

Distributed Smart Cameras

Lorenz Breu

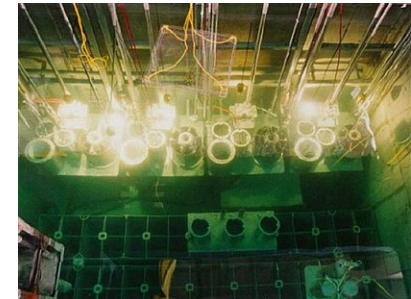
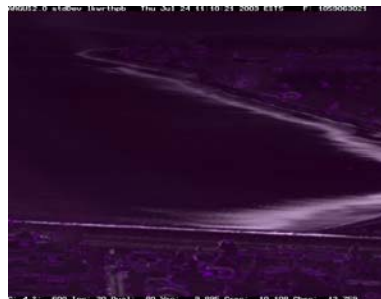


Überblick

- DSC Systeme – Überblick
- Beispiele:
 - (Militärische) Überwachung
 - Fall Detector / Object Finder
 - FACET
- Zusammenfassung

Kamerasysteme heute

- Überall
- IR / RLV
- Überwachung von Umwelt / Technik
- ...



Distributed Smart Camera Systeme

- Traditionell:
 - Kameras als individuelle Einheiten
 - Kameras sammeln Videodaten, der Mensch analysiert
- DSC Systeme:
 - Kameras im Verbund (Knoten in einem Netz)
 - Kameras verfügen über „Intelligenz“

DSC Komponenten

- DSC Systeme bestehen aus 6 Komponenten:
 - Sensoren
 - Algorithmen zur Objekterkennung / Identifikation
 - Hardware
 - Software / Kommunikation
 - User-Interface
 - Algorithmen zur Erkennung von interessanten Ereignissen



Beispiel 1: Militärische Überwachung

Militärische Überwachung

- Automatisierte Überwachung in überbauten Gebieten und auf dem Schlachtfeld
- Aktives Forschungsgebiet seit Mitte der 90er
- DARPA VSAM: Technologien und Konzepte führten zu modernen kommerziellen Systemen

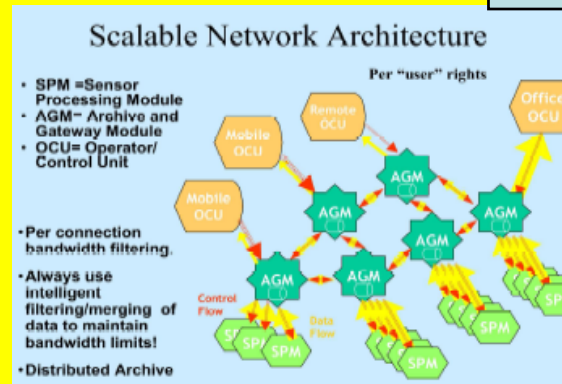
Erkennen von Eindringlingen in 2 Schritten

Omnidirektionale Kamera



-> Objekterfassung

Skalierbares Netz



-> Routing
-> Archivierung
-> SOAP

Pan-Tilt-Zoom Kamera



-> Identifikation

Wahl der Kameraknoten

- Bestehen aus Sensor (Kamera) und Hardware zur Bildbearbeitung und Kommunikation
- Auswahl des Sensors:
 - Auflösung
 - Omni-directional vs. PTZ
 - Nachtsicht, Beleuchtung, Kosten, ...
- Bildbearbeitung am Knoten
 - PC vs. embedded

Schwierigkeiten und Fazit

- Korrekte Erkennung von Menschen, um falsche Alarme zu vermeiden:
 - Wind, Tiere, ...
- Aber, Eindringlinge dürfen nicht übersehen werden...
- Heute: 100 Knoten (88 Kameras), 2km, 1 802.11 Kanal



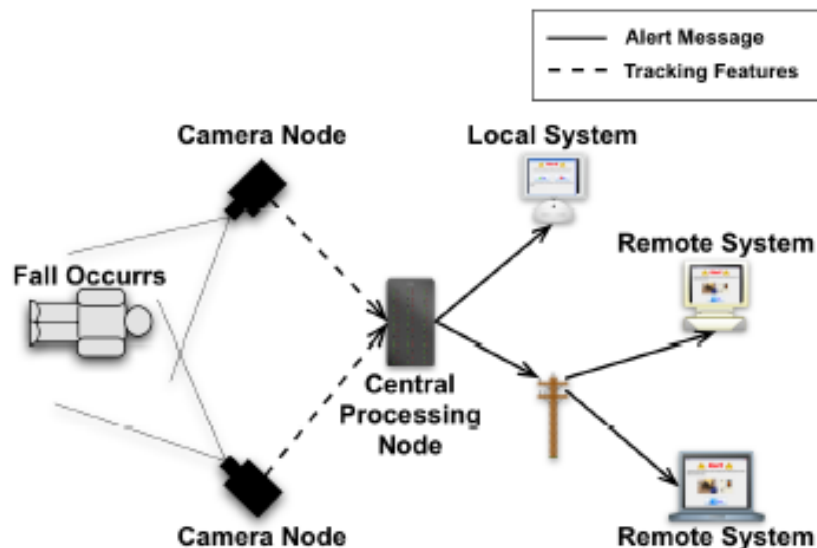
Beispiel 2: Fall Detector / Object Finder

Fall Detector / Object Finder

- Einsatz eines DSC Systems zur Unterstützung älterer Menschen
- Fall Detector
 - Benutzer muss Sensoren nicht mehr am Körper tragen
 - Keine explizite Aktion vom Benutzer benötigt
- Object Finder

Aufbau des Systems

- Kameraknoten: 928MHz CPU, 256MB RAM
- Zentralknoten: zusätzlich 30GB Speicher



Fall Detector - Ablauf (1)

- Kameras detektieren Vordergrundobjekte
- Information über „grosse“ Objekte an Zentralknoten gesendet
- Auflösung von 128x128 Pixel reicht (aber Objekterkennung braucht 320x240)

Fall Detector - Ablauf (2)

- Zentralknoten sucht nach Sturz:
 - Verlauf „Breite / Höhe“ des Menschen im Bild
 - Kalibrierung -> Markov Modelle
- Alarm wird ausgelöst:
 - Lokal
 - Freundesliste
 - Notfallzentrale



Object Finder - Ablauf(1)

- Kameras entdecken bewegliche Pixelmenge die nach einer gewissen Zeit stillsteht
- Zentralknoten lässt Kamera schwenken und zoomen
- Vergleich mit Bildern im Zentralknoten und Eintrag in Datenbank

Object Finder - Ablauf (2)

- Benutzer fragt nach Objekt
- Falls nicht in DB: Kameras suchen Bereich ab (4-5 Bilder) und Zentralknoten sucht
- Ausgabe der wahrscheinlichsten Orte an Benutzer



Schwierigkeiten des Systems

- Wahrung der Privatsphäre
 - Bilder aus der Wohnung des Benutzers
 - Bilder des Benutzers in misslicher Lage
 - Freundesliste, schwarzer Balken, ...
- Prioritäten
 - Zentralknoten sollte nicht nach Schlüssel suchen, wenn der Benutzer gestürzt ist...

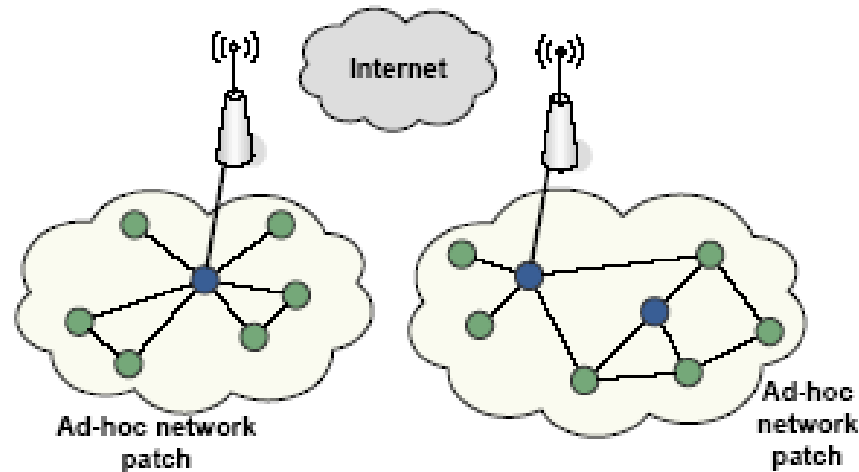
Beispiel 3: FACET

Was ist FACET?

- Einsatz von Mobiltelefonen als Kameraknoten
 - Kameras
 - Kommunikation (Bluetooth, GPRS, UMTS, ...)
 - Java APIs
 - Kostengünstig
 - Allgegenwärtig
- FACET als allgemeines Framework für den Aufbau von DSC Systemen mit Mobiltelefonen

Kommunikations-Infrastruktur

- Lokale „patches“ (Bluetooth) verbunden mit entfernten „patches“ oder anderen Netzen (GPRS)



Aufbau von lokalen Patches

- Kameras entdecken alle Bluetooth Nachbarn
- Bauen lokale Routingtabellen
- Limitationen durch Bluetooth und Java API
 - „Inquiry“ Prozedur dauert lange
 - API erlaubt nur 1 offene Verbindung

Synchronisation und Kalibrierung

- Lokalisierung basierend auf events:
 - In: Objekt betritt Sichtfeld
 - Out: Objekt verlässt Blickfeld
 - Lokale events an alle senden
 - Zusammengehörende in/out events -> Graph
 - Kalibrierungsalgorithmus
- Zeitsynchronisation durch Modifikation von lokalen Timestamps vor dem Senden

Erkennen von beweglichen Objekten

- API erlaubt Bearbeitung von Rohdaten
 - 120x160 Pixel
 - Graustufen
 - 80ms / Bild
- Bild Pixelweise mit Hintergrund verglichen
- Signal filtern
- Viele Pixelunterschiede -> Event melden

Experimentelle Evaluation

- System von 4 Mobiltelefonen im Flur
- Bluetooth-Nachricht ca. 2s
- Zeitsynchronisation: Abweichungen ca. 30ms
- Analyse von ca. 12 Bildern pro Sekunde
- Keine falschen Positiv-Meldungen
- Ca. 20% der Events verpasst
- Limitationen durch Technologie und API

DSC Systeme: Schwierigkeiten

- Auflösung vs. Datenmenge
- Rechenintensive Bildbearbeitung
- Falschmeldungen in vollautomatisierten Systemen
- Klare Prioritäten der Funktionen
- Schutz der Privatsphäre
- Limitationen durch Hard- und Software

DSC Systeme: Zusammenfassung

- Kamerasysteme überall im Einsatz
- Vernetzung und Automatisierung sinnvoll
- Entwicklung von DSC Systemen beeinflusst von verschiedenen Bereichen
- Heutige Hard- und Software erlauben schon Aufbau von grossen spezialisierten Systemen
- Mobiltelefone schon begrenzt einsetzbar



The End

Quellen (1):

- Bolliger, P., Köhler, M. and Römer, K.: Facet: Towards a Smart Camera Network of Mobile Phones. Proceedings of Autonomics 2007 (ACM First International Conference on Autonomic Computing and Communication Systems), Oktober 2007.

Quellen (2):

- Boulton, T.E., Johnson, R.C., Pietre, T., Woodworth, R. and Zhang, T.: A Decade of Networked Intelligent Video Surveillance. ACM Workshop on Distributed Camera Network, Oktober 2006.
- Williams, A., Xie, D. and Ou, S.: Distributed Smart Cameras for Aging in Place. Workshop on Distributed Smart Cameras (DSC 2006), Oktober 2006.