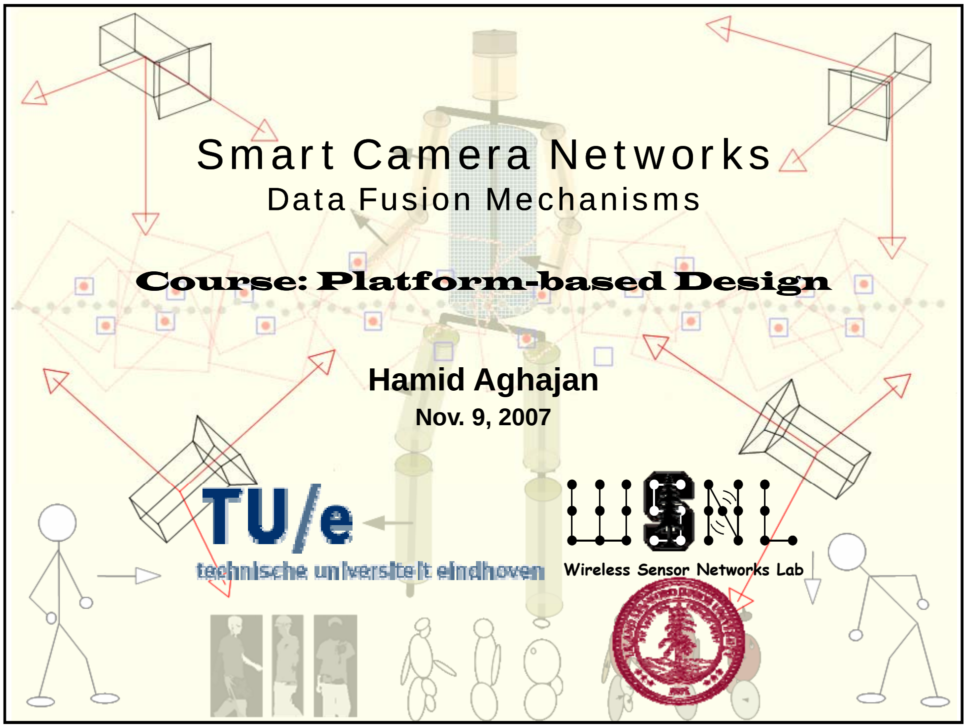


Outline

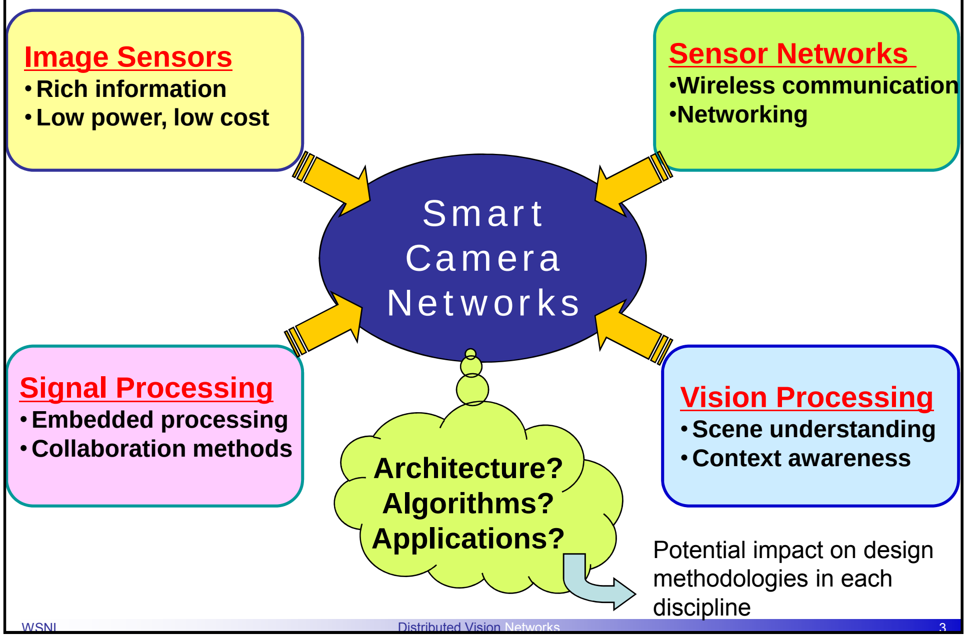
- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

Human pose estimation

Human event detection

Technology Cross-Roads



Distributed Vision Networks

❖ Rich design space utilizing concepts of:
  – Vision processing
  – Signal processing and optimization
  – Wireless communications
  – Networking
  – Sensor networks

❖ Value proposition:
  – Picture better than 1000 words
  – Multiple cameras
  – Be careful about communication bandwidth
  – Be aware of privacy issues

❖ Novel smart environment applications:
  – Interpretive
  – Context aware
  – User centric

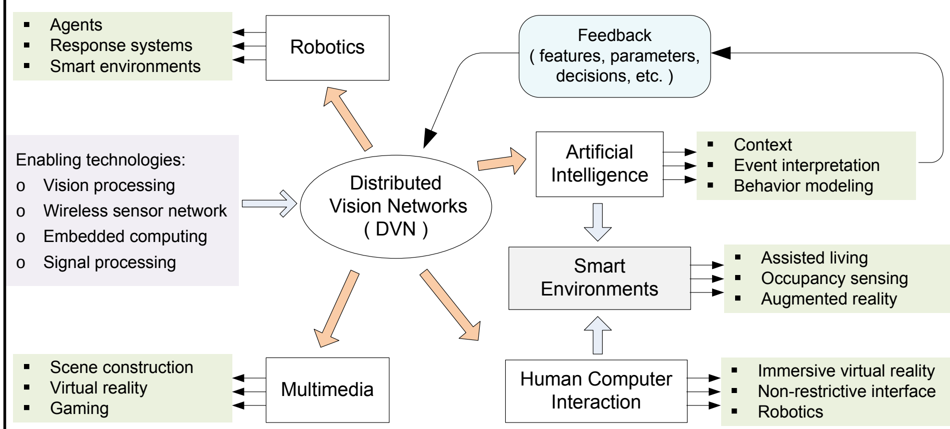
Distributed Vision Networks

- ❖ Processing at source allows:
  - Image transfer avoidance
  - Descriptive reports
  - Scalable networks
- ❖ Design opportunities:
  - Processing architectures for real-time in-node processing
  - Algorithms based on opportunistic data fusion
  - Novel smart environment applications
  - Balance of in-node and collaborative processing:
    - Communication cost
    - Latency
    - Processing complexities
    - Levels of data fusion

Distributed Vision Networks

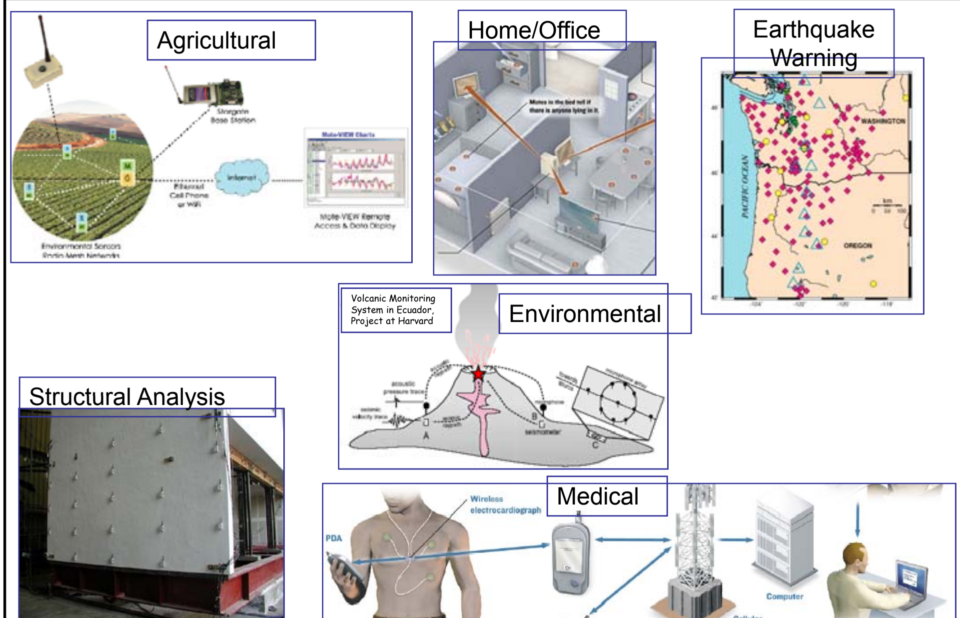
- ❖ Vision sensing requires awareness of:
  - Privacy issues
    - Employ in-node processing
    - Avoid image transfer
    - Applications that provide services not based on monitoring / reporting
  - Bandwidth issues
    - Transmit processed information not raw data
    - Transmit based on information value for fusion / query-based
  - Processing demand
    - Employ separate early vision and interpretive processing mechanisms
    - Layered processing architecture: Features, objects, relationships, models, decisions
      - Employ data exchange and collaboration across different layers

Distributed Vision Networks



Wireless Sensor Networks

Applications



Communication Perspective

Cellular / Mobile Ad-hoc Networks	Wireless Sensor Networks
► Designed to optimize QoS / provide high throughput	► Deployed for common task
► High BW data major part of traffic	► Generally low bandwidth data
► Data flow generally bi-directional	► Data flow uni-directional (source to sink), often broadcasting
► Energy consumption secondary	► Energy consumption primary issue
► Nodes compete for resources	► Nodes work together on resources

Design Perspective

- ➢ Priorities and metrics different
- ➢ Cannot tune traditional methods to special case
- ➢ Need a design paradigm shift

WSN Design Paradigm

- In *wireless* domain:

Other Wireless Networks	Wireless Sensor Networks
1. Network's role: data transport	1. Network's role: information collection and dissemination
2. Network nodes compete for resources	2. Nodes collaborate on resource allocation
3. High data rates (e.g. video streaming)	3. Low data rates (e.g. image attributes transmitted)

Metric:
maximize network throughput

Metric:
Maximize network lifetime

WSN Design Paradigm

- In *processing* domain:

Other Processing Networks	Wireless Sensor Networks
1. Few high-accuracy sensors	1. Many low-accuracy sensors
2. Raw data communicated	2. Data processed first
3. Centralized processing	3. Distributed processing
4. Application relies on high accuracy of measurements	4. Application relies of multiple sources of measurements

Metric:
Optimal solution

Metric:
Energy & BW efficiency,
Sub-optimal solution

WSN Design Paradigm

Communication design perspective

- Perfect processing:
  - "Powerful central processor"
- Design problem:
  - "Maximize rate and throughput to get data there fast"

Data processing design perspective

- Perfect communication:
  - "All data will be available in time"
- Design problem:
  - "Find globally optimal solution"

Wireless Sensor Networks

- Long-distance transmission expensive
  - Limited bandwidth
  - Large correlation/redundancy in data
  - No central processing unit
- Sub-optimal solution ok in many applications

- Local exchange of data
- Distributed processing
- Communicate *information*

WSN: Network-Centric Nature

- Monitoring the environment has been the main application driver
  - Wildlife habitat monitoring
  - Forest fires
  - Surveillance and security applications
  - Tracking assets and people
- *The network* is in charge
  - Measures, computes, makes decisions, reports
  - Everything else is considered data, data source, or data path
- New direction: Put *the user* in charge
  - Move from network-centric design to *user-centric* design
  - Learn behaviors not just measure effects
  - Bring context awareness into the application

WSN: Report-Centric Nature

- Sensor networks mainly a tool to monitor and report
  - Outside observer may decide on actions based on reported data
- New directions:
- Interpretive network:
  - Actively look for useful data
  - Adjust data acquisition based on interpretation
- Context awareness:
  - Provide services based on user's context
    - Location, status, activity, events
- Ambient intelligence:
  - Detect and track context of user and other events

Vision Sensor Networks

Sensor Networks Perspective

❖ Opportunities for novel applications:

- Make *complex interpretation* of environment and events
- Learn *phenomena* and *behavior*, not just measure effect
- Incorporate *context awareness* into the application
- Allow network to *interact* with the environment

- Change of paradigm:
  High-bandwidth sensors (vision)

Vision Processing Perspective

❖ Novel approach to vision processing:

- Use the additional available dimension: space
  - Data fusion across views, time, and feature levels
- Design based on effective use of all available information (opportunistic fusion)
- Utilize multiple views to:
  - Overcome ambiguities
  - Achieve robustness
  - Allow for low complexity algorithms
- Use communication to exchange descriptions - not raw data
  - In-node processing

- Change of paradigm:
  Networked vision sensors

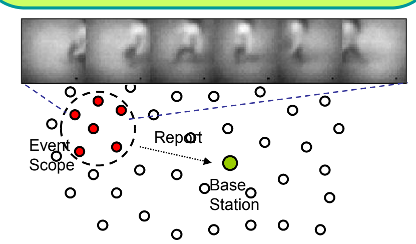
Distributed Vision Networks

New Paradigm

High-bandwidth data
⇩
In-node processing
⇩
Low-bandwidth communication
⇩
Collaborative interpretation

Distributed Vision Systems

Traditional Approach	Image Sensor Networks
• Few high-resolution sensors	• Many low/high-resolution sensors
• Raw images communicated	• Images processed first
• High data rates (visual data transmitted)	• Low data rates (attributes transmitted)
• Centralized processing	• Distributed processing



- Inefficient network use
- Not scalable

- Efficient resource (comm./comp.) use
- Adaptive acquisition/response possible

Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

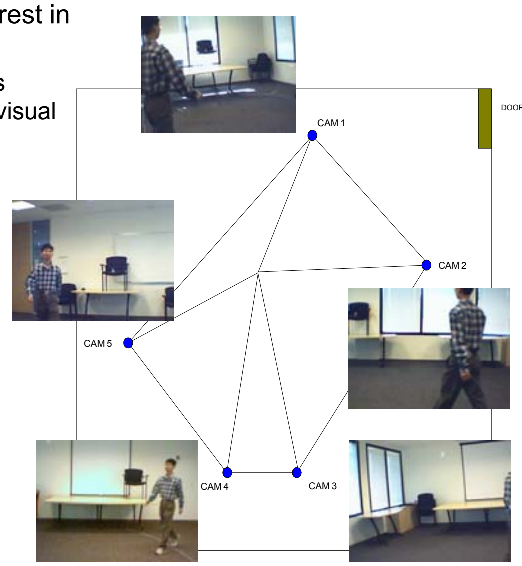
Human face angle estimation

Human pose estimation

Human event detection

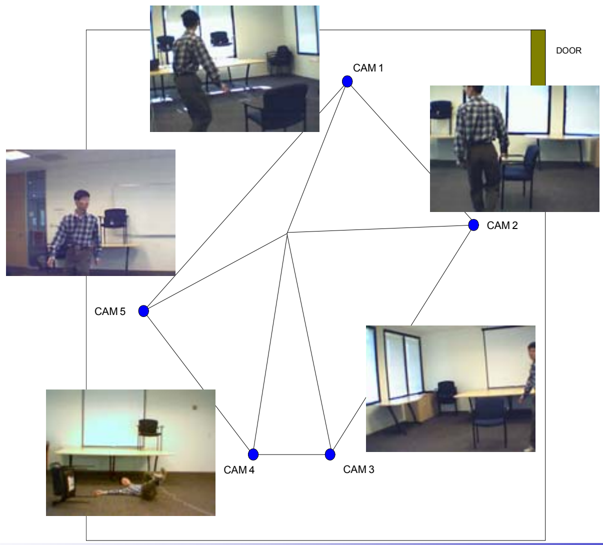
Application Potentials: View Selection

- Select best view of person of interest in real-time tracking
  - Data exchange between cameras determines which one to stream visual data



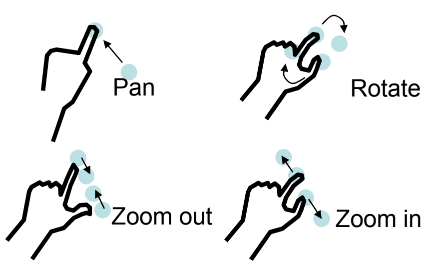
Application Potentials: Fall Detection

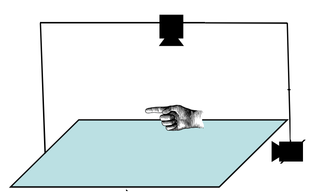
- Detect accidents at home



Application Potentials: Multi-Touch Surface

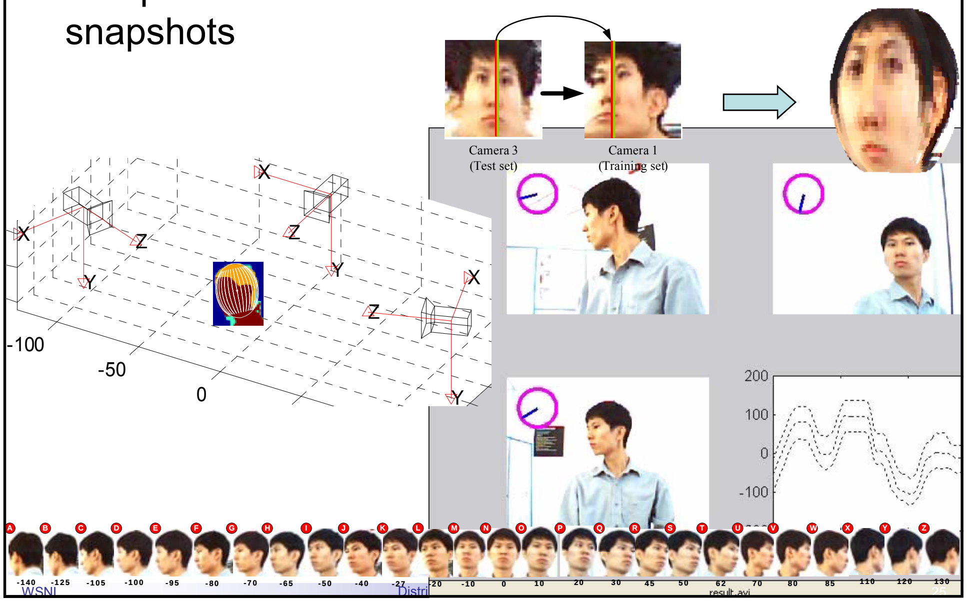
- Manipulate virtual world with free hand gesture



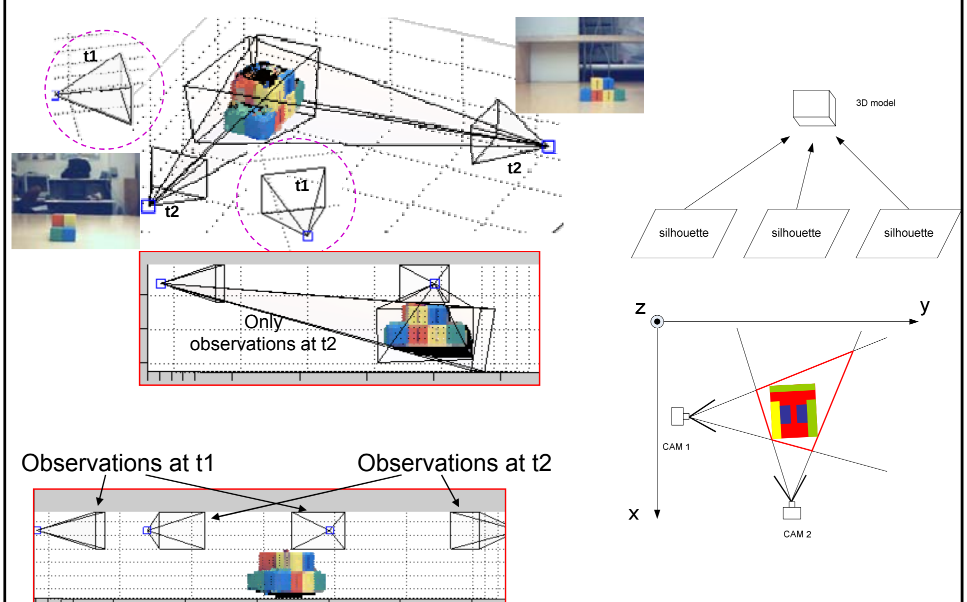


Application Potentials: Face Profiling

- Interpolate and reconstruct face model from a few snapshots

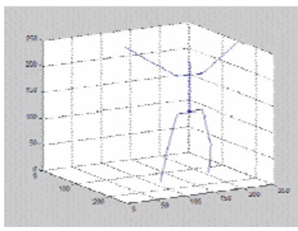


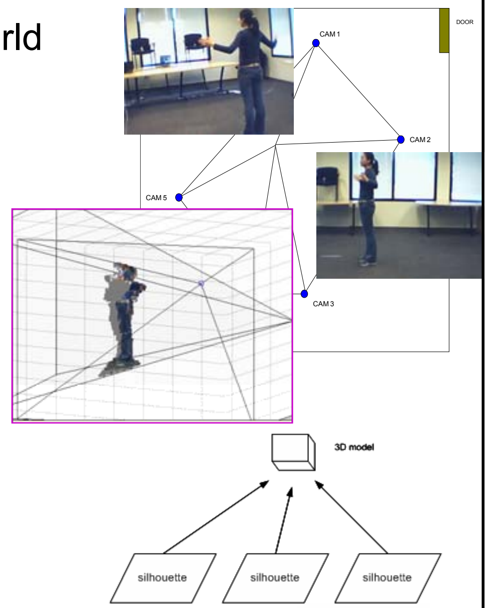
Application Potentials: 3D Model Reconstruction



Application Potentials: Virtual Reality

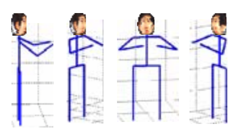
- ➢ Place people in virtual world



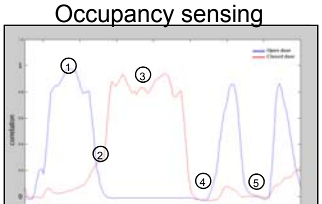


Applications

Gaming



Occupancy sensing

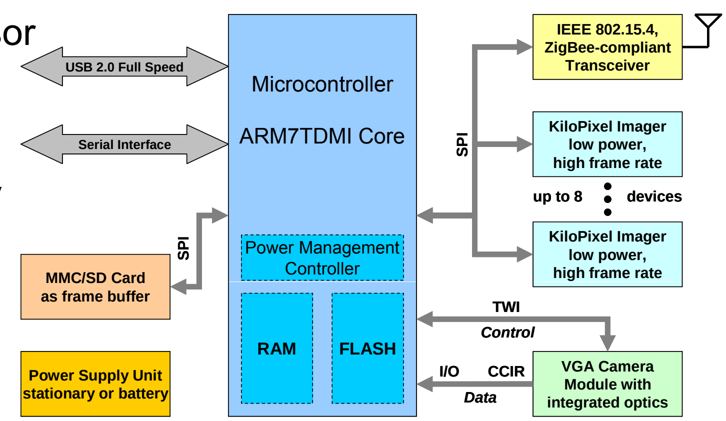


Assisted living

Image Sensor Node

Image Sensor Mote

- General architecture
  - Sensor
  - Processor
  - Radio
  - Power
  - Memory

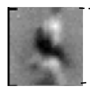


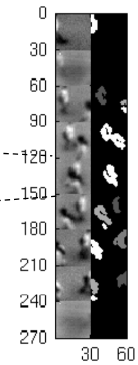
Stanford MeshEye Mote Architecture

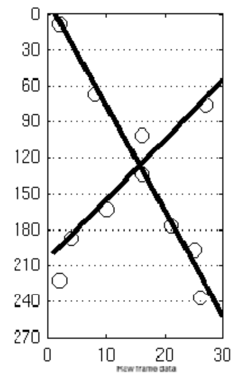
Reference:
• S. Hengstler, H. Aghajan, "A Smart Camera Mote Architecture for Distributed Intelligent Surveillance", Workshop on Distributed Smart Cameras, Oct. 2006

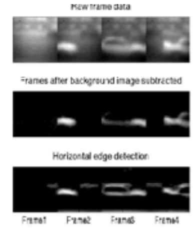
Low (kPix)-Resolution Sensor

- What can it be used for?
  - ➢ Limited information in single frame









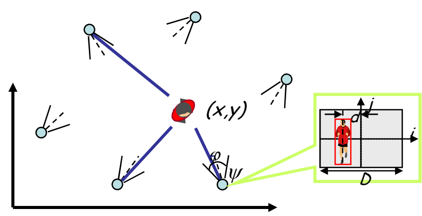
▪ Use kPix camera to:
  - ➢ Detect moving object
  - ➢ Trigger higher-resolution cameras at event

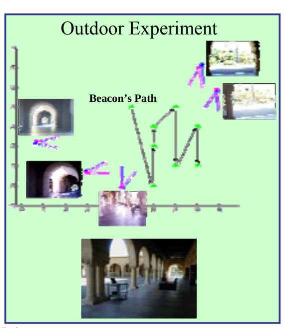
▪ With two kPix cameras:
  - ➢ Provide ROI focus for high-resolution camera acquisition and processing
  - ➢ Provide depth perception for the object

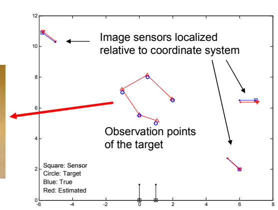
Reference:
• I. Downes, L. Baghaei-Rad, H. Aghajan,, "Development of a Mote for Wireless Image Sensor Networks", Cognitive Systems with Interactive Sensors, March 2006

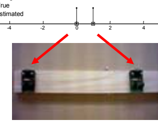
Mid (CIF)-Resolution Sensor

- What can it be used for?
  - ➢ Vision-based network localization
    - ➢ Beacon-assisted
    - ➢ Observations of moving target

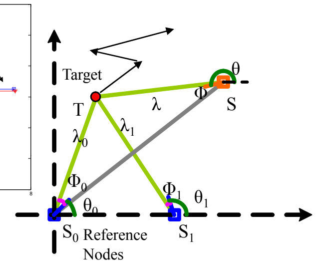








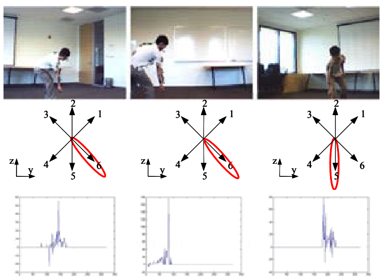
Nodes defining relative coordinate system



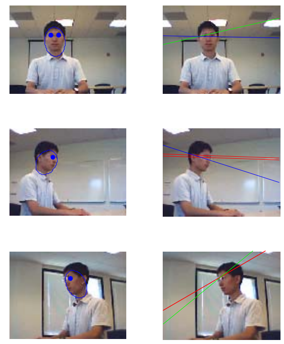
References:
• H. Lee, H. Aghajan, "Collaborative Self-Localization Techniques for Wireless Image Sensor Networks", Asilomar Conference on Signals, Systems and Computers, Oct. 2005
• H. Lee, L. Savidge, H. Aghajan, "Subspace Techniques for Vision-Based Node Localization in Wireless Sensor Networks", ICASSP, May 2006
• H. Lee, H. Aghajan, "Collaborative Node Localization in Surveillance Networks using Opportunistic Target Observations", ACM MM Workshop On Video Surveillance and Sensor Networks, Oct. 2006

“High” (VGA)-Resolution Sensor

- What can it be used for?
  - Event interpretation
  - Human gesture analysis

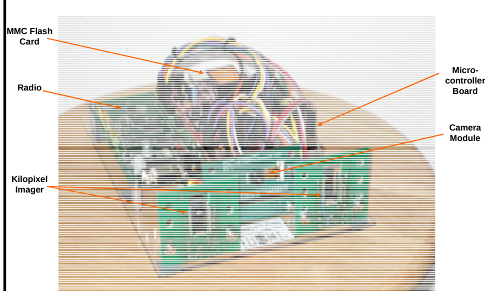


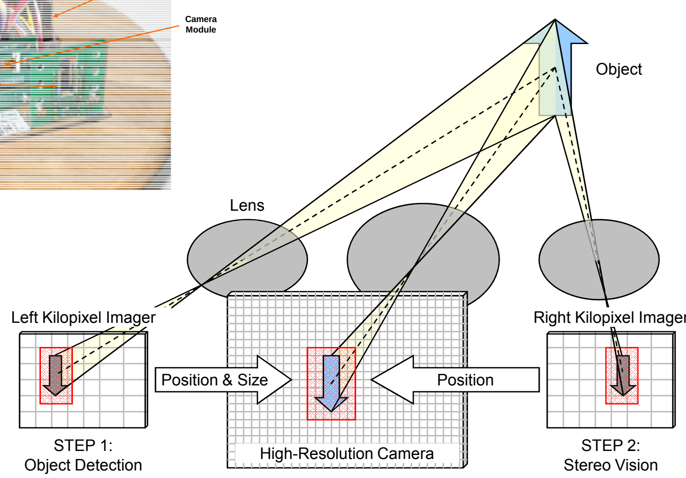
Vertical motion, asymmetric, picking up object



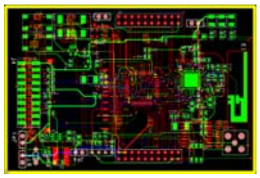
Face feature analysis

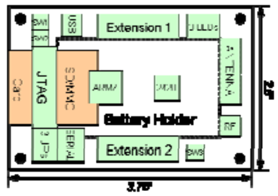
Hybrid-Resolution Vision System



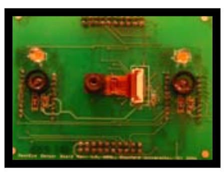


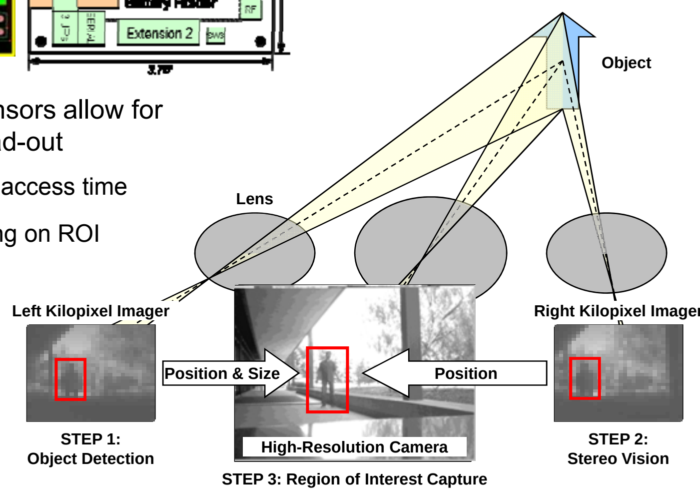
Hybrid-Resolution Vision System



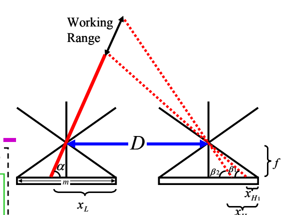


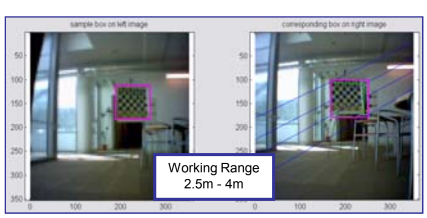
- Modern image sensors allow for ROI extraction at read-out
  - Savings in data access time
  - Vision processing on ROI

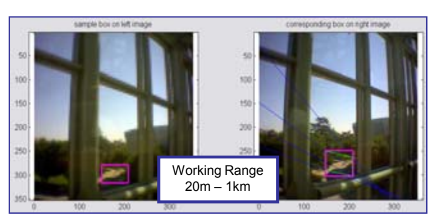




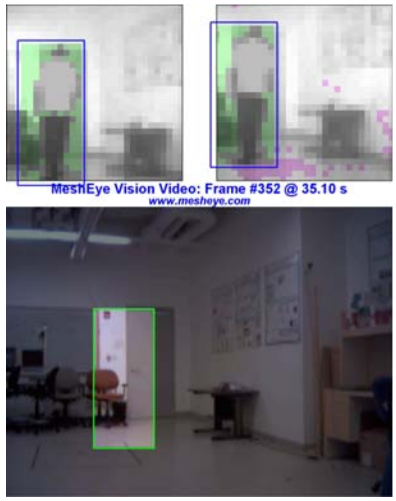
ROI Mapping



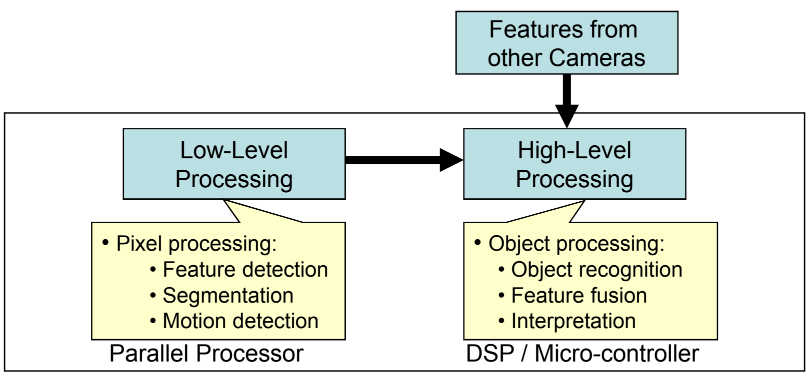


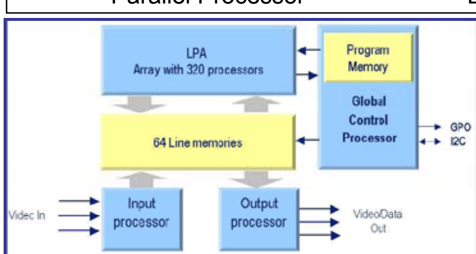


ROI Mapping



Processing

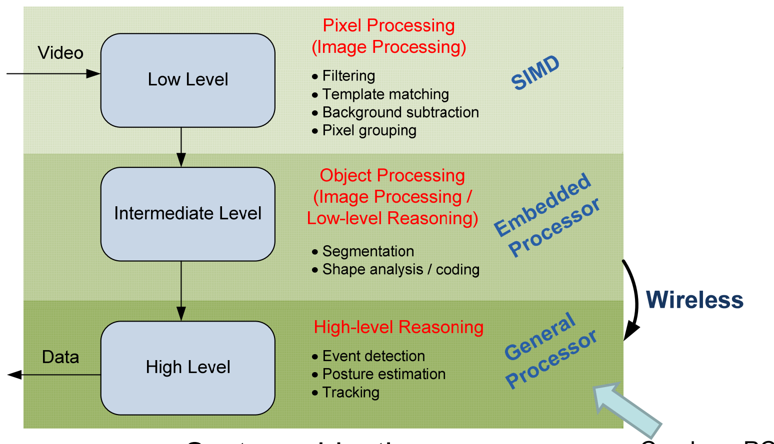




Reference:

• R. Kleihorst, B. Schueler, A. Danilin, M. Heijligers, "Smart Camera Mote with High Performance Vision System", Workshop on Distributed Smart Cameras, Oct. 2006

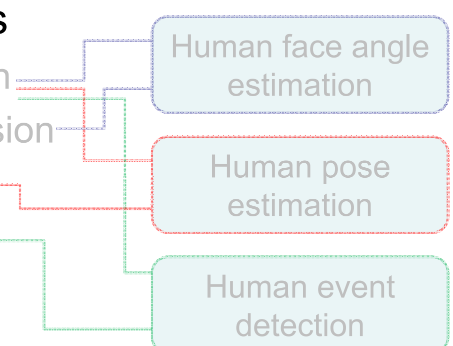
Strategy – Distributed Computation



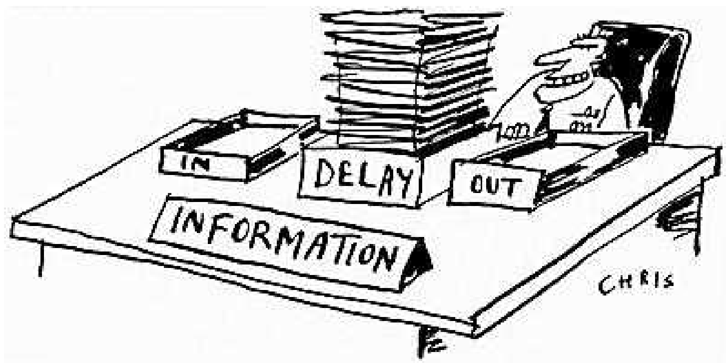
System objectives:
- Wireless transmission
- Real-time computation

Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

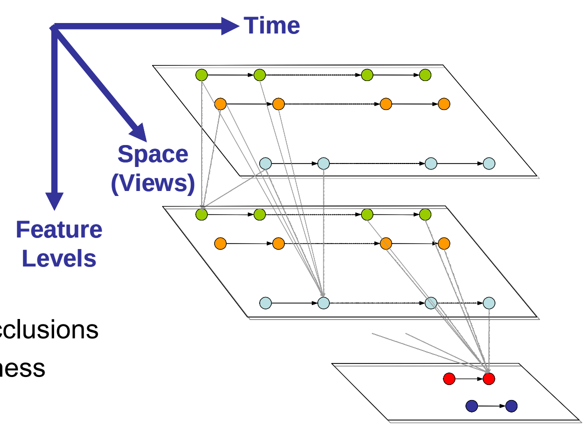


Data Fusion Mechanisms



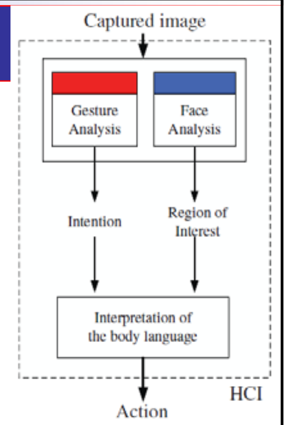
Information In … Delay … and Never Out

Fusion Dimensions



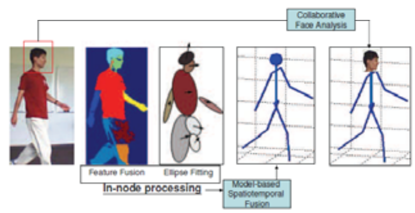
- ❑ Space (views)
  - Overcome ambiguities, occlusions
  - Enhance estimate robustness
- ❑ Time
  - Increase confidence level of estimates
  - Detection of key frames
- ❑ Feature levels
  - Exchange of features with other nodes across algorithmic layers

Objectives



Vision Algorithm
- Face orientation: region of interest
- Body pose : command to the system
- Multiple cameras
- Distributed computation → Spatial distribution, Function modules
- Moderate complexity
- Bandwidth / latency issues

System Requirements
- Real-time
- Wireless links

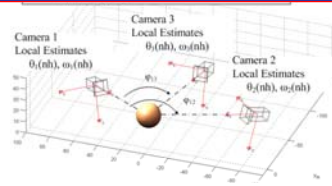


"Walking toward the piano!"

Basic Approach for a Camera Network

- Locally in a single camera:
  - Reduce images to descriptors
- Collaboratively between cameras:
  - Correlation: Mitigate errors (image noise, feature noise)
  - Orthogonality: Multi-view (occlusion, ambiguity, difficult views->easier views)
    - *Synergies*
      - *Image features*
      - *Temporal correction / prediction*
      - *Spatially distributed observation from cameras*

Example: Face Angle Estimation



- Approach:
  - Local estimation
  - Joint refinement / validation
- Fusion of information from 3 dimensions
  - In-node image features (feature fusion)
  - Temporal dynamics (temporal fusion)
  - Spatial consistency (spatial – spatiotemporal fusion)
- Objective:

  Improve robustness & reduce algorithm complexity!

Fusion Mechanisms

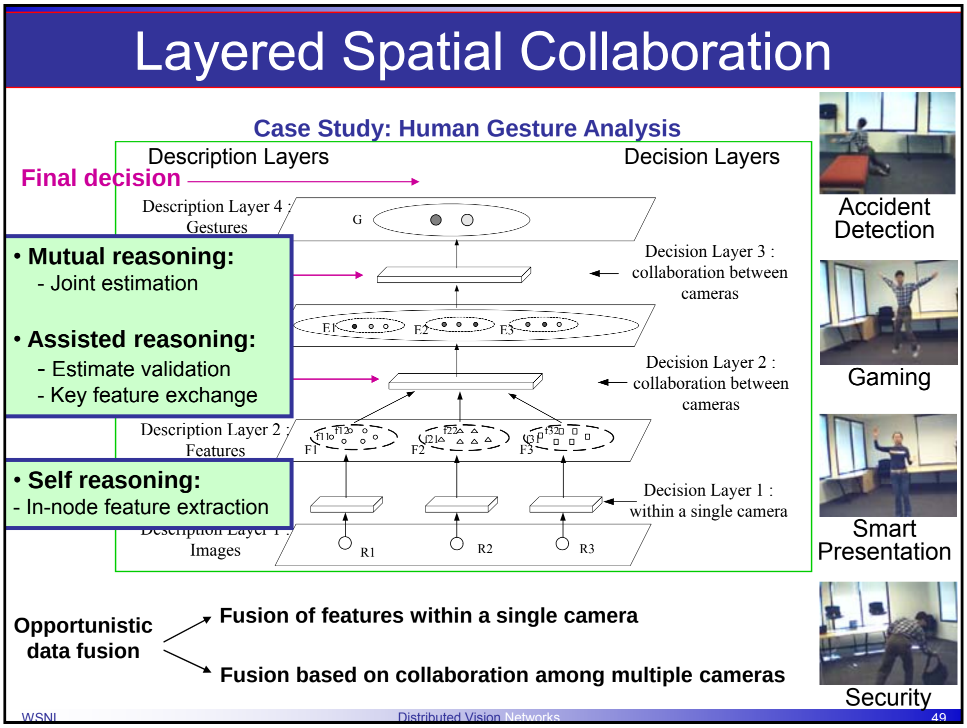
- Types of data fusion:
  - Feature fusion
  - Spatial fusion
  - Temporal fusion
  - Model-based fusion
  - Decision fusion

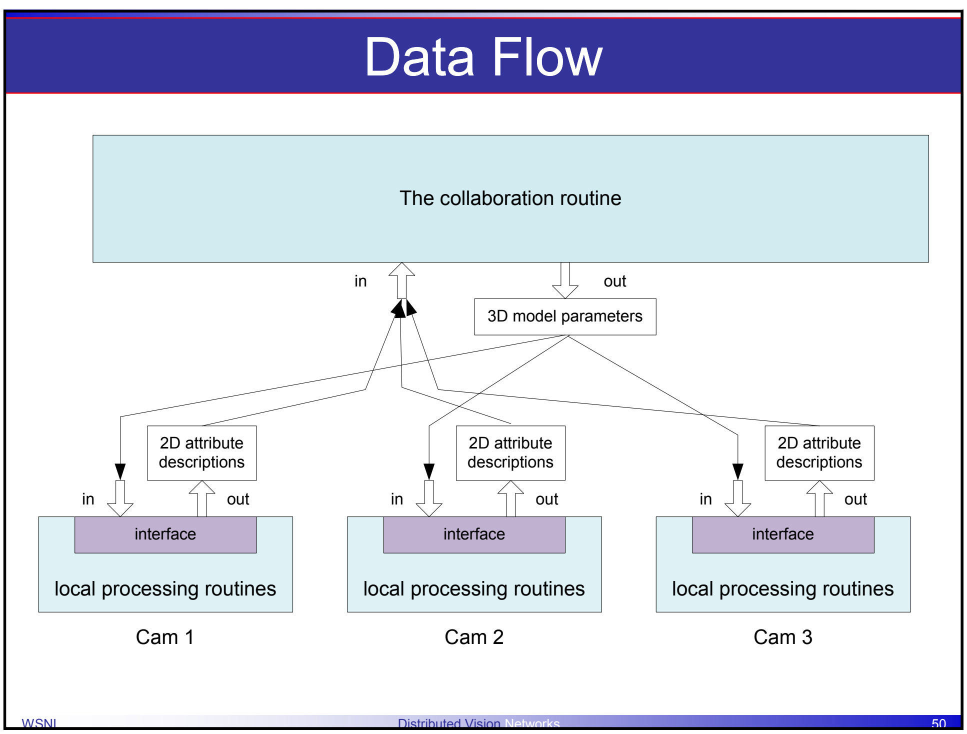
Fusion Mechanisms

- **Feature fusion:**
  - Use of multiple, complementary features within a camera node
- **Spatial fusion:**
  - Localization, epipolar geometry, ROI and feature matching
  - Validation of estimates by checking consistency, outlier removal
  - 3D reconstruction
- **Temporal fusion:**
  - Local interpolation / smoothing of estimates
  - Exchange of updates via spatial fusion
  - Spatiotemporal estimate smoothing and prediction
- **Model-based fusion:**
  - 3D human body reconstruction, human gesture analysis
  - Feedback to in-node feature extraction
- **Key features and key frames:**
  - Information assisting other nodes
- **Decision fusion:**
  - Estimates based on soft decisions
  - Adequate features in own observations
  - Cost, latency of communication

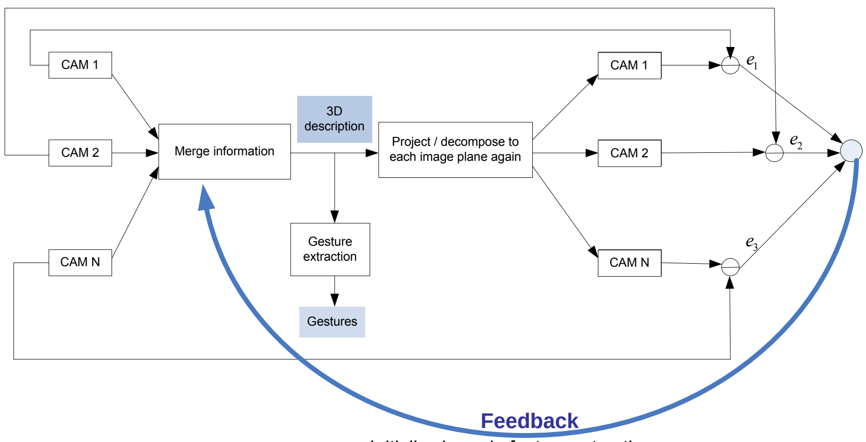
Collaboration Concepts

- Joint estimation
  - Combine measurements obtained by different cameras
- Probabilistic models
  - Associate confidence levels with interpretations
- Collaborative validation
  - Verify results obtained by one camera through further observations by other cameras
- Key frames and key features
  - Observations that help other cameras do better interpretation


Layered Spatial Collaboration
Case Study: Human Gesture Analysis
Description Layers
Decision Layers
Final decision
Description Layer 4 : Gestures
G
• Mutual reasoning:
- Joint estimation
• Assisted reasoning:
- Estimate validation
- Key feature exchange
Decision Layer 3 : collaboration between cameras
E1
E2
E3
Decision Layer 2 : collaboration between cameras
Description Layer 2 : Features
F1
F2
F3
• Self reasoning:
- In-node feature extraction
Decision Layer 1 : within a single camera
Images
R1
R2
R3
Accident Detection
Gaming
Smart Presentation
Security
Opportunistic data fusion
Fusion of features within a single camera
Fusion based on collaboration among multiple cameras
WSNL
Distributed Vision Networks
49


Data Flow
The collaboration routine
in
out
3D model parameters
2D attribute descriptions
2D attribute descriptions
2D attribute descriptions
in
out
in
out
in
out
interface
local processing routines
Cam 1
interface
local processing routines
Cam 2
interface
local processing routines
Cam 3
WSNL
Distributed Vision Networks
50

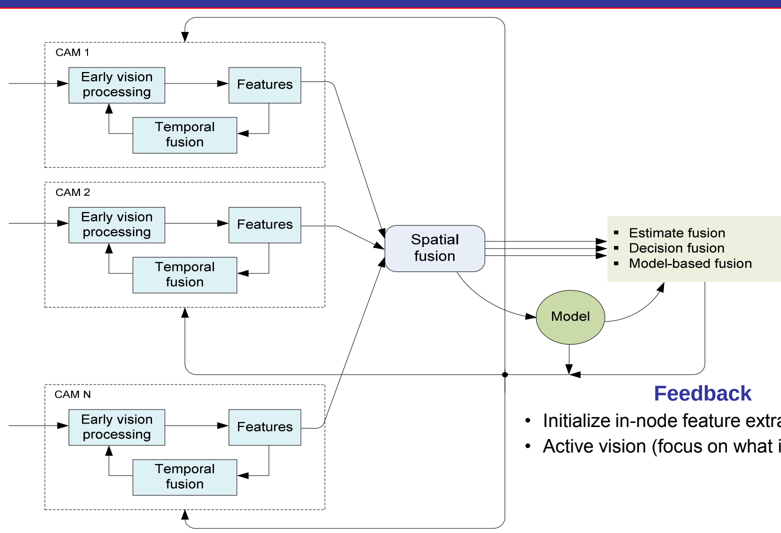
Use of Feedback



Feedback

- Initialize in-node feature extraction
- Active vision (focus on what is important)

Fusion Mechanisms



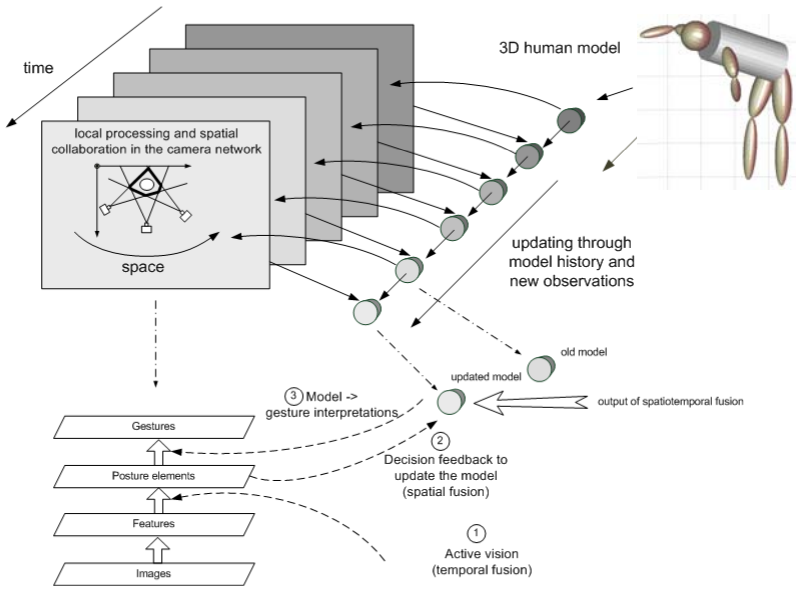
Feedback

- Initialize in-node feature extraction
- Active vision (focus on what is important)

- **Feature fusion**
- **Temporal fusion**

- **Spatial fusion**
- **Spatiotemporal fusion**
  - **Model-based**
  - **Active vision**
  - **Feedback**

The Big Picture



Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

Human pose estimation

Human event detection

Feature Fusion

Feature Fusion

- Extract multiple helpful features in each camera
- Opportunistic approach
  - Various features may be available at different times
- Objective:
  - To achieve robustness in node's description of event / object
- Allows for low-complexity implementation

Feature Fusion

- Generic features:
  - Color
  - Edges and contours
  - Shape geometry
  - Motion
  - Regions

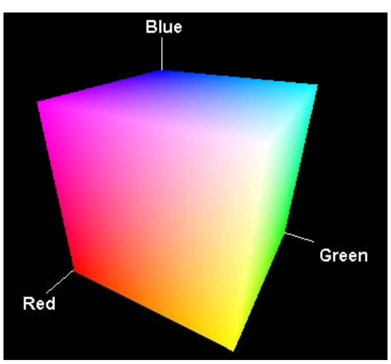
- Other features:
  - Optical flow
  - Invariant features
  - Active contours

▪ Generally useful in many vision applications
▪ Application-specific features may also be defined:
  • Ratios between length measures
  • Positioning of elements with respect to each other

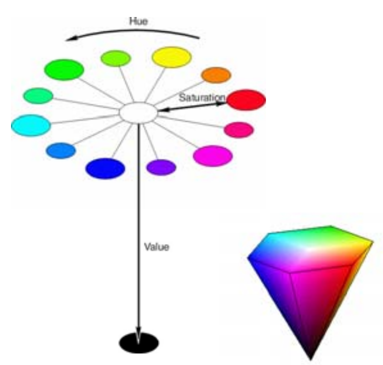
Summary of feature fusion

Color

- Various color spaces:
  - RGB
  - HSV (hue, saturation, value)
  - CIE Lab
    - L*:luminance; a*:red/blue; b*:yellow/blue
  - YC_bC_r



RGB (Red, Green, Blue)

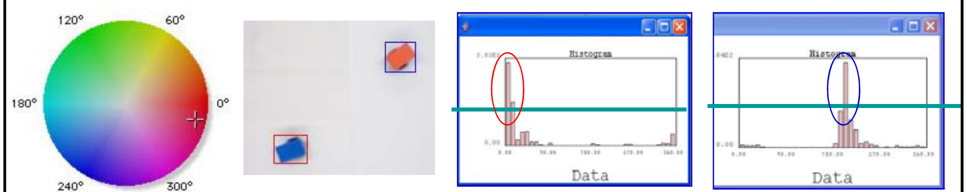


HSV (Hue, Saturation, Value)

Color

- Use of histograms:
  - Color or intensity distribution
  - Detect dominant color and use as label

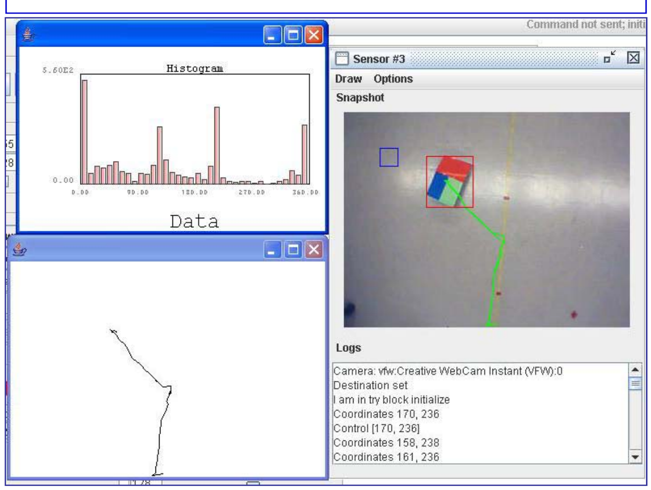
Object tracking based on hue histogram
➤ Peak of histogram used as dominant color attribute



Ref: E. Oto, F. Lau, H. Aghajan, "Color-Based Multiple Agent Tracking for Wireless Image Sensor Networks", ACIVS, Sept. 2006

Color

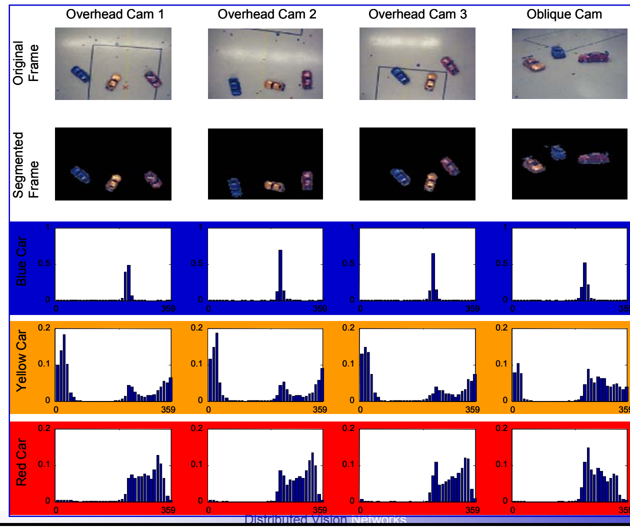
Object tracking based on hue histogram
➤ Histogram used as object's signature



Color

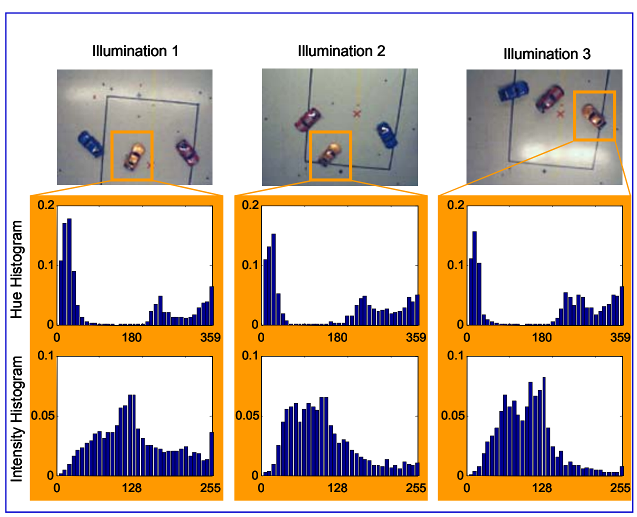
Tracking between camera views needs:

- Distinct signatures
- Color-calibrated cameras



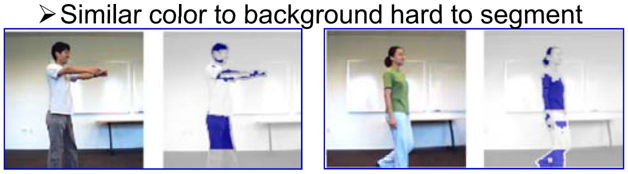
Color

Hue and intensity histograms



Color

- Problems:
  - Not robust for identifying human attributes, such as skin and hair
    - Variation between people
    - Variations in one person's attributes due to environmental factors:
      - Illumination changes
      - Shadowing
    - Variability with camera parameters
  - Similar color to background hard to segment



Color-based segmentation:

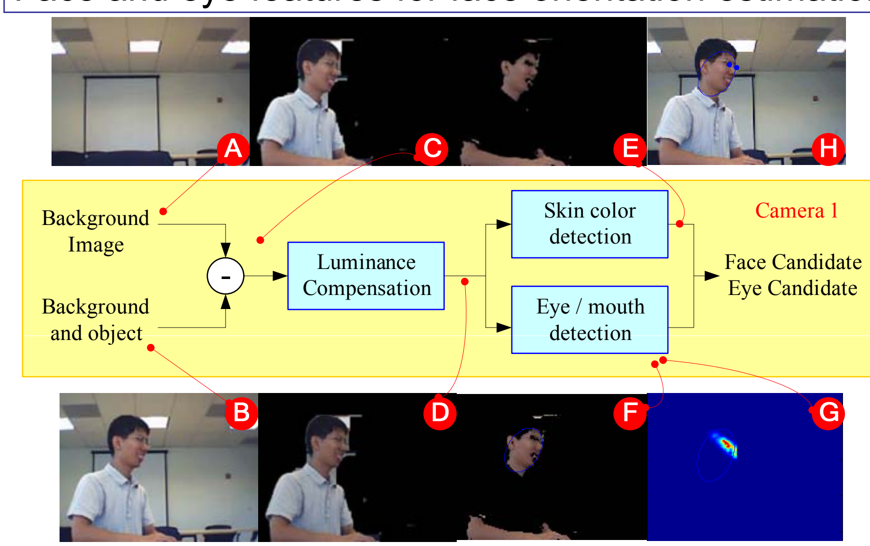
- How many shades of color?

Background subtraction:

- Simple thresholding won't work

Color

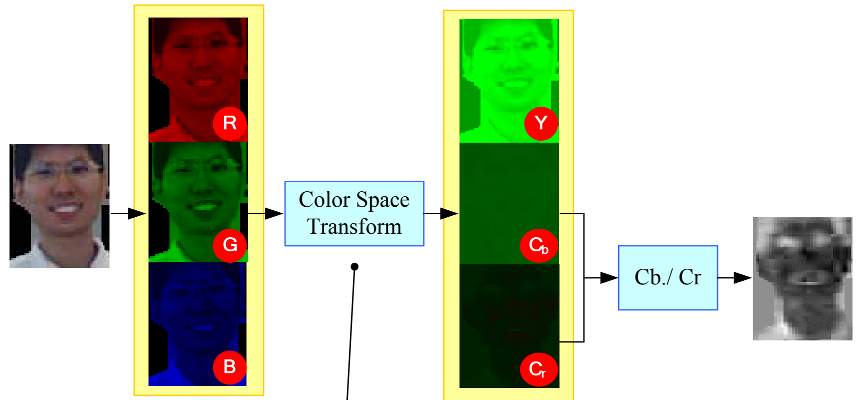
Face and eye features for face orientation estimation



Ref: C. Chang, H. Aghajan, "Collaborative Face Orientation Detection in Wireless Image Sensor Networks", SenSys - Distributed Smart Cameras, Oct. 2006

Color

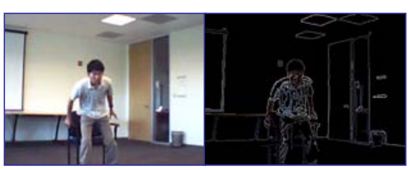
Eye Detection by Cb/Cr ratio



$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \frac{1}{255} \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Edge

- Why we use edges
  - Less susceptible to illumination changes than color or intensity level
  - Can provide shape information

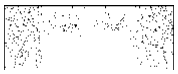


- Problems
  - Sensitivity to texture (e.g. in clothes), usually undesirable
  - Not detected when foreground / background have low contrast
  - Edge fragments require effort to be connected (hard without shape information)

Edge Detection

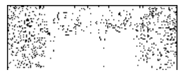
• Different edge detector kernels can be used:

roberts



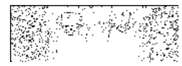
$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

sobel



$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

prewitt



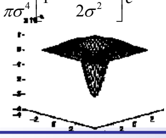
$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Zero crossing of Laplacian of Gaussian

Gaussian (smoothing) → Laplacian (derivative)

Linear operation

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
0	2	3	5	5	5	3	2	0
3	3	5	3	0	3	5	3	3
2	5	3	-12	-23	-11	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	[illegible]	-12	-23	-12	[illegible]	5	2
3	3	5	3	0	3	5	3	3
0	2	3	5	5	[illegible]	3	2	0
0	0	3	2	2	2	3	0	0

canny

Canny Edge Detector

- Widely used as standard edge detection scheme
- Goals:
  - Find true edges: maximize signal-to-noise ratio + true positive detects
  - Good localization: minimize distance between marked edge and real edge
    - Position edge at maximum derivative level
  - Clear response: limit number of detects for a single edge to 1
    - i.e. one response for every real edge

➢ Achieved through smoothing and enhancement of local maxima

Canny Edge Detector

Smoothing → Differencing → Non-maximum suppression → Thresholding / Linking

- Procedure:

$$Gauss(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

1. Smoothing
   - 2D Gaussian smoothing via two 1D Gaussian smoothing filters (separable filter)
2. Differencing
   - Sobel operators (horizontal & vertical)

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

3. Non-maximum suppression (only keep local max)
   - Suppress non-maximum points perpendicular to edge direction
   - Maintain edge strength at local maxima
4. Thresholding and connection
   - upper threshold t1, lower threshold t2
   - Immediate accept if gradient > t1, immediately reject if gradient < t2
   - If t2 < gradient < t1, accept if it can be connected to a strong edge pixel

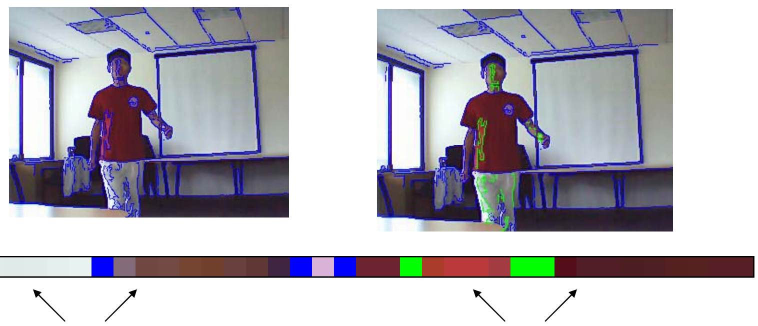
Fusion of Color and Edge Information

- Complementary attributes:
  – Color – <u>region</u> attributes
  – Edge – <u>contour</u> attributes
- Usage issues (example in face/head detection):
  – <u>Color</u>: Difficulty in detection may be caused by shadows or bad illumination
  – <u>Edge</u>: Active contours detect shape from edges, but may fit to outliers



Fusion of Color and Edge Information

Edge detection (small edge pieces) → Look at color both sides → Define inside / outside regions → Classify edges to on border / inside



Different hues

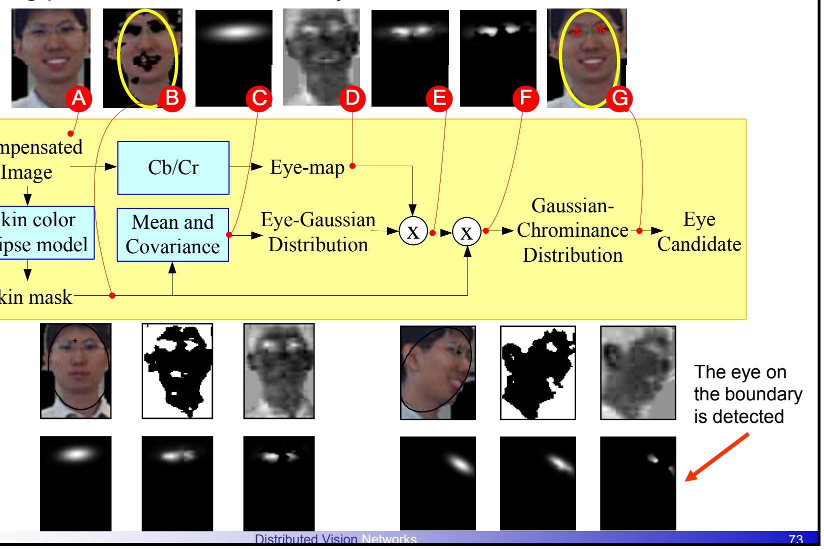
Similar hues

Fusion of Color and Edge Information

- Pixel-based methods
  - Information from immediate neighbors used
- One way to incorporate fusion on pixel level:
  - Define vector of features for a pixel with edge strength, color, etc.
    - Use feature vector to make correspondence between multiple camera images
    - Can also use to generate energy field for active contours
- How to bring in other context information?
  - Shape geometry (positional constraints)

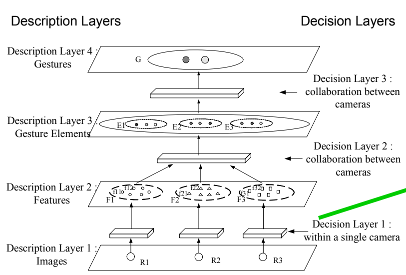
Fusion of Color and Shape Geometry

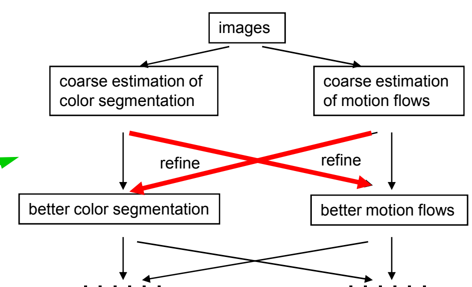
- Eye detection application
  - Adding position constraints for eyes:



The eye on the boundary is detected

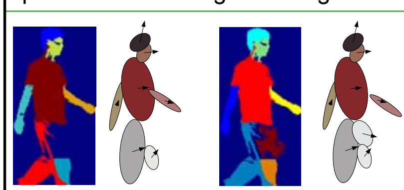
Joint Refinement of Color and Motion





Optical flow assisting color segmentation

Color segmentation assisting optical flow



Clustering close-by points with similar motion vector allows for better segmentation of the leg

Search for fitted ellipse in motion flow allows for effective detection of arm's motion vector

Region-based Fusion

- Problems with pixel-based features:
  - Localized attributes need local thresholds – hard to set
    - Comparing color of foreground / background pixels
  - No information from extended neighborhood considered
    - Knowledge about extent of neighborhood not available
      - That is the objective in many cases – segmentation
- Objects often contain correlated attributes in a region
  - Idea: Grow regions based on correlated attributes

Segmentation

- Motivation:
  - Foreground-background
  - Body parts
  - Face/hair
- Approaches:
  - Watershed
  - K-Means
  - Expectation Maximization (EM)

➢ Use of complementary features
  - Edge and color
  - Color and motion

➢ Combine pixel-based and region-based methods

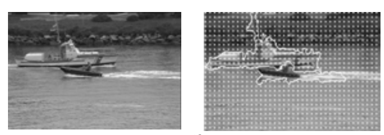
Summary of feature fusion

Segmentation

- Segment the image into *meaningful* groups
- What's meaningful?
  - Type of similarity that defines groups (attributes, neighborhood size)
- What to use?
  - Usually one feature is chosen (color, edge, motion, texture)
  - Interaction of different features
  - How to incorporate knowledge of object model
- Balance between image observations and target attributes

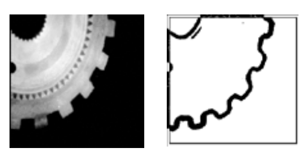
Segmentation

color



motion

texture

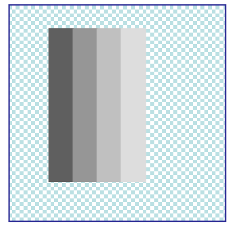


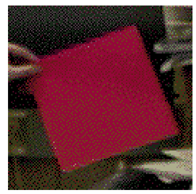
edge

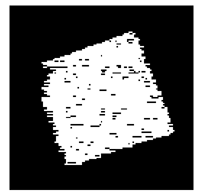
➢ Some heuristics on features
  - Helpful to use both region and edge information
  - Color is a useful cue, texture is better
  - Possible to detect texture boundaries instead of texture regions
  - Shadows and gradients (shades) are usually misleading
  - Different features may be complementary

Segmentation

- Method: <u>thresholding</u>
  - Typical procedure:
    - Choose an image criterion
    - Binarize image
    - Do clean-up operations
  - Methods for adaptive thresholds
    - Usually based on uniformity within region, not relationship between regions
    - Susceptible to local noise
  - Often used in background removal



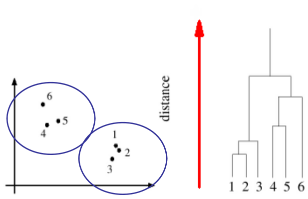


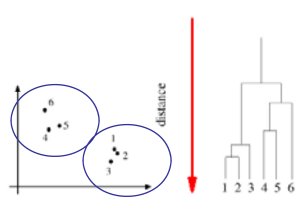


Segmentation

- Method: <u>region growing</u>
  - Take each point as a cluster
  - At each step:
    - Merge two clusters according to some metrics:
      - E.g. similar color

- Method: <u>region splitting</u>
  - Take the whole image as a cluster
  - At each step:
    - Split a cluster into two smaller ones according to some metrics:
      - E.g. average motion vector

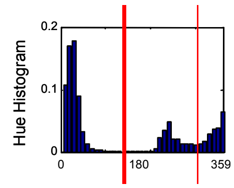




These may yield different results!

Segmentation

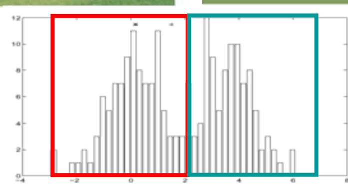
- Method: K-means
  - Divide all colors into K groups of color
  - Each color defines a region, may not be connected
  - Color histogram
  - Mode search is done iteratively, minimizing the ratio:
    - Intra-group variance / Inter-group variance



K=2 or 3?

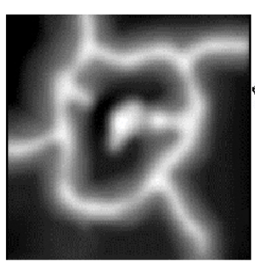
Relation in image across time may provide clues

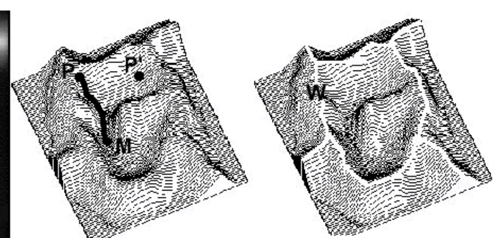
K=2



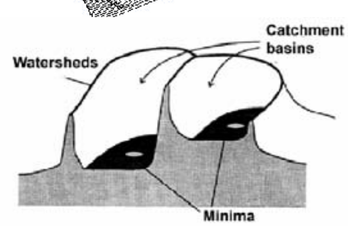
Watershed Segmentation – Topology Analogy

- Image data interpreted as a topographic surface with gray levels as heights
- The idea is to move from single-pixel background removal to region-based background removal and segmentation

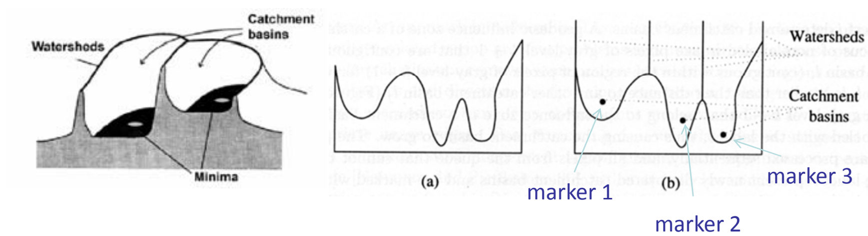




- Region edges correspond to watersheds
- Low-gradient region interiors correspond to catchment basins



Marker-based Watershed Segmentation



Markers: a set of pixels specified to be in the basins

- Imagine there are holes in marker pixels, and water comes out at the same velocity to immerse the topology
- Water first starts to fill the basins
- When two sources of water meet (from different markers), the two regions merge
- Highest walls maintain the boundary of regions

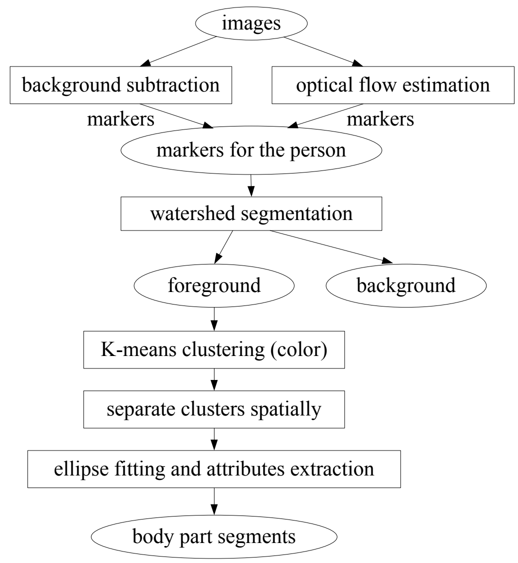
Feature Fusion for Segmentation

Watershed segmentation using K-means

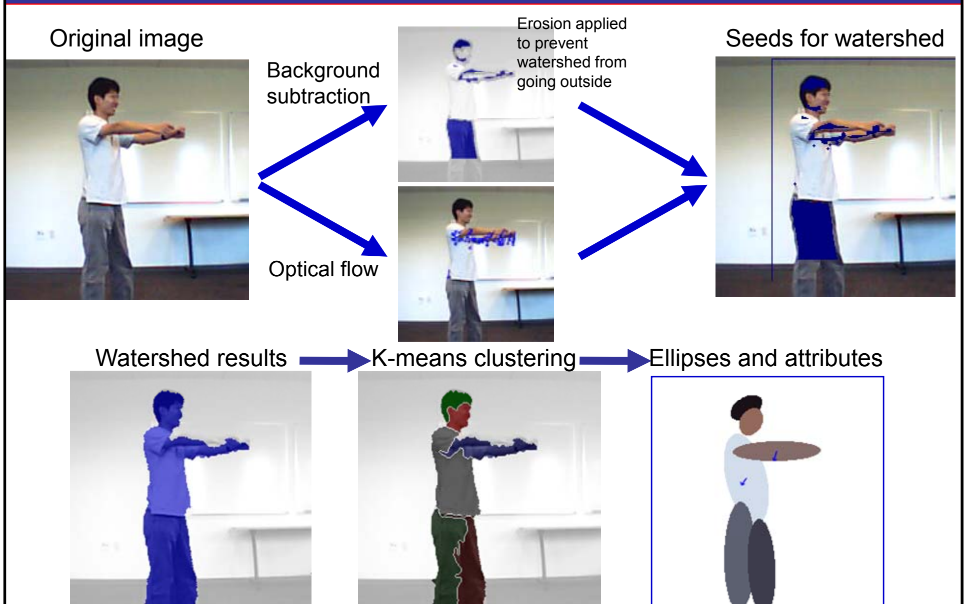
Original image → Background subtraction / Another foreground discriminator → Seeds for watershed → Watershed → K-means clustering → Segments



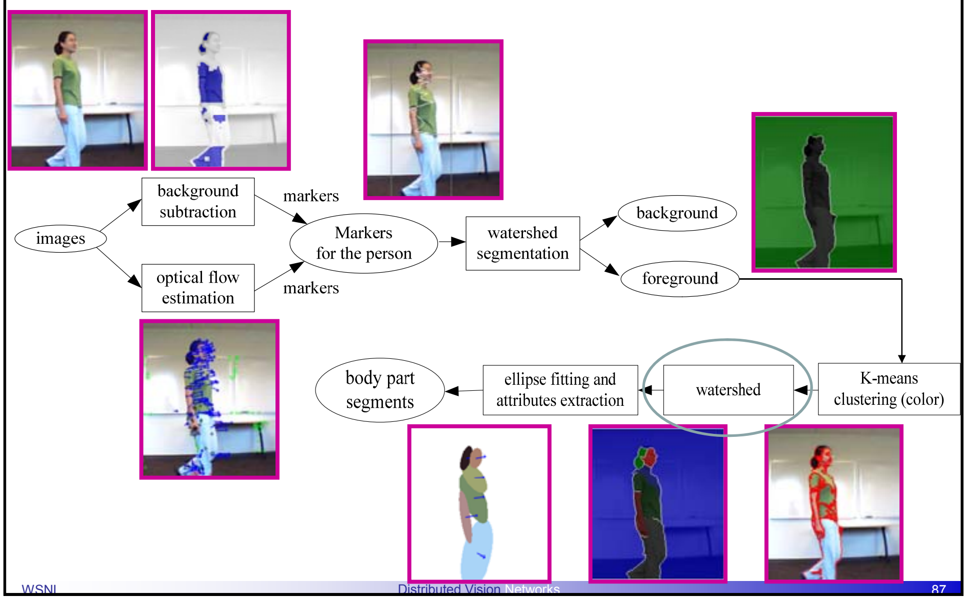

Fusion of Optical Flow and Color



Feature Fusion: Optical Flow and Color

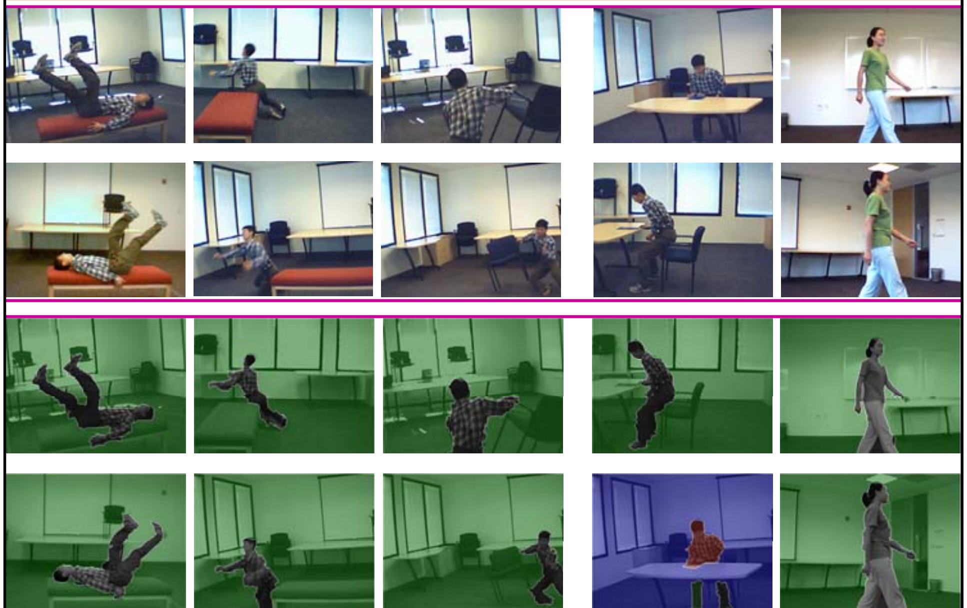


Feature Fusion: Optical Flow and Color



Feature Fusion: Optical Flow and Color

Watershed Background Removal

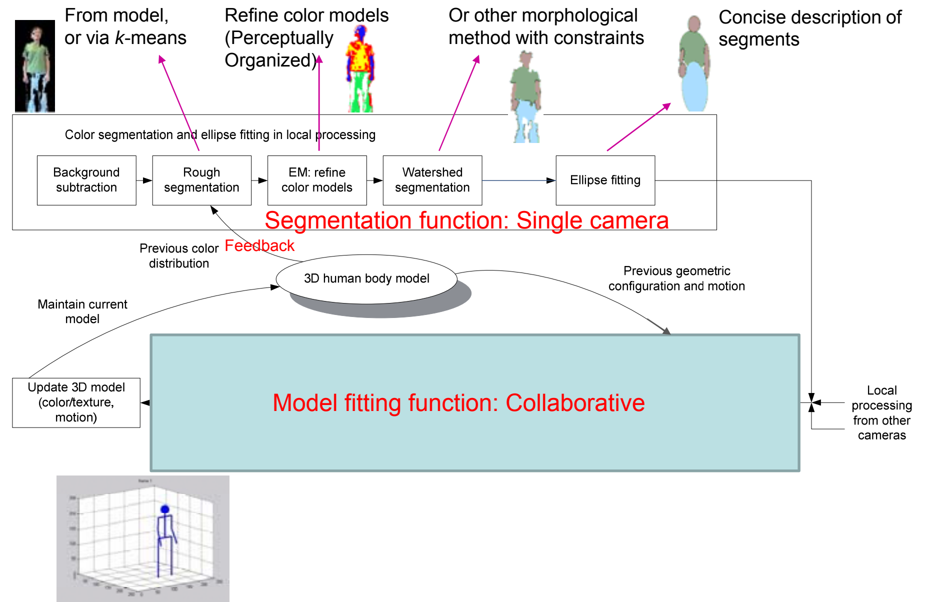


Segmentation

- Approaches:
  - Watershed
  - K-Means
  - Expectation Maximization (EM)
    - Number of segments unknown or varying in time

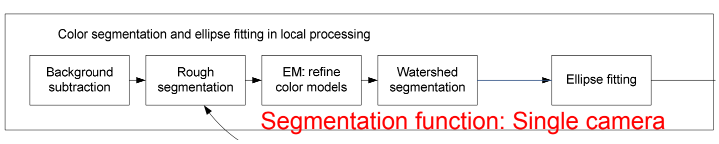
Summary of feature fusion

Human Pose Reconstruction



Segmentation

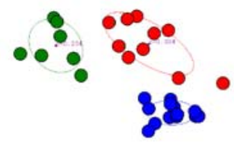
- In-node function based on:
  - Feature fusion
  - Feedback from model



- Feedback allows for incorporation of spatiotemporal fusion outcome into local analysis
- Rough estimate of segments provided by:
  - Local initialization
  - Adoption of spatiotemporal model
- Expectation Maximization (EM) methods use new observation to refine local color distributions
  - EM produces markers (collection of high-confidence segment islands) for watershed
  - Also helps with varying color distributions between cameras
- Watershed enforces spatial proximity information to link the segment

EM Segmentation

- Mixture model:
  - Each pixel is produced by a density associated with one of the N image segments
  - Segmentation is to find the generating segment for every pixel
- "Missing data ⟺ Hidden parameters" problem:
  - Missing data:
    - Need label y (to segment the image) : Which segment the pixel comes from $p(y_i = l \mid x_i)$



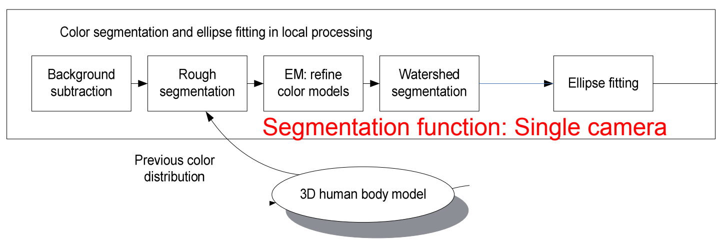
  - Hidden parameters:
    - Parameters of each segment $\Theta = \{\theta_1, \ldots, \theta_N\}$
    - Mixing weights $A = \{\alpha_1, \ldots, \alpha_N\}$ (the likelihood of each segment)

EM Segmentation

- The challenge:
  - Missing data >> hidden parameters
    - If we know the segment from which the pixel comes $p(y_i = l \mid x_i)$
    - Then it will be easy to determine its parameters $\Theta = \{\theta_1, \ldots, \theta_N\}$ and $A = \{\alpha_1, \ldots, \alpha_N\}$
  - Missing data << hidden parameters
    - If we know the segments $\Theta = \{\theta_1, \ldots, \theta_N\}$
    - We can determine $A = \{\alpha_1, \ldots, \alpha_N\}$ and $p(y_i = l \mid x_i)$
  - BUT, we know neither missing data nor hidden parameters
- Strategy:
  - Estimate missing data $p(y_i = l \mid x_i)$ from an estimate of hidden parameters Θ and A
  - Update Θ and A using current estimate of missing data $p(y_i = l \mid x_i)$
  - Iterate
  - ➢ Employ initialization to get close to a reasonable solution

EM Segmentation

- Initialization:
  - Not a good idea to arbitrarily specify an initial estimate
    - EM may be trapped to local optima
  - Ways to obtain initial estimates:
    - K-means
      - Centers of clusters are taken as the initial estimations for EM
    - Segment parameters from the 3D body model
      - Assumes appearance doesn't change very quickly



EM for Gaussian Mixture Models

- Gaussian mixture model (GMM)
  - Enforce a model on the data structure
  - Gaussian hidden parameters: $\theta_l = \{\mu_l, \Sigma_l\}$
  - Need to "label" x_i , i.e. determine $p(y_i = l \mid x_i)$

$$P_{\theta_l}(x_i) = \Pr(x_i \mid \theta_l) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_i-\mu_l)^T \Sigma_l^{-1} (x_i-\mu_l)}$$

- E step: compute "expected segment" for every data point

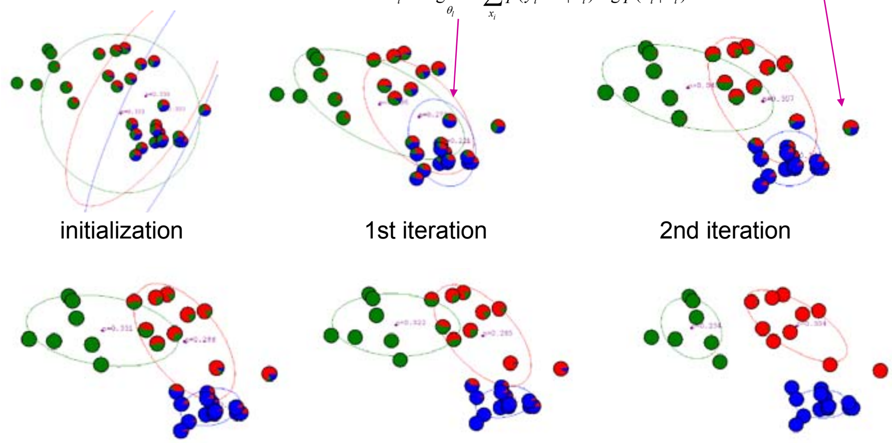
$$\left.\begin{aligned} & p^{(k+1)}(y_i = l \mid x_i) \propto \alpha_l^{(k)} P_{\theta_l^{(k)}}(x_i),\ l = 1,\ldots N \\ & \sum_{l=1}^{N} p^{(k+1)}(y_i = l \mid x_i) = 1 \end{aligned}\right\} \Rightarrow p^{(k+1)}(y_i = l \mid x_i)$$

- M step: maximize the log-likelihood $L(x;\Theta) = \sum_{i=1}^{M}\left(\sum_{l=1}^{N} p(y_i = l \mid x_i) \log p(x_i \mid \theta_l)\right)$

EM for Gaussian Mixture Models

- E step: compute "expected segment" for every data point $\sum_{l=1}^{N} p^{(k+1)}(y_i = l \mid x_i) = 1$
- M step: maximize the log-likelihood

$$\theta_l = \arg\max_{\theta_l} \sum_{x_i} p(y_i = l \mid x_i) \log p(x_i \mid \theta_l)$$



initialization | 1st iteration | 2nd iteration

3rd iteration | 4th iteration | 20th iteration

Perceptually Organized EM (POEM)

- Regular EM method:
  - A pixel-based method
    - Doesn't use spatial relationship between pixels / segment islands
  - May also leave some pixels unclassified

- POEM:
  - Segments are continuous, so consider a pixel's neighborhood
  - Use a measure of expected grouping: $w(x_i, x_j) = e^{-\frac{\|x_i - x_j\|}{\sigma_1^2} - \frac{\|coord(x_i) - coord(x_j)\|}{\sigma_2^2}}$
  - The neighborhood votes for (x_i in segment l):

$$V_l(x_i) = \sum_{x_j} \alpha_l(x_j) w(x_i, x_j), \quad \text{where } \alpha_l(x_j) = p(y_j = l \mid x_j)$$

Perceptually Organized EM (POEM)

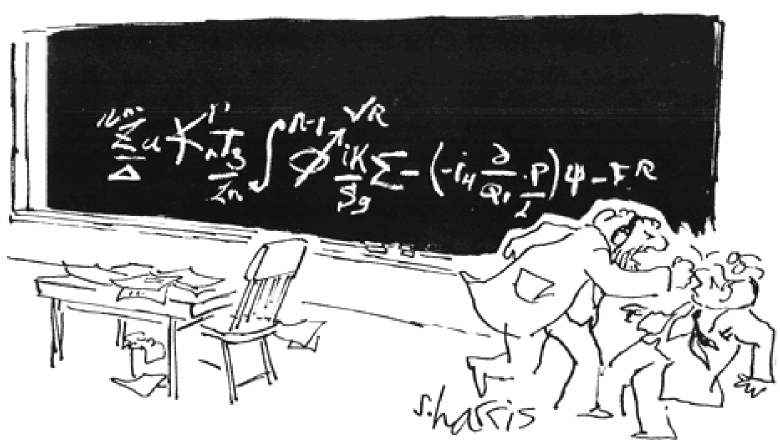
- Key difference with EM:
  - In EM mixing weights α_l are the same for every pixel x_i
  - In POEM mixing weights differ from pixel to pixel, and are influenced by pixel's neighbors
- E step: compute "expected segment" for every data point

$$\alpha_l^{(k)}(x_i)$$

$$\left.\begin{aligned} p^{(k+1)}(y_i = l \mid x_i) &\propto \alpha_l^{(k)} P_{\theta_l^{(k)}}(x_i),\ l = 1, \ldots N \\ \sum_{l=1}^{N} p^{(k+1)}(y_i = l \mid x_i) &= 1 \end{aligned}\right\} \Rightarrow p^{(k+1)}(y_i = l \mid x_i)$$

$$\alpha_l^{(k)}(x_i) = \frac{e^{\eta \cdot V_l(x_i)}}{\sum_{l=1}^{N} e^{\eta \cdot V_l(x_i)}}$$

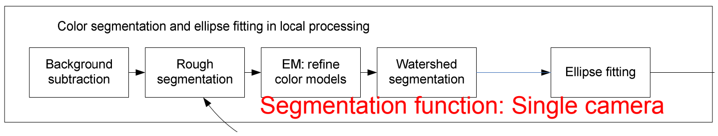
η controls "softness" of the voting combination



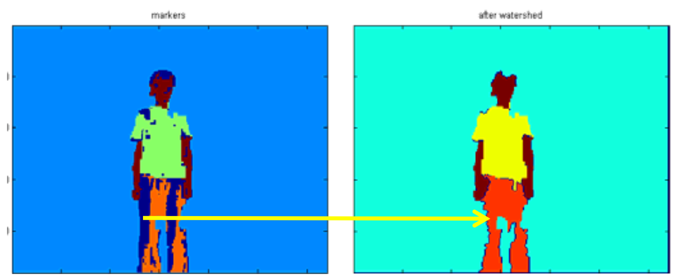
You want proof? I'll give you proof!

Watershed Segmentation

- Removing “vague” pixels is important before watershed, since wrong seeds/markers would compete with correct ones and cause false segments



Red: undecided pixels

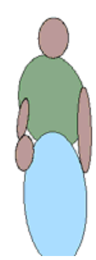


Assigns labels to undecided (dark blue) pixels

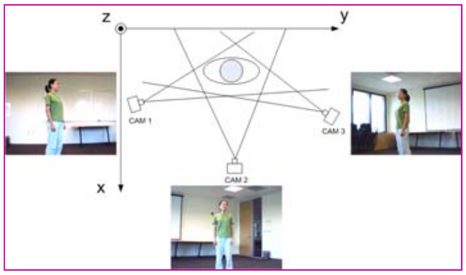
Ellipse Fitting

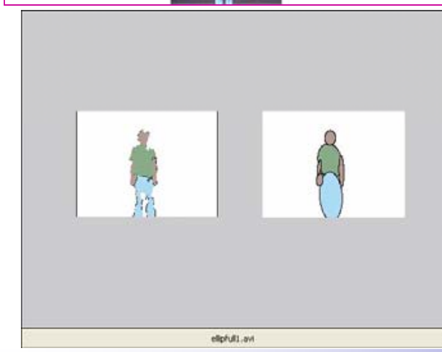
- Motivation:
  – Concise descriptions of segments
  – Each ellipse should represent a segment with similar shape
  – Not necessarily correspond to body parts

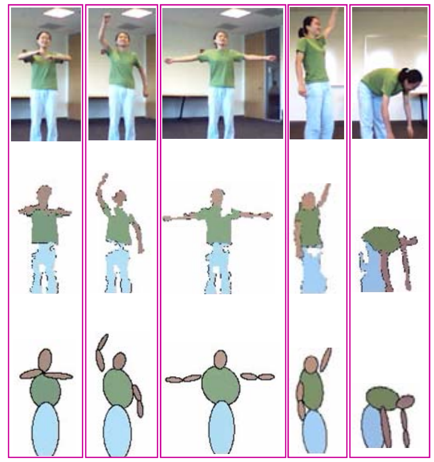
- Goodness of fit measures control ellipse fitting:
  1. Occupancy of the ellipse
  2. Coverage of the segment



In-Node Segmentation for Pose Estimation

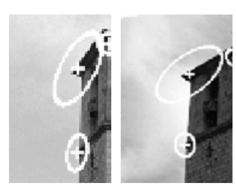


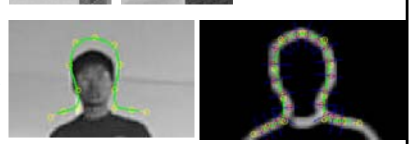




Feature Fusion

- Generic features:
  - Color
  - Edges and contours
  - Shape geometry
  - Motion
  - Regions
- Other features:
  - Optical flow
  - Invariant features
  - Active contours





Summary of feature fusion

Optical flow

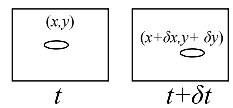
- Optical flow -- motion of brightness patterns

Optical flow

- Applications:
  - Global motion detection
    - Detection of a moving object
  - Segmentation based on motion
    - Segmentation of foreground from background
    - Segmentation of parts of object with different motion vectors
- Approaches:
  - Pixel-based
  - Feature-based
    - Edge points, corner points, other features

Optical flow

Brightness $I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$



$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t \sim 0$$

Optical flow constraint equation $I_x u + I_y v + I_t = 0$

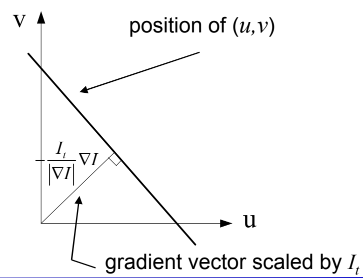
- (u,v): x and y components of optical flow
- (I_x, I_y, I_t): intensity derivatives

2D Motion Constraint Equation:

$$\nabla I^T \vec{u} = -I_t, \text{ where } \nabla I = \begin{pmatrix} I_x \\ I_y \end{pmatrix}, \vec{u} = \begin{pmatrix} u \\ v \end{pmatrix}$$

1 equation in 2 unknowns

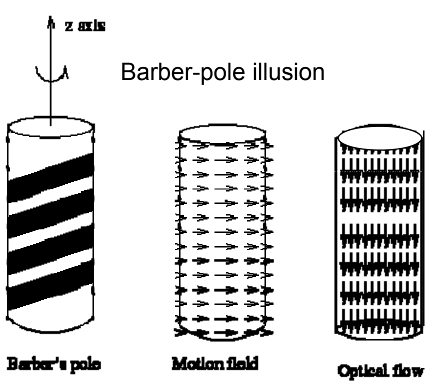
(u,v) is on the line orthogonal to image gradient, but we do not know its exact location (aperture problem)

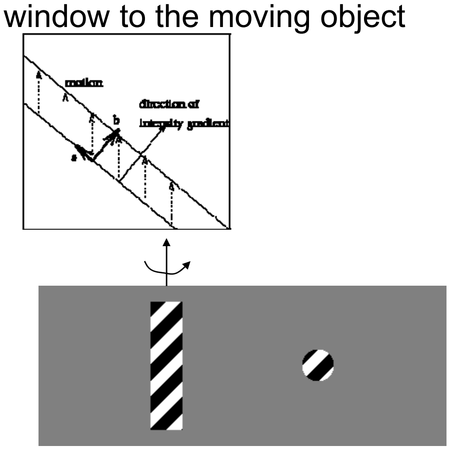


Optical flow

Aperture problem:

- Can only measure the component of optical flow along the direction of intensity gradient (normal to edge)
  - <u>Motion component along the edge cannot be detected</u>
- The reason is we look at small window to the moving object





Optical flow

- How to avoid the aperture problem?
  - Use more constraints for a pixel
  - Consider a 3x3 window

$$\begin{pmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{x9} & I_{y9} \end{pmatrix} u = -\begin{pmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{t9} \end{pmatrix}$$

$$Au = b \quad \text{minimize } \|Au - b\|$$

- Lucas-Kanade equation $A^T Au = A^T b$ Least-Squares Problem

$$u = (A^T A)^{-1} A^T b = \begin{pmatrix} \sum I_{xi} I_{xi} & \sum I_{xi} I_{yi} \\ \sum I_{xi} I_{yi} & \sum I_{yi} I_{yi} \end{pmatrix}^{-1} \begin{pmatrix} -\sum I_{xi} I_{ti} \\ -\sum I_{yi} I_{ti} \end{pmatrix}$$

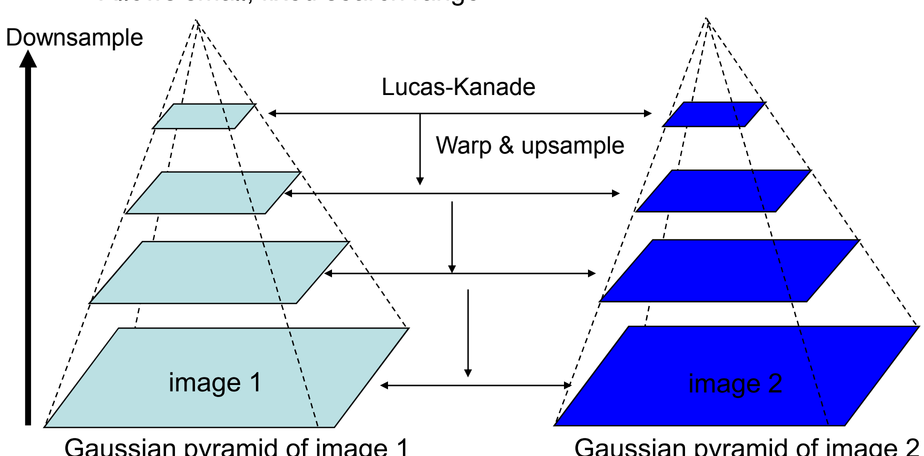
Solvable when $A^T A$ invertible $\longrightarrow$ no aperture problem

- If an edge exists, motion component along edge won't show up $\rightarrow$ $A^T A$ not full rank
- Increasing window size an option, but large window may include multiple motions

Two approaches: Pyramid searches, feature-based methods

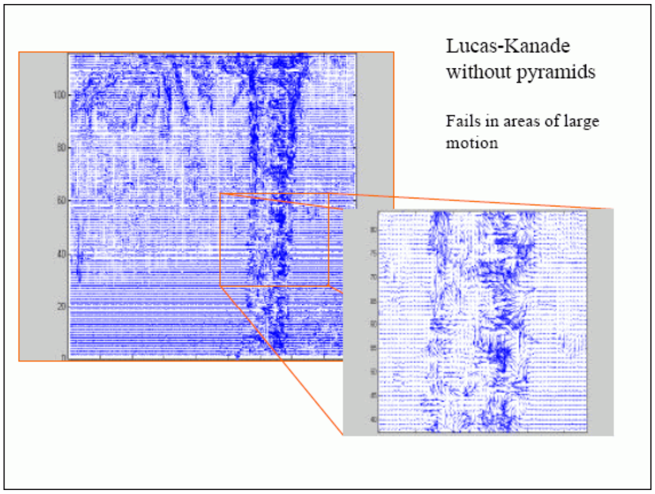
Optical flow

- Another source of problem: Large motion vector
  - Increases size of search window
- Multi-scale pyramid
  - Allows small, fixed search range



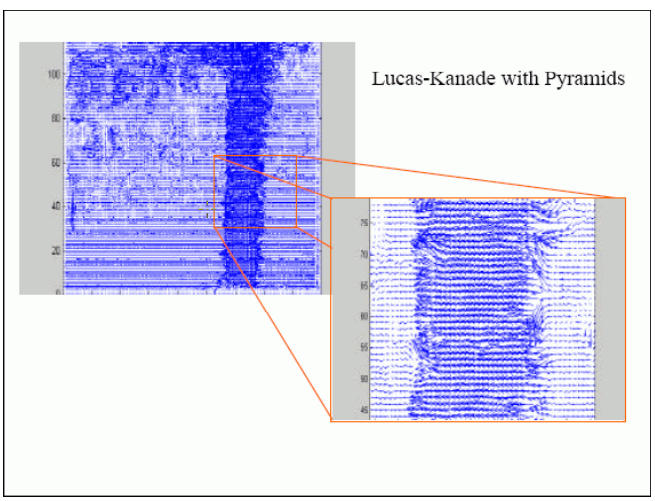
Optical flow

- Pyramid example



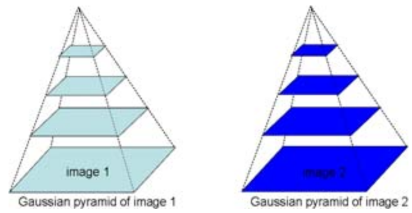
Optical flow

- Pyramid example



Optical flow

- Issues with pyramids:
  - Brightness constancy may not hold when down-sampling the 2 frames
    - E.g. with shadowing
  - Fails when neighboring pixels do not move in the same way
    - E.g. non-rigid motion of body parts
  - When motion is large, error in coarse scales will propagate to fine scales
    - E.g. fast motion in human gestures



- How to make the method selective to quality?
  - Pixels with no good matches can be excluded form motion field

Optical flow

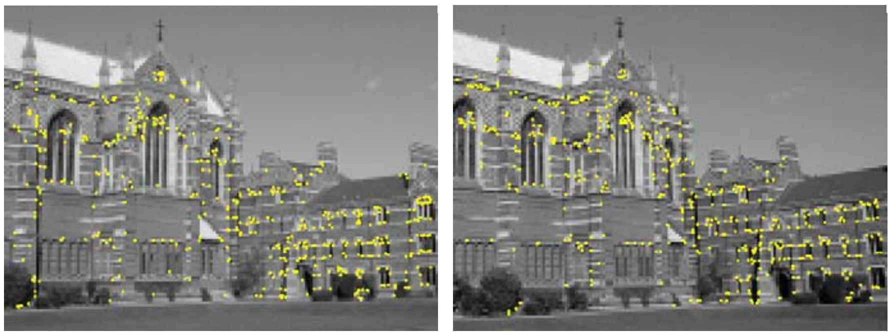
- Feature-based approaches
  - Find features in each image
  - Match between features
  - Find motion vectors

➢ Advantage
  - Reduce information to be processed
    - Only compute optical flow for feature points
  - Robust estimation for global relation between images
    - Called structure from motion
  - Higher level interpretation of contents in the images
    - Since they work with object features

➢ Requirements:
  - Features present and prominent in both images
  - Define descriptors of features for matching
  - Features have to be distinctive in descriptors (so the match can be found)
  - Need to assume certain motion model (affine, perspective) in matching

Optical flow

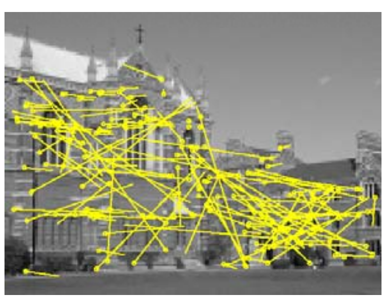
- Feature-based: corners



Detected features

Optical flow

- Cross-correlation matching



Initial matches

After global constraints

- Use behavior of majority to delete outliers

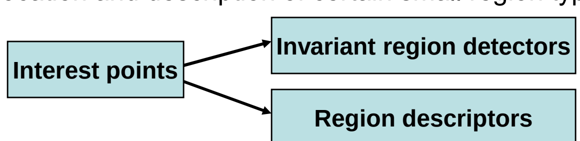
Feature Fusion

- Generic features:
  - Color
  - Edges and contours
  - Shape geometry
  - Motion
  - Regions
- Other features:
  - Optical flow
  - Invariant features
  - Active contours

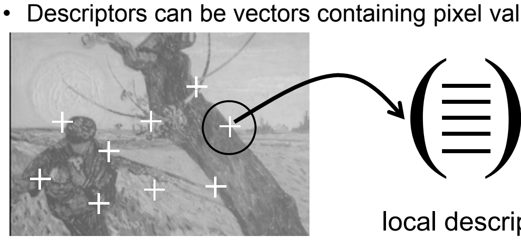
Summary of feature fusion

Local Invariant Features

- Based on location and description of certain small region types



- Harris corner detector
  - Corner: Significant derivative in both directions
  - A descriptor defined for the interest points
    - Descriptors can be vectors containing pixel values, gradients, etc.



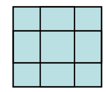
local descriptor

This is beyond vector of features for a single pixel, and uses region information (e.g. SIFT)

Local Invariant Features - Detector

- Harris corner detector
  - Auto-correlation matrix of intensity derivatives

$$\begin{bmatrix} \sum_{(x_k, y_k) \in W} (I_x(x_k, y_k))^2 & \sum_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) \\ \sum_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) & \sum_{(x_k, y_k) \in W} (I_y(x_k, y_k))^2 \end{bmatrix}$$



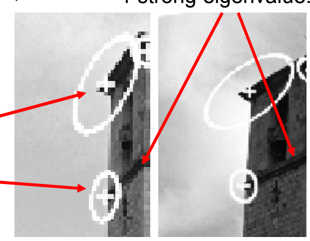
Captures the structure of the local neighborhood

- Measure defined based on eigenvalues of this matrix
  - 2 strong eigenvalues $\longrightarrow$ interest point (corner)
  - 1 strong eigenvalue $\longrightarrow$ contour (edge)
  - 0 eigenvalue $\longrightarrow$ uniform region

1 strong eigenvalue: edge

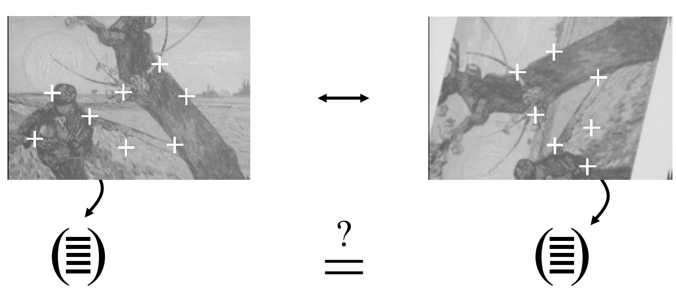
2 strong eigenvalues: ratio ~1 (strong corner)

2 strong eigenvalues: ratio >>1 (weak corner)



Local Invariant Features - Detector

- Harris corner detector
  - correspondence



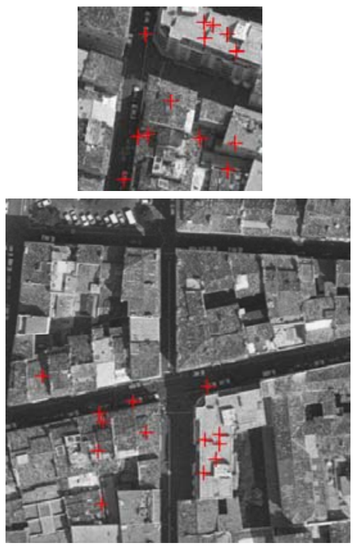
Vector comparison using some distance:

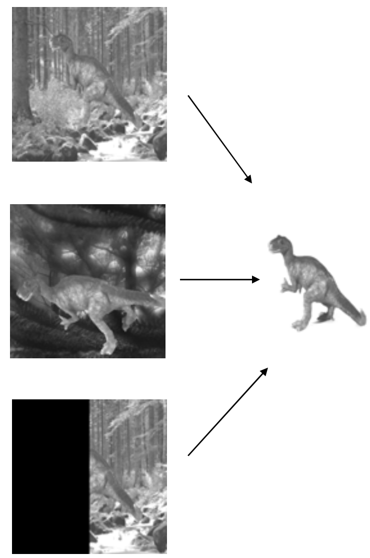
- The Mahalanobis distance

$$dist_M(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^T \Lambda^{-1} (\mathbf{p} - \mathbf{q})}$$

Local Invariant Features - Detector

- Harris corner detector



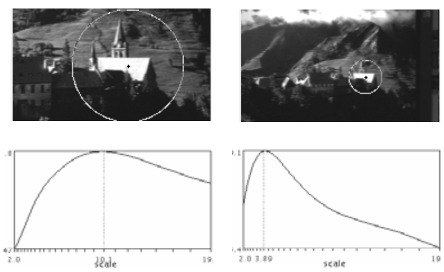


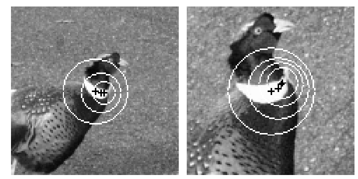
Local Invariant Features - Detector

- Harris corner detector
  - Strength:
    - Good detection in the presence of occlusion
      - Uses <u>many corners</u> of the object of interest
      - Based on <u>localized information</u>
      - <u>Invariant</u> to rotation and illumination change
  - Weakness:
    - Not invariant to scale and affine changes
  - Approach:
    - Extend from corners to interest points or regions
      - <u>Multi-scale</u> to provide scale invariance
      - For affine invariance:
        - ➢ Use direction of max. gradient as <u>reference</u>
        - ➢ <u>Normalize</u> the principal axes according to their characteristic scale
    - Develop good descriptors

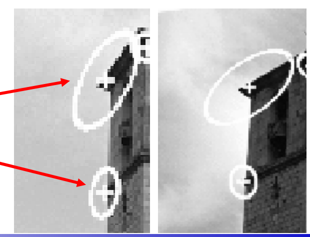
Local Invariant Features - Detector

- Extension: Multi-scale extraction of Harris interest points
  - Selection of points occurs at characteristic scale
    - E.g. the scale with max. gradient levels, or corner strengths



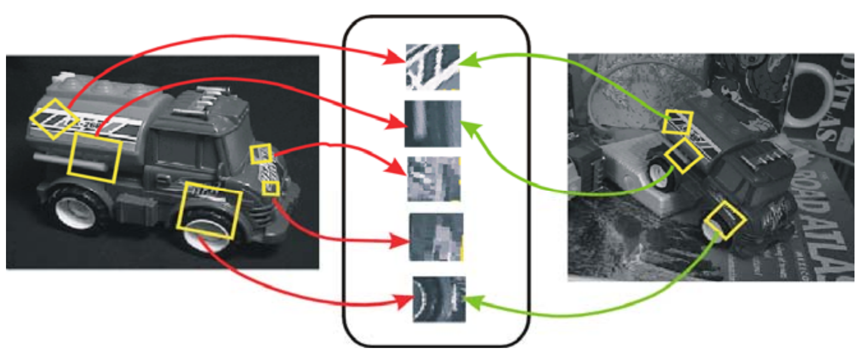


Best scale for each axis is used to size the ellipse



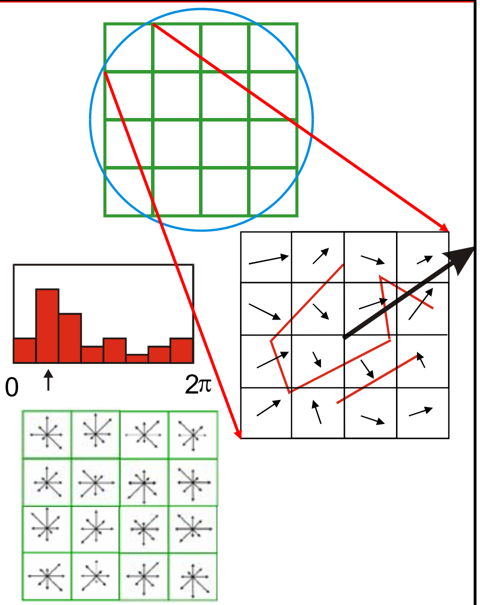
Local Invariant Features – Descriptors

- Descriptors – SIFT (Scale Invariant Feature Transform)
  - Image content is transformed into local features invariant to translation, rotation, scale

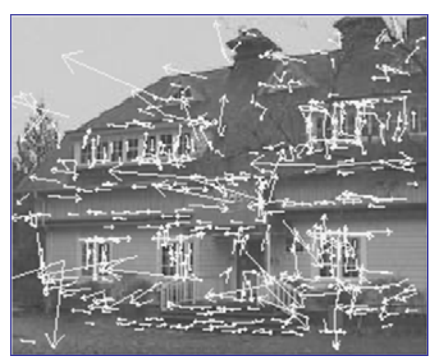


Local Invariant Features – Descriptors

- SIFT
  - In image at original scale:
    - Canonical orientation chosen for each feature
      - Computed at selected scale
    - Divide feature region into 4x4 blocks
  - Create histogram of local gradient directions:
    - For 4x4 windows within each block
    - 8 bins in histogram
  - Compose descriptor vector for feature:
    - Descriptor vector of 128 elements (8 x 16)



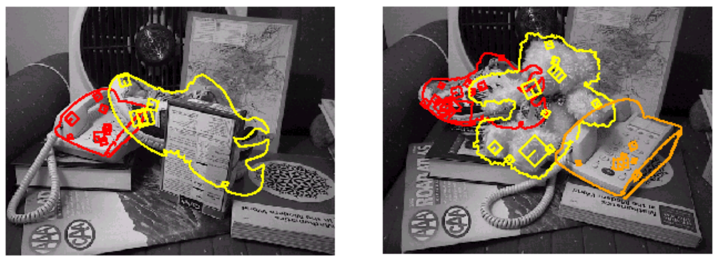
Local Invariant Features – Descriptors



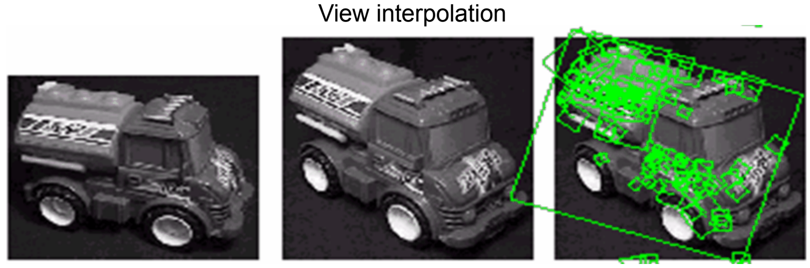
Arrows indicate “canonical orientation” of the features

Local Invariant Features

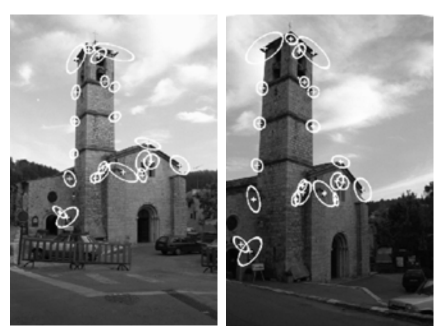
Recognition under occlusion



View interpolation



Local Invariant Features



The photo tourism example
http://phototour.cs.washington.edu/

Feature Fusion

- Generic features:
  - Color
  - Edges and contours
  - Shape geometry
  - Motion
  - Regions
- Other features:
  - Optical flow
  - Invariant features
  - Active contours

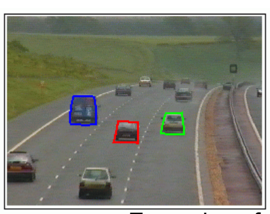
Summary of feature fusion

Active Contours

- Model-based segmentation:
  - Active contours
    - Use of prior object knowledge / model
    - Represents an object boundary or shape feature as a parametric curve
    - An energy functional E is associated with the curve
    - Finding the boundary is cast as an energy minimization problem

(Diagram courtesy "Snakes, shapes, gradient vector flow", Xu, Prince)

Energy minimization



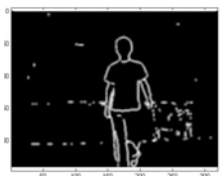
Examples of object models

Active Contours

Initialize template size → Generate energy field → Sample the image → Initialize template position → Iterate: match image

Compatible size

High-score matches



Edge map

• Use other information (other frames, other object model info)
• Very challenging part

Active Contours

- The contour is defined in the (x, y) plane of an image as a parametric curve

$$\mathbf{v}(s)=(x(s), y(s))$$

- Contour is said to possess an energy (E) which is defined as the sum of three energy terms:

$$E = E_{internal} + E_{external} + E_{constraint}$$

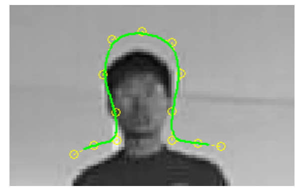
➢ Constraints of the contour:
  • E.g. relation of control points w.r.t. each other

➢ The measured field from the image:
  • E.g. the gradient field

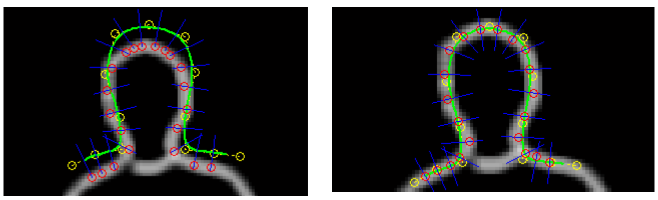
- The terms are defined to make final position of the contour have minimum energy
  - Energy minimization problem

Active Contours

- Deformable shapes – control points
  - The contour is represented by a set of <u>control points</u>
  - The curve is interpolated piecewise with the control points
    - Linear, B-splines, etc.
  - Control points are moved by the energy force



Yellow: Control points p
Green: Curve fitted to control points



Blue lines: Search line for every sample on the curve
Red: Optimal positions on a blue line, determine next position of control points

$$p' = A\,p + b$$

A and b are determined by position of Red points

Active Contours

- Issues:
  - Initialization of the shape:
    - A bad initialization may lead to the shape trapped in local minima
  - Convergence:
    - Hard to predict whether the shape will converge to the desired image features
  - Energy field:
    - How to define a global field and handle local features?
      - **Edge fragments**
    - What are the image features to look for?
  - Image noise may deform the shape in an undesired way
    - Solution:
      - **Dynamic models to predict and consider shape deformations**

Summary of Feature Fusion

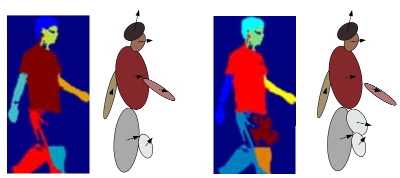
- Use any one or multiple features based on:
  - Global knowledge, object model
  - Image properties
  - Adaptive learning of the effectiveness of the selected features

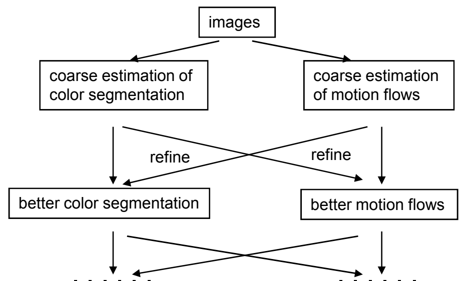
Summary of Feature Fusion

- Extract multiple helpful features in each camera
- Opportunistic approach
  - Various features may be available at different times
- Joint feature refinement
- Objective:
  - To achieve robustness in node's description of event / object
  - Allows for low-complexity implementation

Fusion of Features in Segmentation

- Summary
  - Segmentation based on different image features and the object model
  - Utilize flexibility in choice of features and interactions between them
    - Example: color & motion segmentation for human body





Summary of Feature Fusion

Pixels, Regions, Attributes

- Pixel-based feature analysis methods:
  - Information from immediate neighbors used
    - Thresholding, segmentation
  - Localized attributes need local thresholds – hard to set
    - Comparing color of foreground / background pixels
  - No information from extended neighborhood considered
    - Knowledge about extent of neighborhood not available
      - Which is the objective in many cases – segmentation

➢ Two ways to extend:

- Attribute-based methods:
  - Define vector of features for a pixel:
    - Edge strength, color, etc.

Both try to utilize similarity in one or more attributes

- Region-based methods:
  - Objects often contain correlated attributes in a region

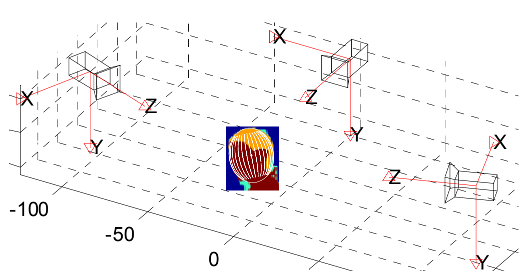
Summary of Feature Fusion

Pixels, Regions, Attributes

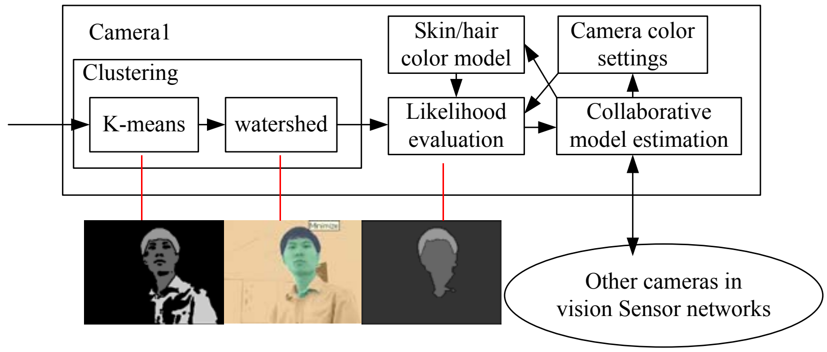
- These can be combined:
- Methods based on attributes of small regions
  - Define *invariant features* that can be used for:
    - Object detection
    - Matching between images
    - Measuring motion of objects across frames
    - Object recognition in presence of occlusion
  - Small number of invariant features used instead of pixel-level density

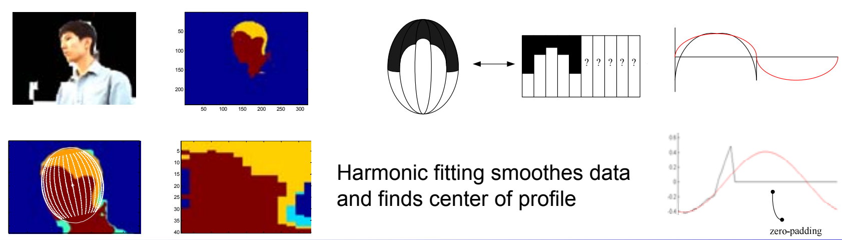
Face Orientation Analysis

- Methods:
  - Color and geometry-based method
  - Spatial / temporal validation method
  - Spatiotemporal fusion method



Hair-Face Ratio





Harmonic fitting smoothes data and finds center of profile

Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

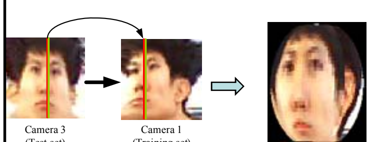
Human pose estimation

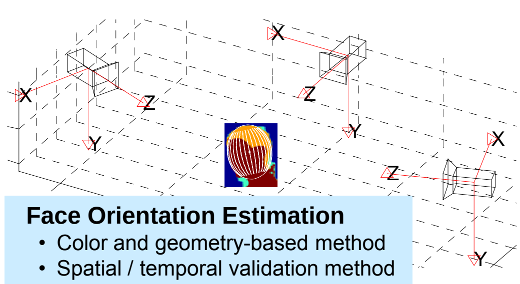
Human event detection

Spatial Fusion

Spatial Fusion

- Geometric fusion
  - Making correspondences
  - Tracking
  - Reconstruction of 3D models
  - Camera network calibration
  - Use of epipolar geometry to:
    - Feature matching
    - Outlier removal
    - ROI mapping between camera views
- Mutual reasoning
  - Joint estimation
  - Joint refinement
  - Decision fusion
- Assisted reasoning
  - Estimate validation
  - Key frame exchange

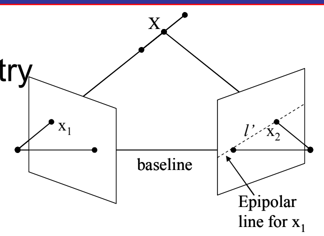


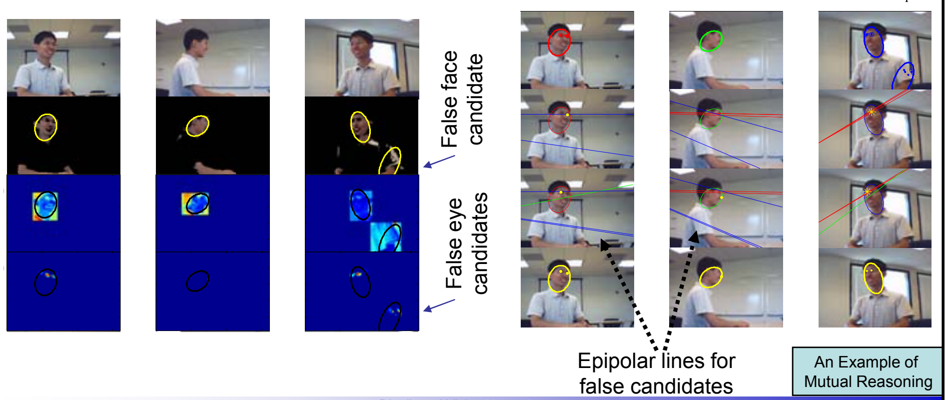


Face Orientation Estimation
- Color and geometry-based method
- Spatial / temporal validation method

Color and Geometry Fusion

- Face orientation analysis
  - Feature matching with epipolar geometry
    - Use geometry of cameras to:
      - Match features
      - Remove false feature candidates

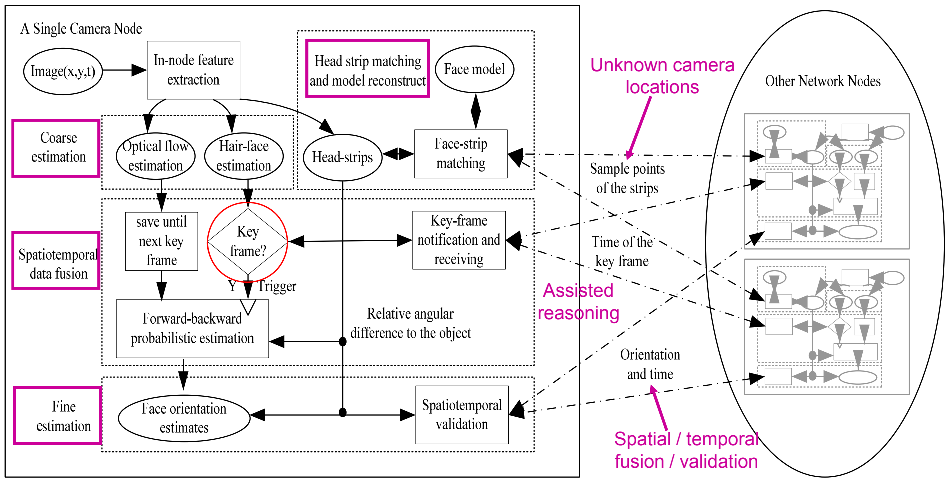




Spatial / Temporal Fusion Method

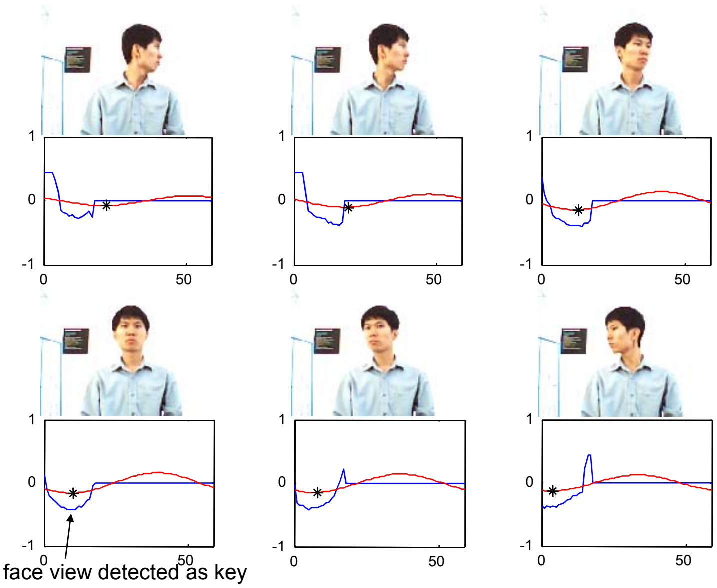
- An *assisted reasoning* method:
  - Key frame exchange
    - Value observations of frontal view
  - In-node feature fusion
    - Local angle estimates
  - Temporal fusion
    - Local interpolation of angle between key frames
  - Spatial / temporal validation
    - Face orientation estimates exchanged and validated
      - Spatial: outlier removal
      - Temporal: smoothing

Data Exchange



Region-based Fusion: Optical Flow and Color

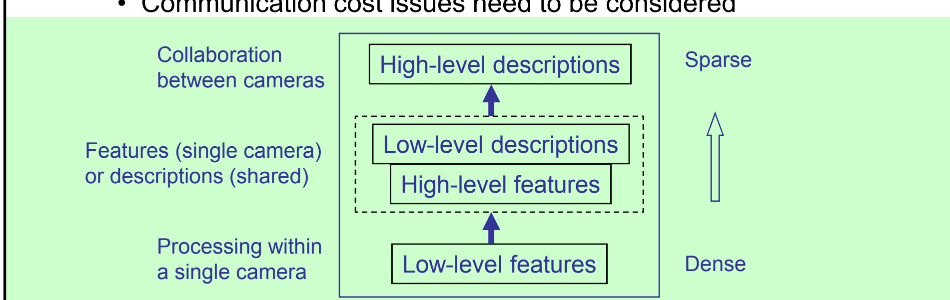
Hair-Face Ratio



Frontal face view detected as key frame info

Feature Fusion

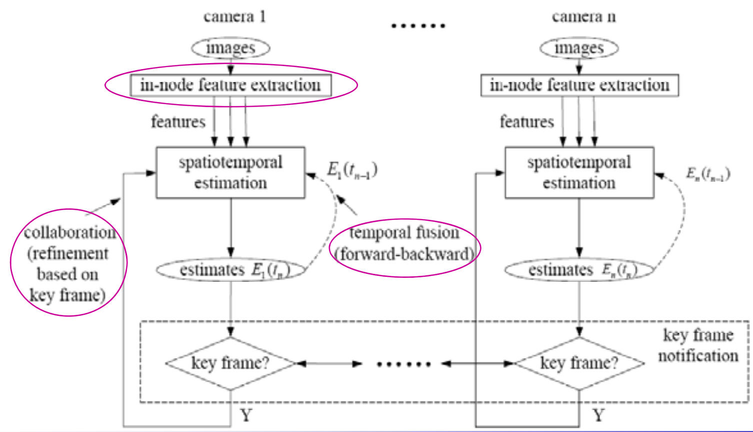
- Level of features for fusion between cameras?
  - Features are typically dense fields
    - Edge points, motion vectors
  - They are locally fused to derive descriptions (sparse)
    - Descriptions are exchanged
  - Valuable features may be exchanged as dense descriptors
    - Communication cost issues need to be considered



- <u>Key features</u> and <u>key frames</u> allow selective sharing of dense features

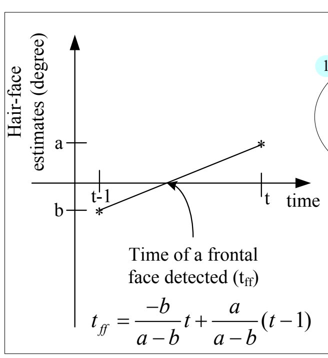
Key Frames

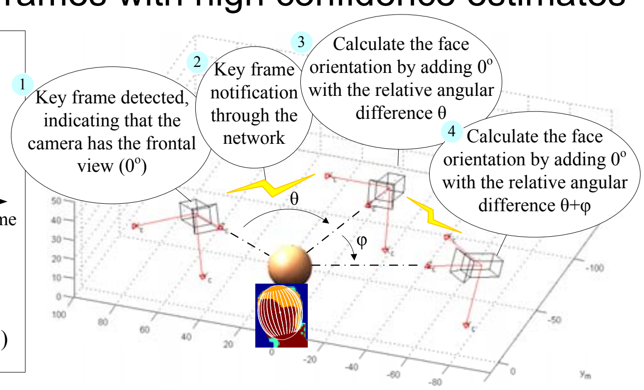
- Frames with high confidence estimates
  - Node with key frame observation broadcasts derived information
  - Other nodes use them to refine their local estimates



Key Frame Notification

- Key frames are frames with high confidence estimates

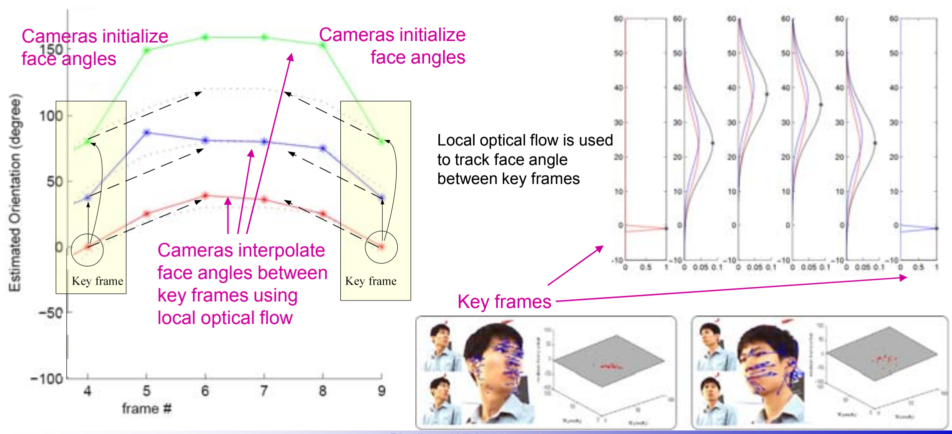




- If cameras calibrated:
  - ➢Other nodes can use received key frame information to:
    - Re-initialize their face angle tracking method
    - Calculate a weighted average for the face angle using received estimates

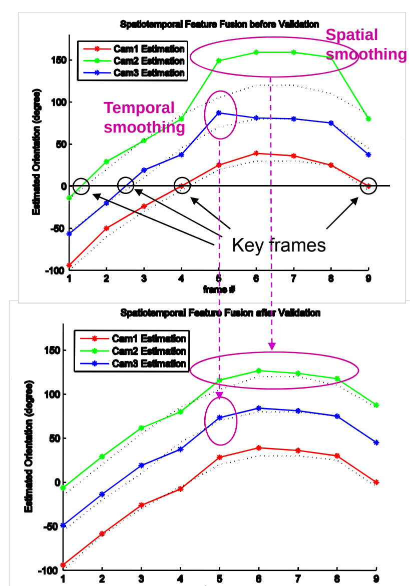
Temporal Fusion

- Use key frames to re-initialize local face angle estimate
  - Use angle estimates close to zero (frontal view)
- Aims to limit error propagation in time
  - Use optical flow to locally track angle changes between frames
  - Interpolate between two key frames to limit optical flow error propagation

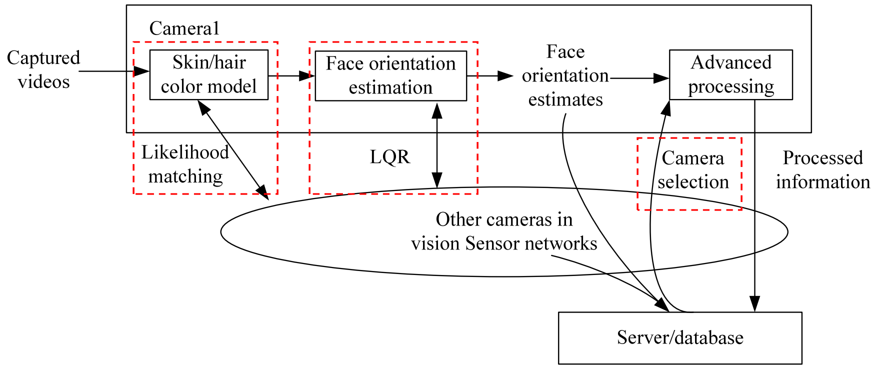


Spatial / Temporal Validation

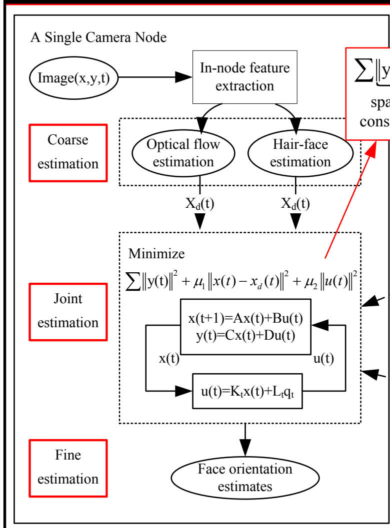
- Estimates between key frames are corrected by:
  - Temporal smoothing (one camera)
  - Outlier removal (multiple cameras)
- Can this be done more effectively?
  - ➢ Spatiotemporal filtering

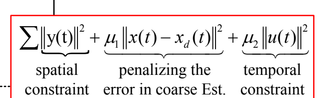


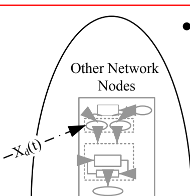
Spatiotemporal Fusion

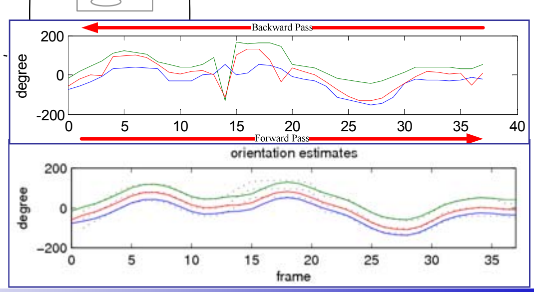


Spatiotemporal Fusion



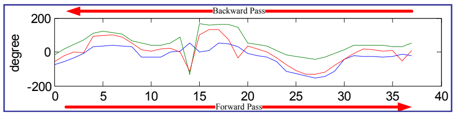


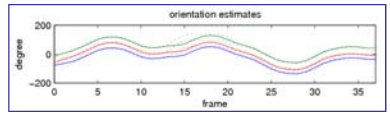




- Joint estimation by LQR:
  - Spatiotemporal filtering by minimizing a cost function

Spatiotemporal Fusion

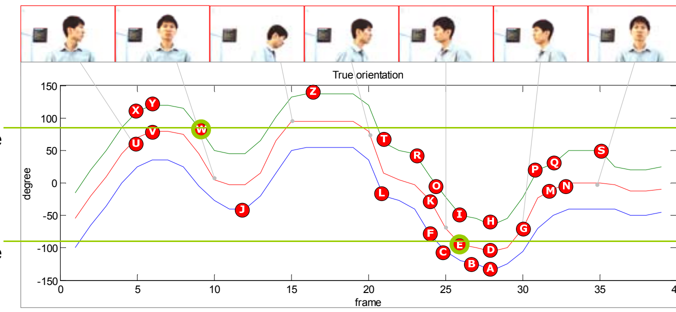




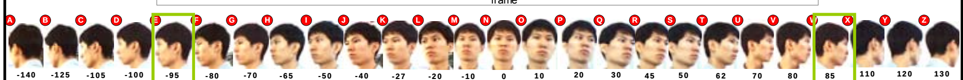
- Opportunistic creation of face profile

Right Profile

Left Profile

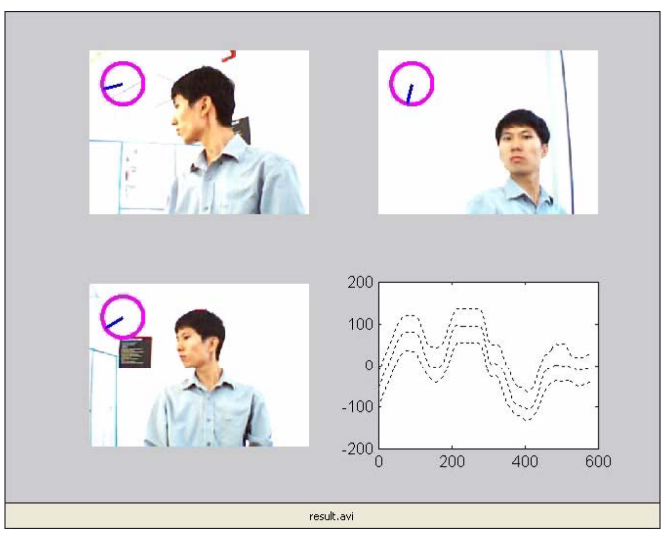


Mapped to ellipsoid



Query result examples: side profiles

Spatiotemporal Fusion



Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

Human pose estimation

Human event detection

Model-based Fusion

Model-based Fusion

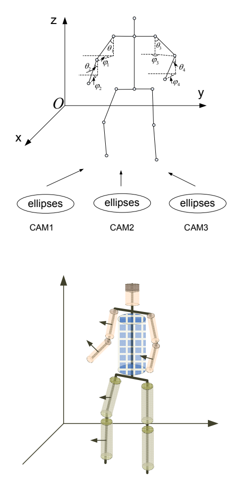
- Motivation to build a human model:
  - ➢ A concise reference for merging information from cameras
  - ➢ Universal interface for different gesture interpretation applications
  - ➢ Allows new viewing angles in virtual domain
  - ➢ Facilitates active vision methods:
    - Focus on what is important
    - Exchange descriptions only relevant to the model
    - Develop more detail in time
    - Initialize next operations (segmentation, motion tracking, etc)
  - ➢ Helps address privacy concerns in various applications

Vision ⇄ Human Model ⇄ Reasoning / Interpretations

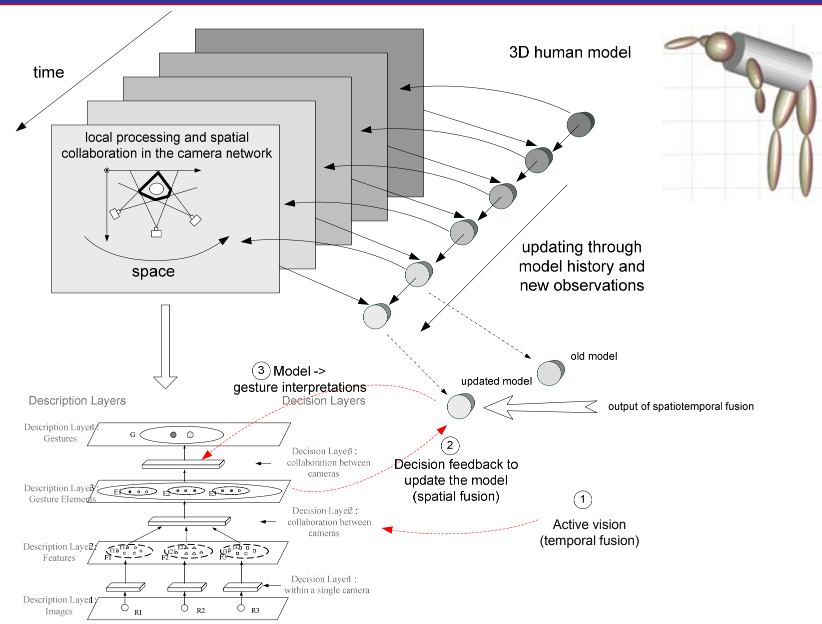
Human Model
- Kinematics
- Attributes
- States

Model-based Fusion

- Approach:
  - ➢ Exchange segments and attributes, combine to reconstruct a 3D model
  - ➢ Subject's information mapped and maintained in the model:
    - Geometric configuration: dimensions, lengths, angles
    - Color / texture / motion of different segments

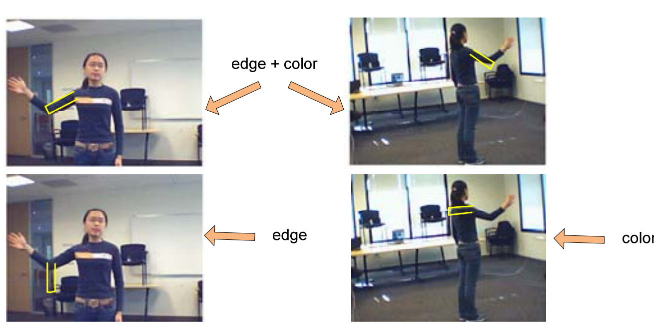


Spatiotemporal Fusion



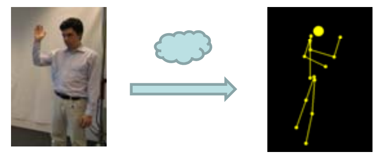
Feature Fusion

- **Edge**
  - Templates
  - Chamfer distance (distance, orientation)
- **Color**
  - skin color
  - adaptively learned color
- **Motion**
  - Structure
  - Object boundaries / edges
- **No single method is robust !**
  - Point / line features vs region features



Posture Estimation – Review

- Discriminative -> *template-based*
- Generative -> *model-based*
  - Bottom-up
  - Top-down
- Combined
  - Discriminative for body parts
  - Generative for whole-body configuration

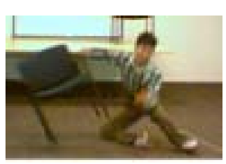


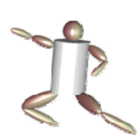
Multi-view Challenges

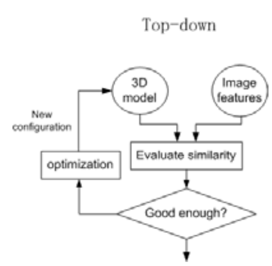
- redundancy
- misleading info in some images
- correspondence
- communication (*images?*)

Posture Estimation – Top-Down Approach

– 3D model -> 2D projections of edges and silhouettes
– Validate 2D projections with image observations
+ Easy to handle occlusions
- Difficult to optimize: non-convex
- Time consuming in calculating projections and evaluating them

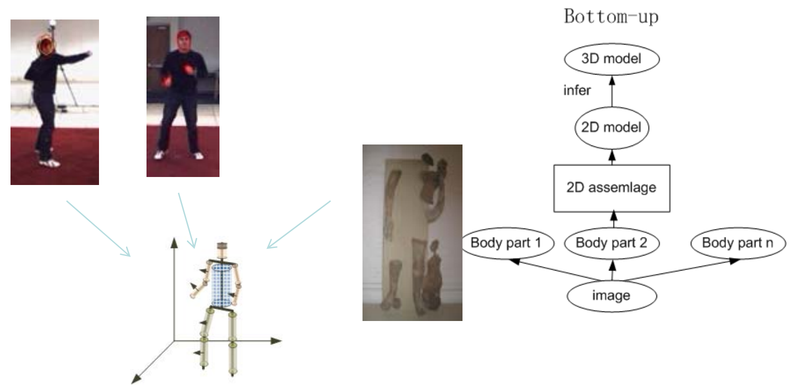






Posture Estimation – Bottom-Up Approach

– Looking for body part candidates in images
– Assemble 2D/3D models from body part candidates
+ Distribute more computation in images (i.e. body part candidates, local assemblage)
- Difficult to handle occlusions without knowing relative configurations of body parts
- Not direct to map from 2D assemblage to the 3D model



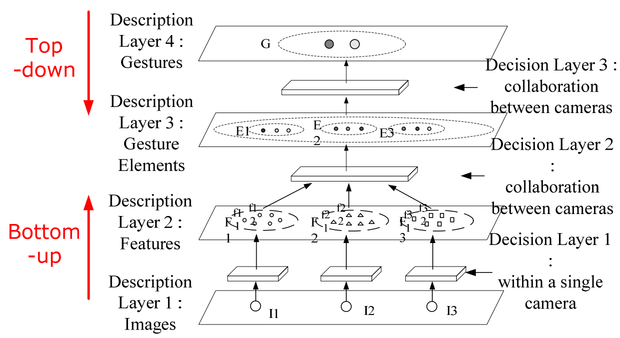
Multi-View Camera Network

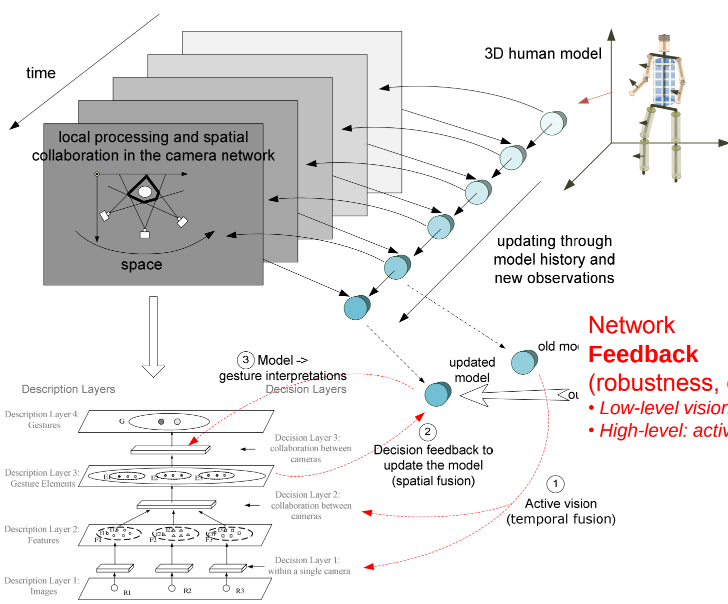
- Basic Assumption and Constraint

  -- *Powerful local image processor, limited communication*
  – Reduce local information
  – Maximally utilize multi-views:
    - to compensate for partial observations and reduced descriptions
- Ideas
  – Combine bottom-up and top-down approaches
    - Concise and informative local deduction
  – Choose best view for different purposes
    - Optimally combine
    - Reduce redundancy
  – *Challenge: Can we learn adaptively?*
    - *Model (size, appearance)*
    - *Behaviors -> prediction & validation*

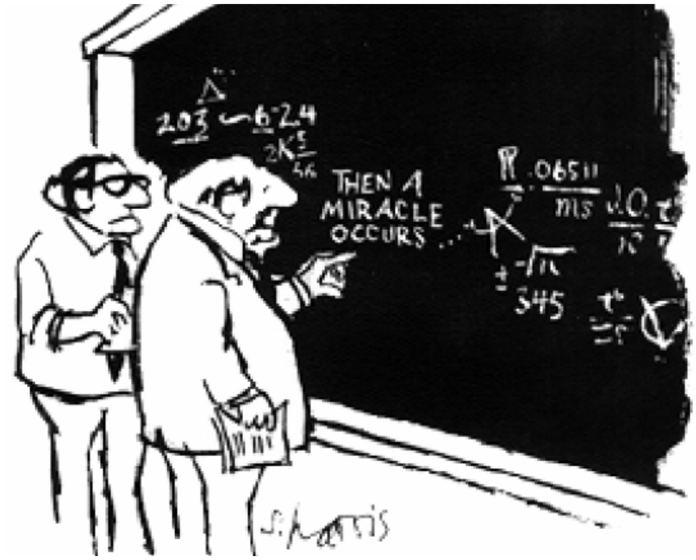
Posture Estimation – Strategies

- Combine bottom-up and top-down approaches to
  - Locally : Image -> descriptions
  - Hierarchical search for full body geometric configuration



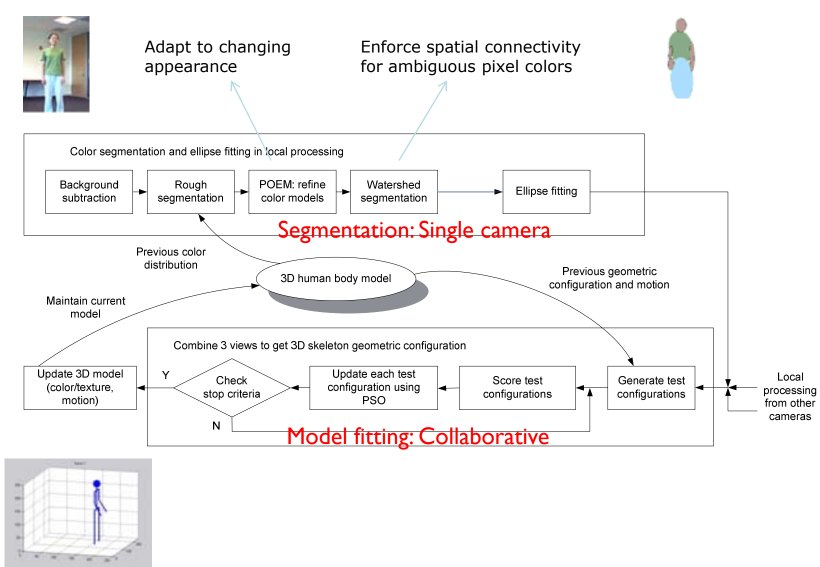


Network **Feedback**
(robustness, efficiency)
- *Low-level vision*: appearance
- *High-level: activity interpretation*



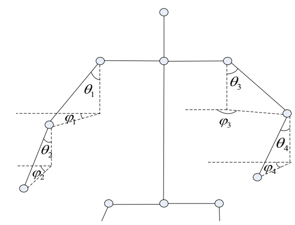
" I think you should be more explicit here in step two."

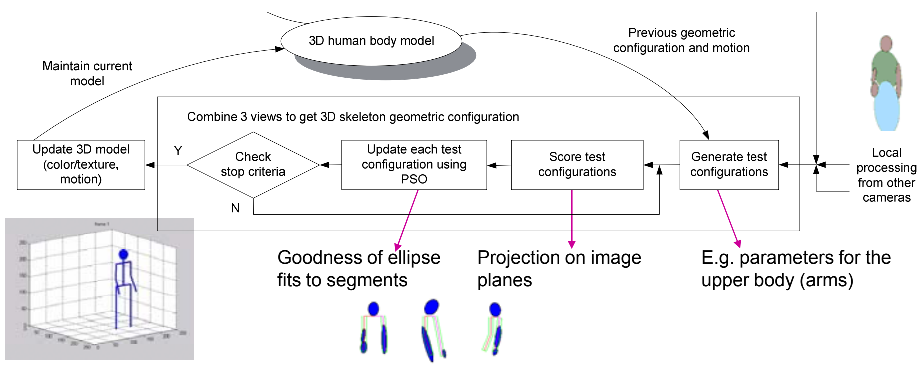
Posture Estimation



Collaborative Model Fitting

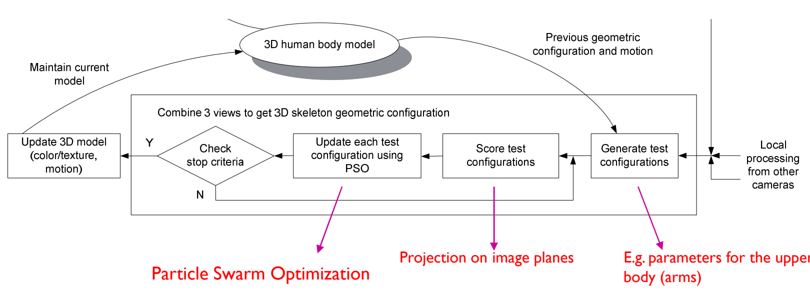
- Ellipse parameters are exchanged between cameras
  - Reduced data communication load for collaboration
- A server node or each of cameras collect the data and create a "virtual skeleton"





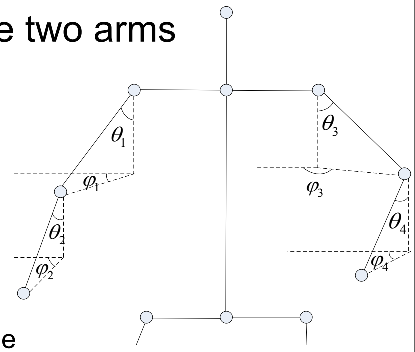
Posture Estimation – Optimization

- Key problem
  - Explore possible local optima as candidates for the global optimal
  - Determine the global optimal
- Techniques
  - Particle filtering: multiple hypothesis
  - Graphical models: exponential -> linear
- Particle Swarm Optimization (social /inertia coefficient)



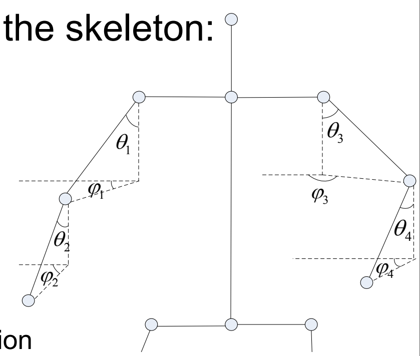
Skeleton Fitting

- A simple example: fitting for the two arms
  - 8 parameters:
    - Elevation angles: θ
    - Azimuth angles: φ
  - Assumptions:
    - Known projection from 3D to 2D image planes (localization information)
    - Normalize the 2D projection to size and position of ellipses in the image
      - Use subject's orientation and geometric shape

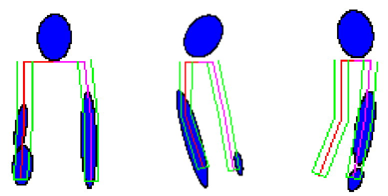


Skeleton Fitting

- Options to find parameters for the skeleton:
  - ➢Solve for θ's and φ's based on geometry
    - Need to first establish correspondence between camera observations
      - A hard problem especially under occlusion
    - Ambiguity on 3D positions exists even if we have 2D projections of several cameras
  - ➢Cast as an optimization problem and find θ's and φ's to minimize an objective function
    - Non-linear and non-convex
      - Difficult to solve
  - ➢Sample the solution space and find the best sample (particle filtering)
    - Not so intelligent if involves exhaustive search
    - Can model constraints be used to determine the search space?
      - A feasible solution

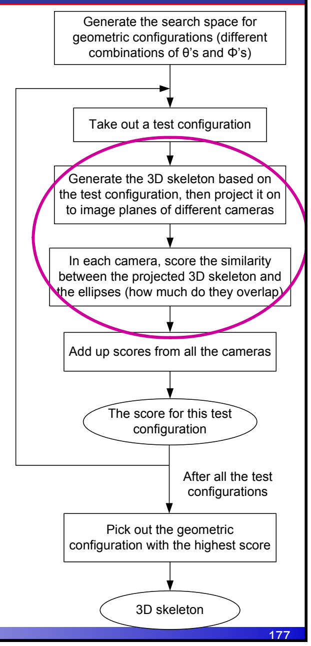


Skeleton Fitting



• Red: projection of skeleton on image plane
• Green: region of arms grown from red lines
• Blue: ellipses from segmentation

➢ Score = Area (ellipses falling within green polygons) / Area (green polygons)



Collaborative Model Fitting

Frame 1

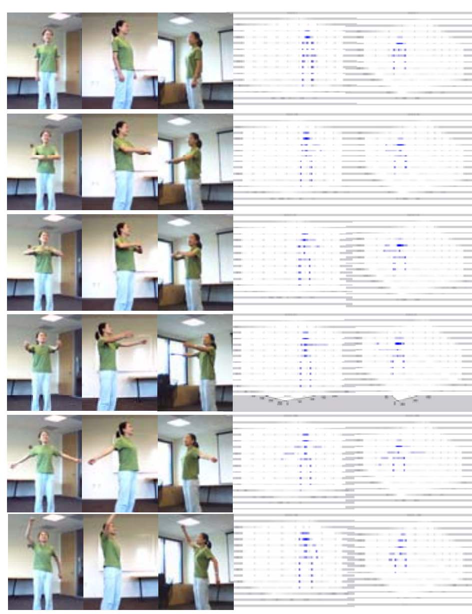
Frame 28

Frame 70

Frame 81

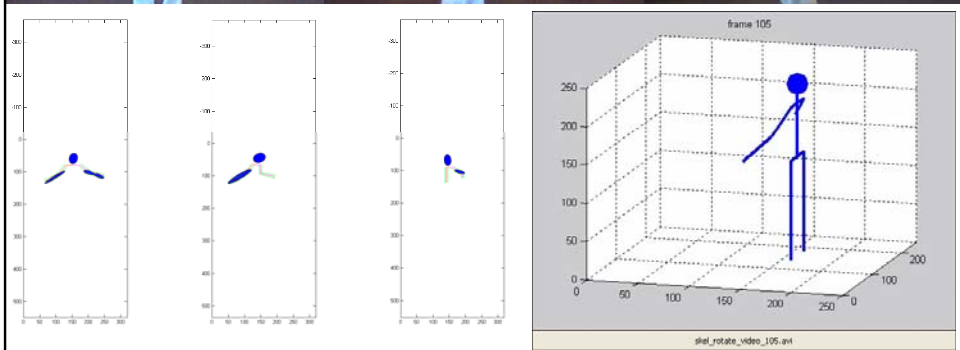
Frame 105

Frame 148

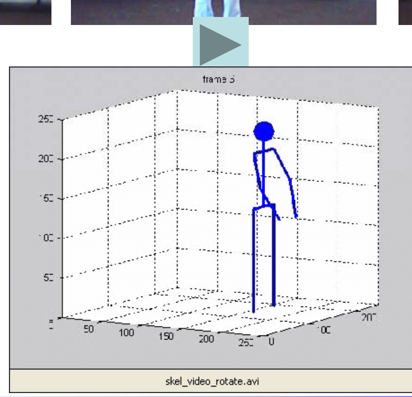


Collaborative Model Fitting

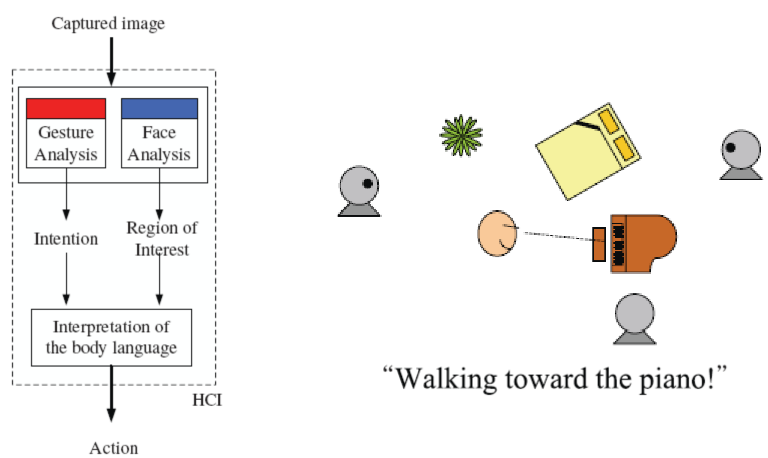
Frame 105



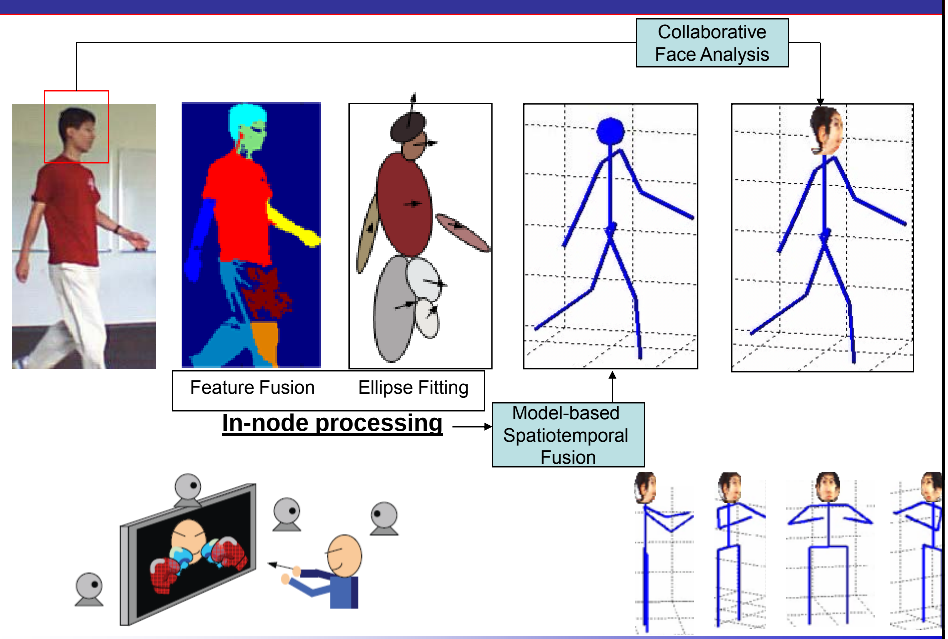
Collaborative Model Fitting



Virtual Placement



Virtual Placement



Towards *real-time* + *wireless* communication

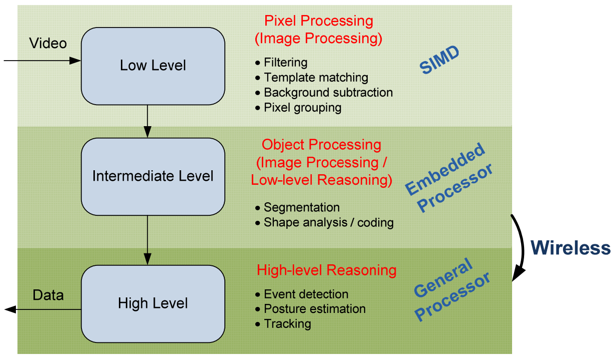
Challenges

- Wireless (ZigBee) + Real-time vision
  - ~100Kbit / 30 fps ~ 400B/frame
- Computation capacity
  - How fast is the whole system given enough comm bandwidth?

Strategy

Distributed Computation!

Hardware support ⟺ Balance! ⟺ Algorithm

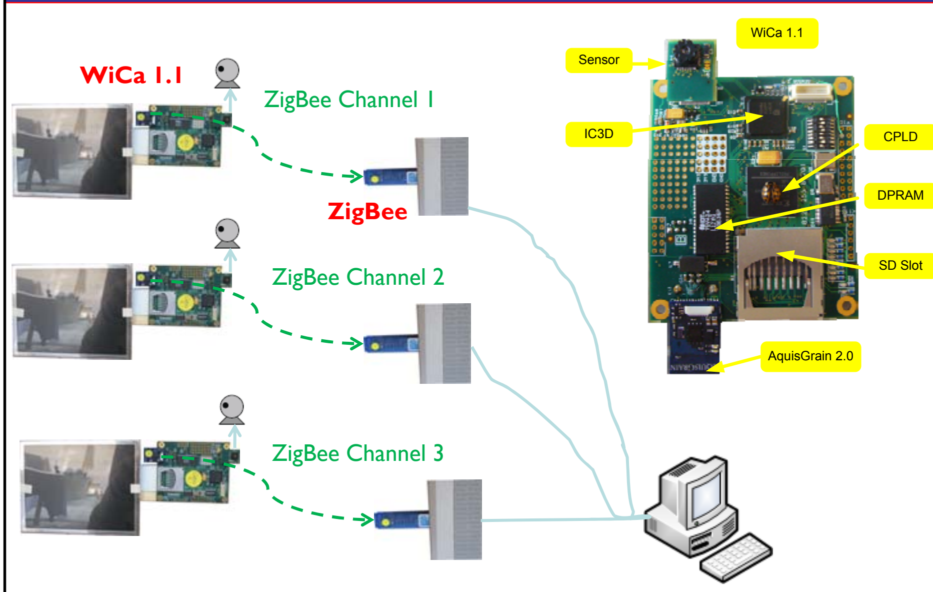


SIMD(Single Instruction Multiple Data)
- high performance
- low power

	Xetal-II SIMD : 320PE@150MHz	Pentium4 2.4GHz
Peak Performance	100 GOPS	6 GOPS
Peak Power Consumption	1.0 Watt	59 Watt

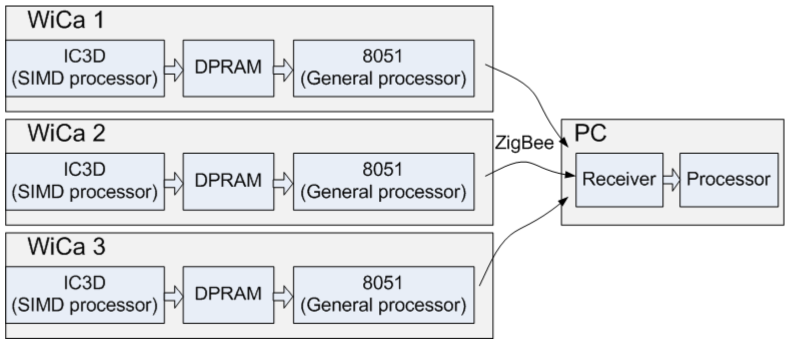
Joint work with NXP Semiconductors, the Netherlands

The System

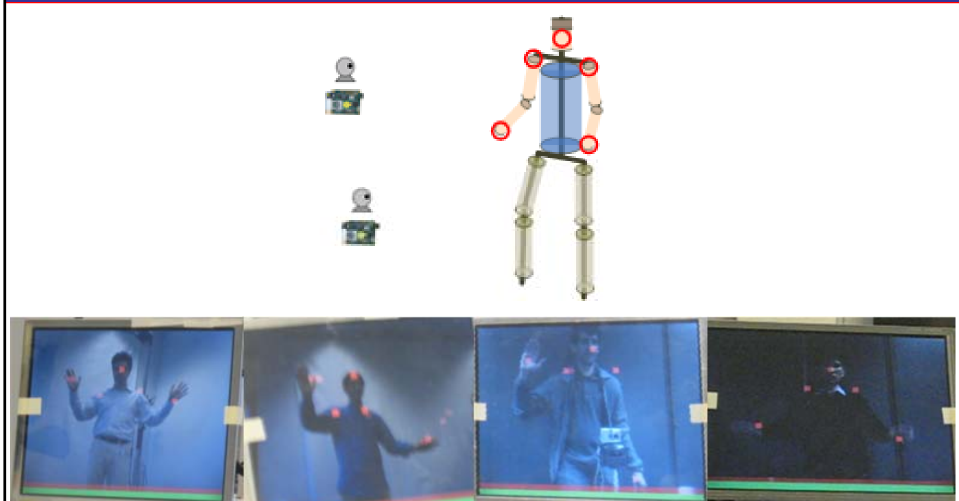


Joint work with NXP Semiconductors, the Netherlands

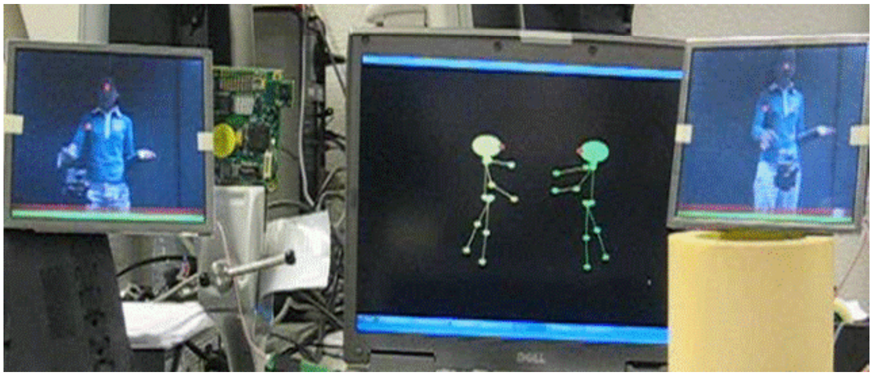
Data Flow



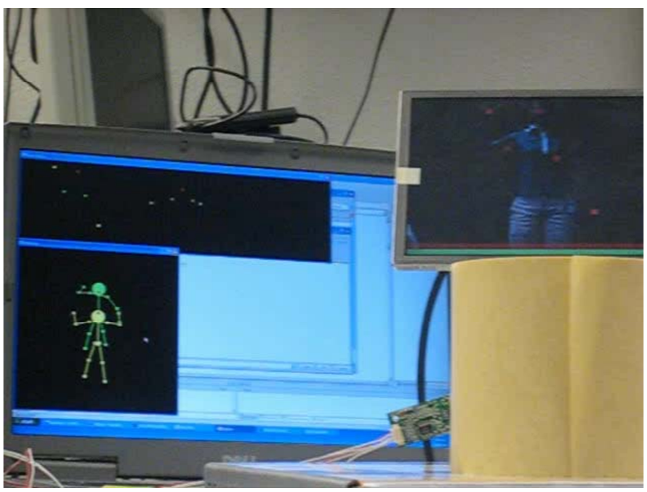
Model



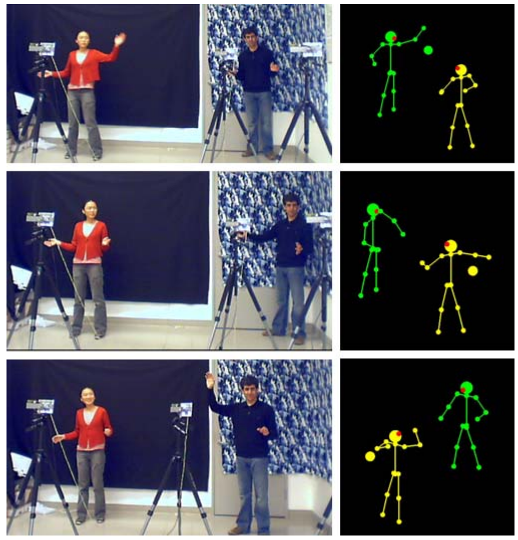
Demo



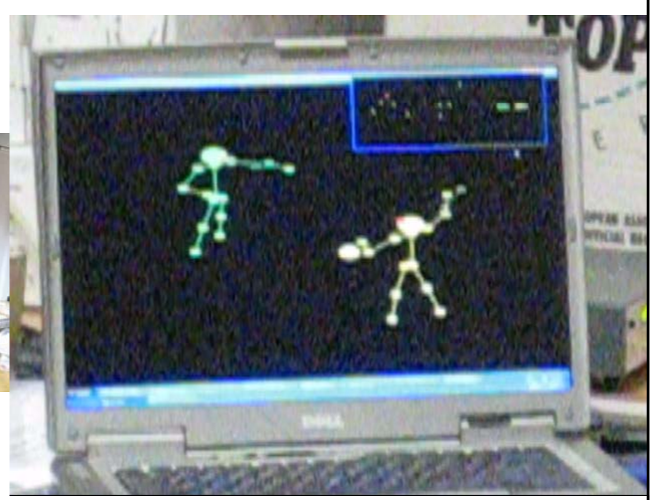
Demo



BallGame Application

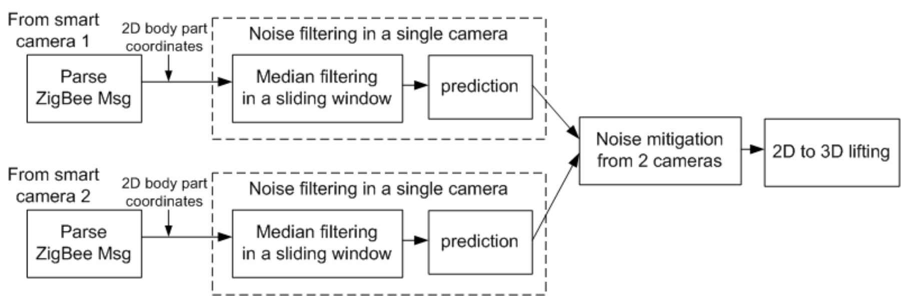


BallGame Application

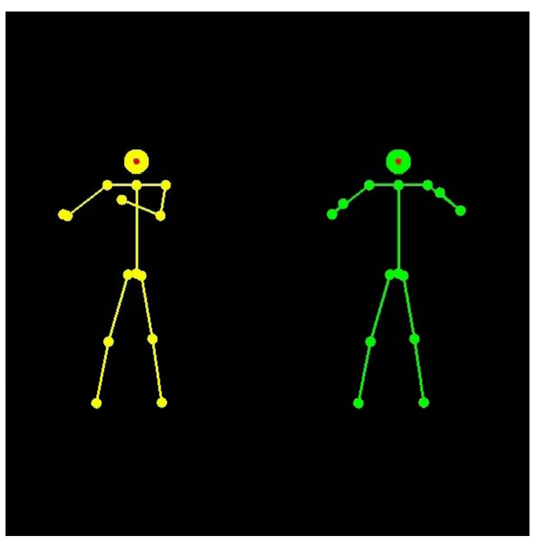


ICDSC (International Conference on Distributed Smart Cameras)
Sept 2007, Vienna, Austria

Spatiotemporal Smoothing



Spatiotemporal Smoothing



No smoothing

Two-camera feature fusion and temporal smoothing

Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

Human pose estimation

Human event detection

Decision Fusion

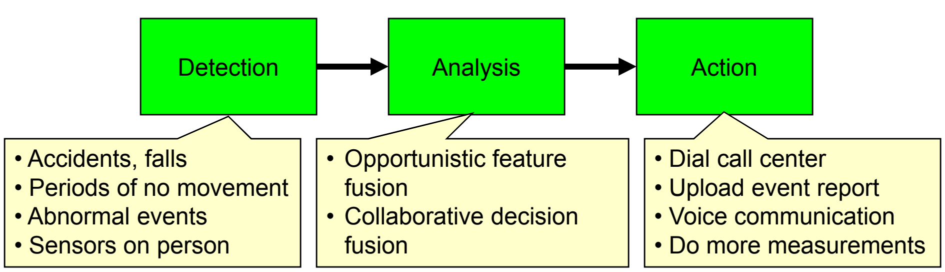
Decision Fusion

- Cameras may independently do multiple feature level processing due to:
  - Adequate features in own observations
  - Cost, latency of communication
  - Lack of event observation in some cameras due to spatial distribution

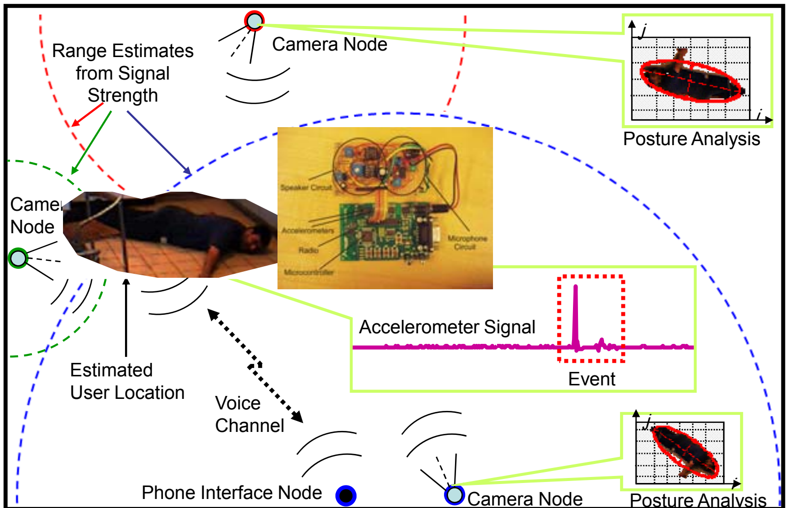
- Processing models based on:
  - Opportunistic feature fusion in each camera
    - Use of all available information to make decision
  - Soft decision exchange
    - Through the use of detected states and event priority levels
  - Event subscription data exchange model
    - Allows participation by all interested nodes
  - Certainty assignment module
    - Provides basis for comparing node decisions

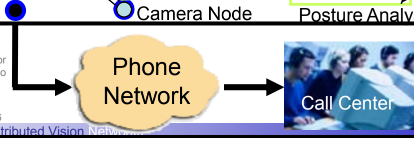
Smart Home Care Network

- Objectives:
  - Home care monitoring system
  - Allowing ***independent*** living
  - ***Access*** to help when needed
  - Event ***analysis*** and reporting
  - ***Low false alarm*** via multi-modal analysis



Smart Home Care Network



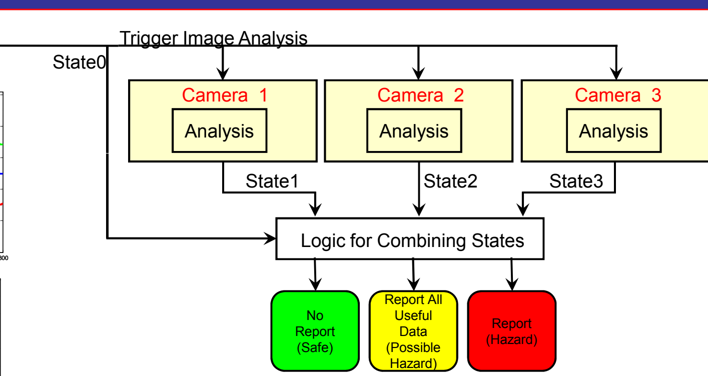


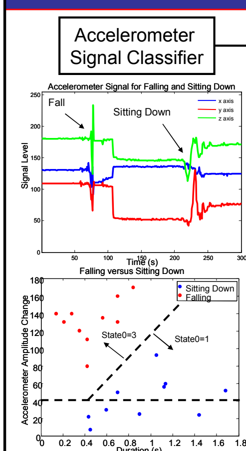
References:

• A. Maleki-Tabar, A. Keshavarz, H. Aghajan, "Smart Home Care Network using Sensor Fusion and Distributed Vision-Based Reasoning", ACM Multimedia Workshop On Video Surveillance and Sensor Networks, Oct. 2006

• A. Keshavarz, A. Maleki-Tabar, H. Aghajan, "Distributed Vision-Based Reasoning for Smart Home Care", ACM SenSys Workshop on Distributed Smart Cameras, Oct. 2006

Decision Fusion Model





- Accelerometer signal:
  - Hard to reliably classify into fall / no-fall
    - Large variation from person to person
    - May have similar signature with sitting down, bending down
  - Can be used to detect sudden movements
  - Triggers vision analysis
  - Severity of signal can be used at decision fusion logic

Opportunistic Feature Fusion

Frame Subtraction & Blob Extraction

Identify Human/Non-human Blobs

Human

Non-human

Log the Event

Posture Detection

Head Detection

Certainty Assignment Module

Exchange State with Other Nodes

Report Event Based on State

Opportunistic Feature Fusion

Frame Subtraction & Blob Extraction

Identify Human/Non-human Blobs
- % of skin color in blob
- % of straight edges in blob

Human

Non-human

Log the Event

Posture Detection

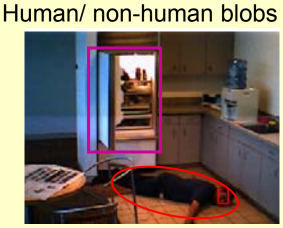
Head Detection

Certainty Assignment Module

Exchange State with Other Nodes

Report Event Based on State

Human/ non-human blobs



Opportunistic Feature Fusion

Frame Subtraction & Blob Extraction

Identify Human/Non-human Blobs
- % of skin color in blob
- % of straight edges in blob

Human → **Posture Detection**
- **Body orientation**
- **Aspect ratio**

→ Head Detection

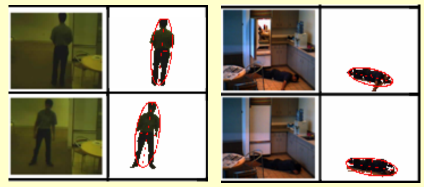
Non-human → Log the Event

Certainty Assignment Module

Exchange State with Other Nodes

Report Event Based on State

Posture Detection:



Opportunistic Feature Fusion

Frame Subtraction & Blob Extraction

Identify Human/Non-human Blobs
- % of skin color in blob
- % of straight edges in blob

Human → Posture Detection
- Body orientation
- Aspect ratio

→ **Head Detection**
- **Skin color**
- **Shoulder/neck profile**

Non-human → Log the Event

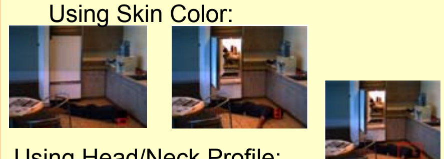
Certainty Assignment Module

Exchange State with Other Nodes

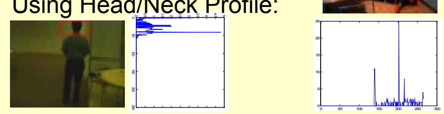
Report Event Based on State

Head Detection:

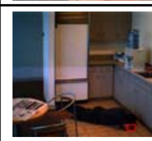
Using Skin Color:



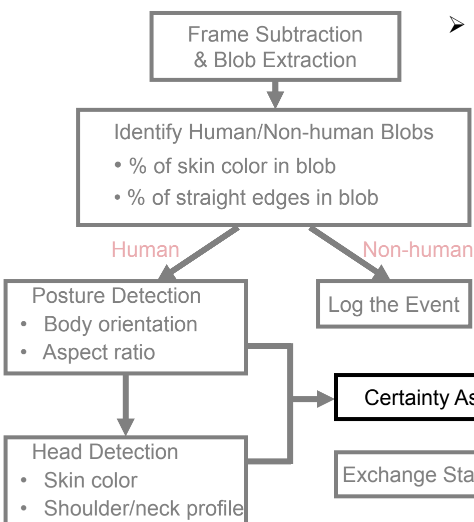
Using Head/Neck Profile:



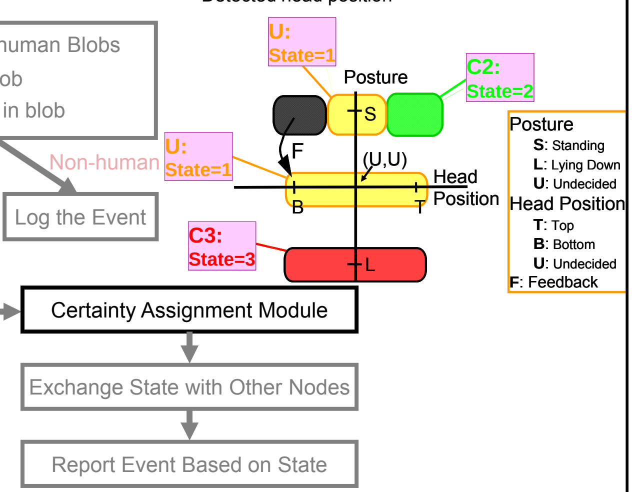
Opportunistic Feature Fusion

Image	Body Mask	Head Mask	Posture
			Standing
			Standing
			Lying Down
			Lying Down

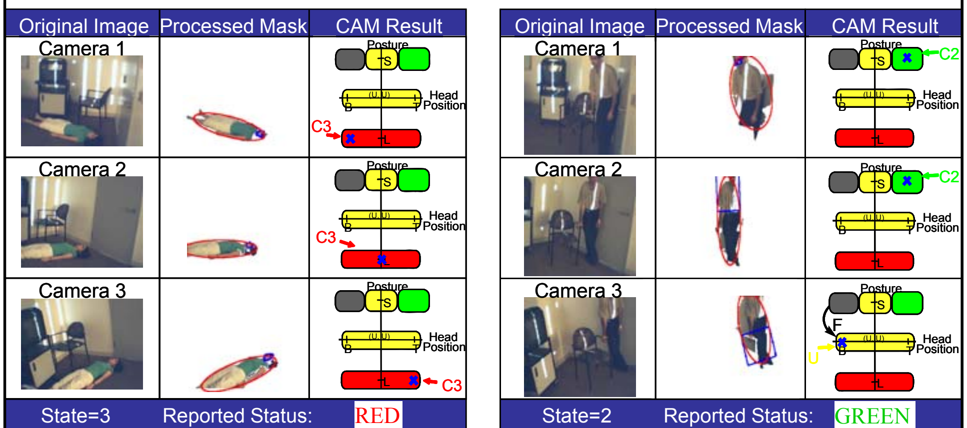
Certainty Assignment Module



- Each camera produces a state based on:
  - Detected posture
  - Detected head position

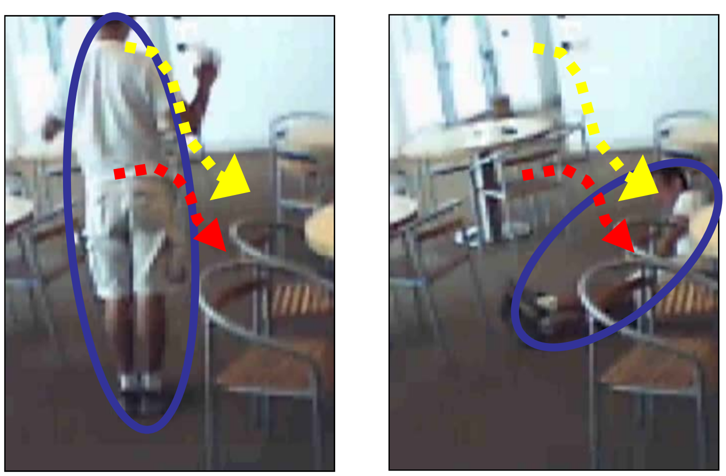


Decision Fusion

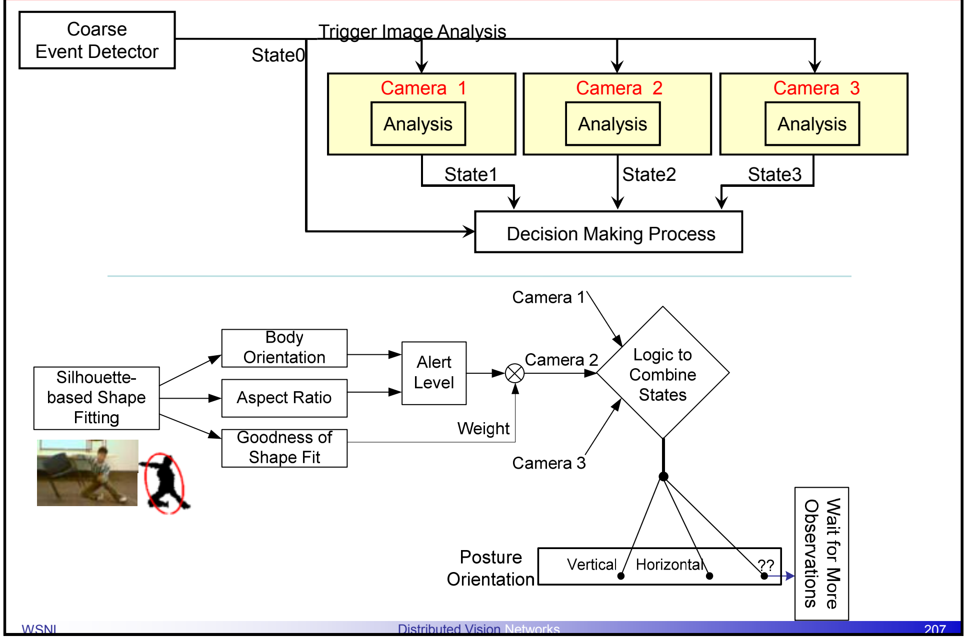


Other Reports

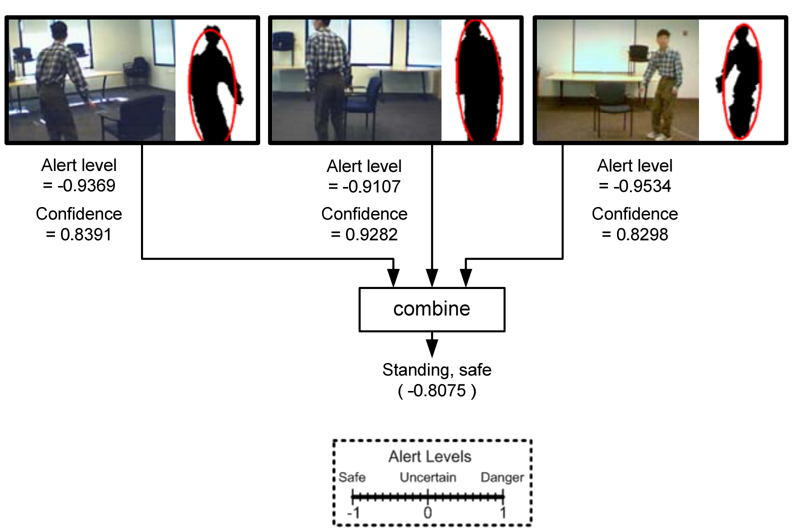
- Each camera produces trajectory of body mask and head during a fall



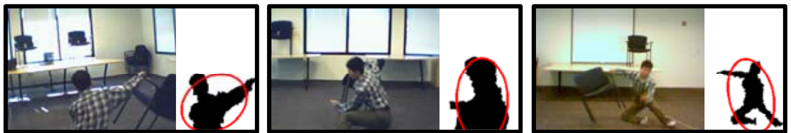
Event Analysis



Event Analysis



Event Analysis



Alert level
= -0.1651
Confidence
= 0.7346

Alert level
= -0.7920
Confidence
= 0.8153

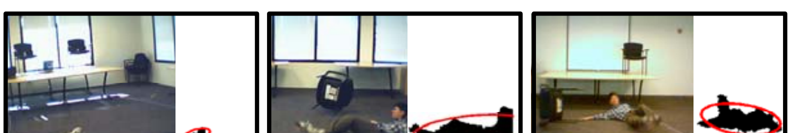
Alert level
= -0.5945
Confidence
= 0.6517

combine

Uncertain
(-0.3039)

Alert Levels
Safe Uncertain Danger
-1 0 1

Event Analysis



Alert level
= 0.6598
Confidence
= 0

Alert level
= 0.8370
Confidence
= 0.7389

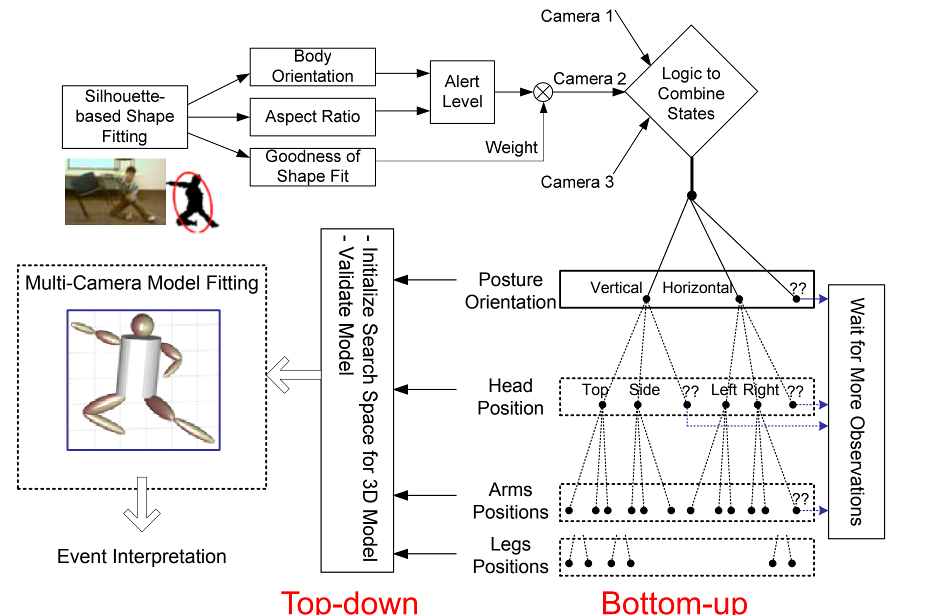
Alert level
= 0.8080
Confidence
= 0.7695

combine

Lying down, danger
(0.6201)

Alert Levels
Safe Uncertain Danger
-1 0 1

Hierarchical Reconstruction



Outline

- Introduction
- Application potentials
- Data fusion mechanisms
  - Features and feature fusion
  - Spatial / spatiotemporal fusion
  - Model-based fusion
  - Decision fusion
- Outlook

Human face angle estimation

Human pose estimation

Human event detection

Summary

- Smart camera networks:
  - Towards novel user-centric applications:
    - Interpretive
    - Context aware
    - Generalized HCI
  - Processing at source allows:
    - Image transfer avoidance
    - Scalable networks
    - Descriptive reports
  - Privacy issues:
    - Awareness of user choices
    - In-node processing and image transfer avoidance
    - Model-based or silhouetted images to reconstruct event

Summary

- Opportunistic data fusion:
  - Within one camera
  - Between cameras
  - Use of all available information
  - Lower complexity methods
- Key features and key frames:
  - Information assisting other nodes
- Spatial fusion:
  - Locations, angles, movements, matching features
  - Validation of estimates by checking consistency, outlier removal
  - Occlusion handling, ambiguity resolution
  - Handling short events, time limits in estimation
  - 3D reconstruction, model-based, feedback
- Temporal fusion:
  - Local interpolation of estimates
  - Collaborative estimate smoothing
  - Iteration towards better estimates with new observations

Summary

- Distributed vision networks:
  - Algorithm design is key in efficient use of computing resources
    - In-node feature extraction and opportunistic fusion
    - Use of key features in the data exchange mechanism
    - Model-based approach provides feedback / initial points for in-node processing
  - Balance issues between in-node and collaborative processing
    - Communication cost
    - Latency
    - Processing complexities
    - Levels of data fusion

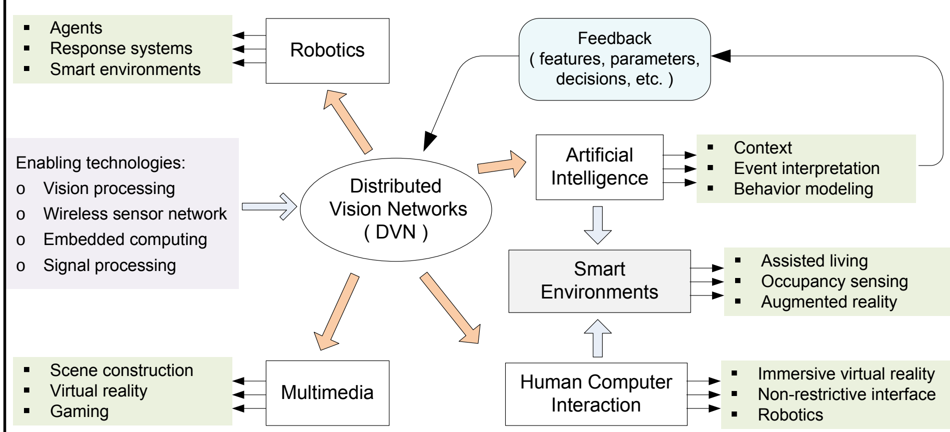
Towards Active Vision

- Active vision in feature extraction:
  - Detection of prominent color / texture attributes
  - Use of features that matter instead of generic features
  - Use of spatiotemporal fusion results to learn key features
- Active vision in modules with processing load:
  - Instead of avoiding methods with high processing cost / latency:
    - Define what the methods should look for
    - Perform initialization to restrict searches
- Active vision in gesture analysis:
  - Use prior knowledge to guide vision network:
    - History of subject
    - Semantic meanings of gestures
    - Context of the observed event

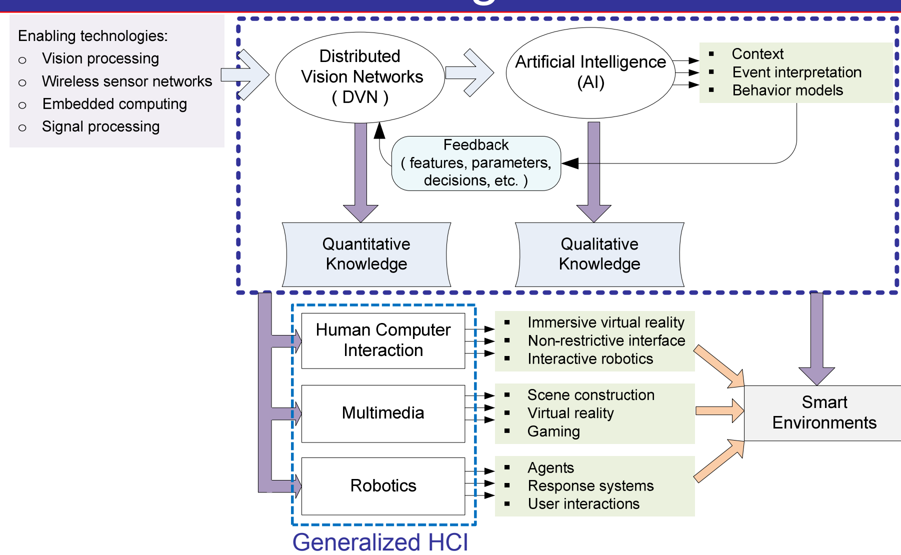
Open Questions

- How much advantageous over monocular? In what ways? How to use them in the correct way?
- Capability limit of the camera network (how well can it understand the scene, how many views are needed)?
- Balance and trade-off : In-node v.s. collaborative processing
- Networking: Data exchange v.s. latency

Outlook

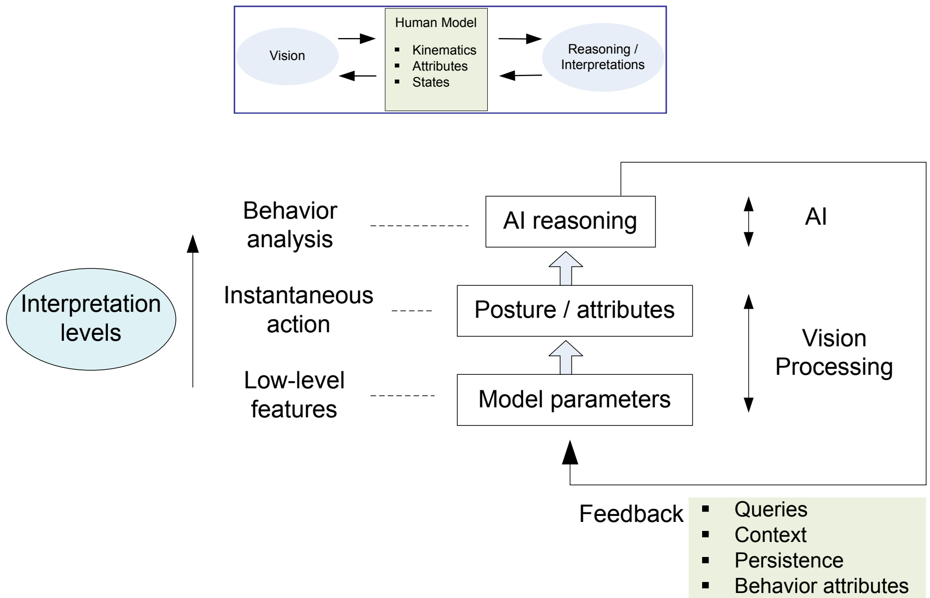


Interfacing Vision



- Quantitative knowledge provides specific distinctive information for the AI module
- Qualitative representation offers clues to the features of interest to be extracted
- This can lead to active vision approaches

Interfacing Vision



Behavior Model

References

- H. Aghajan and C. Wu, From Distributed Vision Environment to Human Behavior Interpretations, Behaviour Monitoring and Interpretation Workshop, 30th German Conference on Artificial Intelligence, Sept. 2007.
- C. Chang, C. Wu, and H. Aghajan, Pose and Gaze Estimation in Multi-Camera Networks for Non-Restrictive HCI, Int. Conf. on Computer Vision – ICCV Workshop on HCI, Oct. 2007.
- C. Chang and H. Aghajan, Linear Dynamic Data Fusion Techniques for Face Orientation Estimation in Smart Camera Networks, ICDSC 2007.
- C. Wu and H. Aghajan, Model-based Human Posture Estimation for Gesture Analysis in an Opportunistic Fusion Smart Camera Network, AVSS 2007.
- C. Chang and H. Aghajan, A LQR Spatiotemporal Fusion Technique for Face Profile Collection in Smart Camera Surveillance, AVSS 2007.
- C. Chang and H. Aghajan, Spatiotemporal Fusion Framework for Multi-Camera Face Orientation Analysis, ACIVS 2007.
- C. Wu and H. Aghajan, Model-based Image Segmentation for Multi-View Human Gesture Analysis, ACIVS 2007.
- Hamid Aghajan and Chen Wu, Layered and Collaborative Gesture Analysis in Multi-Camera Networks, ICASSP 2007.
- Chen Wu and Hamid Aghajan, Opportunistic Feature Fusion-based Segmentation for Human Gesture Analysis in Vision Networks, IEEE SPS-DARTS 2007.
- Chen Wu and Hamid Aghajan, Collaborative Gesture Analysis in Multi-Camera Networks, ACM SenSys Workshop on Distributed Smart Cameras 2006.
- Chung-Ching Chang and Hamid Aghajan, Collaborative Face Orientation Detection in Wireless Image Sensor Networks, ACM SenSys Workshop on Distributed Smart Cameras 2006.
- A. Maleki-Tabar, A. Keshavarz, H. Aghajan, "Smart Home Care Network using Sensor Fusion and Distributed Vision-Based Reasoning", ACM Multimedia Workshop On Video Surveillance and Sensor Networks 2006.
- A. Keshavarz, A. Maleki-Tabar, H. Aghajan, "Distributed Vision-Based Reasoning for Smart Home Care", ACM SenSys Workshop on Distributed Smart Cameras 2006.

http://wsnl.stanford.edu/publications.php