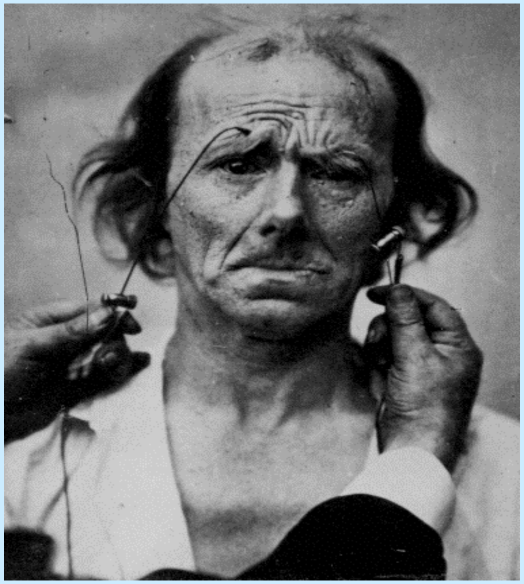
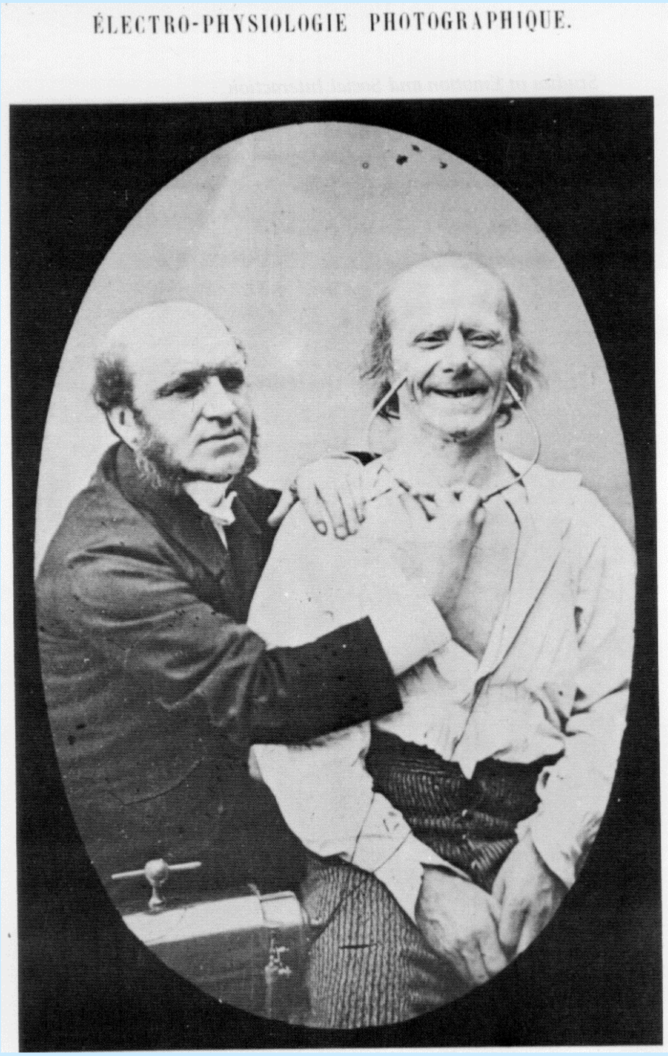
Real Time Video Surveillance of Human Activity

Larry Davis
Computer Vision Laboratory
University of Maryland
College Park, Maryland

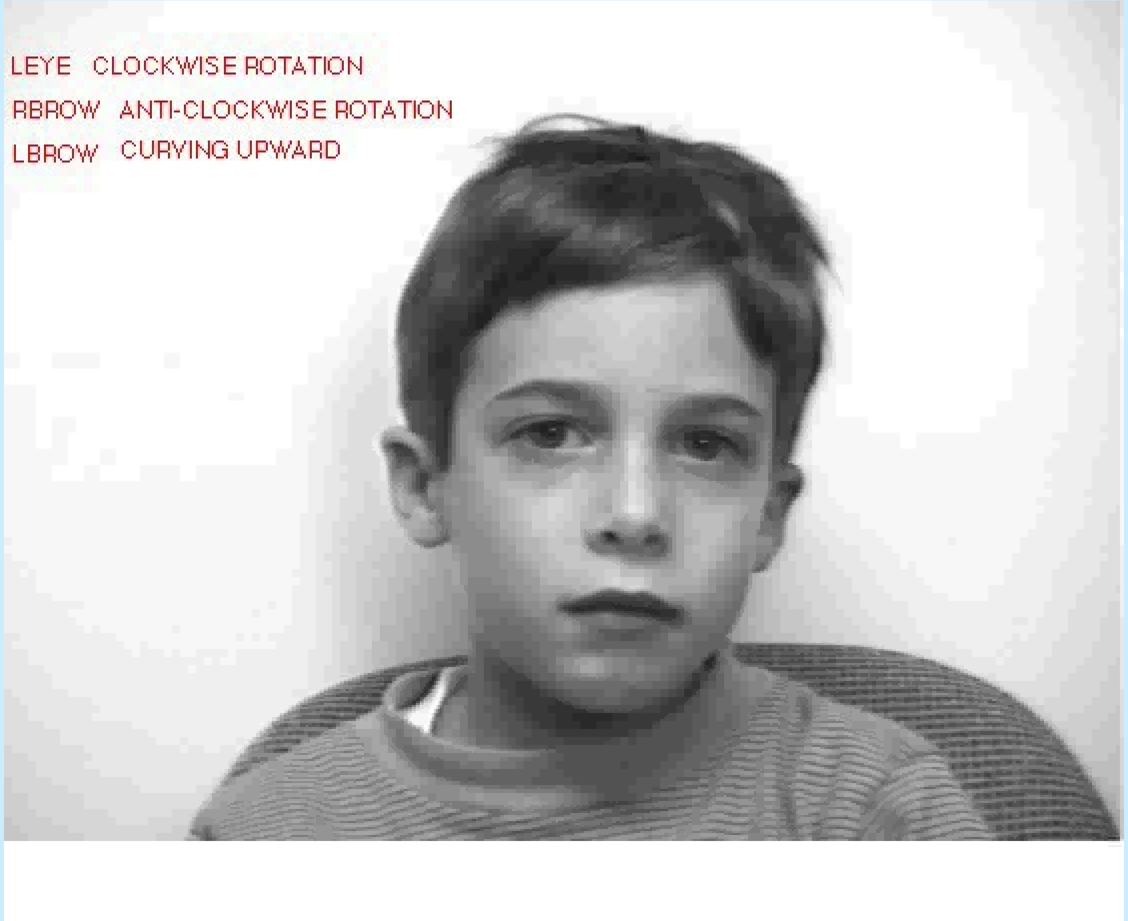
Recognition of facial expressions

- Black and Yacoob
- Recognize expressions based on nonrigid motions of facial features
  - separate head "flow" into rigid head motion and facial feature motion
  - applied to real video (Amadeus, 2001, …





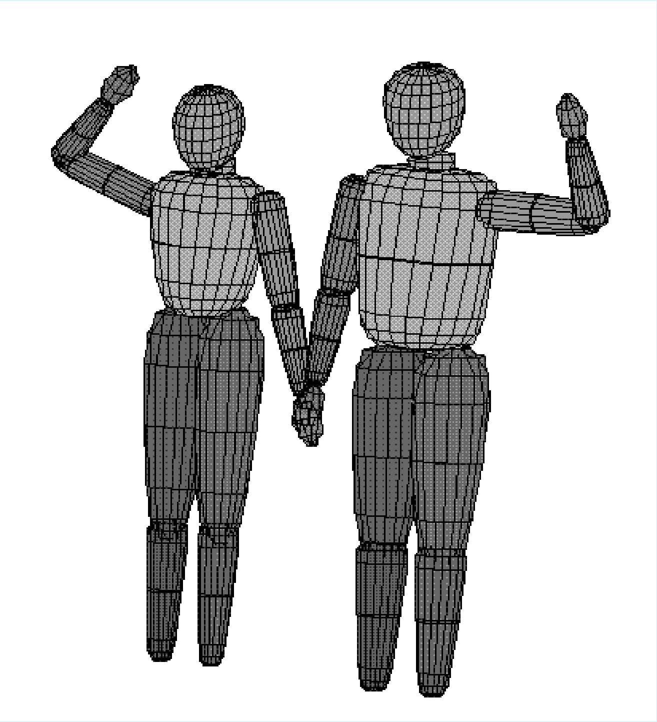
Recognizing facial expressions



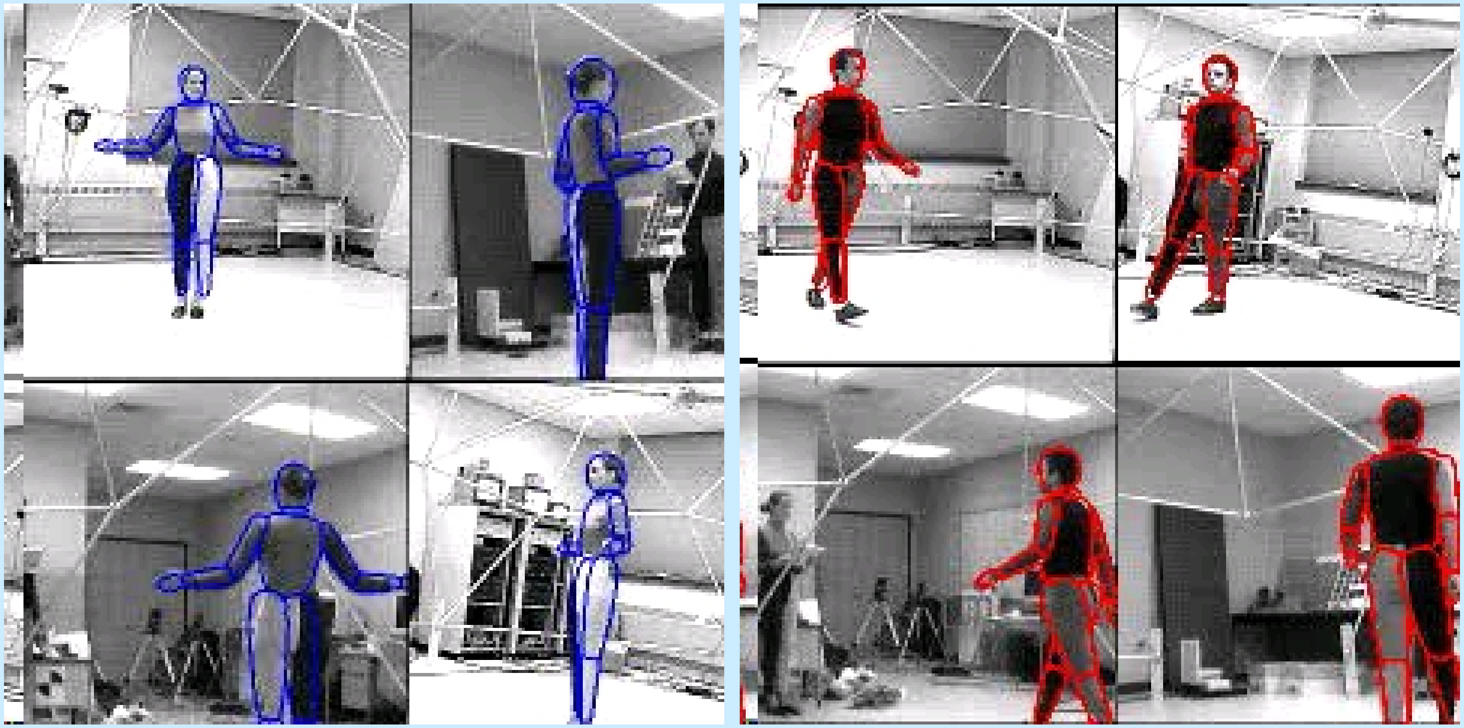
More examples

Multi-camera recovery of 3-D body pose

- Gavrila and Davis
- Match articulated body part model to 4-7 views of a person in motion

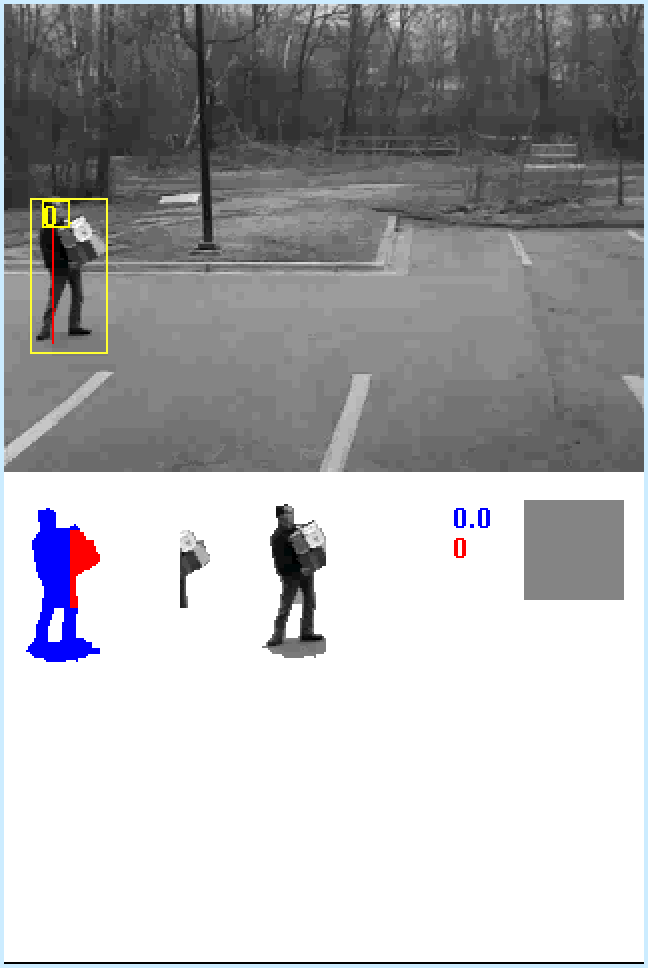


3-D Body Articulation Recovery



Visual Surveillance-Goals

- Detection of moving and fixed objects
- Classification as people, animals, vehicles
- Recognition of specific individuals and vehicles
- Recognition of actions and interactions
  - between people
  - between people and objects



W^4

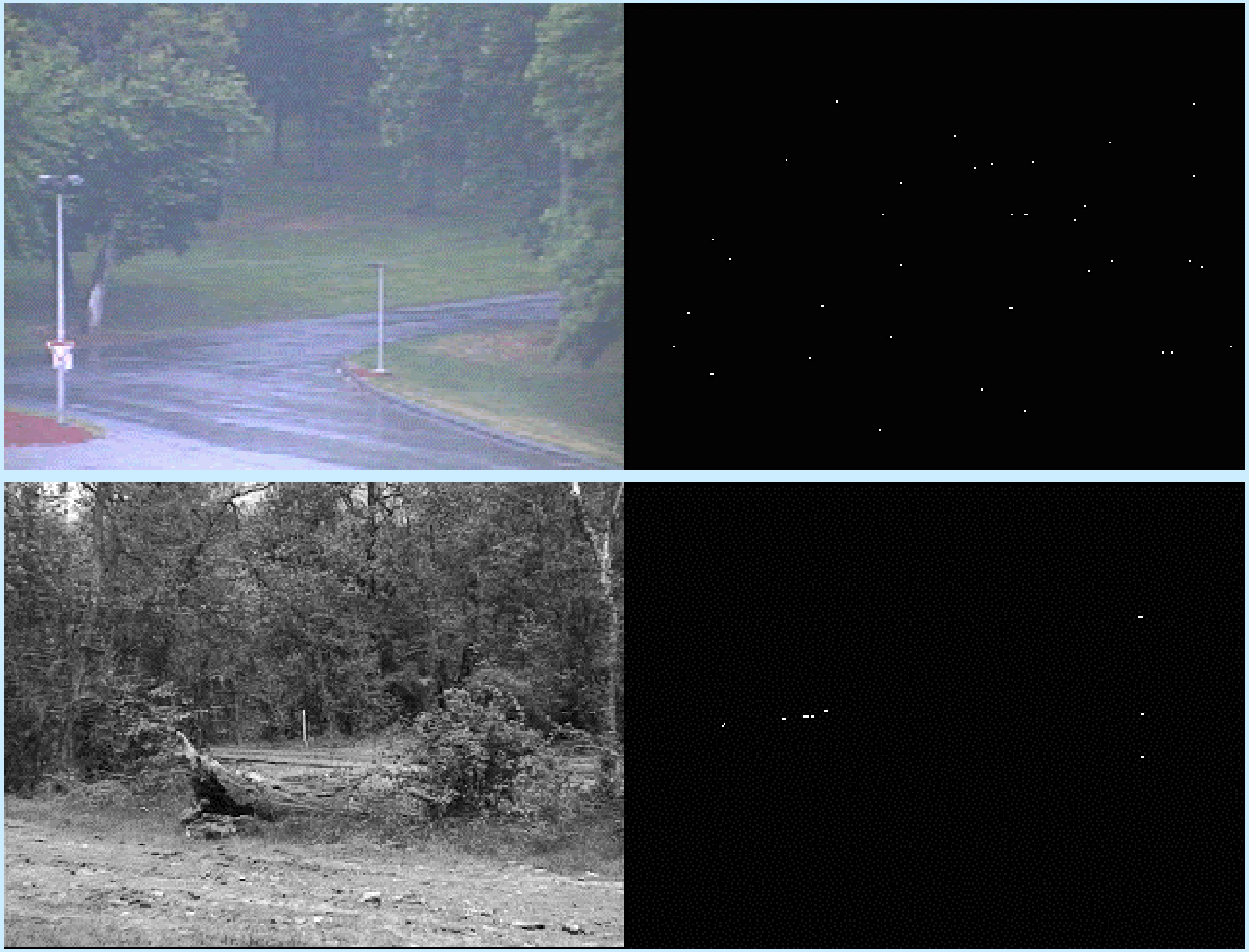
- Detects and tracks people and their body parts
  - Real Time (~15-30 fps)
  - Monochromatic video camera (visible or infrared)
  - Stationary camera with pan/tilt/zoom
  - People can appear in a variety of poses and in small groups
  - Tracks people, recognizes people carrying and exchanging objects
  - No special hardware - dual 450 MHz PC

Detection: Background Modeling

- Sources of Difficulty
  - Camera jitter
  - Motion of background objects

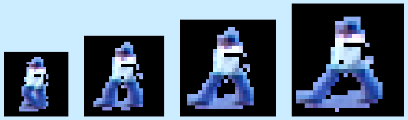
- Model of background variation while the scene contains no people
- Updated during tracking

Background Subtraction Example

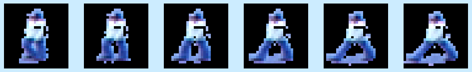


Classification of people using periodic motion

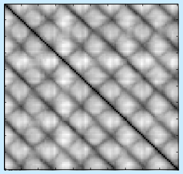
Track objects



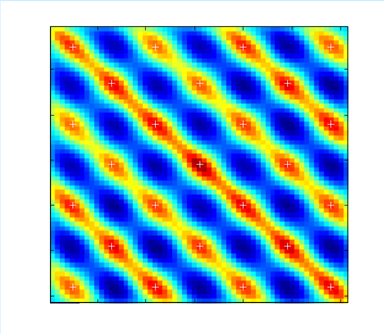
Align and scale objects



Compute similarity matrix *S*

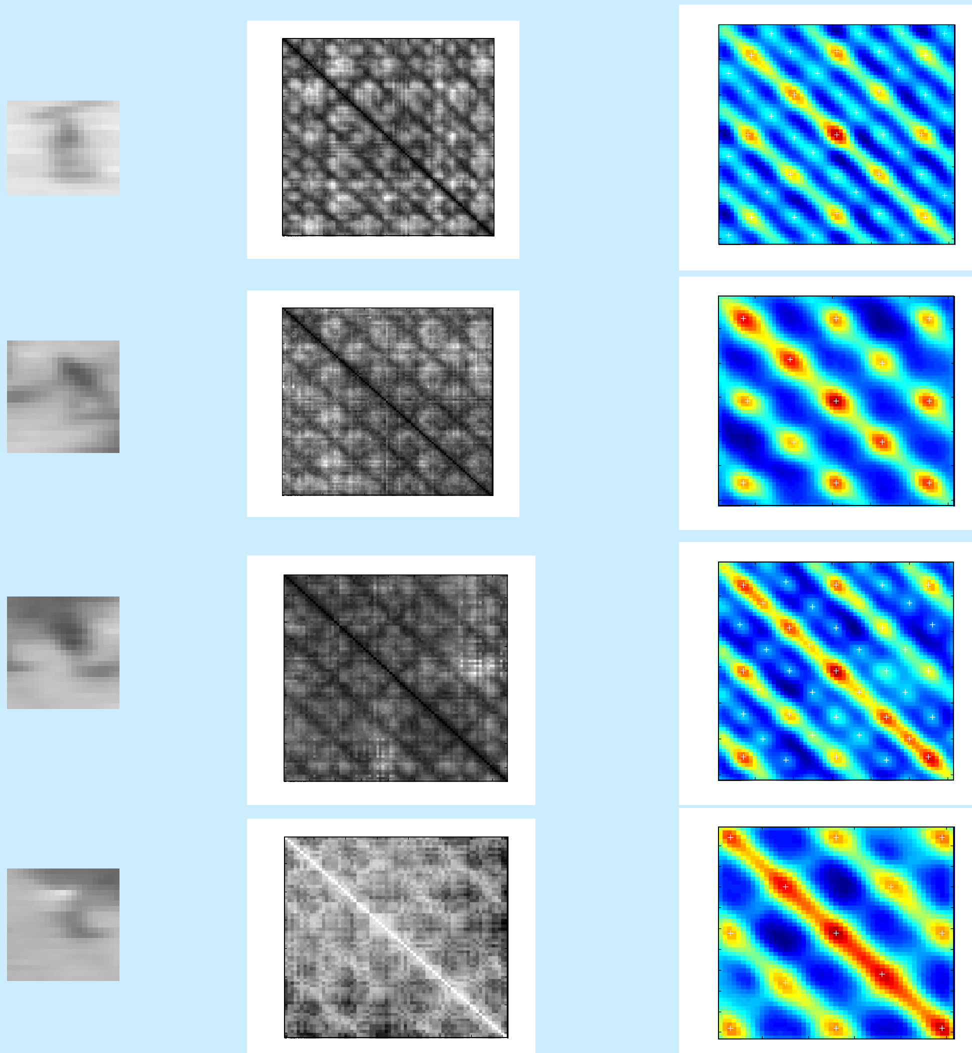


Autocorrelate *S*

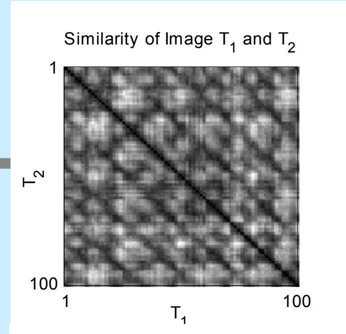


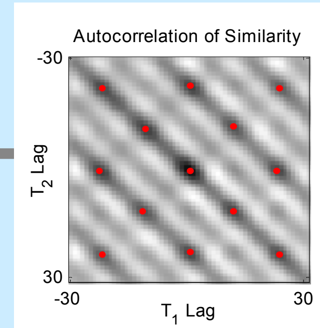
Template fit peaks of *S*

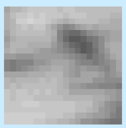
Periodic Motion: People

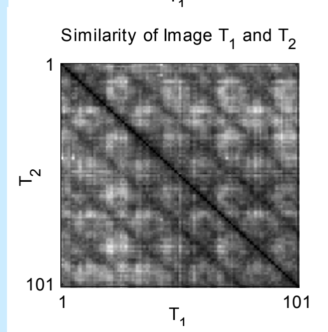


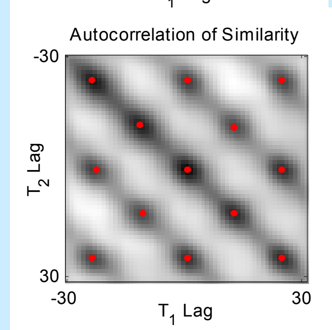
Person Classification

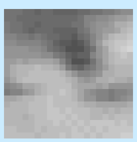


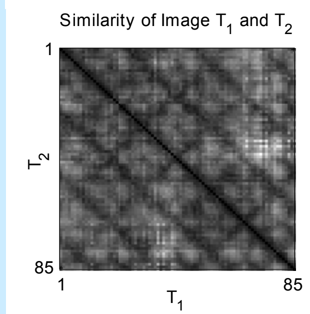


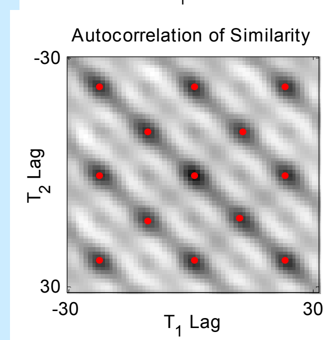


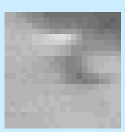


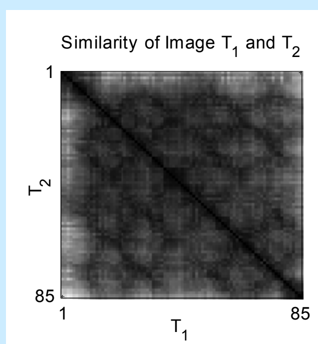


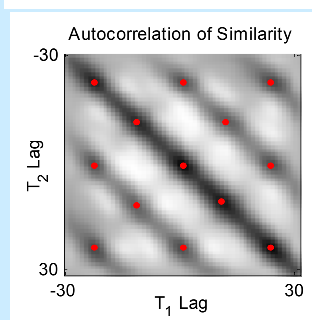




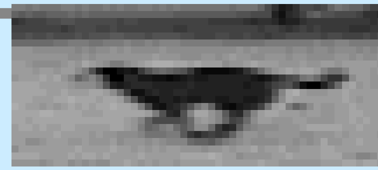


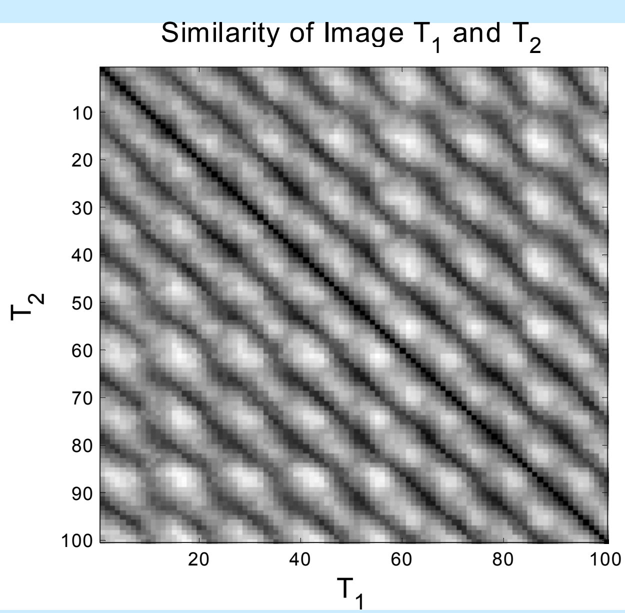


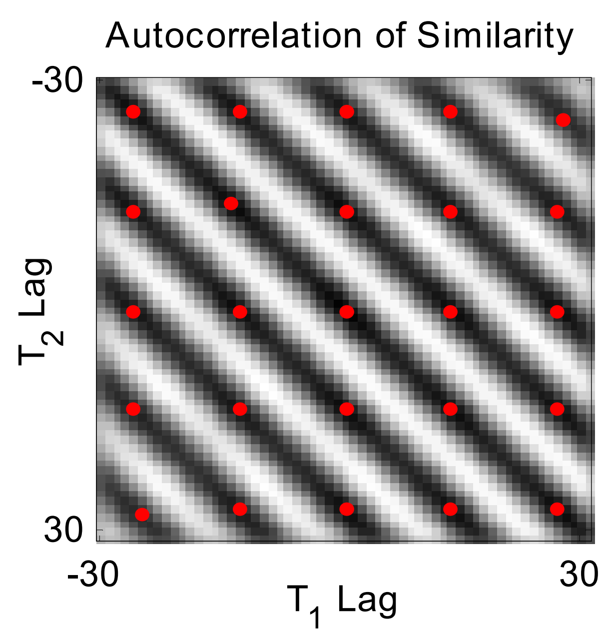




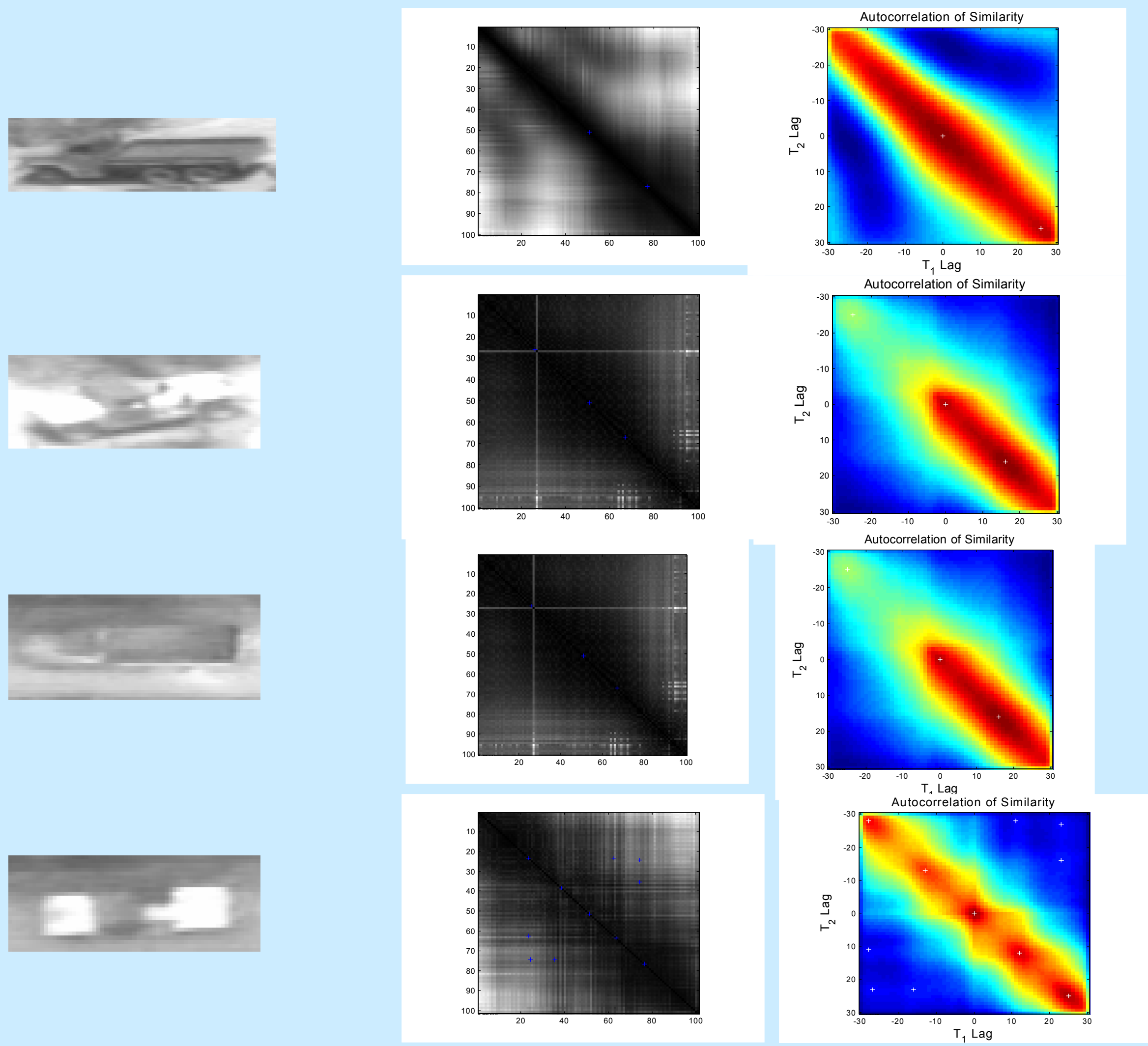
Motion Symmetry of Running Dogs







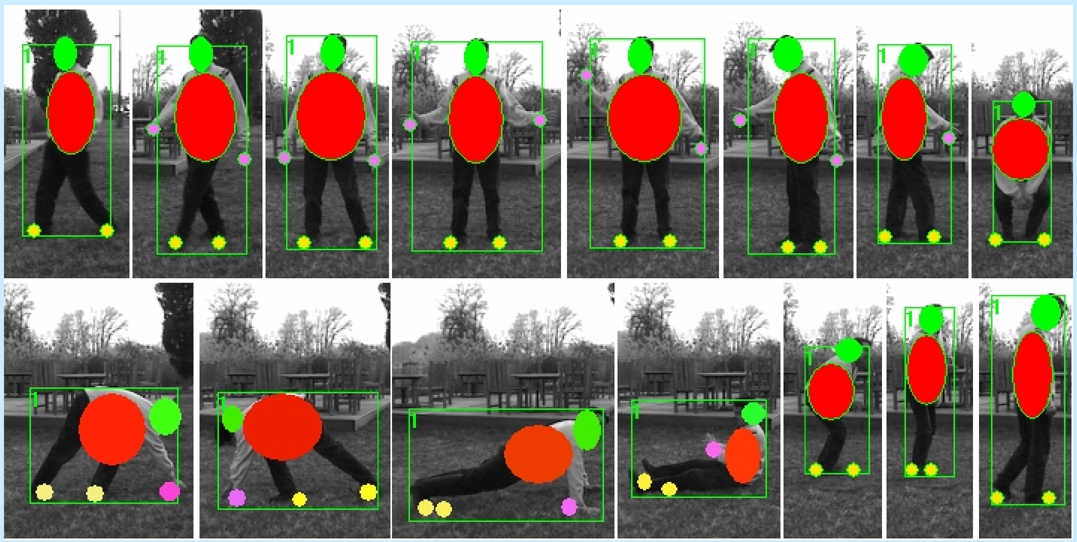
No Periodic Motion: Vehicle

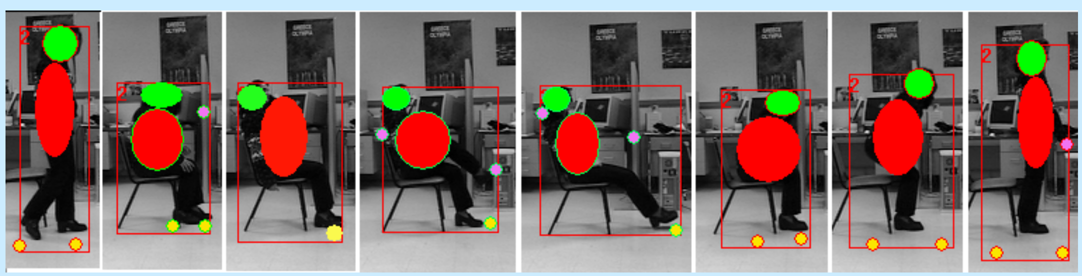


AVS example:
Periodic motion detection

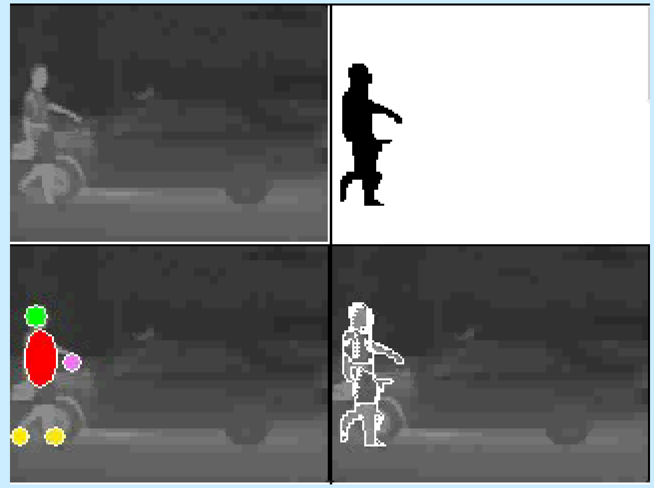
Airborne
Video

Ghost: Body Part Labeling

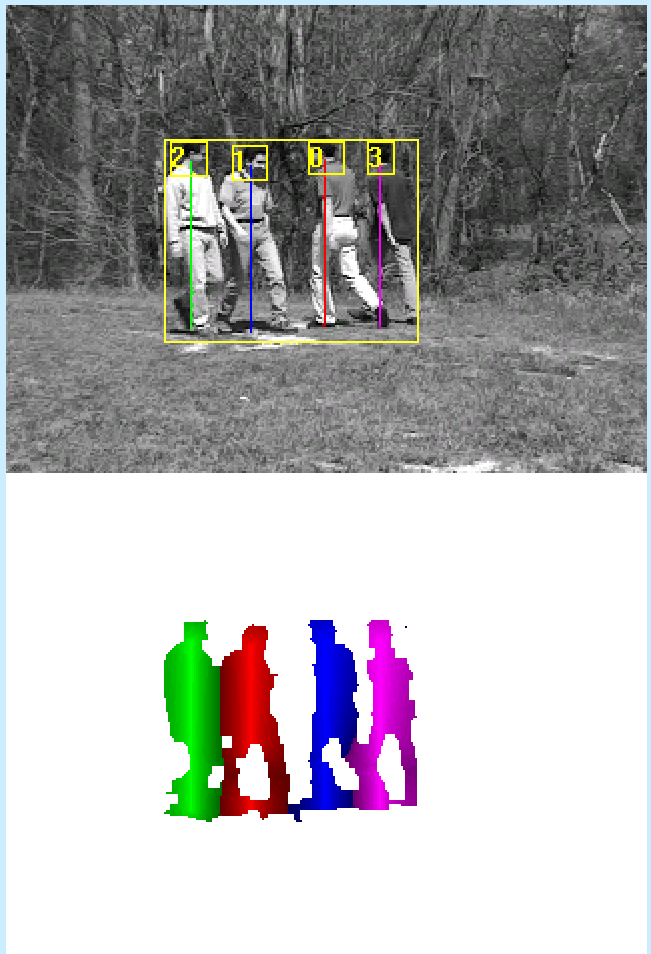




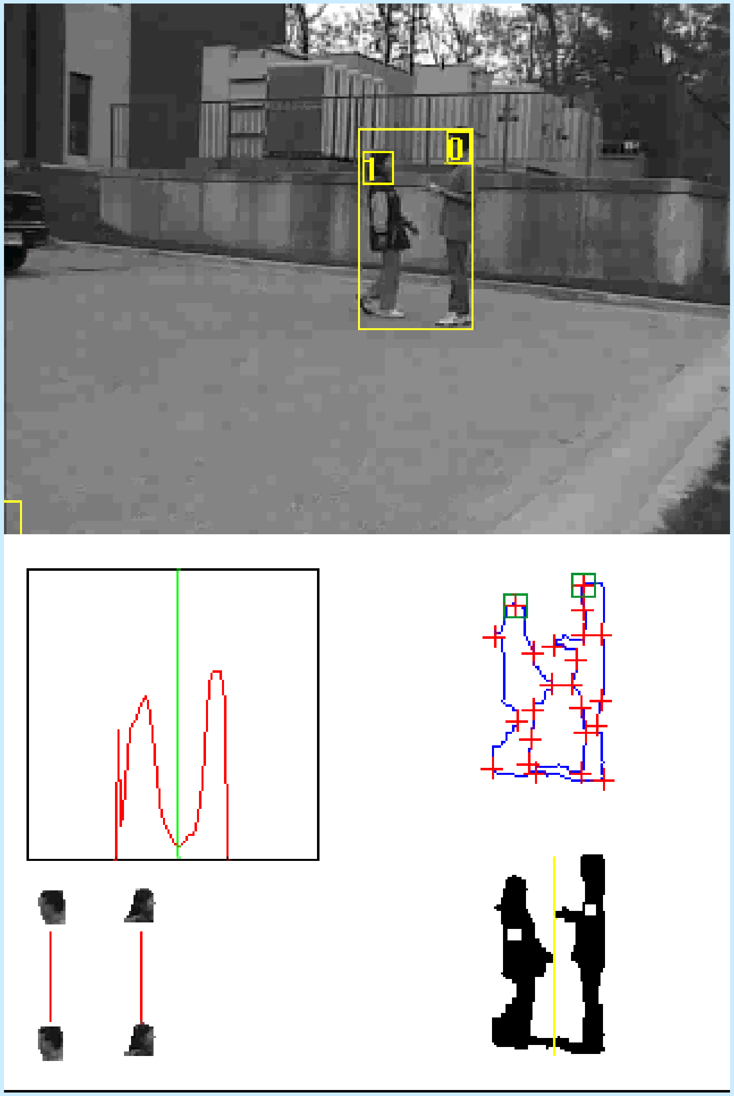
Tracking examples



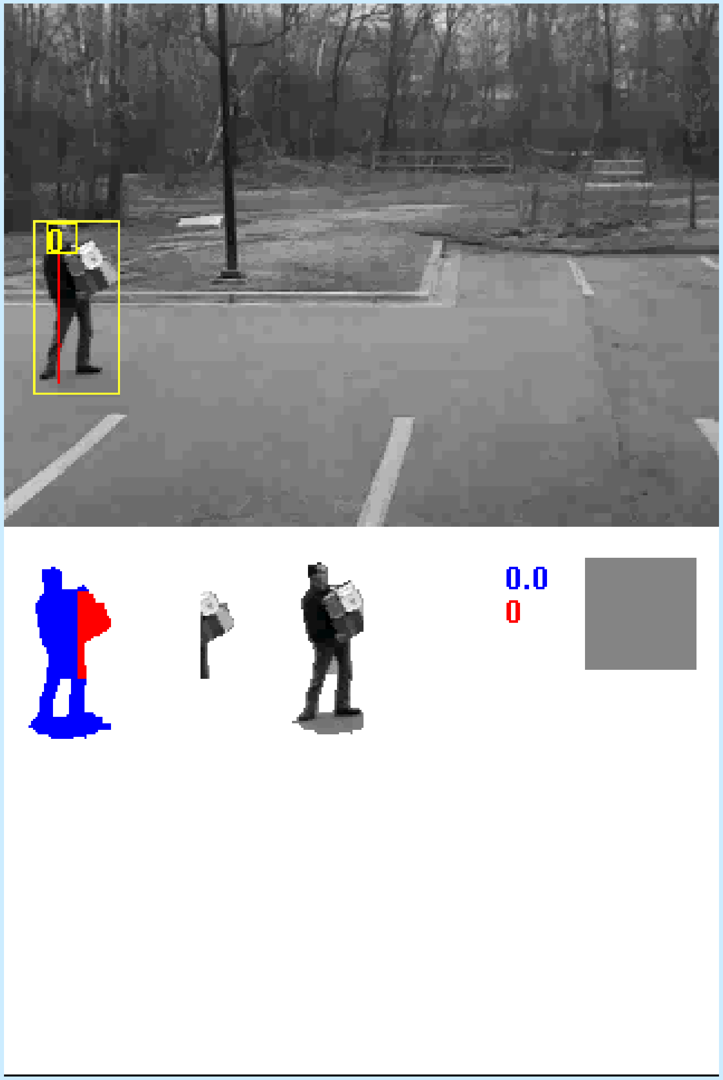
Analyzing small groups



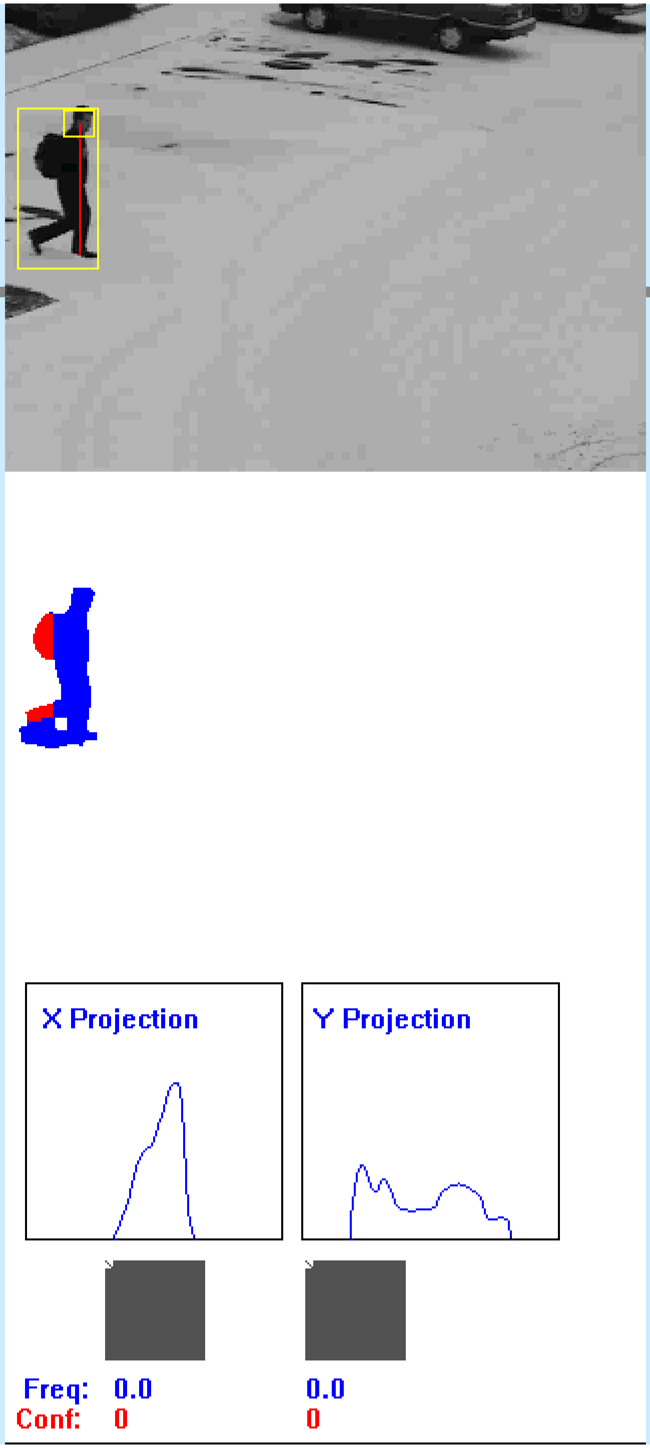
Detailed example



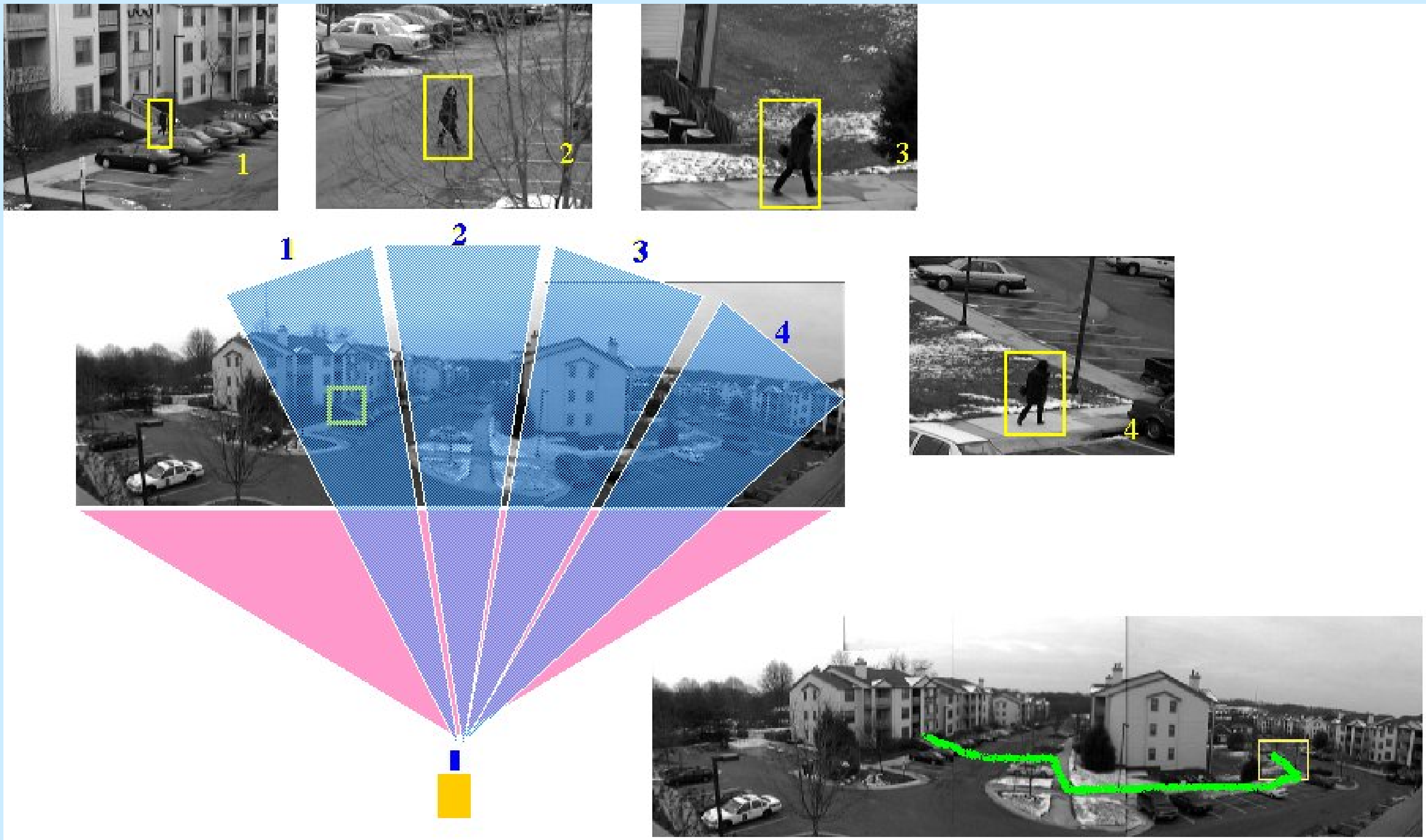
Recognizing interactions between people and objects - carrying and exchanging



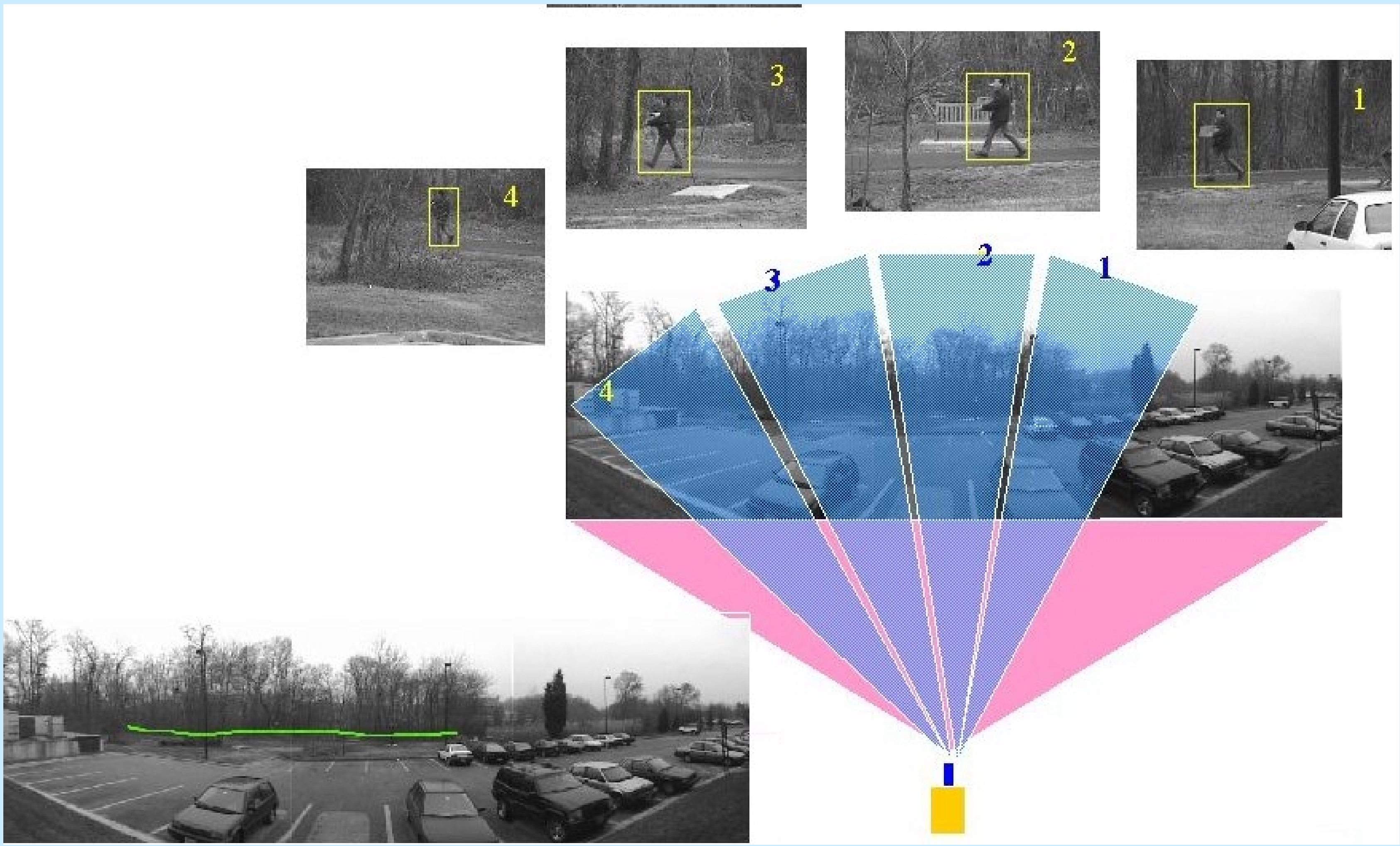
Backpack



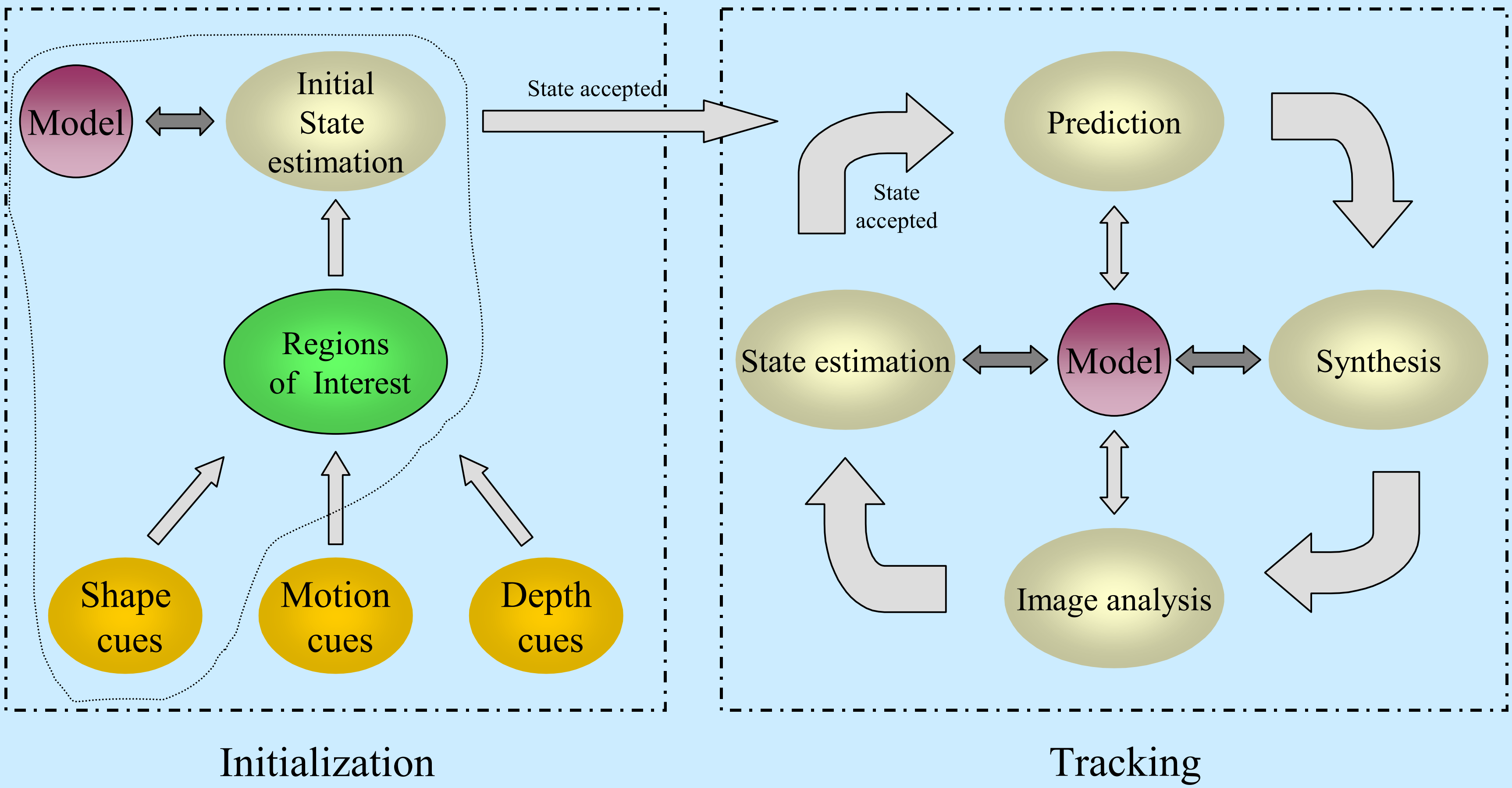
Active tracker



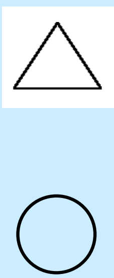
Active tracker

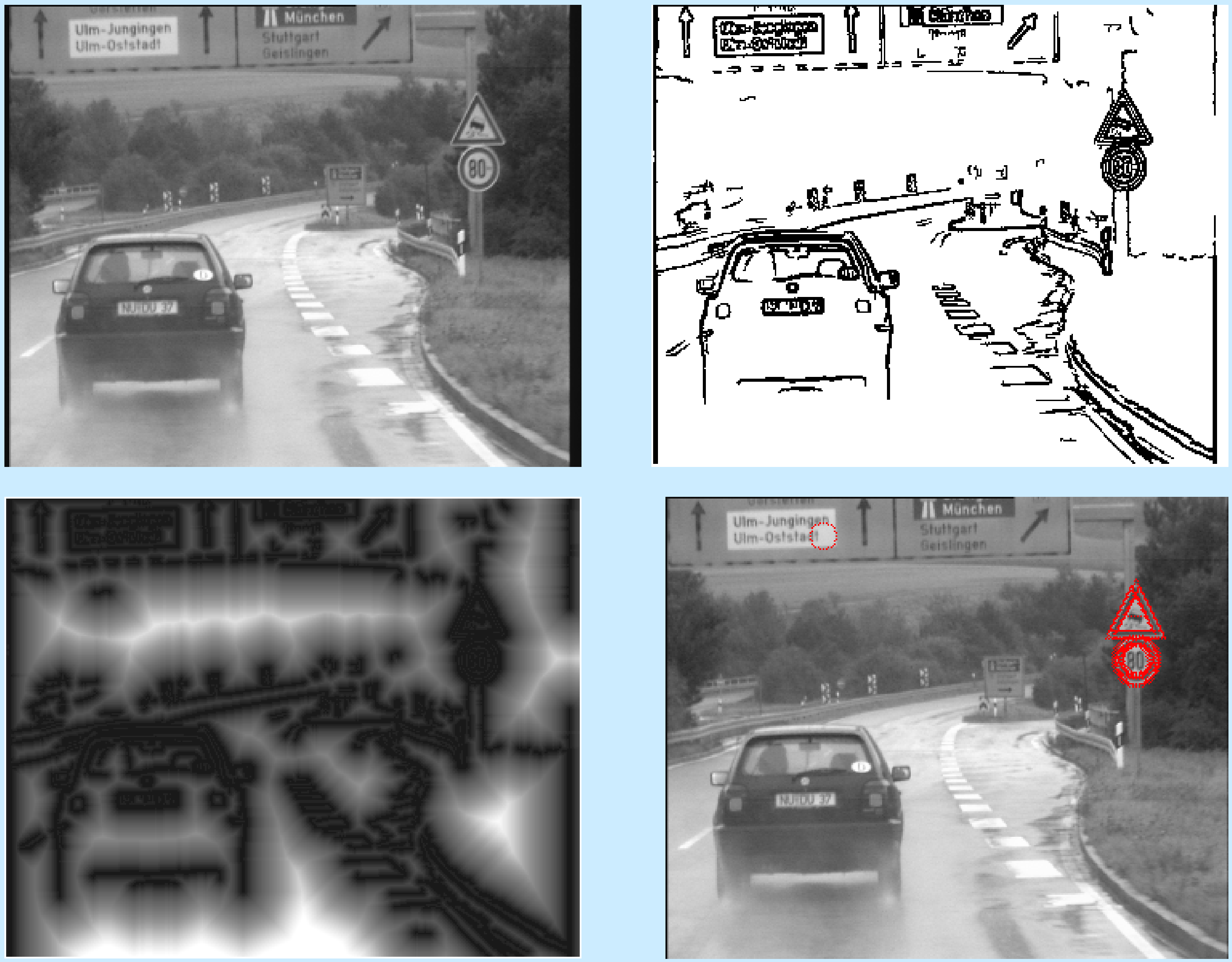


An object detection and tracking framework



DT based matching

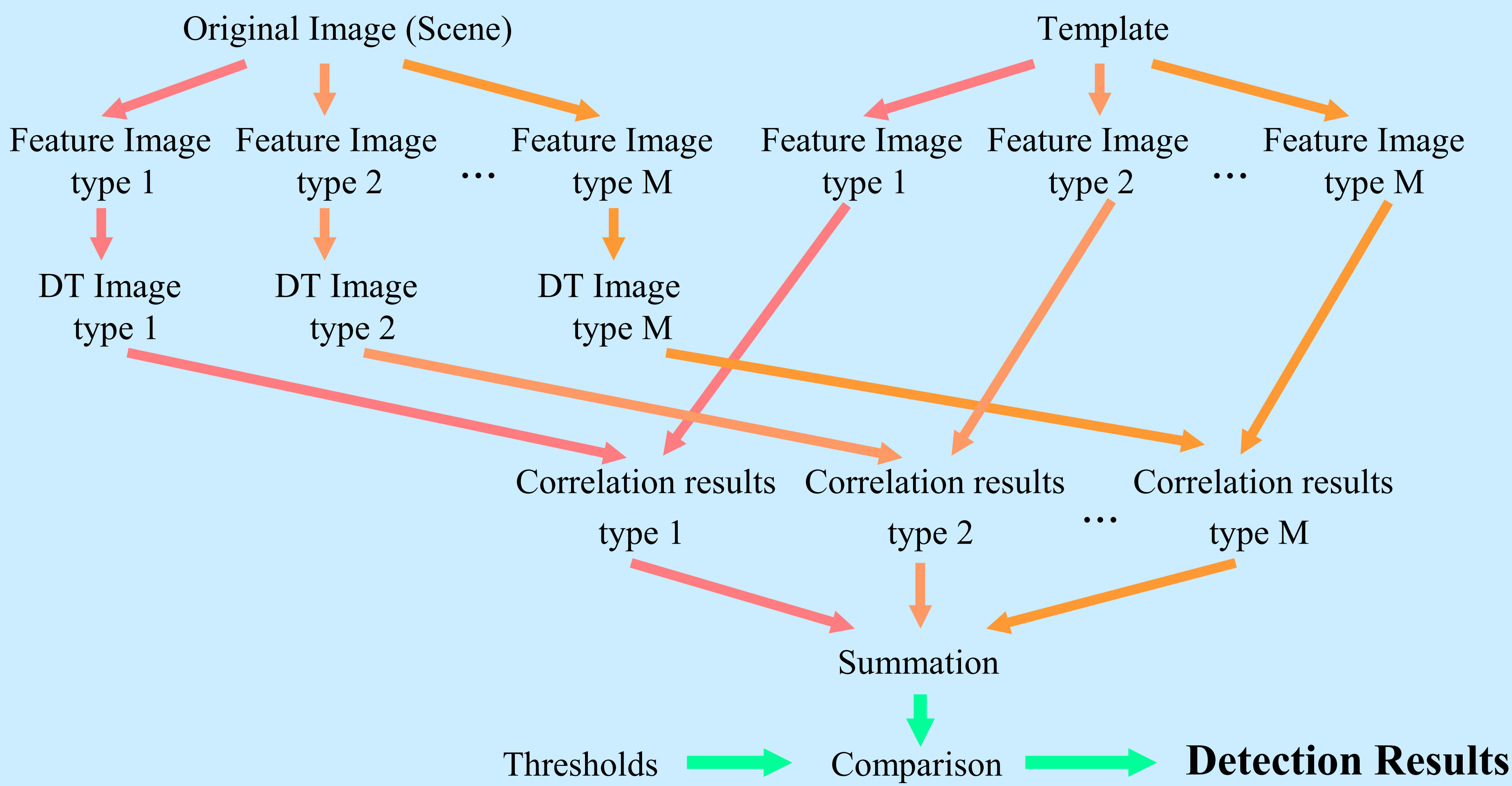




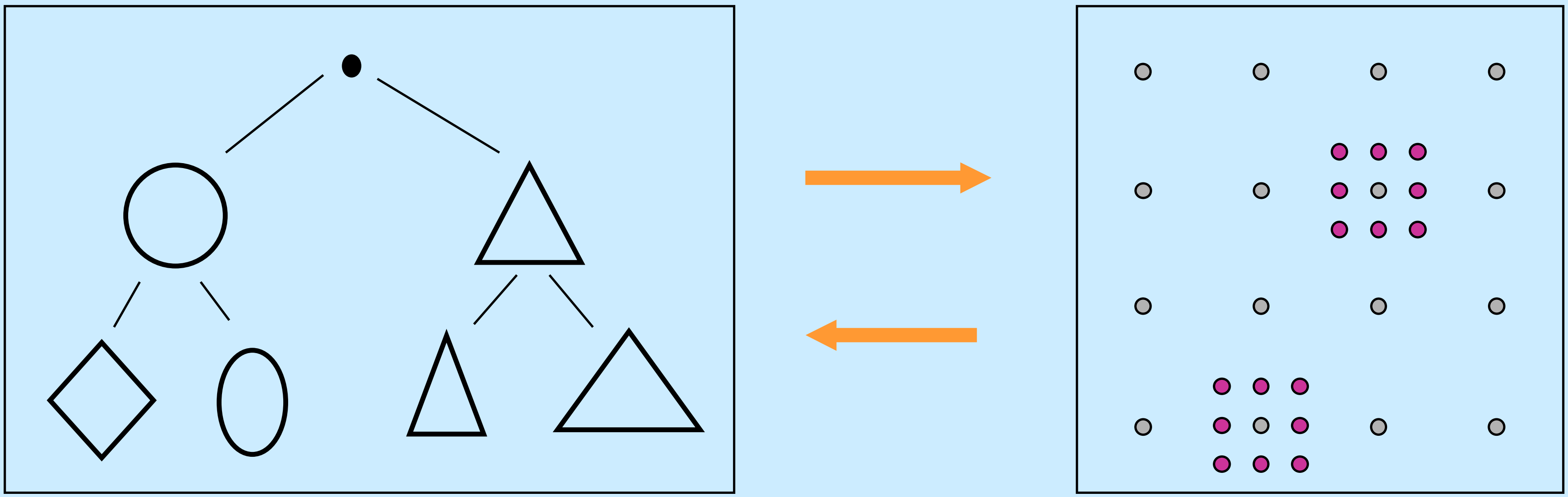
Extensions

- use of **multiple feature types**
- matching **multiple** templates using a **template hierarchy**
- automatically **grouping** templates to construct the hierarchy

Multiple feature types



Multiple templates and template hierarchy



Factors determining the appropriate distance thresholds during matching

- size search grid
- distance of parent template to its children templates
- segmentation errors
- object variability

Grouping of templates into a hierarchy

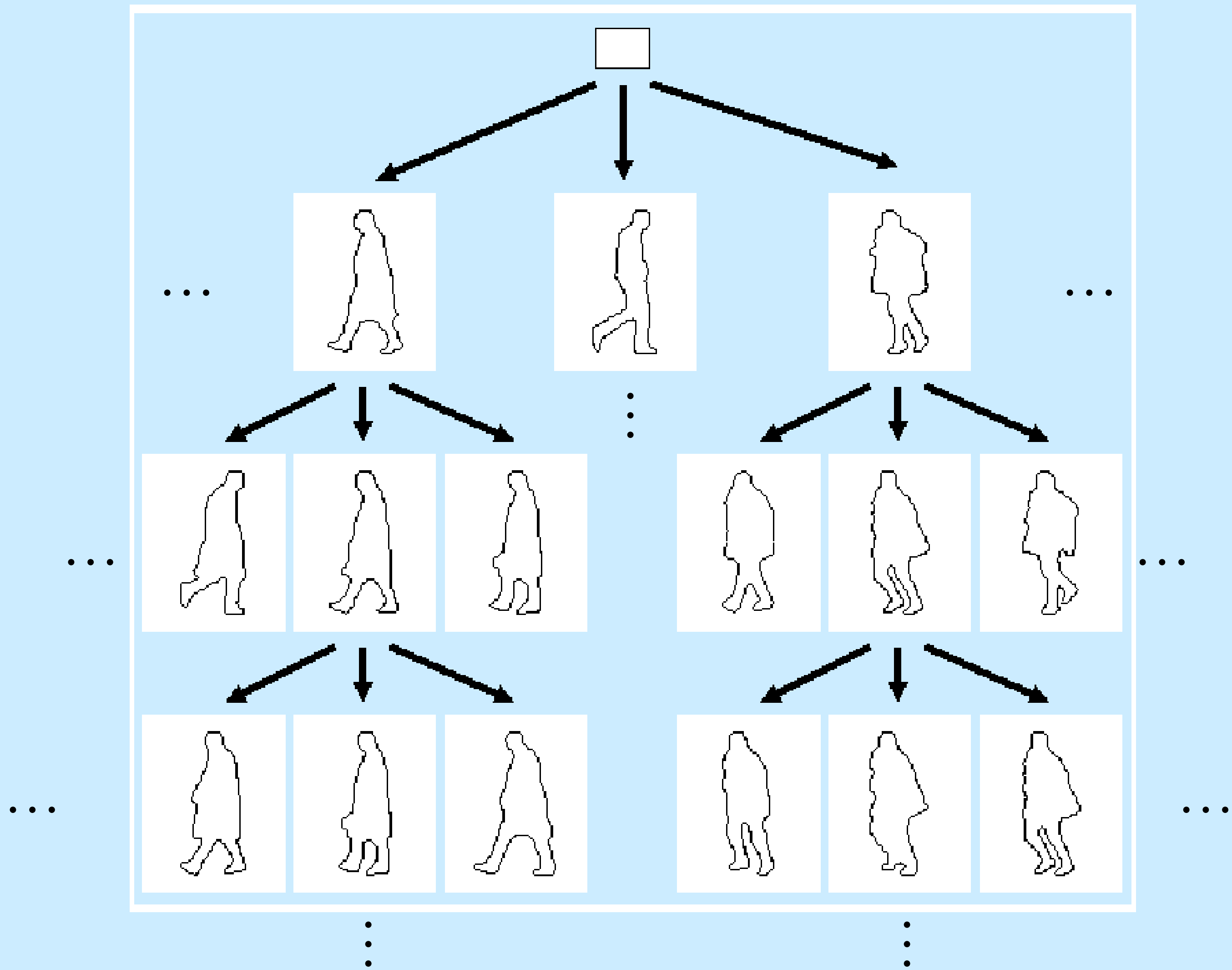
- K-means like clustering algorithm
- Input - Number of clusters K and a set of templates
- Output - K partitions and prototypes for each group
- Compute distance matrix
- Minimize $E = \sum_{k=1}^{K} \sum_{t_i \in S_k} D(t_i, p_k^*)$
- Two passes at every iteration
  - k-means pass
  - forcing pass
- Simulated annealing

Results - Traffic sign detection

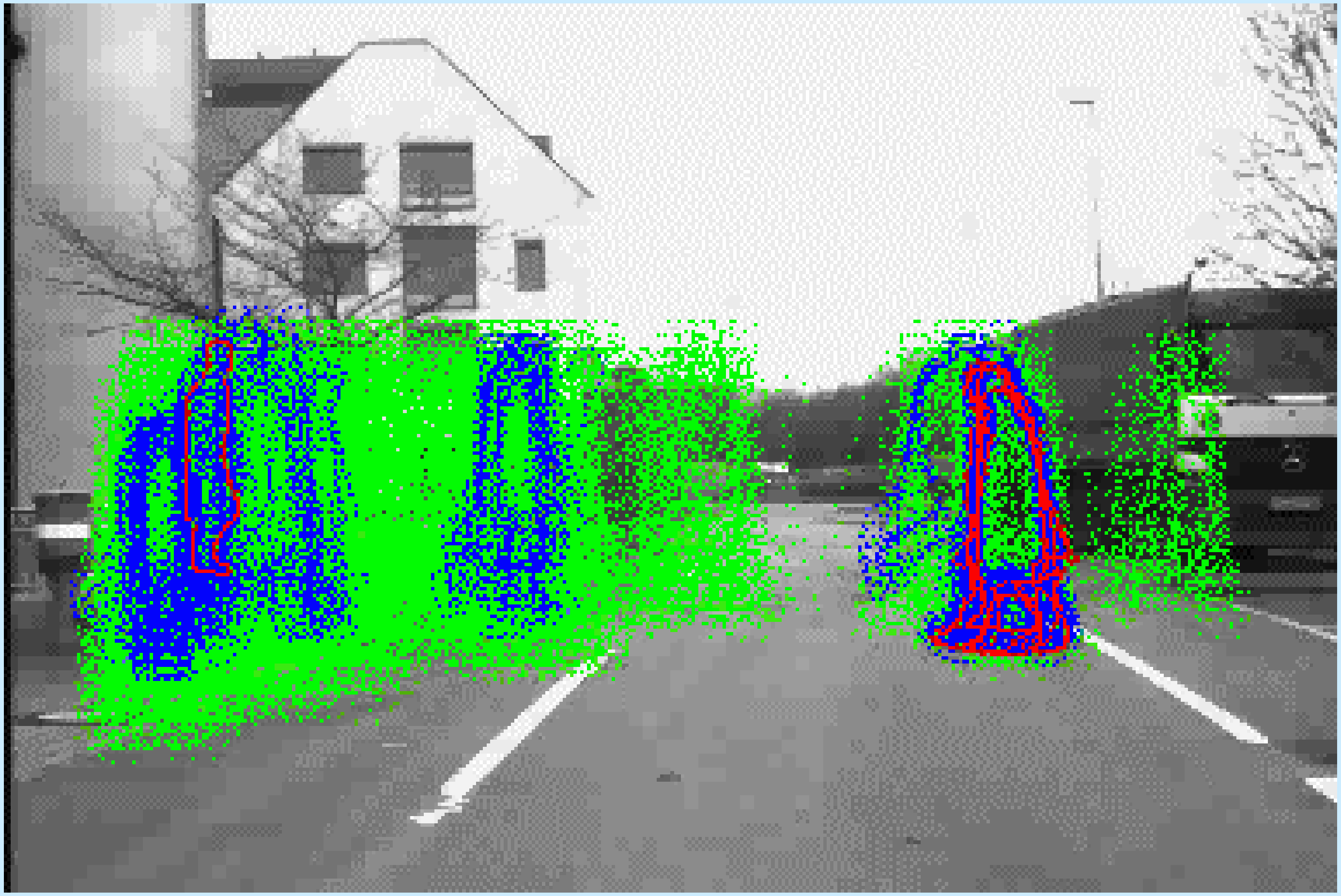
- detection rate > 90 % (single frame)
- false positives < 2 per image
- speed-up factor 200-400 compared to brute-force approach (not including the SIMD implementation)
- 400% increase in speed over standard optimized C code due to SIMD implementation
- processing speed > 11 Hz on dual-Pentium II 333 MHz

Detection results

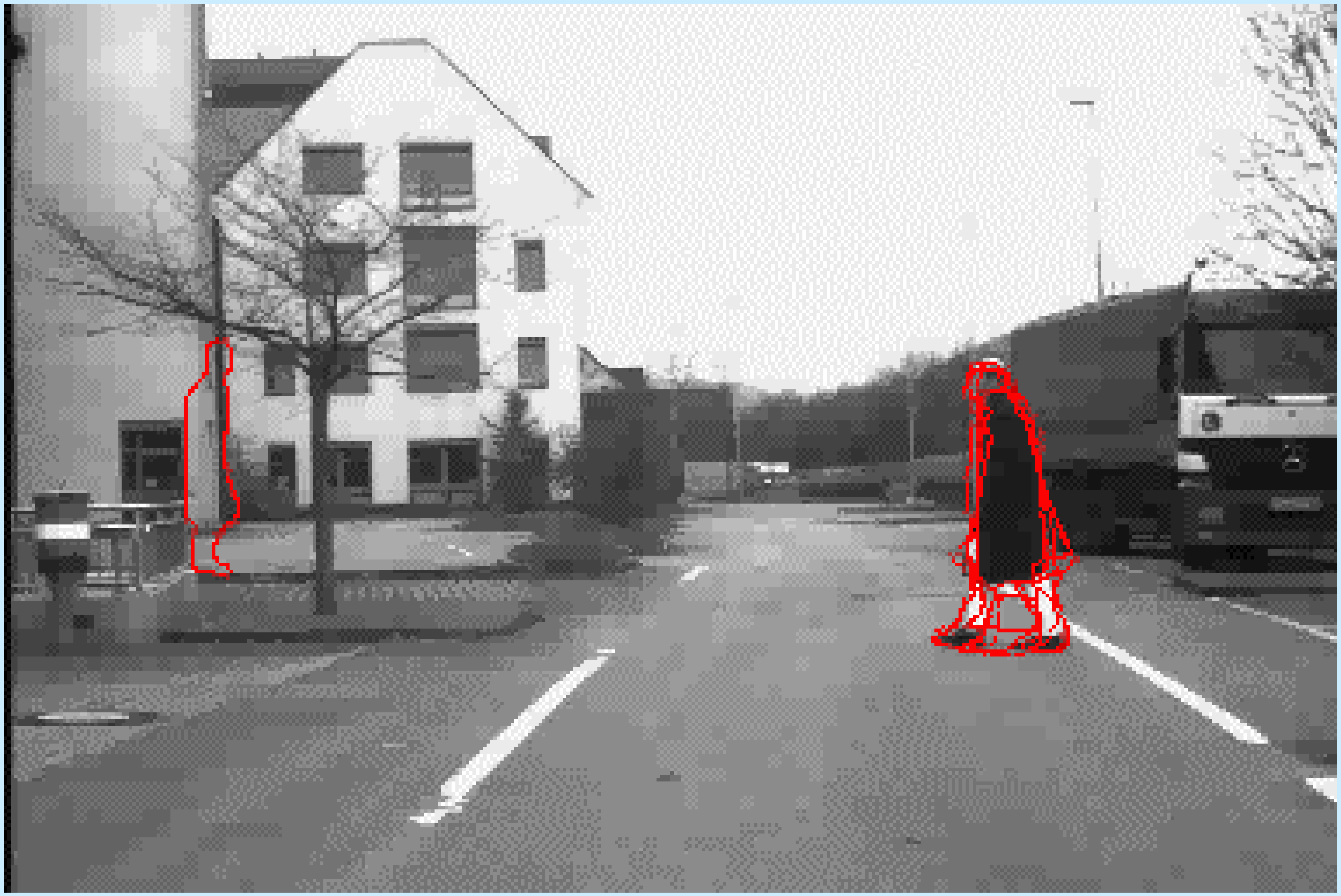
Pedestrian Detection



People detection from static shape models



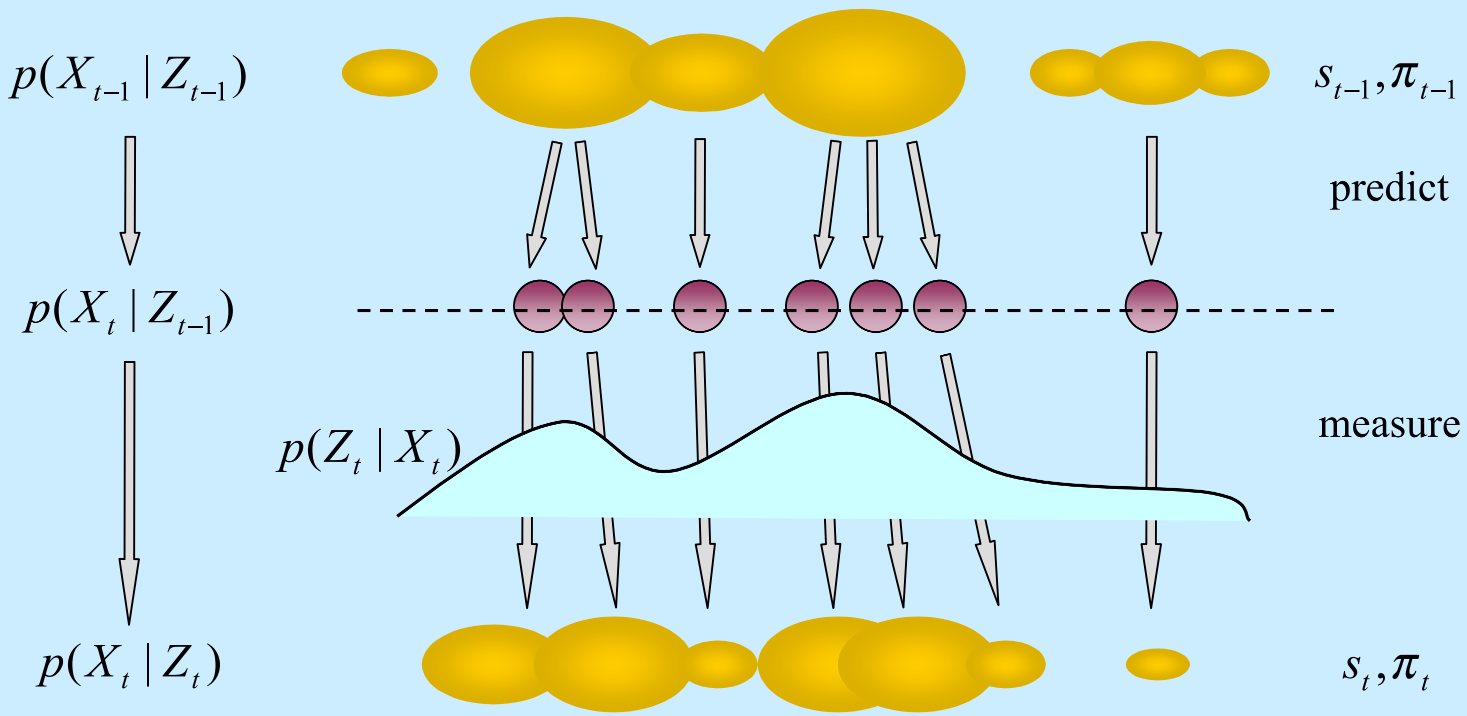
Detecting people from a moving camera



Tracking

- Condensation algorithm
  - Pdf represented by a set of random samples (Monte Carlo approach)
  - Propagate samples (using a motion model as a predictor) and resample
  - Update sample probabilities based on measurements

The Condensation algorithm



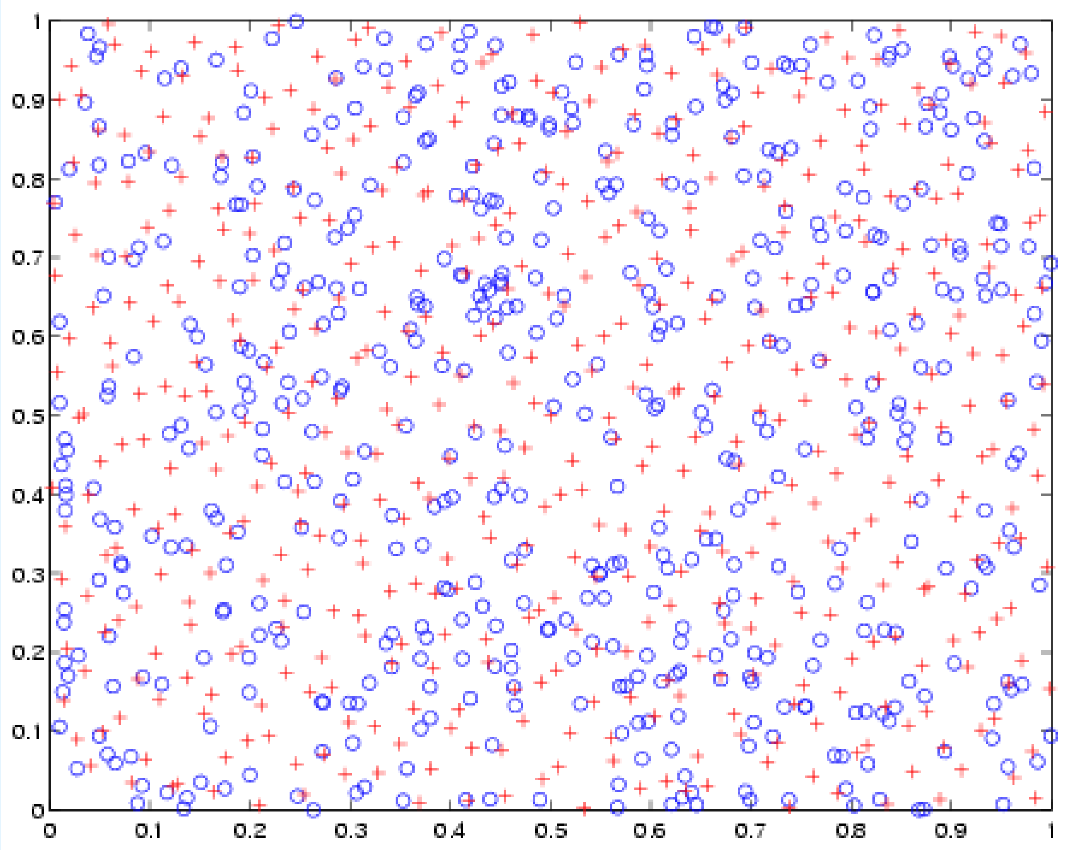
Difficulties

- Learned motion model must be accurate for robust tracking
- Unknown motion model
- High-dimensional state space (4 Euclidean + 8 deformation)
- Sub-optimal and inaccurate sampling
  - Sampling error for N points for a ‘perfect’ pseudo-random generator decreases only as $O(N^{-1/2})$
  - Rand() is not free of sequential correlation on successive calls
  - Modulus operator – least significant bits less random

Proposed Extensions

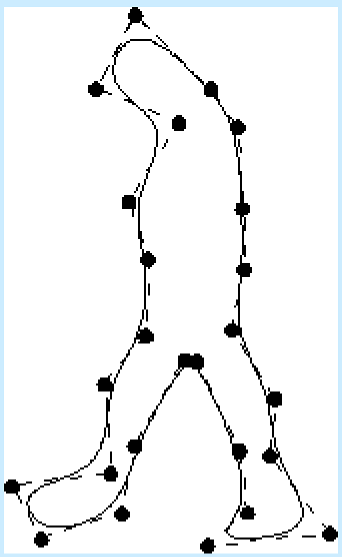
- Quasi-Random sequences
  - Want to pick sample points "at random", yet spread out in some self-avoiding way
  - Sequences of k-tuples that fill k-space more uniformly than pseudo-random points
  - Improve asymptotic complexity of search and well spread in multiple dimensions
  - Sampling error decreases as $O(N^{-1})$ as opposed to $O(N^{-1/2})$ for pseudo-random
- Zero-order motion model with large process noise
  - Sample more densely in the gaussian neighborhood of high probability samples from the previous time step

Pseudo-random vs. Quasi-random points



Gaps left by pseudo-random points are filled in by the quasi-random points

Learning a linear pedestrian model



- Extract a training set of pedestrian contours
- NURB fit to point data $\{Q_k\}$, k=0, …, n using least squares
  - Parameterize the curve using the centripetal method

$$\bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{\sum_{k=1}^{n} \sqrt{|Q_k - Q_{k-1}|}} \quad k = 1, \ldots, n-1$$

  - Solve for the control points P_i from $Q_k = \sum_{i=0}^{n} N_{i,p}(\bar{u}_k) P_i$
- Represent each shape by the shape vector consisting of the control points P_i (“landmark” points in PDM)
- Align the training shapes using Weighted Generalized Procrustes Analysis (more significance to stable landmark points)
- Use PCA to find the modes of variation

Pedestrian tracking

Sample with maximum probability

Mean estimate of the posterior

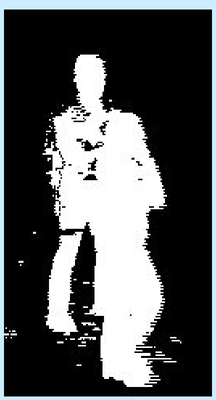
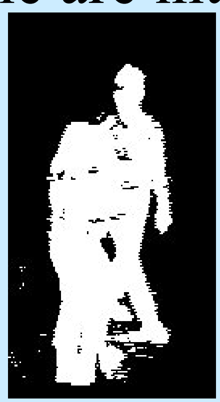
Surveillance

Modal state (maximum probability)

Mean estimate of the posterior

Probabilistic Framework for Segmenting People Under Occlusion

- Motivation:
  - What people do while they are interacting is essential for surveillance systems.
  - Do not want to lose targets when they are partially occluded by other people.
- Objective:
  - Build representation of different people when they are isolated that enables the **segmentation of foreground regions** when people are interacting as a group.


- Assumptions:
  - People are isolated before the occlusion (so can a representation can be created for each one).
  - Foreground regions are detected.
- Approach:
  - Model the color of the major parts of the body (torso, bottom, head).
  - Localize the color features with respect to the person.

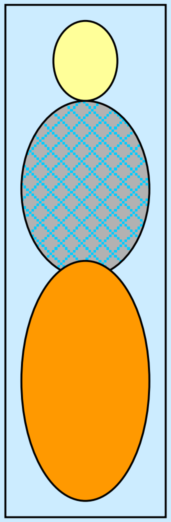
Representation

- Model the person as a vertical set of blobs.

$$M=\{A_i\}$$

- Each blob has the same color distribution everywhere inside the blob. (color distribution is independent of the location within the blob) i.e.,

$$h_A(c \mid x,y) = h_A(c)$$



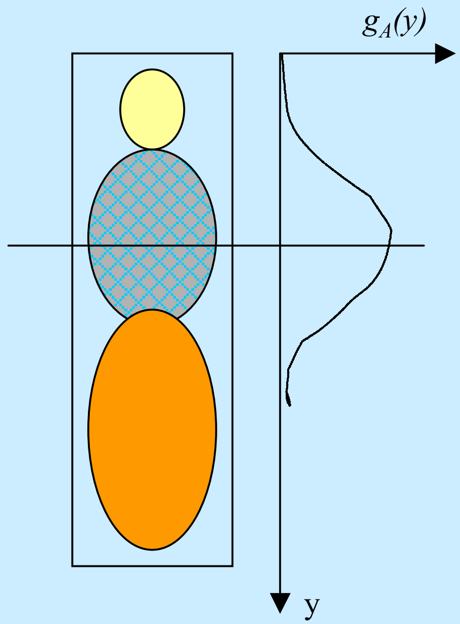
Representation

- The vertical location of each blob w.r.t. the person is independent of the horizontal location.

$$g_A(y \mid x) = g_A(y)$$

$\Rightarrow$The joint distribution within the blob:

$$P_A(x,y,c)=f_A(x)\, g_A(y)\, h_A(c)$$



- Given *M*, the probability of color *c* at location *x,y* is:

$$P(x,y,c|M) = \frac{f(x)}{C(y)} \sum_i g_{A_i}(y) \cdot h_{A_i}(c)$$

Where $C(y) = \sum_i g_{A_i}(y)$

- If the Model origin moves to (x_o, y_o) , then

$$P(x,y,c|M(x_o,y_o)) = \frac{f(x-x_o)}{C(y-y_o)} \sum_i g_{A_i}(y-y_o) \cdot h_{A_i}(c)$$

- Three blobs: Head, Torso & Bottom.
  $M=\{H,T,B\} \Rightarrow$

$$P(x,y,u|M)=\frac{f(x)}{C(y)}\left(g_H(y)\cdot h_H(c)+g_T(y)\cdot h_T(c)+g_B(y)\cdot h_B(c)\right)$$

- To discriminate between blobs:

$$P(x,y,u|H)\propto(g_H(y)\cdot h_H(c))$$

$$P(x,y,u|T)\propto(g_T(y)\cdot h_T(c))$$

$$P(x,y,u|B)\propto(g_B(y)\cdot h_B(c))$$

Segmentation under Occlusion

- Given 2 Models M_1, M_2
- Hypothesis:
  - Person 1 origin (x_1, y_1)
  - Person 2 origin (x_2, y_2)

For each Foreground pixel $X_i = (x_i, y_i, c_i)$ use maximum Likelihood classification:

$$X_i \in M_k \; s.t. \; k = arg_k \, max \, P(X_i | M_k)$$

Segmentation under Occlusion

- Each choice (x_1, y_1, x_2, y_2) represents a classification surface between two classes.
- Optimal solution: minimize Bayes error
- Generally, for N persons we have a search problem in $2N$ dim
- Exhaustive search will require $O(w^{2N})$

$\Rightarrow$ *Not Practical…*

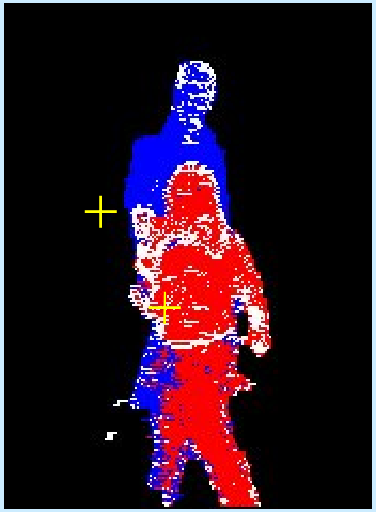
Practical Solution

- Look for a good choice for (x_1,y_1,x_2,y_2)
- Use an origin that is always detectable in a robust way. (e.g. Top of the head)

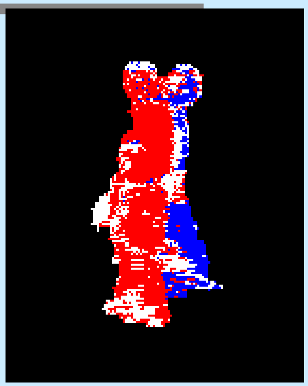
For each new frame t

1. Given origin location (x_i^{t-1},y_i^{t-1}) at frame t-1
2. Classify each pixel using $P(X|M\ (x_i^{t-1},y_i^{t-1}))$
3. Detect new origin location (x_i^t,y_i^t)

Iterations through 2,3 might lead to a better solution.

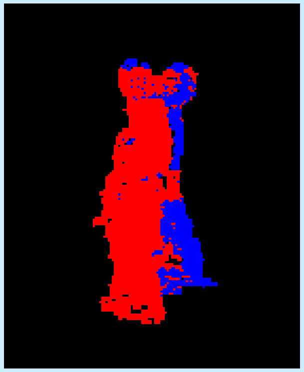


Labeling



- Misclassifications are common at very low likelihood probabilities.
- Consider only strong probabilities:

$X_i \in M_k$ *s.t.* $k = \arg_k \max \; (P(X_i|M_k) > th)$



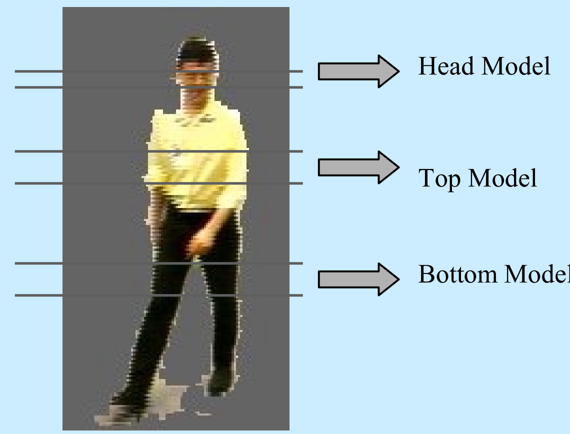
- Fill in with lower probability pixels.

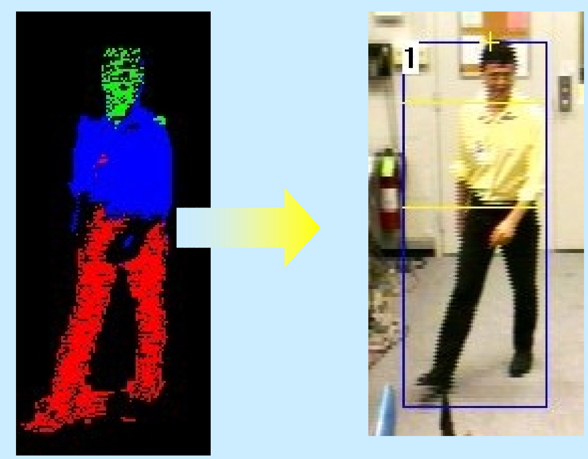
(Spatial localization constraint)

Learning

Learning Color distribution $h_A(c)$

- Initialize blob model with regions at relative locations of the person.
- Classify the whole FG area accordingly.
- Determine blob separators that minimize the misclassifications.
- Recapture blob models.
- Re-segment at each new frame to determine blob separators.

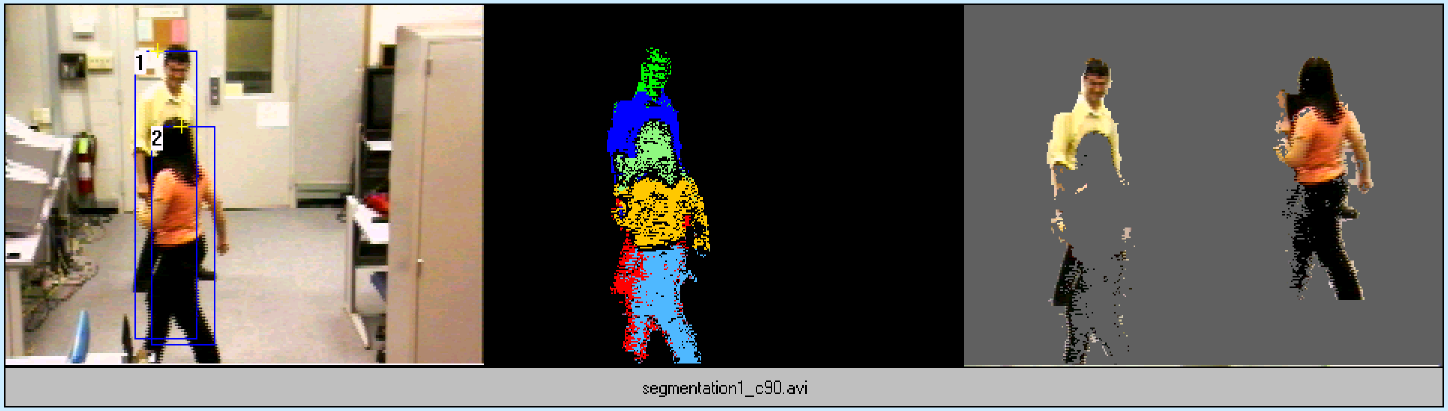




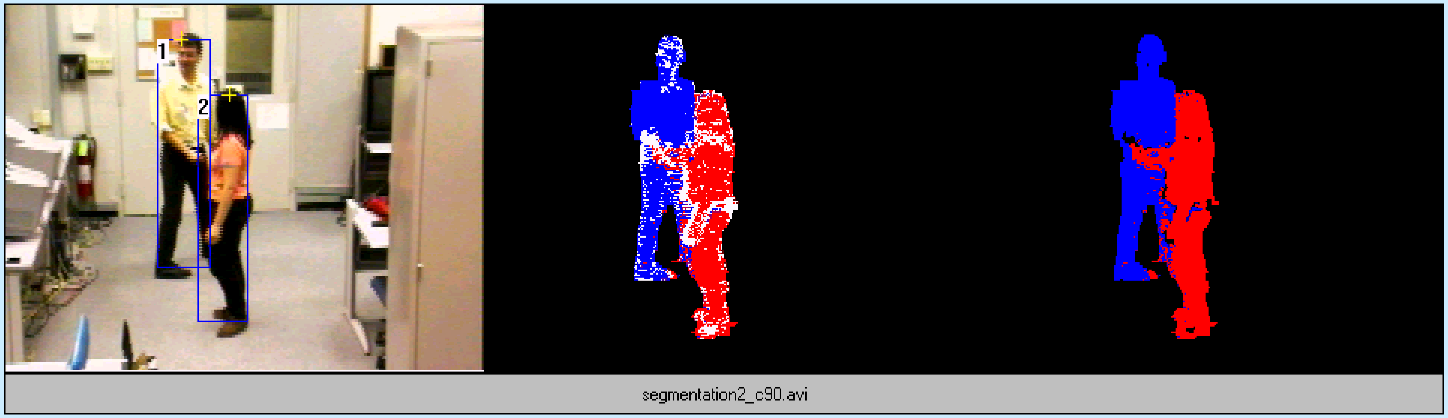
Learning

- Learning Vertical Density $g_A(y)$
  - For each new frame find the histogram of detected blob pixels $H_t(y)$
  - Update density: $g_t(y) = (1-\alpha)\, g_{t-1}(y) + \alpha\, H_t(y)$
  - Align densities using a robust feature (we use torso-bottom separator)
- Horizontal Density $f(x)$
  - Assume Normal density.
  - Fit $N(\mu,\sigma)$ to the person detected pixels

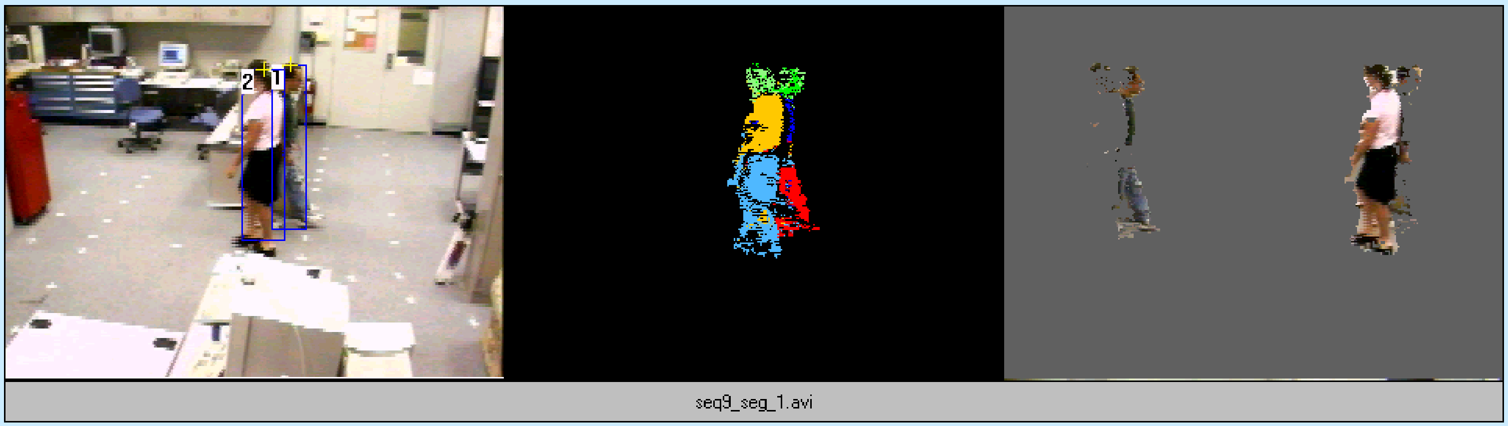
Results



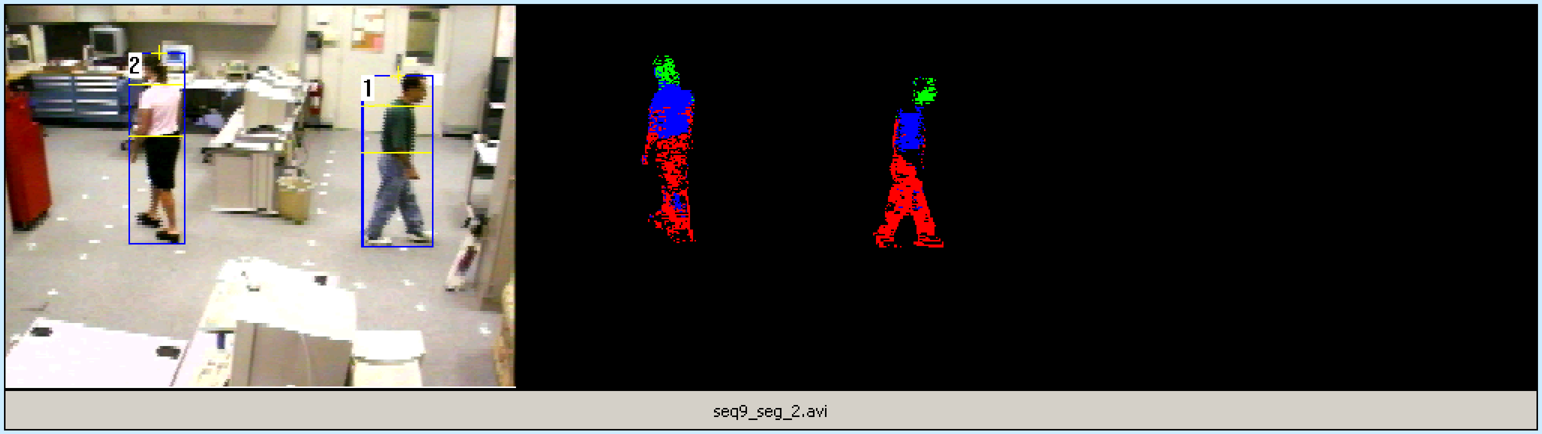
Results



Results



Results



Acknowledgements

- W^4: Ismail Haritaoglu, David Harwood
- Periodic motion analysis: Ross Cutler
- Kernel estimation methods for background modeling: Ahmed Elgammel, David Harwood
- Detection and tracking: Vasanth Philomin, Dariu Gavrila, Ramani Duraiswami