- non-volatile memory for special cryptographic keys
- volatile memory where currently used keys are stored



TPM Date Encryption

- TPM can be used to securely store data
- Private TPM keys can only be used in the chip
- Keys can be migratable or non-migratable
- Data binding:
  - guarantees that data can only be decrypted by the TPM that has the private key
  - If key is non-migratable it is guaranteed that only this specific TPM can decrypt the data

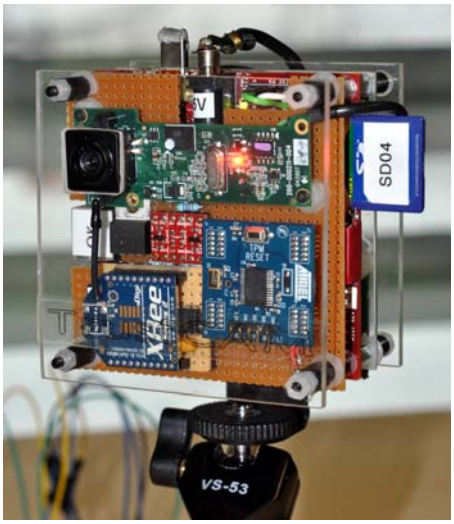
TPM Data Signing

- Digital signature using the TPM protected key
- Using non-migratable keys
- Signature can only be created inside the TPMs
- Evidence that data comes from the system that the TPM is a part of

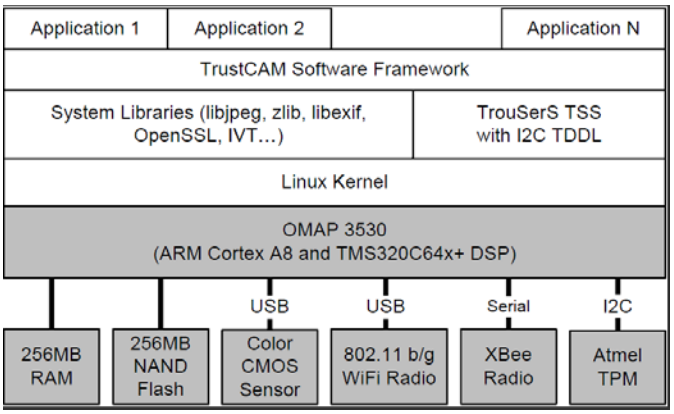
Research Prototype and System Architecture

Hardware 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



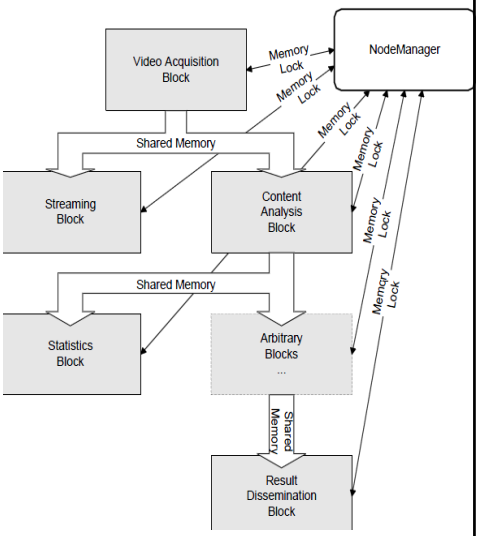
Hardware/Software Stack



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

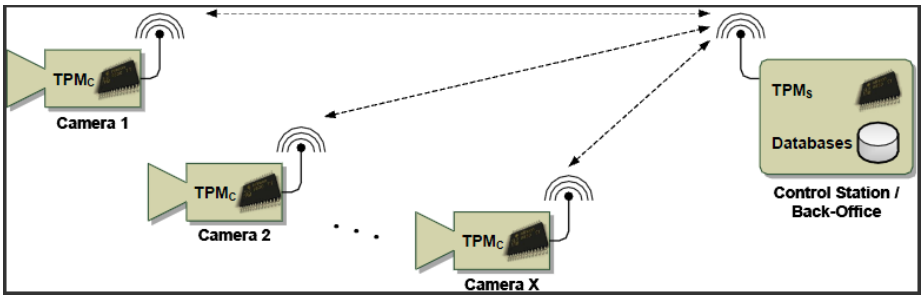
Application Development Framework

- software framework based on reusable components ("blocks")
- blocks are independent processes connected via shared memory
- actual applications consist of several blocks that are configured as required
- block management is done by the NodeManager (one instance per camera)



Architecture Overview

- Each Camera is equipped with a TPM called TPM_C



- Cameras are controlled from central back-office

Camera Setup

- camera is under full control of operating personnel
- a non-migratable signing key KSIG is generated in TPM_C
  - the public part K_{SIGpub} is exported and stored in the CS database
- multiple non-migratable binding keys $K_{BIND1...X}$ are generated in TPM
  - the public parts $K_{BIND1...X}$ are exported and stored on the camera
- note that the private keys can not be exported from their TPMs and hence can only be used inside their TPMs

Security Functionality
Design, Implementation and Results

Video Integrity and authenticity

Approach

Cryptographic signing of frames before they are streamed

- Control station requests video streamed
- Camera signs outgoing frames with signing key K_{SIG}
- Control station loads the expected the public signing key K_{SIGpub} from database
- Verification of image signature using K_{SIGpub}
- Successful verification ensures:
  - Frame was not modified during transport (integrity)
  - Frame comes from the intended camera (authenticity) – KSIG can only be used inside the TPM of the camera

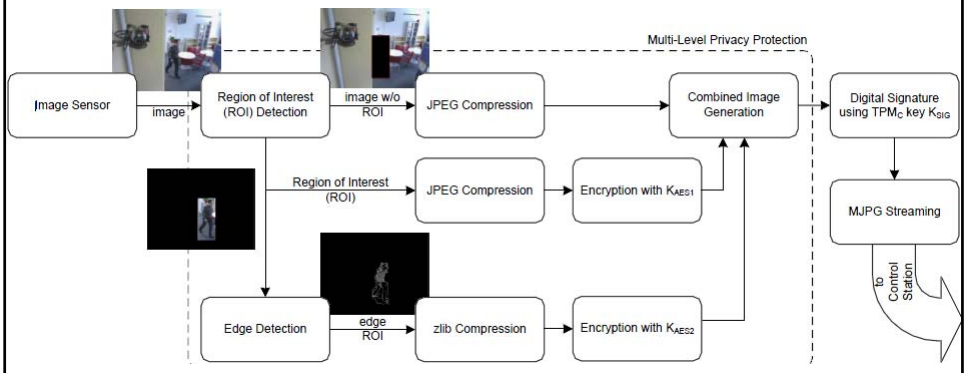
Confidentiality and Access Control

Approach

Encryption of the frames before they leave the camera

- Encryption with public parts $K_{BIND1...Xpub}$ of non-migratable binding keys of control station
- Encryption of full image vs. selected regions of interest (ROIs) to ensure privacy of monitored people
- On TrustCAM we encrypt ROIs at two different levels
  - $K_{BIND1pub}$ and $K_{BIND2pub}$ are used to bind plain ROIs
  - access to plain ROIs requires passwords of K_{BIND1} and K_{BIND2} (e.g., supervisor authorization)
  - $K_{BIND3pub}$ is used to bind an abstracted version of the ROIs
  - access to abstracted ROIs requires password of K_{BIND3}

Processing Flow



- K_{AES1} is bound with $K_{BIND1pub}$ and $K_{BIND2pub}$
- K_{AES2} is bound with $K_{BIND3pub}$

Implementation and Results

Signature Performance

- SHA1 runtime: less than 2ms for less than 30kB of Data
- TPM signature runtime: approx. 800ms
- additional TPM overheads: approx. 50ms

- Image signing using TPM: SHA1 of image + TPM signature
- TPM too slow to sign every frame
- Approach: accumulate the SHA1 hash of F frames and use TPM to sign this accumulated sum
- Verification also has to be done for the frame groups
- Additional property: group signature ensures correct frame order

Overall Results

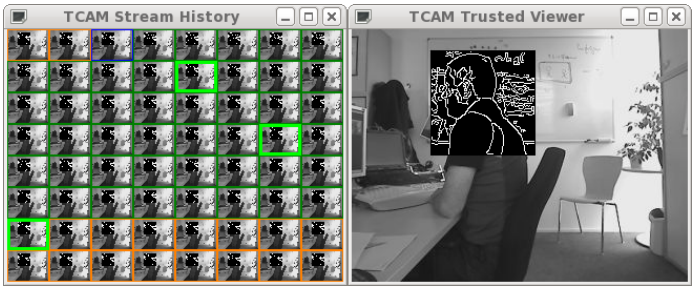
Input Format	Internal Format	Plain Streaming	ROI Encr. (200 x200) + Image Signing
320 x 240 YUYV	Gray	24.4 fps	20.6 fps
	RGB24	23.8 fps	12.1 fps
640 x 480 YUYV	Gray	12.8 fps	9.1 fps
	RGB24	6.5 fps	5.0 fps

- Acceptable impact for image singing and ROI encryption

Input Format	Internal Format	Color Conv	ROI Extract	JPEG Comp. Backgr.	JPEG Comp. ROI	Zlib comp Edge.	ROI	Edge	Sig.	Total
320 x 240 YUYV	Gray	2.4ms	9.7ms	16.1ms	9.8ms	5.2ms	1.6ms	1.2ms	1.0ms	47.0ms
	RGB24	6.6ms	12.8ms	31.8ms	18.7ms	5.3ms	1.8ms	1.2ms	1.0ms	79.2ms
640 x 480 YUYV	Gray	8.8ms	8.9ms	63.9ms	9.1ms	3.9ms	1.2ms	0.9ms	1.7ms	98.4ms
	RGB24	27.6ms	11.5ms	125.9ms	17.8ms	3.8ms	1.3ms	0.9ms	1.9ms	190.7ms

- Minor impact from AES encryption and SHA1 hashing
- No TPM impact: Batch signatures, parallel to main CPU
- Performance bottleneck: image compression→ use DSP!

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

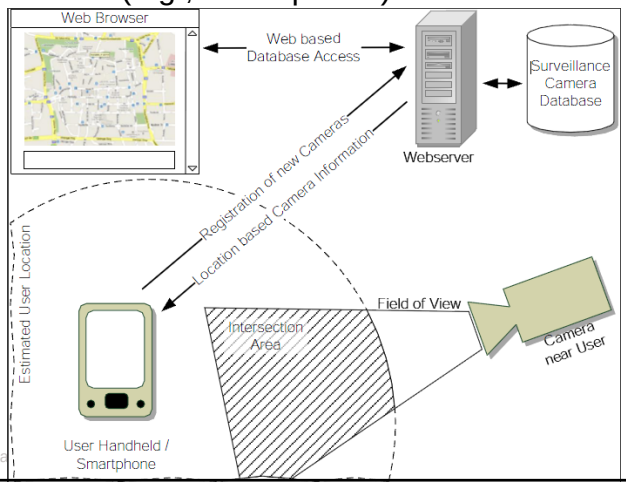
User-centric Privacy

Privacy-aware Camera Networks

- What about users (i.e., monitored people)?
  - users usually do not care much about integrity, authenticity of time stamping
  - users (hopefully!) care about confidentiality and privacy!
- Question 1: How can we increase privacy awareness?
- Question 2: How can we demonstrate that (our) cameras protect the privacy of users?

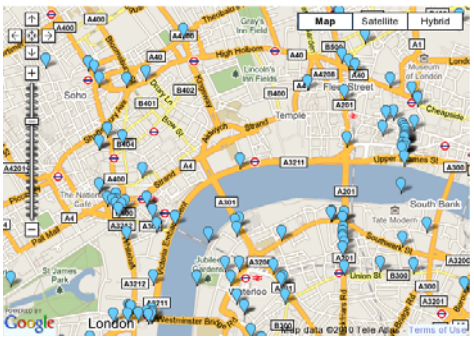
Raising Privacy Awareness

- let users know if there are cameras in their environment
- use user's handheld (e.g., smart phone) for location-based notifications



Data Collection

- community-based registration and mapping of cameras
- similar to community-based efforts like Wikipedia
- first community databases are already available
  (e.g., OpenStreetMap.org and MapCams.org)



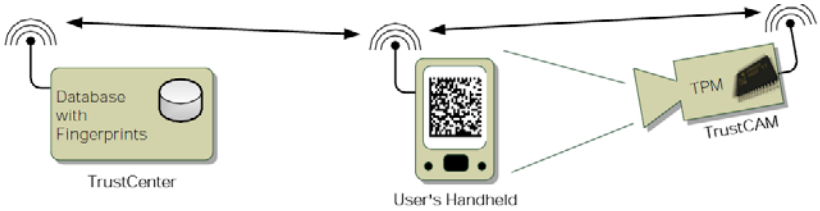
Level of Privacy Awareness

- Level 0 - No Awareness
  - cameras are advertised by notes or stickers
  - most users unaware about cameras in their environment
- Level 1 - Basic Awareness
  - location-based notification with data of community databases
  - issue an alarm if approaching an area under surveillance
- Level 2 - Extended Awareness
  - requires availability of additional information (e.g., who owns a camera, for what purpose, how long is data stored, . . .)
  - who can provide this information? (users vs. operators)
- Level 3 - Full Awareness with Direct User Feedback
  - provide hard evidence about what a camera is doing
  - based on Trusted Computing attestation techniques

Handheld Usage Scenarios

- register and map video surveillance cameras
  - GPS based
  - map preview: sketch camera orientation and estimated FoV
  - register (externally visible) camera properties such as type of camera, . . .
- users can define individual privacy policies
  - users decide which properties are relevant for them
  - selective notification / alert system
- use handheld as client for direct user feedback
  - give users direct feedback and hard evidence what a specific camera is doing

Direct User Feedback



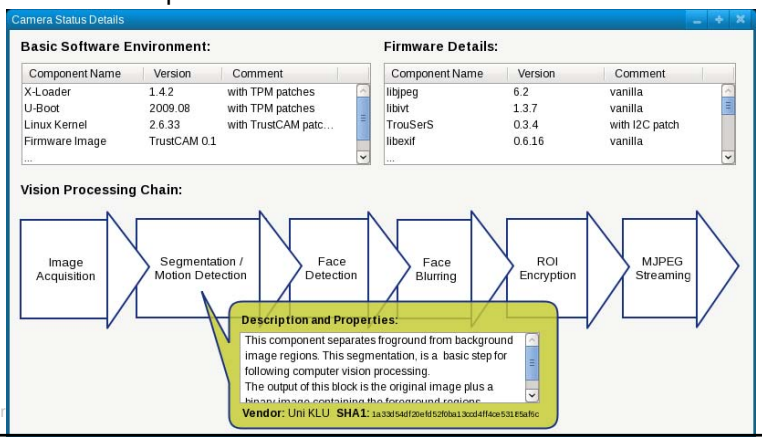
- establish an authentic communication channel to camera
- wireless communication is problematic
  - how to assert that response comes from the intended camera?
- use visual communication for device pairing
  - direct line of sight - attackers are easy to spot
  - intuitive way to select the intended 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

User Experience

- make detailed results available to users
- show running vision processing blocks and their interactions
- present description and check sums of blocks



Summary of TrustCAM

- TrustCAM prototype demonstrates feasibility of integration of TC into an embedded computer vision system
- authenticity, confidentiality and integrity protection for images and videos
- multi-level access control
- performance impact on vision system is relatively small
- future work:
  - using DSP for image processing and video compression
  - freshness and time stamping of images
  - focus on privacy aspects and user involvement
  - using TPM for system integrity checks, reboot detection, …

Collaborative Microdrones: an aerial VSN

Battery-powered UAVs

- **Quadcopter** platform with onboard sensors and electronic for flight stabilization
- Attached cameras for sensing the environment
- GPS receiver for autonomous **waypoint flights**
- **Limitations** on payloads, flight time, weather conditions

www.microdrones.de

www.asctec.de

Disaster Relief with UAVs

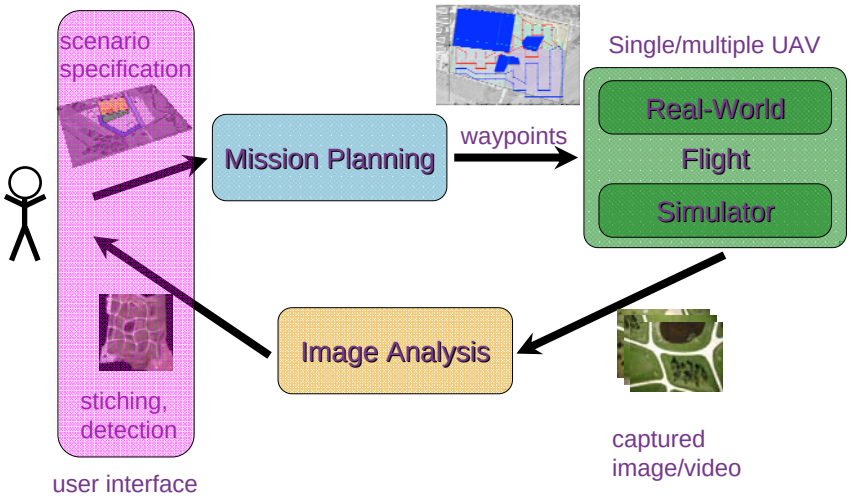
General idea

- Support first responders in disasters with multiple UAVs
- Provide latest and relevant information about the scene
- Autonomously flying, networked, collaborating UAVs

Use case: Generate overview image

- Cover the disaster area and take images at individual points
- „Stitch“ individual images to generate scene overview (mosaic)
- Provide intuitive user interface

Autonomous UAV Operation



Issues

1. How to **generate & adapt movement routes** for the UAVs?
   - Achieve multiple optimization goals
   - Deal with changes in the environment
   - Compare centralized versus distributed approaches
2. How to **stitch the individual images**?
   - Apply incremental image stitching
   - Tradeoff between good geo-referencing and visually appealing overview
3. **System integration and demonstration**

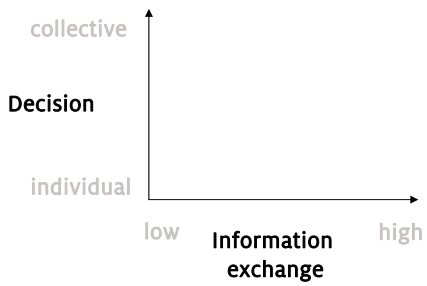
1: Generation of Routes

	Non-cooperative	Cooperative
Deterministic	UAV has a **predefined route** that is **independent** of other UAV paths.	UAV has a **predefined route** that **depends** on other UAV paths.
Dynamic	UAV has an *a priori* **unknown route** that is **independent** of other UAV paths.	UAV has an *a priori* **unknown route** that **adapts** to other UAV paths.

Adaptive Routes

Meeting event

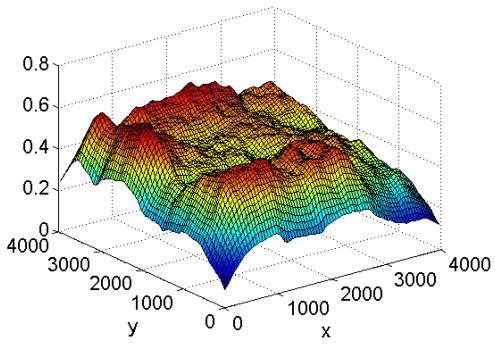
How to adapt the route?



Analysis of the Area Coverage

Approaches

- Simulation-based studies
- Discrete stochastic processes



2: Image Mosaicking

- **Problem definition**
  - Given n individual images Ii, find **image transformations** Ti for each Ii

$$I_{overview} = \bigcup_{i=1}^{n} T_i(I_i)$$

  which maximizes some quality function $\lambda(I_{overview})$

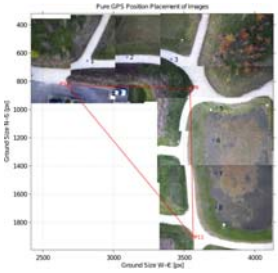
- **Two fundamental approaches for finding the transformations**
  1. Exploit auxilliary data, i.e., camera's position and orientation (**meta data based approach**)
  2. Exploit corresponding points within image overlaps (**image based approach**)

Challenges for Mosaicking

- **Low altitude** and **non-planar surface** introduce high perspective distortions
- Light-weight UAVs are vulnerable to wind resulting in **non-nadir view**
- **Inaccurate position and orientation** data due to small, low-cost GPS, IMU and altimeter sensors
- Strong **resource limitations** wrt. onboard processing, power, communication etc.

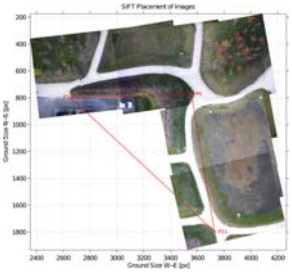
Incremental Image Mosaicking

- **Start with meta data approach, refine with image-based approach**



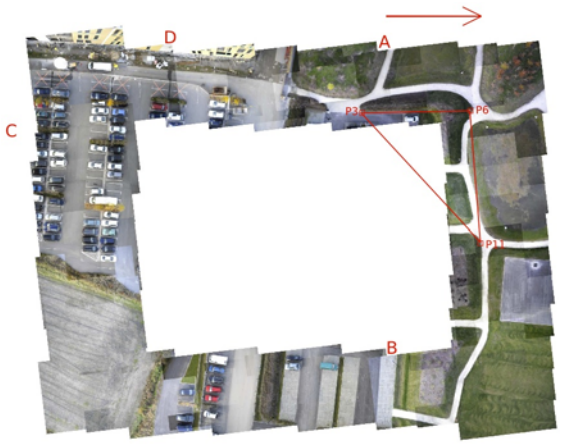
Position data
(GPS meta data)

Position&orientation data
(GPS & IMU)



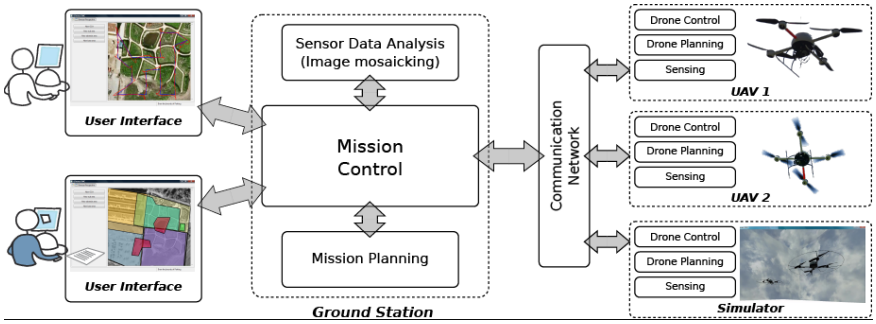
SIFT feature points
(Image data)

Overview image

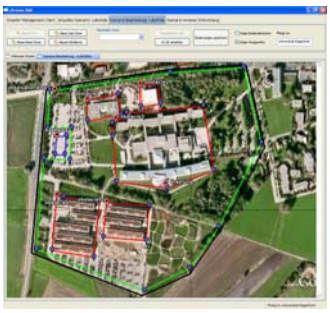


3: System Integration

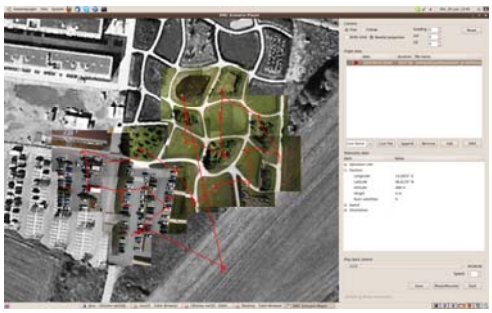
FAMUOS - Fully Autonomous Multi-UAV Operation System



„Google-like“ User Interface

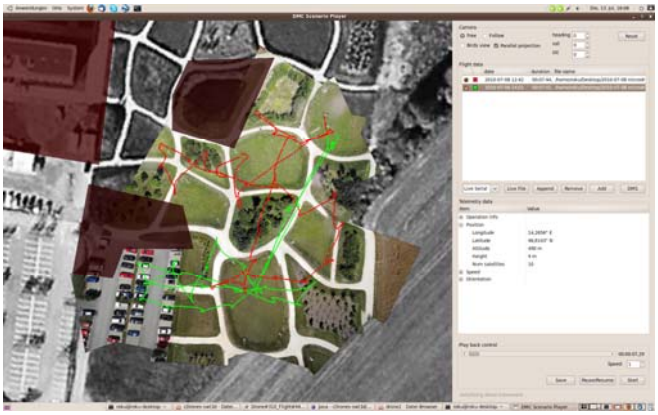


Specifying the **scenario description**



Visualizing the **most recent overview image** and the flight route

Demonstration (Video)



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

Tutorial Agenda

1. Introduction
2. Smart cameras
   - Architecture of Smart Cameras
   - Prototypes
3. Visual Sensor Networks
   - Advantages & Challenges
   - Characteristics of Visual Sensor Networks
   - Research Directions
4. Applications
   - Security- and privacy-awareness in Smart Camera Networks
   - Aerial Visual Sensor Networks
5. Conclusion