SIEMENS

SCR

Engineering, Statistical Modeling and Performance Characterization

…of a Real-Time Dual Camera Surveillance System

Michael Greiffenhagen

Supervisors:
Prof. H. Niemann, Dr. D. Paulus, Uni Erlangen
Dr. Visvanathan Ramesh, SCR, Princeton, USA

Outline

- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key-Modules:
  - Statistical Modeling / Performance Analysis
  - Validation of Assumptions and Models, Testing
- Evolution of Existing System:
  - How to relax constraints / add requirements
  - How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

Characteristics of Current Problems in Computer Vision

- *Compete* against other techniques (price, robustness).
- *Predictable* and *robust systems* (quantitative performance characterization within predefined bounds)
- We need "*simple*" and *fast* algorithms $\Rightarrow$ analyze tradeoff between *computation time* versus *accuracy*.
- Simplification only possible through extensive use of *context*

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Outline

- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key- Modules:
  – Statistical Modeling / Performance Analysis
  – Validation of Assumptions and Models, Testing
- Evolution of existing System:
  – How to relax constraints / add requirements
  – How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

SIEMENS

SCR

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Application Auto Cameraman: Track moving people, zoom onto faces

- Dual camera approach:
  - constantly monitor large area of interest (Shree Nayar's OmniCamera)
  - simultaneously: high resolution images of faces (e.g. face recognition)

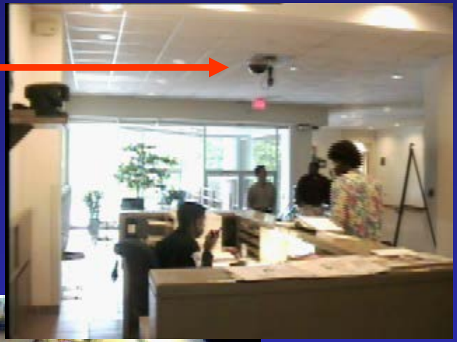
⇒ Monitoring / surveillance

⇒ Trigger alarms (sensitive areas)

⇒ Log information (time, location, face)

⇒ Post-processing (recognition/data-base)

Lobby scene

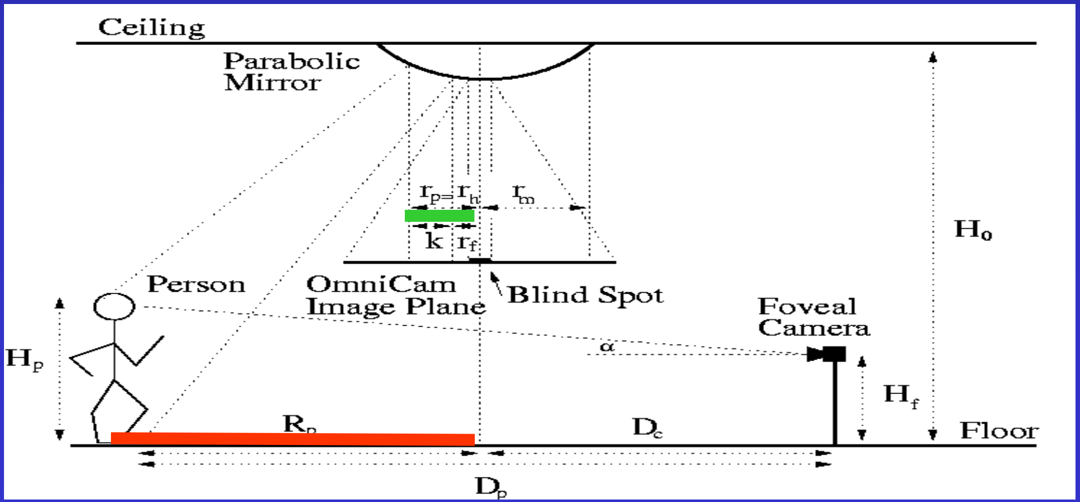


Omni-view

Foveal-view

“Simple” Problem to Solve...

- Goal is to control foveal camera (high resolution) by evaluating omni image:
  - *find* foot-(or head) position in omni-image
  - *transform* image coordinates into 3D world coordinates
  - *transform* 3D world coordinates into foveal-camera control parameters (pan, tilt, zoom)



Nonlinear Transform

$$R_p(r_p,\ldots) =$$

$$= 2\frac{r_p r_m}{r_p^2 r_m^2}(H_0 - H_p)$$

Unfortunately: Not that simple!

Brute-forth feasibility study (Master's, ACCV'98) showed:

⇒System unpredictable and unreliable

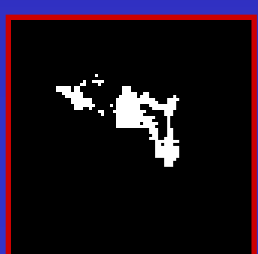
- Zoomed *in* too much - face cut off / not in captured in frame
- Zoomed *out* too much - face too small

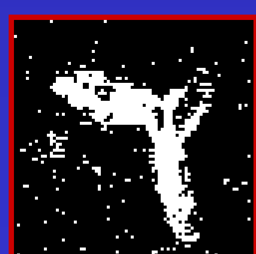
⇒Performance depending on

- Object location
- Mounting position of cameras
- Sensor noise
- Spatially varying Discrimination Power (background / cloths)
- Time of the day (different light conditions / shadow)

Solution: On-line Performance Analysis

Why the effort? *Because*, always errors involved; small errors can be fatal; *best estimate not good enough*: we need to know quantitatively *how good* it is!



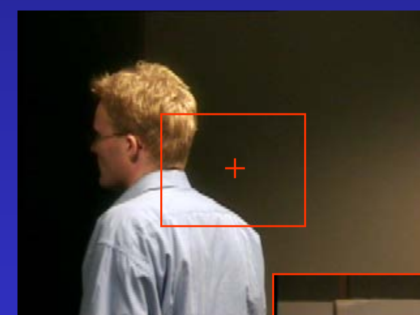


Where are the feet?

Quiet *uncertain*…

We better zoom accordingly!

zoomed out



zoomed in

Preview: The System

- Arbitrary positions, varying discrimination power.
- Different setting – no system parameter changes
- Self-adjusting tuning parameters: Adaptive zooming!

Approach: Statistical Modeling

- Decompose System into Modules
- Model input statistically (true + perturbation)
- Model output statistically as f(input, transform, parameters)

⇒ Propagate distributions / uncertainties through each module (calibration, sensor, segmentation, object localization)

⇒ Perform on-line analysis to estimate uncertainties

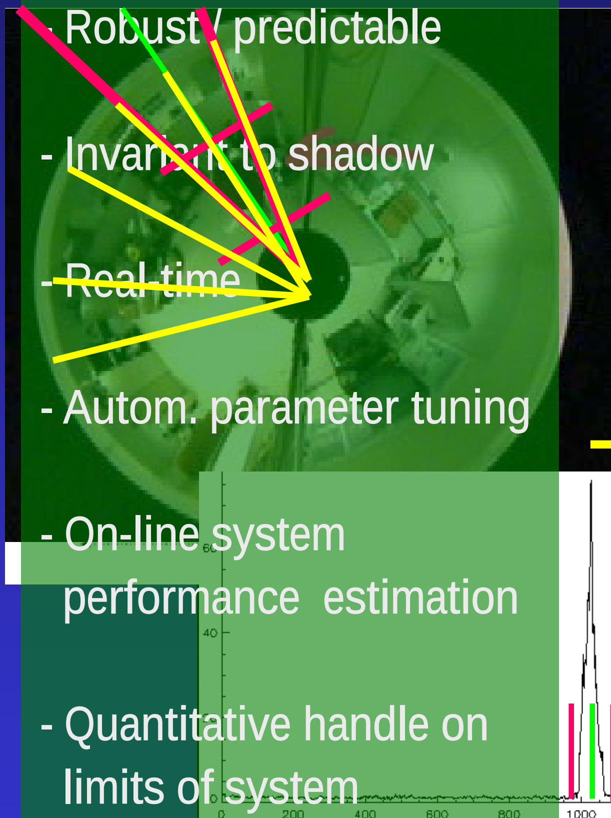
<u>Advantages:</u>

AutoCameraMan:	- Not only best estimate on location
	- also quantitative measure on quality of estimate
	- used for optimizing zoom
General:	- estimate parameters / thresholds adaptively
	- estimate optimal performance at any position/time given data and task

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

System Requirements + Prior Knowledge = Sys.-Architecture

Uncertainty / Modules / Trafos
Propagation



- Robust / predictable
- Invariant to shadow
- Real-time
- Autom. parameter tuning
- On-line system performance estimation
- Quantitative handle on limits of system

Sensor Noise: Gaussian in RGB
$N(RGB,\sigma_R\sigma_G\sigma_B)$
Covar-propagation
Illumination: n sources: $A_i\ p(\lambda)$
$N(rg,\Sigma_{rg})$
Distrib. propagation
Scene: Background static
χ^2
Distrib. Propagation
Projection: upright → on floor
χ^2_{2m} or $\chi^2_{2(m+n)}$
bootstrap
Geometry: OmniCam projection
$N(r,\varphi,\sigma_r\sigma_\varphi)$
Covar.-propagation
Calibration: dual cam setup
$N(\alpha,\beta,\sigma_\alpha\sigma_\beta)$
Covar.-propagation
Object: head centered
Foveal camera: optical features
$N(zoom,\sigma_{zoom})$

Sensor
Model: ideal + noise
Illumination compens.
Intensity normalization
Change detection
Mahalanobis distance
Fast indexing
Integration along lines
Foot position estimation
2D image ⇒ 3D world
Control parameter estim.
Simple trigonometry
Zoom parameter estim.
Simple trigonometry

Example: Distribution Propagation to Estimate System Performance On-line

Given: - Sector ROI from previous step (change detection module)
- Input distribution, here: difference image pixel $\chi^2(c)$ distributed

Goal: - Generate "some" accumulative (stable) measure M (ft-pos estimation)

Trafo: - Sum values within region $\Rightarrow$ get input for next module

Key: - This trafo allows to derive output *distribution*: $\chi 2_{2n}(c)$

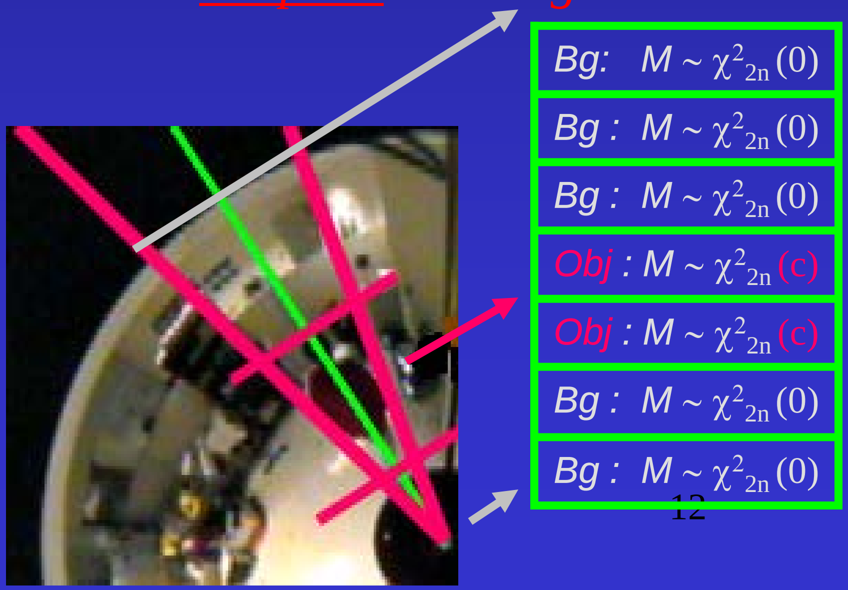
$\Rightarrow$ Allows to estimate *uncertainty* in foot position for *adaptive zooming*

Known Sector of Interest

Known Distribution $M \sim \chi^2_{2n}(0)$ *or* $\chi^2_{2n}(c)$

Estimated Centrality Parameter c
from current data

Estimated Degrees of Freedom n
from geometry, segmentation result



Radial Foot Position

- Use prior knowledge of geometry (interval partitioning)
- Find best hypothesis h_i which minimizes Bayes error (hypothesis test).
- h_i : i^{th} out of multiple foot position hypothesis.
- Profile: m: measurement M.

$$
\begin{aligned}
&P(h_i|m) \\
&= P(h_i^b|m^b)P(h_i^o|m^o) = P(h_i^b|m^b)\left(1 - P(\bar{h}_i^o|m^o)\right) \\
&= \frac{p(m^b|h_i^b)P(h_i^b)}{p(m^b)} \frac{p(m^o) - p(m^o|\bar{h}_i^o)P(\bar{h}_i^o)}{p(m^o)}
\end{aligned}
$$

⇒ Values known, maximize P(r_f | profile) ⇒ approximate P(partition | profile):

Test: for

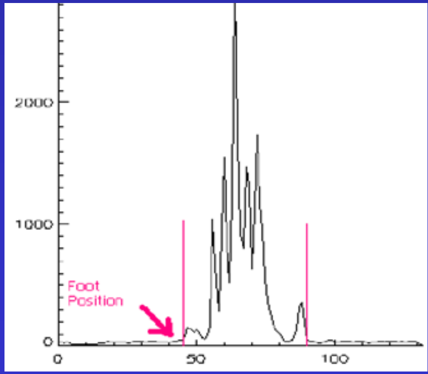
Prior | Hypothesis test

Distribution known | From projection model | Find foot position that

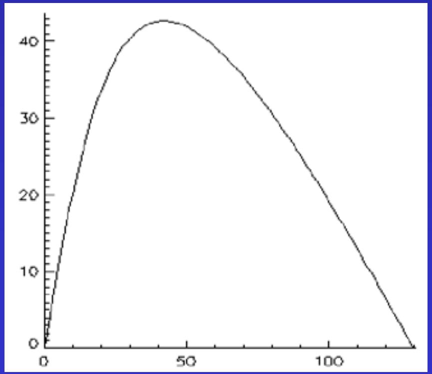
Where is object? | Kernel used for partitioning | corresponds to lowest cost

$$
r_f = \underset{r'_f}{\operatorname{argmin}} \log\left(\frac{p(m^o|\bar{h}_i^o)}{p(m^b|h_i^b)}\right)
$$

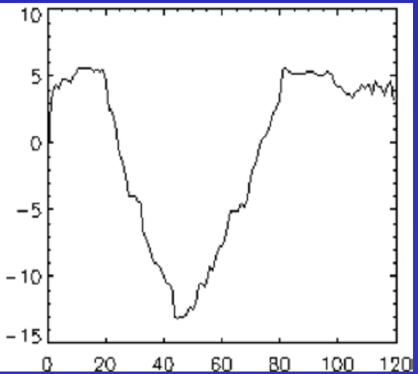
$$
\log\left(\frac{p(m^o|\bar{h}_i^o)}{p(m^b|h_i^b)}\right) = \left(\sum_{r=0}^{r'_f-1} \hat{M}_{r,\theta_f}^\top + \sum_{r=r_n(r'_f)}^{r_m} \hat{M}_{r,\theta_f}^\top - \sum_{r=r_f}^{r_n(r'_f)-1} \hat{M}_{r,\theta_f}^\top\right)
$$



Sum of profiles along radial line within sector of interest



Projected length of object



Cost function: foot position estimation

⇒ Since distributions known: Estimate uncertainty using bootstrap/sampling technique

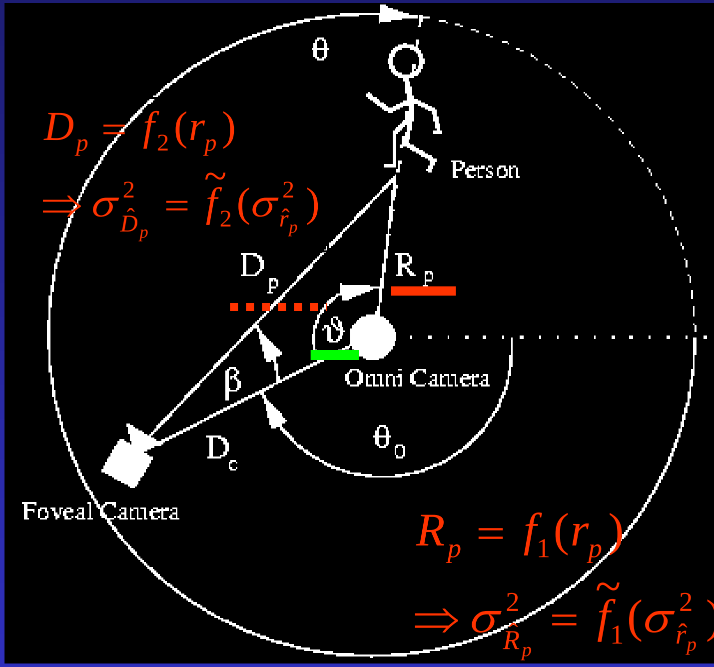
Foveal Camera Parameter Estimation

Now, all parameters available:

- from current image: foot position
- from calibration: remaining parameters
- $r_p \Rightarrow R_p$ (see geometry model)

3-D & covariances (*covar. propagation*)

$\Rightarrow$ foveal camera control-parameters & σ^2



$$D_p = \sqrt{D_c^2 + R_p^2 - 2 D_c R_p \cos(\vartheta)}$$

$$\tan(\alpha) = \frac{H_p - R_h - H_f}{D_p}; \qquad \sin(\beta) = \frac{R_p}{D_p}\sin(\vartheta)$$

$$\sigma^2_{\tan\hat{\alpha}} = \frac{\sigma^2_{\hat{D}_p}}{D_p^4}\left((H_p - R_h - H_f)^2 + \sigma^2_{\hat{H}_p} + \sigma^2_{\hat{R}_h} + \sigma^2_{\hat{H}_f}\right) + \frac{\sigma^2_{\hat{H}_p} + \sigma^2_{\hat{R}_h} + \sigma^2_{\hat{H}_f}}{D_p^2}$$

$$\sigma^2_{\sin\hat{\beta}} = \frac{R_p^2 \sigma_{\hat{\vartheta}}^2 \cos^2\vartheta}{D_p^2} + \left(\sin^2\vartheta + \sigma_{\hat{\vartheta}}^2 \cos^2\vartheta\right) * \left(\frac{R_p^2 \sigma^2_{\hat{D}_p}}{D_p^4} + \frac{\sigma^2_{\hat{R}_p}}{D_p^2} + \frac{\sigma^2_{\hat{R}_p}\sigma^2_{\hat{D}_p}}{D_p^4}\right)$$

Model:

-Calibration parameters normal distributed

Assumptions:

-small errors (Linearization)

-parameters independent

-single level floor

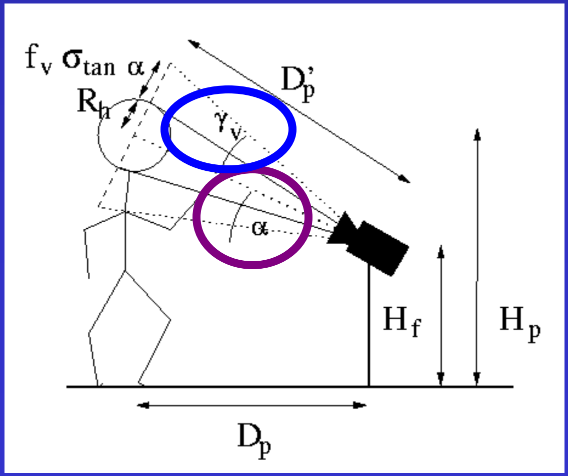
Zoom Setting

Models involved:

- Function of **geometry**: relative position of object to foveal camera
- Function of **calibration**, **signal to noise** ratio
- Function of **location** and **quality of segmentation**: uncertainty in pan and tilt

  ⇒ Function of relative angular position of object / omni-camera (uncertainty in ϑ)

  ⇒ Function of discrimination power between background / object (uncertainty in r_p)



$$\gamma_v = 2\ \mathrm{atan}\left(\frac{\hat{R}_h + f_v\,\sigma_{\tan\hat{\alpha}}\,\hat{D}'_p}{\sqrt{\hat{R}_h^2 + \hat{D}_p'^2}}\right) \quad \text{with} \quad \hat{D}'_p = \frac{\hat{D}_p}{\cos\alpha}$$

where factor f_v solves for $\int_0^{\frac{f_v}{2}} N(0,1)\,d\xi = \frac{x_z}{2}\%$

Requirement: guarantee, that entire head is in $x_z\%$ in frame, given segmentation, stationary (→ outlook)

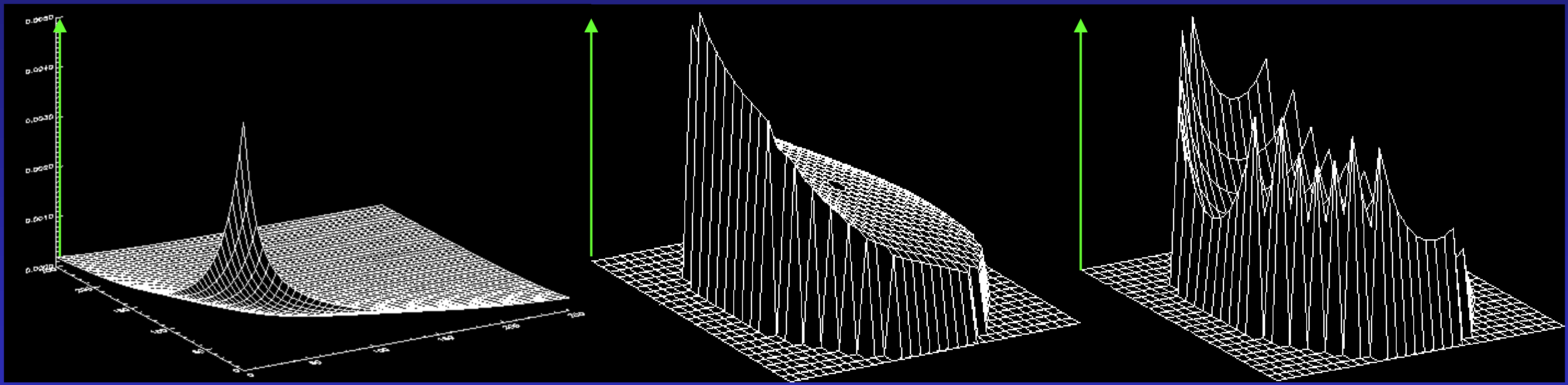
Overview

- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key- Modules:
  - Statistical Modeling / Performance Analysis
  - Validation of Assumptions and Models, Testing
- Evolution of existing System:
  - How to relax constraints / add requirements
  - How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

Validation of Assumptions: Need to know *where* models hold

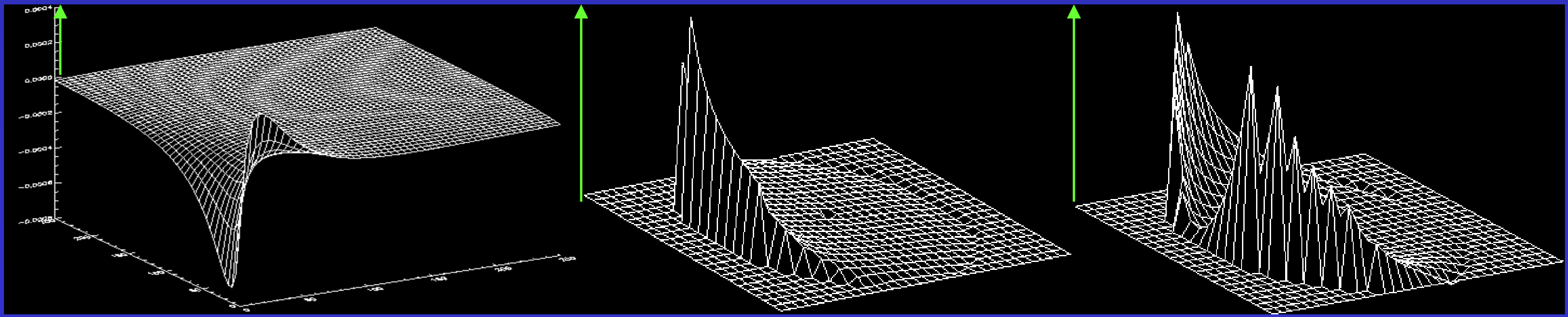
Theoretical variances for: color normalization, tilt, pan over scene

0.0050 0.0014 0.0025



Difference between simulation and theory (see scale!)

0.0004 0.0001 0.0020



Validation of Models

- With best theory, and nicest models we can not "buy" anything; *unless we verify correctness of models* used!

Here, 9 positions:

Ground-truth against model-estimates (tilt, pan):

$\times 10^{-5}$	P1	P2	P3	P4	P5	P6	P7	P8
$\hat{\sigma}^2_{\tan\hat{\alpha}}$	2.10	2.12	1.57	1.40	1.35	1.31	1.31	1.32
$\tilde{\sigma}^2_{\tan\hat{\alpha}}$	2.05	2.04	1.60	1.34	1.36	1.32	1.40	1.31
$\hat{\sigma}^2_{\sin\hat{\beta}}$	28.9	26.1	21.3	17.9	15.3	15.2	18.4	20.1
$\tilde{\sigma}^2_{\sin\hat{\beta}}$	25.9	24.1	19.5	15.1	14.9	15.0	18.1	19.3

Overview

- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key- Modules:
  - Statistical Modeling / Performance Analysis
  - Validation of Assumptions and Models, Testing
- Evolution of existing System:
  - How to relax constraints / add requirements
  - How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

New System-Requirements Relaxed Constraints

Previous system

- Invariant to shadows
- Constant background
- Constant illumination
- Single mode background
- Restricted operational in black and saturated background areas

Extended system

- Invariant to shadows
- *Variable* background
- *Variable* illumination
- *Multi* modal background
- Operational on *full range* of sensor input

Approach:

- Add new background adaptation module
- Fusion with existing system

Requirements for Fusion

- ***Reuse*** of old system modules & analysis.
- ***Maintain*** *design and statistical behavior* of old modules after new modules are added to system.

⇒ *"Open" chain of transforms, "insert" additional module*

⇒ ***Maintain*** *interface distribution* requirements (sequential case)

⇒ ***Maintain*** *output distribution* of module that takes extended feature vector as input (parallel case)

⇒ ***Maintain*** *output feature space* (shadow assignment).

⇒ Find module and stable features that fulfill these requirements

⇒ Describe it statistically and fuse it accordingly

… preferably, existing 3rd party module!

Multi-Modal Background Adaptation

Third party module: Stauffer/Grimson: “Adaptive Background mixture model for real-time tracking”, CVPR 1999

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} * \zeta(X_t, \mu_{i,t}, \sigma_{i,t})$$

⇒ Sum of *K* weighted Gaussians.

For *i*<*B*: mode *i* labeled background with:

$$B = argmin_b \left(\sum_{k=1}^{b} w_k > T \right)$$

w_k ordered
T application depedent
(not illumination dependent)

⇒ Online parameter update (mean, variance, weight).

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

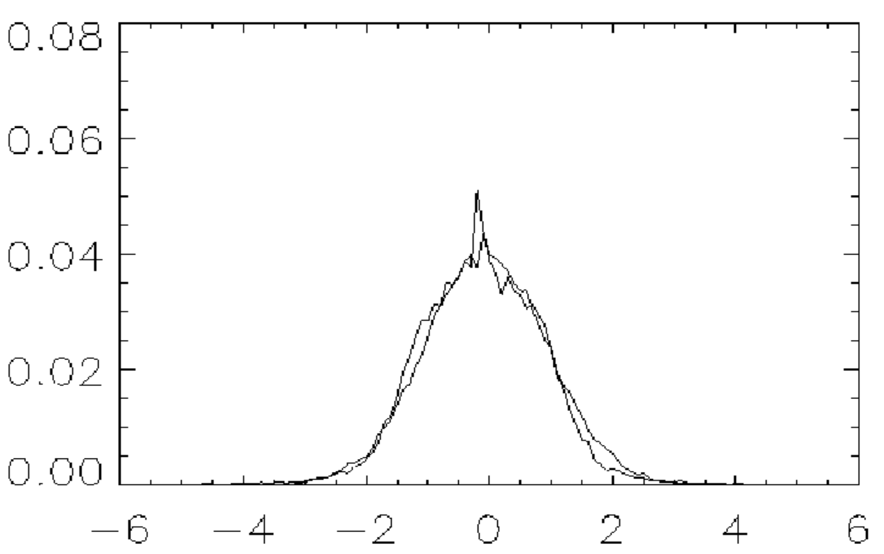
Only Mean is Stable Feature!

$$X_{b,t} = \mu_{j,t} | j = \underset{k}{\operatorname{argmin}} (\mu_{k,t} - X_t)^2 \, \forall k \in \{0 \ldots B\}$$

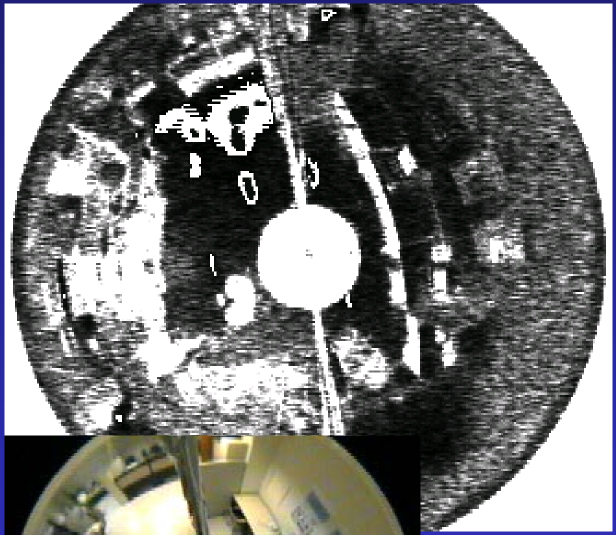
$$\Delta_t = X_{b,t} - X_t \sim N(0, \sqrt{2}\sigma_{\hat{\mu}_X, j, t})$$

Difference between current value and <u>*closest*</u> background mean.

Linear error propagation not feasible: $P(X_t)$ is no stable feature!



Fusion 1: *Update* Normalized Color Based Change Detection Module



Stauffer/Grimson applied to *R,G,B* bands provides stable mean estimate, which is used to update background parameters in normalized color space (mean and covariance matrix)

$$\hat{d}^2 = (\hat{\mu}_\mathbf{b} - \hat{\mu}_\mathbf{c})^T \left(2\Sigma_{\hat{\mathbf{r}}_\mathbf{b},\hat{\mathbf{g}}_\mathbf{b}}\right)^{-1} (\hat{\mu}_\mathbf{b} - \hat{\mu}_\mathbf{c})$$

$$\Sigma_{\hat{\mathbf{r}}_\mathbf{b},\hat{\mathbf{g}}_\mathbf{b}} = \frac{\sigma_I^2}{S^2} \begin{pmatrix} \frac{\sigma_R^2}{\sigma_I^2}\left(1 - \frac{2R}{S}\right) + 3\frac{R^2}{S^2} & -\frac{\sigma_G^2 R + \sigma_R^2 G}{\sigma_I^2 S} + 3\frac{RG}{S^2} \\ -\frac{\sigma_G^2 R + \sigma_R^2 G}{\sigma_I^2 S} + 3\frac{RG}{S^2} & \frac{\sigma_G^2}{\sigma_I^2}\left(1 - \frac{2G}{S}\right) + 3\frac{G^2}{S^2} \end{pmatrix}$$

with $\sigma_I^2 = \sigma_S^2/3 = (\sigma_R^2 + \sigma_G^2 + \sigma_B^2)/3$

Fusion 2: *Add Intensity* to Normalized Color Based Change Detection Module

Probability of pixel being background ⇒ Mahalanobis distance

$$\hat{d}'^2 = (\hat{\mu}'_{\mathbf{b}} - \hat{\mu}'_{\mathbf{c}})^T \left(2\Sigma_{\hat{\mathbf{r}}_{\mathbf{b}},\hat{\mathbf{g}}_{\mathbf{b}},\hat{\mathbf{I}}_{\mathbf{b}}}\right)^{-1} (\hat{\mu}'_{\mathbf{b}} - \hat{\mu}'_{\mathbf{c}}) \qquad \text{with}$$

$$(\hat{\mu}'_{\mathbf{b}} - \hat{\mu}'_{\mathbf{c}}) = \begin{pmatrix} \hat{r}_b - \hat{r}_c \\ \hat{g}_b - \hat{g}_c \\ \hat{I}_b - \hat{I}_c \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{\mu}_{\mathbf{b}} - \hat{\mu}_{\mathbf{c}} \\ \hat{I}_b - \hat{I}_c \end{pmatrix}, 2\Sigma_{\hat{\mathbf{r}}_{\mathbf{b}},\hat{\mathbf{g}}_{\mathbf{b}},\hat{\mathbf{I}}_{\mathbf{b}}} \right)$$

$$\hat{d}'^2 = (\hat{\mu}_{\mathbf{b}} - \hat{\mu}_{\mathbf{c}})^T (2\Sigma_{\hat{\mathbf{r}}_{\mathbf{b}},\hat{\mathbf{g}}_{\mathbf{b}}})^{-1} (\hat{\mu}_{\mathbf{b}} - \hat{\mu}_{\mathbf{c}}) + (\hat{I}_b - \hat{I}_c)^T \frac{1}{2\sigma^2_{\hat{I}_b}} (\hat{I}_b - \hat{I}_c)$$

$$\hat{d}'^2 = \hat{d}^2 + \frac{(\hat{I}_b - \hat{I}_c)^2}{2\sigma^2_{\hat{I}_b}}$$

Background: $d^2 \sim \chi^3(0)$

Object $d^2 \sim \chi^3(c)$

⇒Distribution form <u>maintained</u>; only degrees of freedom increased!

Maintain Feature space

Problem:

- Normalized color space segments into:

  object vs. background *including shadow*

- Stauffer/Grimson scheme segments into:

  object *including shadow* vs. background

Solution:

⇒ Assign shadow mode to background classes by hypothesis testing (feature space has to be maintained!).

Hypothesis Testing [K.V. Mardia]

Shadow: $H^0: \quad \vec{\mu}_{\mathbf{t-1}} = k\vec{\mu}, \quad \vec{\mu}, \mathbf{\Sigma} \quad \text{known.}$ ...all bands experience same change

m.l.e. for $\hat{k} = (\mathbf{v_{t-1}}^T \mathbf{\Sigma}^{-1} \mu)/(\mu^T \mathbf{\Sigma}^{-1} \mu)$

Non-Shadow: $H^1: \vec{\mu}_{\mathbf{t-1}} = \mathbf{K}\vec{\mu}, \quad \mathbf{K} = \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{pmatrix}; \quad \vec{\mu}, \mathbf{\Sigma} \text{ known.}$ ...all bands experience different change

m.l.e. for $\hat{k}_i = \dfrac{\mathbf{v_{t-1}}^T \mathbf{\Sigma}_i^{-1}}{\mu^T \mathbf{\Sigma}_i^{-1}}$

Likelihood ratio test for $\hat{d}_t = ln(p(shadow) / p(non\text{-}shadow))$:

$$\hat{d}_t = (\hat{\mathbf{v}}_{\mathbf{t-1}} - \hat{k}\vec{\mu})^T \mathbf{\Sigma}^{-1} (\hat{\mathbf{v}}_{\mathbf{t-1}} - \hat{k}\vec{\mu}) - (\hat{\mathbf{v}}_{\mathbf{t-1}} - \hat{\mathbf{K}}\vec{\mu})^T \mathbf{\Sigma}^{-1} (\hat{\mathbf{v}}_{\mathbf{t-1}} - \hat{\mathbf{K}}\vec{\mu})$$

Known distribution: chi-square distributed with 2 degrees of freedom: $\hat{d}_t \sim \chi^2(0)$

Revised Background Estimate

Background estimation for each band (R,G,B,I):
Similar: Choose closest background mode but *including* shadow mode!

old background modes

$$X_{b,t} = \mu_{j,t} | j = \underset{k}{\operatorname{argmin}} (\mu_k - X_{c,t})^2 \forall k \in \{0 \ldots B, K+1\}$$

added shadow mode

Stauffer/Grimson model parameter update:

- Object model parameters: weighted by *P(non-shadow)*
- Shadow model parameters: weighted by *P(shadow)*

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Overview

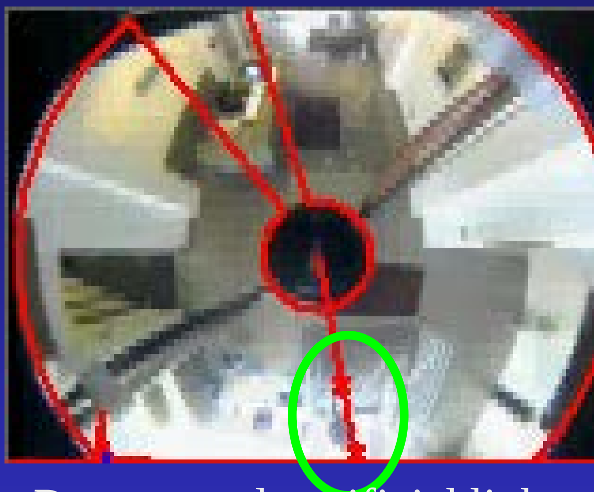
- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key- Modules:
  - Statistical Modeling / Performance Analysis
  - Validation of Assumptions and Models, Testing
- Evolution of existing System:
  - How to relax constraints / add requirements
  - How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

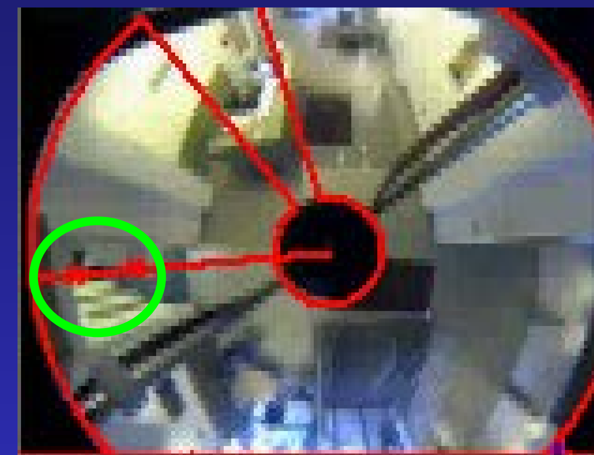
SIEMENS

SCR

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Continuous Changes over 24h

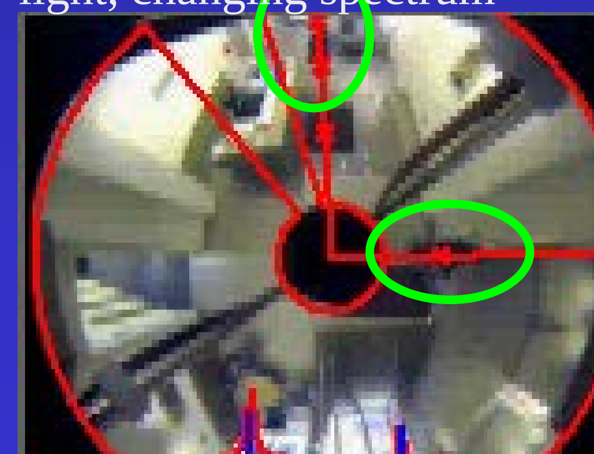




__Day__: natural+artificial light; saturation

__Night__: artificial light low contrast regions

__Morning/afternoon__: mixed light, changing spectrum



Results (Hybrid Setting)

- Experiment: 24h a day, 2 weeks (day and night)
- High / low contrast
- Changing background / shadow
- Varying object size

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Overview

- Introduction
- Application: Auto Cameraman
- Demonstrate Approach on Key- Modules:
  - Statistical Modeling / Performance Analysis
  - Validation of Assumptions and Models, Testing
- Evolution of existing System:
  - How to relax constraints / add requirements
  - How to fuse existing and new modules correctly
- Results, Demos
- Summary / Outlook

Chair for Pattern Recognition
FAU Erlangen-Nuremberg

Summary / Contributions

- **Showed how to systematically design a complete real-time vision system that is predictable, robust & guarantees performance within pre-defined bounds in a real-world setting**
  - ⇒ Decompose system into modules
  - ⇒ Statistical modeling and analysis of each module
  - ⇒ Propagation of uncertainties from input data to final output
  - ⇒ Complete engineering cycle: design, analysis, validation, test
  - ⇒ Optimize performance given data and task
  - ⇒ Optimize setup / camera position

- **Demonstrated how to evolve an existing system incrementally to meet added requirements**
  - ⇒ without redesign of existing modules
  - ⇒ fusion of existing and 3rd party module combining strength of both
  - ⇒ maintaining analysis of prior system valid

⇒ **Built working system – *stable* and in lobby at SCR, Princeton, *in use***

Outlook / Future Work

- Sudden light changes that require *new* state
- Multiple people along *same* radial line
- Analysis of tracker to improve foveal camera control dynamics
- Feedback / evaluate foveal image

Thank you!

Acknowledgements

Many thanks to
Prof. Heinrich Niemann,
Dr. Visvanathan Ramesh,
and my future wife, *Katja*.

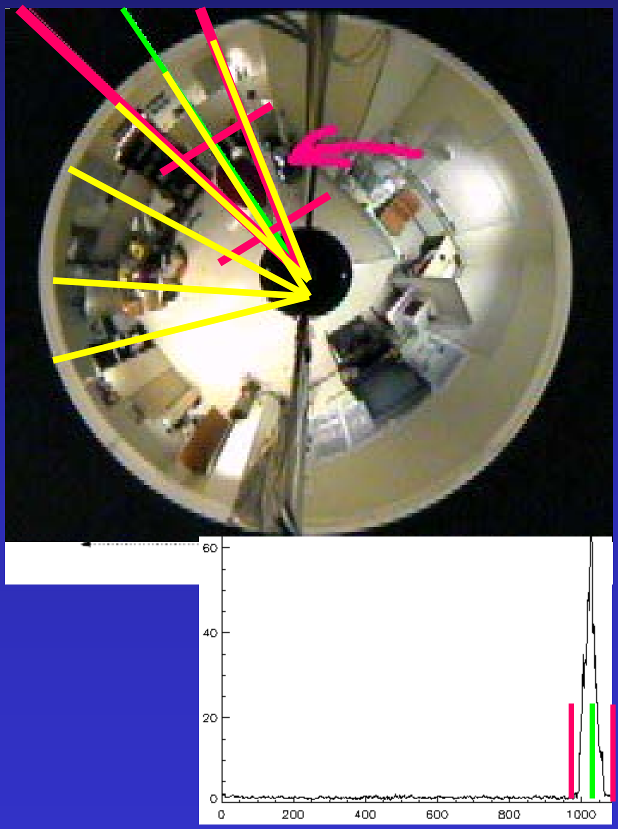
They gave me the opportunity and supported the idea of performing this research work abroad – it was a great experience, not only from the academic point of view!

Modules / Requirements		Priors: Constraints		Transforms / High-Level Algo.
Sensor-noise model		Sensor Noise: Gaussian in RGB		Model: ideal + noise
Illumination compensation		Illumin.-sources: n, A_i p(λ)		Intensity Normalization
P(change) on pixel level		Background: static		Mahalanobis distance
Indexing (real-time)	+	Object : person upright on floor	=	Integration along lines
Hypothesis generation: BG?		Scene : mainly background		Dynamic thresholding
Foot position estimation		Geometry: OmniCam projection		2D image $\Rightarrow$ 3D world
Control parameter estim.		Calibration: dual-cam setup		Simple trigonometry
Zoom parameter estim.		Object geometry: head centered Foveal camera: optical features		Simple trigonometry

LME Chair for Pattern Recognition FAU Erlangen-Nuremberg

Uncertainty Propagation

Transforms / High-Level Algo.



Uncertainty Propagation	Transforms / High-Level Algo.
	Model: ideal + noise
$N(RGB,\sigma_R\sigma_G\sigma_B)$	
Covar.-propagation	Intensity Normalization
$N(rg,\Sigma_{rg})$	
Distrib. propagation	Mahalanobis distance
χ^2	
Distrib. Propagation	Integration along lines
χ^2_{2m} or $\chi^2_{2(m+n)}$	
---	Dynamic thresholding
region of interest	
bootstrap	2D image $\Rightarrow$ 3D world
$N(r\varphi,\sigma_r\sigma_\varphi)$	
Covar.-propagation	Simple trigonometry
$N(\alpha\beta, \sigma_\alpha\sigma_\beta)$	
Covar.-propagation	Simple trigonometry
$N(r\varphi,\sigma r\sigma\varphi)$	