Distributed Vision Processing in Smart Camera Networks

CVPR-07

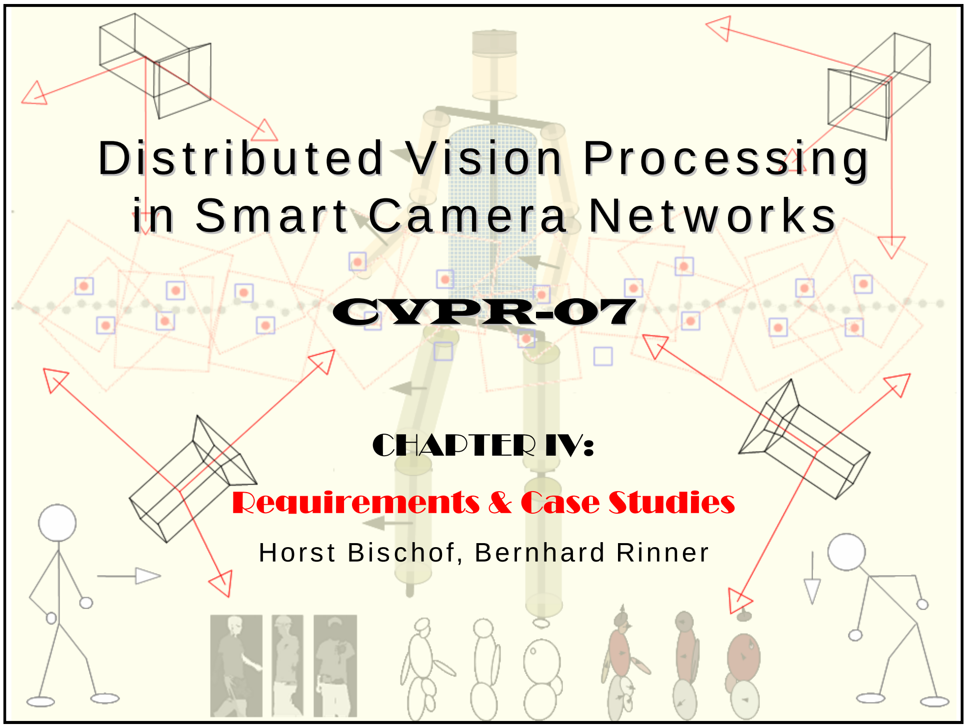
Hamid Aghajan, Stanford University, USA
François Berry, Univ. Blaise Pascal, France
Horst Bischof, TU Graz, Austria
Richard Kleihorst, NXP Research, Netherlands
Bernhard Rinner, Klagenfurt University, Austria
Wayne Wolf, Princeton University, USA

March 18, 2007
Minneapolis, USA

Course Website – http://wsnl.stanford.edu/cvpr07/index.php

Outline

I. Introduction
II. Smart Camera Architectures
  1. Wireless Smart Camera
  2. Smart Camera for Active Vision
III. Distributed Vision Algorithms
  1. Fusion Mechanisms
  2. Vision Network Algorithms
IV. Requirements and Case Studies
V. Outlook



Requirements and Case Studies

Contact: Horst Bischof, Bernhard Rinner

bischof <AT> icg.tugraz.at

bernhard.rinner <AT> uni-klu.ac.at

Overview

- Prerequisites
- Image processing pipeline
- Case studies
  - Automotive applications
  - Tracking
- Summary

Prerequisites

Hardware Challenges of SmartCams

- Embedded platforms with resource limitations
  - Processing (fixed-point, dedicated HW)
  - Memory
  - Power
- Limitations concerning optics and sensor
  - Sensor resolution
  - Optical performance

Software Challenges of SmartCams

- (Lack of) system-level software
  - Operating system, i.e., communication primitives and memory management
  - Middleware for distributed system
  - (Some) knowledge of underlying hardware required
- Image processing libraries
  - Reduced availability and functionality

Networking Challenges of SmartCams

- Temporal and spatial calibration
  - Registration
  - Synchronization
- Cooperation among cameras
  - Data abstraction and communication
  - Distributed control

Requirements for SmartCams Apps

- Online / real-time algorithms
- Memory-efficiency
- Fixed-point implementation
- Embedded software development

Image Processing Pipeline

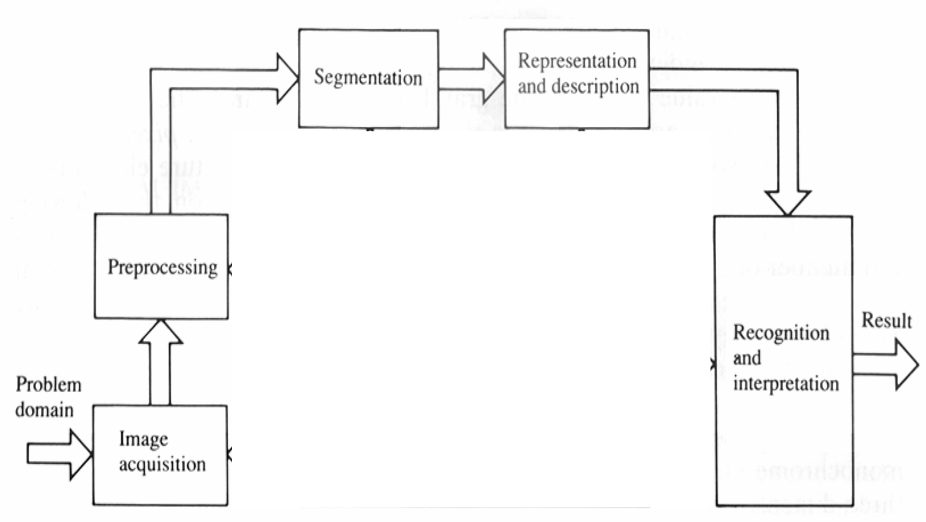
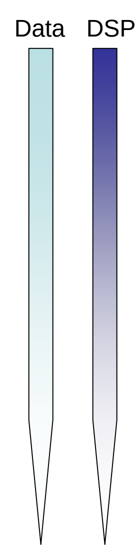


Image Processing Pipeline

- Preprocessing / filtering
  – Pixel-based, intra-frame operations
  – Output: images
- Segmentation / motion detection
  – Output: object parts
- Localization / tracking / classification
  – Output: objects / events
- Event understanding



Case Study – Automotive Applications

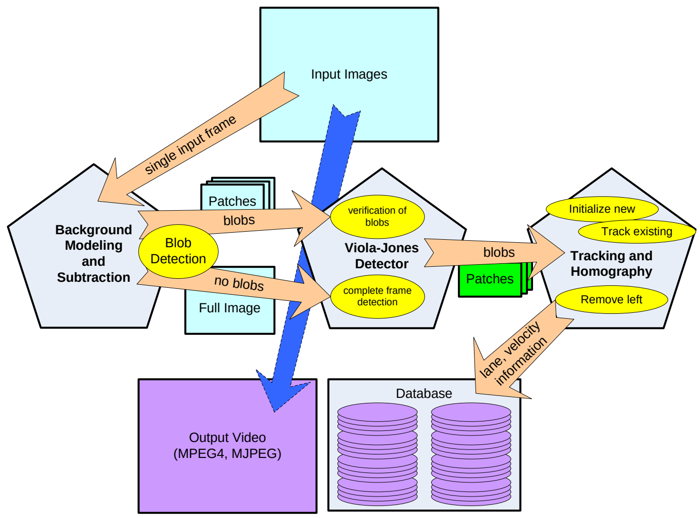
Applications in Traffic Surveillance

- Vehicle counting, tracking, speed estimation, classification

- License plate detection and OCR

- Wrong-direction-driver detection, traffic jam detection and alerting

Vehicle Detection



Background Modeling and Subtraction

- Approximated median filter

[*McFarlane/Schofield 1995*]

$$m_c[x,y] = \begin{cases} m_c[x,y]-1 & \text{if} \quad v_c[x,y] < m_c[x,y] \\ m_c[x,y]+1 & \text{if} \quad v_c[x,y] > m_c[x,y] \end{cases}$$

$$\text{diffImage}[x,y] = \begin{cases} 1 & \text{if} \quad \text{sum} > \text{Threshold} \\ 0 & \text{else} \end{cases}$$

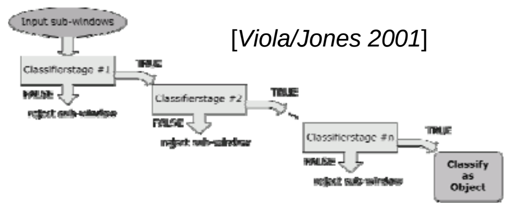
$$\text{sum} = \sum_{c \in \{R,G,B\}} abs(v_c[x,y] - m_c[x,y])$$

- Stationary and slow moving objects
- Occlusions and shadows
- Camera shaking

+ No floating-point arithmetic
+ Moderate amount of memory
+ Good performance, even under adverse environmental conditions

Viola-Jones Detector

- Based on ADA-Boost
- Exhaustive search
- Integral images
- Calculation of simple features and cascading



[*Viola/Jones 2001*]

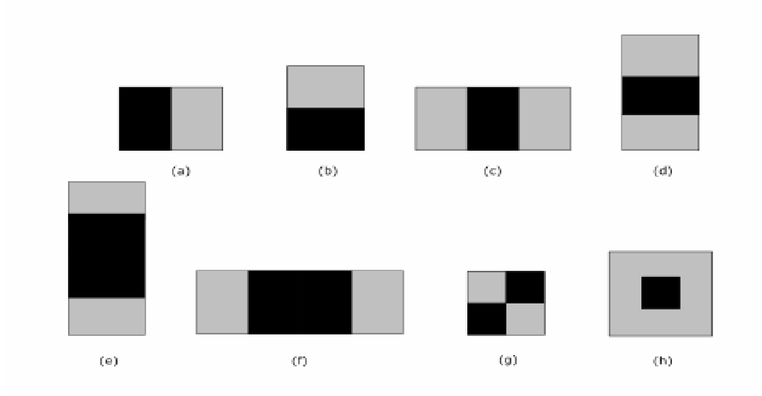
➢ Minimizing the number of weak classifiers and the number of cascades

➢ Simultaneously keeping a good level of performance

- Integral images in *Integer* units, thresholding using *Floats*
- RealBoost (instead of discrete boosting)
- Inter-stage feature propagation
- Selection of scene dependent negatives for training

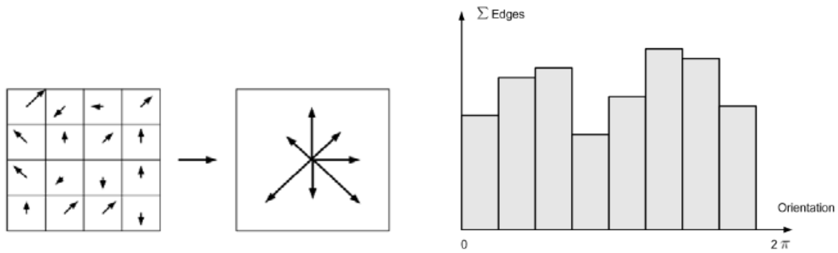
Object Detection System

- Appearance features (1909 Alfred Haar)



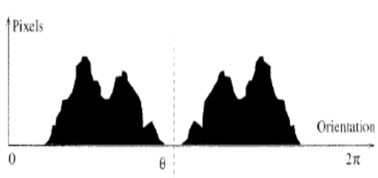
Orientation Features

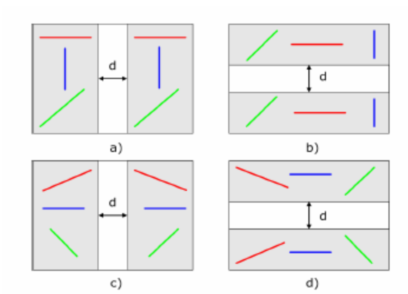
- Gradient image (Sobel)



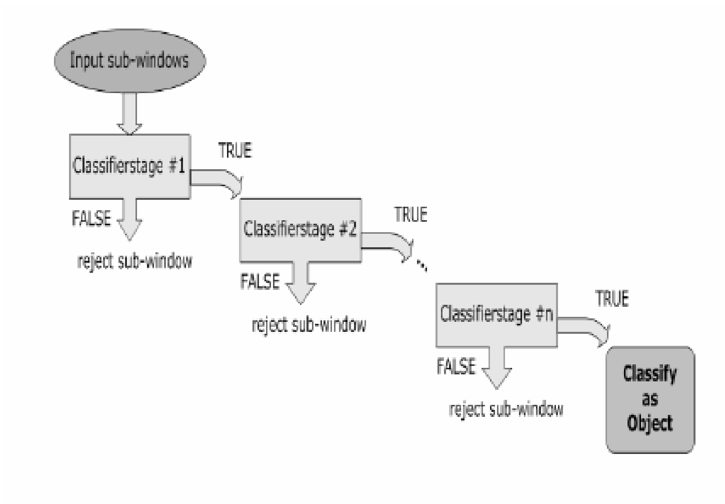
- Three different feature types
  ⟹ EOHs and full orientation histogram features

Symmetry Features





Cascade Classifier and Inter-Stage Feature Propagation



Comparison of Different Feature Types

UIUC

- 7 stages
- 11 features
- 56% F-measure

InfoA10

- 4 stages
- 8 features
- 85% F-measure

Performance Evaluation

Kernel Functions

Function	PC [ms]	DSP [ms]
CalculateIntegralImage()	1.828	2.661
CreateGradientIntegralImages()	40	429
CreateMasks()[5]	20	630

Function	Stage	#Haar-Features	#EOH-Features	PC [ms]	DSP [ms]
EvaluateStageOnlyHaar()	1	3	-	0.006	0.006
EvaluateStageOnlyHaar()	2	1	-	0.003	0.002
EvaluateStageEOH()	1	1	1	0.004	0.007
EvaluateStageEOH()	2	1	4	0.014	0.028

System Performance

Experiment	Detector		Region of Interest	PC [ms]	DSP [ms]
	#Stages	#Features			
1	10	27	(0,0)	4260	2041
2	10	27	(180,40)	1884	709
3	10	27	(180,40)	64	38

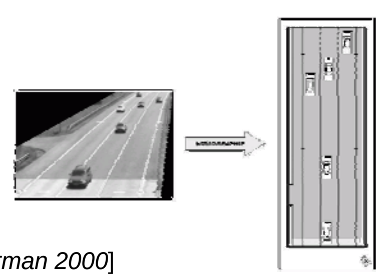
Tracking and Homography

- Kalman Tracker
  - Initialize for new vehicle
  - Estimating using a motion equation of 2nd order (p_x,v_x,a_x, p_y, v_y, a_y)
  - History of movements

[*Kalman 1960*]

Floating point

- Homography
  - Mapping image plane to ground plane
  - Just calculated once at startup



Floating point

[*Hartley/Zisserman 2000*]

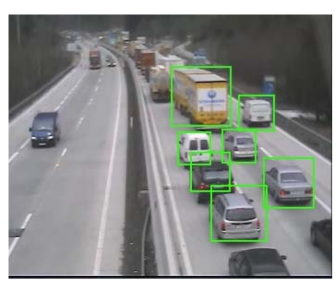
Single Algorithm Time Consumption

– Full frame / ROI
– BGM threshold: 45
– VJ parameters: 27/7 (vehicles) and 37/8 (license plates) features / stages

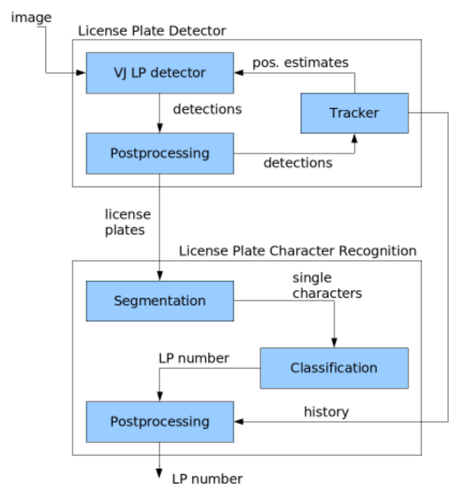
Module	Full Frame 352x288		Region Of Interest 192x220	
	avg. time [ms]	std. dev.	avg. time [ms]	std. dev.
Background Modelling and Subtraction	1.768	0.0	0.75	0.0
Region Labelling	2.7	0.87	1.32	0.68
Viola-Jones Detector (3 scales) (Vehicles)	92.38	10.58	44.14	5.67
Viola-Jones Detector (5 scales) (License Plates)	141.62	3.27	-	-
Non-Maximum Suppression (max. 150 detections)	0.18	0.16	0.034	0.041
CamShift (max 40 iterations)	208.53	222.43	139.7	93.86
	Single Execution			
Kalman Tracker (Estimate/Update)	0.346	0.01		
Homography	0.008	0.0		

Enhanced Vehicle Detection

- No more merging due to shadows
- Static object detection

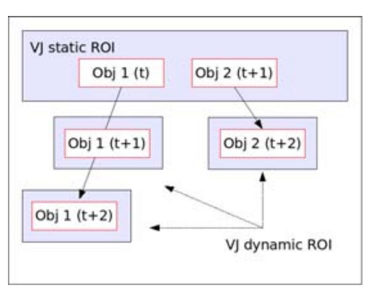



License Plate Detection

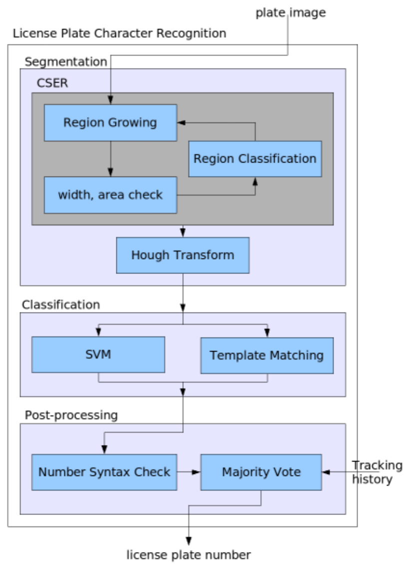


Tracker

- Kalman tracker [Kalman 60]
- Limit detector search to certain areas
- Create history of tracked objects



License Plate Character Recognition



Segmentation

- Crucial step for subsequent classification
- Isolate individual characters
  - Region growing
  - Classify character specific features [Matas and Zimmermann 05]
    - Compactness
    - Entropy of gray-scale histogram
    - Central invariant statistical moments
  - Hough transform

Character Classification

- Template matching
  - Compare segmented regions with templates
  - Take class with highest correlation value
- Support vector classification
  - Linear discriminant classifier
  - Direct pixels as features
  - Multi-class classification
    - ONE against ALL
    - Tree

Test Data

- Digital camera
  - 260 plates
  - 1400 chars (120x40)
  - 1200 chars (90x30)
- Video frames
  - 210 plates (~90x30)
  - Char size ~7x11

Results – Character Classification

- Template matching

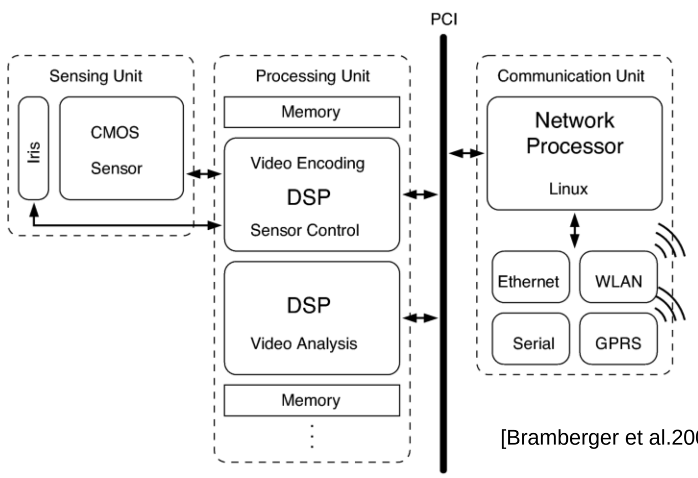
test set	CSER	post-process level	per char	per plate
1	N	0	79%	38%
1	N	1	82%	40%
1	N	2	**84%**	**54%**
1	Y	0	82%	40%
1	Y	1	81%	39%
1	Y	2	83%	53%
2	N	0	58%	21%
2	N	1	75%	26%
2	N	2	**76%**	**38%**
2	Y	0	76%	28%
2	Y	1	75%	28%
2	Y	2	76%	37%

Case Study – Autonomous Multi-Camera Tracking

Autonomous Multi-Camera Tracking

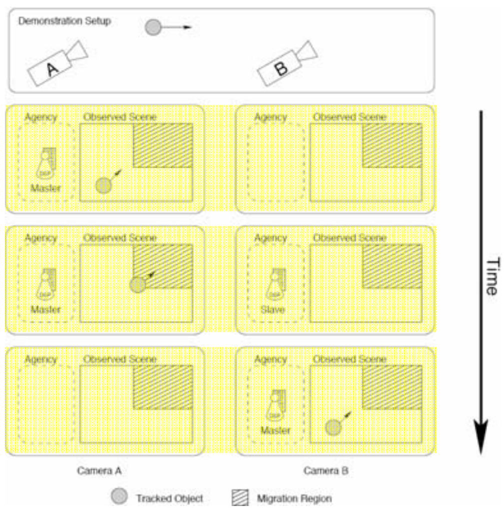
- Develop autonomous multi-camera tracking
  - On embedded smart cameras [Quaritsch et al.2007]
  - Using an arbitrary tracking algorithm
  - Without central coordination
- Tracking algorithm [Bradski 1998, Comaniciu et al. 2000]
  - Standard ("color-based") CamShift tracker
  - Tracker encapsulated in mobile agent
  - One tracking agent for each tracked object / person

Our SmartCam Architecture



[Bramberger et al.2006]

Master / Slave Handover Strategy



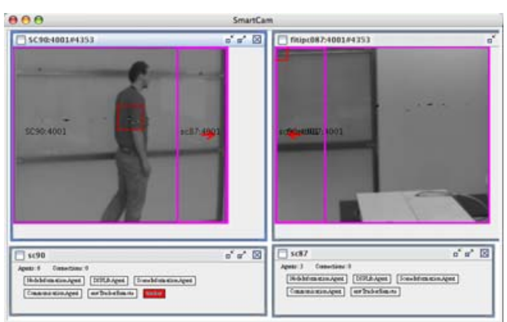
Master / Slave handover

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 as initialization data

Implementation and Results



Visualization

- Migration region (magenta)
- Tracked object (red rectangle)
- Tracking agent (red box)

Code size	15 kB
Memory requirement	300 kB
Internal state	256 B
Init color histogram	< 10 ms
Identify object	< 1ms

CamShift (single camera)

Loading dynamic executable	0.18 s
Initializing tracking algorithm	0.25 s
Creating slave on next camera	2.13 s
Reinitializing tracker on slave	0.04 s
Total	**2.57 s**

Multi-camera performance

Summary

- Smart camera applications
  - On embedded platforms
  - Examples: traffic, vehicles, persons
  - Domains: monitoring, surveillance, entertainment, compression
- Multi-camera setup
  - Local (pre)-processing
  - Collaboration among cameras
  - Bandwidth reduction by avoiding raw data streaming

Summary

- Advantages
  - Power efficiency
  - Resource utilization, availability
  - Bandwidth reduction, real-time
- Limitations
  - Resource limitations (memory, computing)
  - SW development (tools, libraries)

References

- Arth Clemens, Leistner Christian, Bischof Horst. Robust Local Features and their Application in Self-Calibration and Object Recognition on Embedded Systems. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07) (Embedded Computer Vision Workshop - ECV), to appear, 2007.
- Arth Clemens, Limberger Florian, Bischof Horst. Real-Time License Plate Recognition on an Embedded DSP-Platform. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07) (Embedded Computer Vision Workshop - ECV), to appear, 2007
- Arth Clemens, Leistner Christian, Bischof Horst. TRICam - An Embedded Platform for Remote Traffic Surveillance. In Proceedings of the 2nd Workshop on Embedded Computer Vision , IEEE International Conference on Computer Vision and Pattern Recognition, 2006-June
- Grabner Helmut, Bischof Horst. On-line Boosting and Vision. In IEEE Conference on Computer Vision and Pattern Recognition, Volume 1 (CVPR'06), pages 260-267, 2006
- H. Schwabach, M. Harrer, A. Waltl, Bischof Horst, A. Tacke, G. Zoffmann, C. Beleznai, B. Strobl, Grabner Helmut, G. Fernández. VITUS: Video based Image analysis for Tunnel Safety. In International Conference on Tunnel Safety and Ventilation, 2006

References

- Bernhard Rinner, Milan Jovanovic, Markus Quaritsch. Embedded Middleware on Distributed Smart Cameras. In Proc. of IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing. Hawaii USA, 2007.
- Markus Quaritsch, Markus Kreuzthaler, Bernhard Rinner, Horst Bischof, Bernhard Strobl. Autonomous Multi-Camera Tracking on Embedded Smart Cameras. EURASIP Journal on Embedded Systems. (special issue on Embedded Vision Systems). 2007.
- Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, Helmut Schwabach. Distributed Embedded Smart Cameras for Surveillance Applications. IEEE Computer 39(2) pages 68-75, February 2006.
- Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, Helmut Schwabach. Distributed Embedded Smart Cameras for Surveillance Applications. IEEE Computer 39(2) pages 68-75, February 2006.
- G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface”, Intel Technology Journal, vol. 2, no. 2, p. 15, 1998.
- D. Comaniciu, V. Ramesh, and P.Meer, “Real-time tracking of non-rigid objects using mean shift,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’00), vol. 2, pp. 142–149, Hilton Head Island, SC, USA, June 2000.

Outline