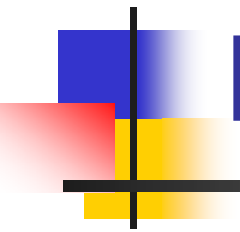
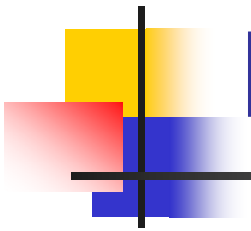


Multi-Camera Tracking

CS 428 Presentation

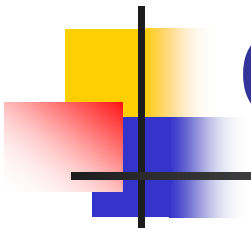
Man-Cho Anthony So

May 29, 2003



Problem Setting

- A (possibly variable) number of people in some environment
- A system of cameras and/or sensors
- Goals:
  - Detect and track the people in the environment
  - Estimate the number of people in the environment

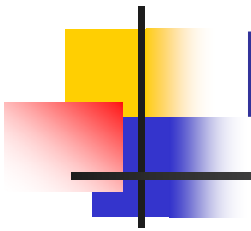


Common Issues

- Distinguish the desired objects from the background
- Combine the information obtained from the system of cameras/sensors
- Determine the value (or possible values) of some attribute (e.g. Where is person A? What is the number of people in the room?)

Detecting and Tracking

- Q. Cai and J. K. Aggarwal, "*Tracking Human Motion Using Multiple Cameras",* Proc. 13th Intl. Conf. on Pattern Recognition, 68-72, 1996.
- Anurag Mittal and Larry Davis, "*Unified Multi-Camera Detection and Tracking Using Region-Matching",* IEEE Workshop on Multi-Object Tracking, 2001.



Preprocessing

- A crucial component of any detection and tracking system is to separate human subjects from the background
- The basic procedures are
  - determine the background
  - determine the non-background objects by differencing and/or grouping pixels with similar intensities or color characteristics

Preprocessing (Con't)

Figure 1: Eight images from a 16-perspective sequence at a particular time instant.

Figure 2: The images from Figure 1, background-subtracted and segmented. The segments are colored randomly and the background is black.

Basic Tracking Procedure [CA96]

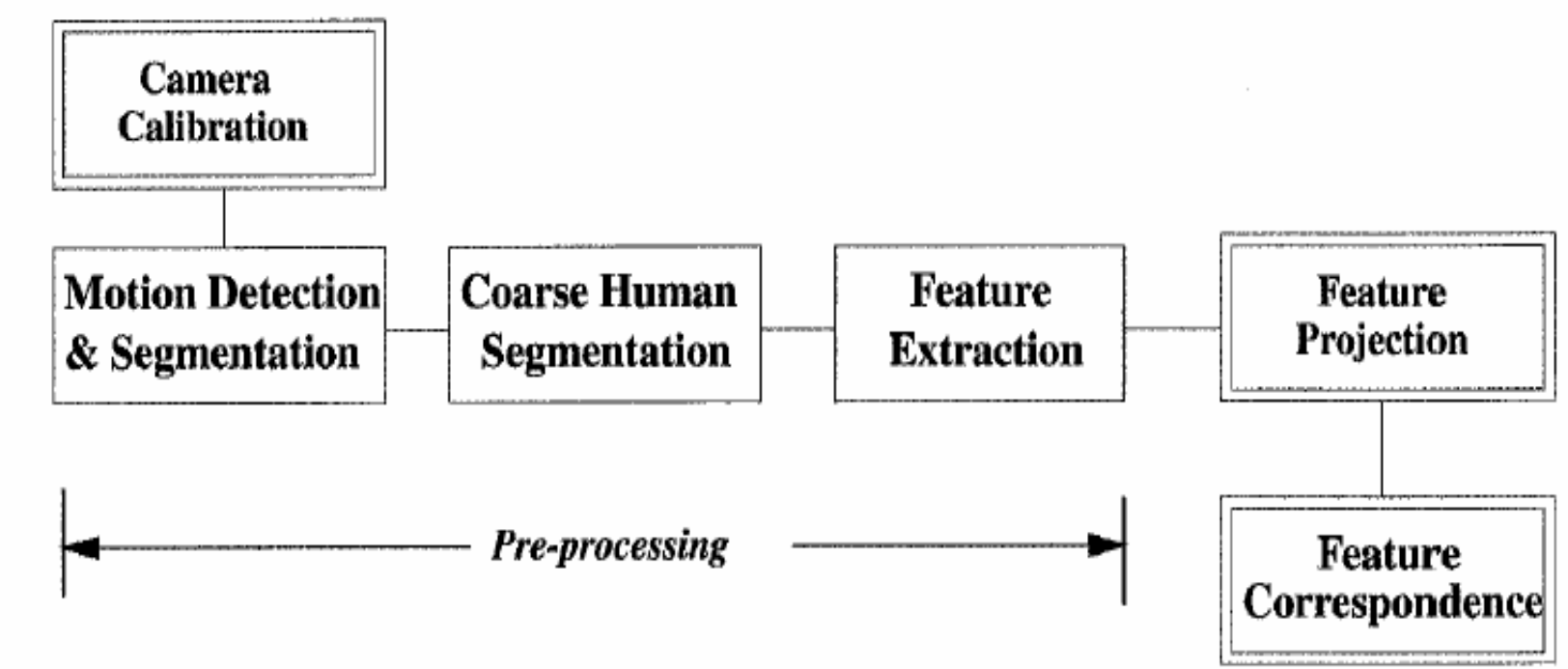
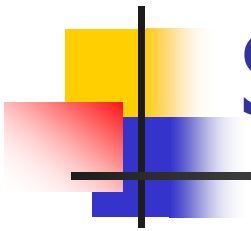


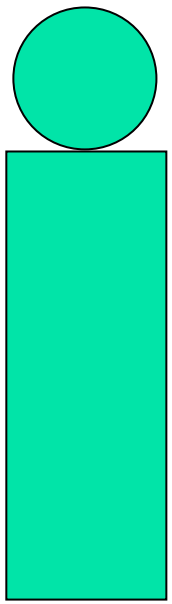
Fig. 1. The basic procedure of transition tracking.



Setup

- Multiple fixed cameras mounted in the area of interest
- Subject in the area will be in the view of at least one camera
- Cameras provide monocular grayscale images
  - cannot use color information
- Tracking via low level recognition of human motion

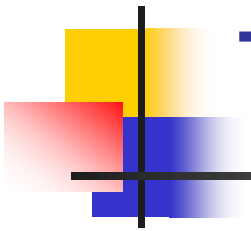
2D Human Model



- Head: ellipse with a height to width ratio between 1 to 1.5
- Human Trunk: rectangle with a height to width ratio between 1 to 3
- Height of Trunk to Height of Head ≈ 4:1

Preprocessing

- Detection of human subjects from the segmented non-background objects
  - Look for a blob whose area is consistent with the quantity $\pi ab/4$
  - Check if there is also a rectangular trunk region
- Feature extraction from the segmented human objects
  - geometric: *n* points on the medial axis of the upper body as the feature for tracking
  - visual: average intensity of neighborhoods of the above points



Tracking

- Find closest match of features in adjacent frames based on physical constraints
- Assume each feature point of a subject is independent of each other
- Assume the geometric and visual features are independent of each other

Tracking (Con't)

- Use Bayes' rule to maximize

$$\Pr(Z_t \mid \Theta) = \Pr(X_t \mid \Theta_X) \cdot \Pr(Y_t \mid \Theta_Y)$$

where :

$$Z_t = [X_t, Y_t];\ \Theta = [\Theta_X, \Theta_Y]$$

Z_t = feature vector of a subject at time t

Θ = feature vector of the tracked subject at time $t-1$

X = geometric feature vector

Y = visual (avg nhd intensity) feature vector

Tracking (Con't)

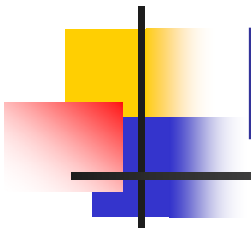
- Assume a multivariate Gaussian model for the feature distributions

$$\Pr(X_t \mid \Theta_X) = \prod_i \Pr(x_{it} \mid \Theta_X)$$

$$= \prod_i \frac{1}{2\pi\sigma_i^2} \exp\left[- \frac{(u_{it} - \overline{u_{it}})^2 + (v_{it} - \overline{v_{it}})^2}{2\sigma_i^2} \right]$$

- Maximizing $\Pr(Z_t \mid \Theta)$ is now equivalent to minimizing the corresponding *Mahalanobis distance*

Mittal and Davis' Approach [MD01]

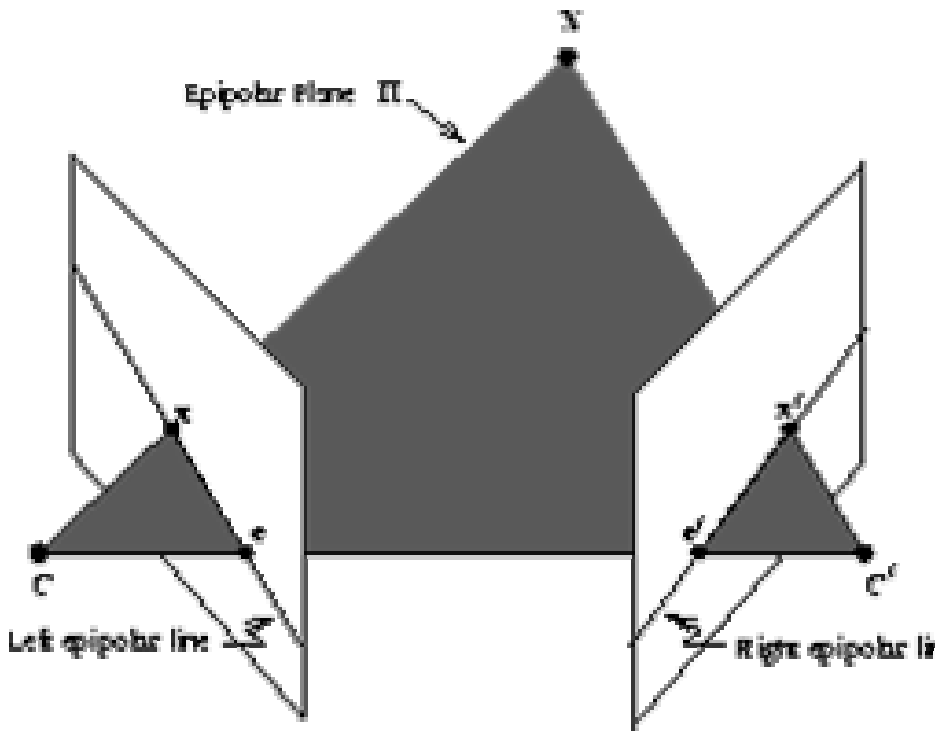


- Information is matched across multiple cameras
- Hypothesized object locations and tracks are combined in a robust manner in 3D

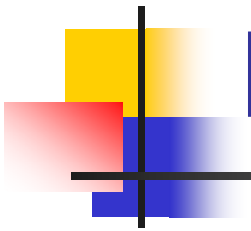
The Unified Framework

- background subtraction and image segmentation on each camera view
- match regions along epipolar lines in pairs of camera views
- produce probability distribution map of object location from a single camera pair
- combine results from multiple cameras pairs

Some Background



- The epipolar line is the image in one camera of a ray through the optical centre and image point in the other camera.



Matching Across Pairs of Views

- Along each epipolar line
  - match all segments from one camera view to the segments in the other view based on color characteristics
- For each matched pair of segments
  - take the midpoints of the segments along the epipolar line
  - backproject the midpoints to obtain a 3D point
  - project the 3D point onto the ground plane

Matching Across Views (Con't)

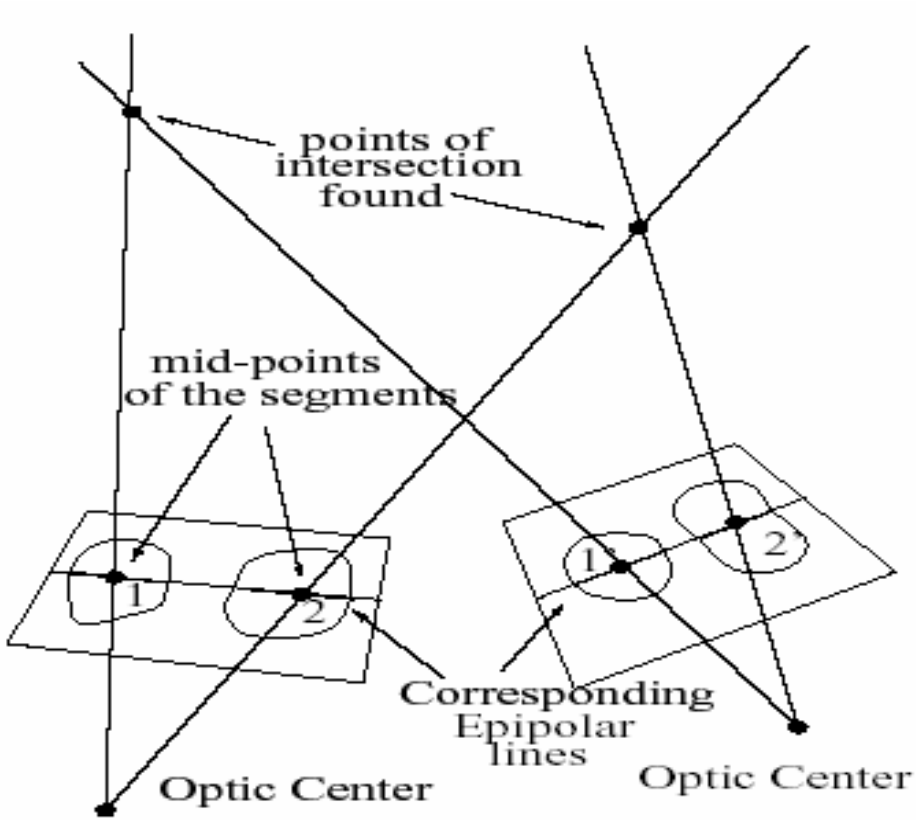


Figure 3: The mid-points of the matched segments are back-projected to obtain 3D points lying inside the objects. The matching segments are 1 and 1', and 2 and 2' respectively.

Probability Estimates

- For each 2D ground point $\boldsymbol{x_i}$, add a Gaussian kernel G_i:

$$G_i(x) = \frac{1}{2\pi\sigma_i^2} \exp\left[-\frac{\|x_i - x\|^2}{2\sigma_i^2} \right]$$

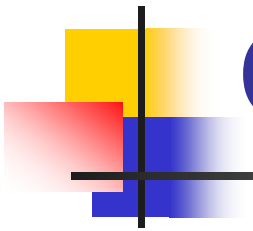
- For each point $\boldsymbol{x}$ on the plane, the probability that an object is present is given by:

$$\Pr(x) = C \cdot \sum_i G_i(x)$$

Probability Estimates (Con't)

- The standard deviation σ of the kernel depends on the minimum width of the segment, and the camera instantaneous field of view

Combining Results from Many Camera Pairs



- Observation: The probability values at the true locations of the objects reinforce each other.
- Simple Solution: Add all the probability values
- More sophisticated: Use a Gaussian weighting scheme

Combining Results (Con't)

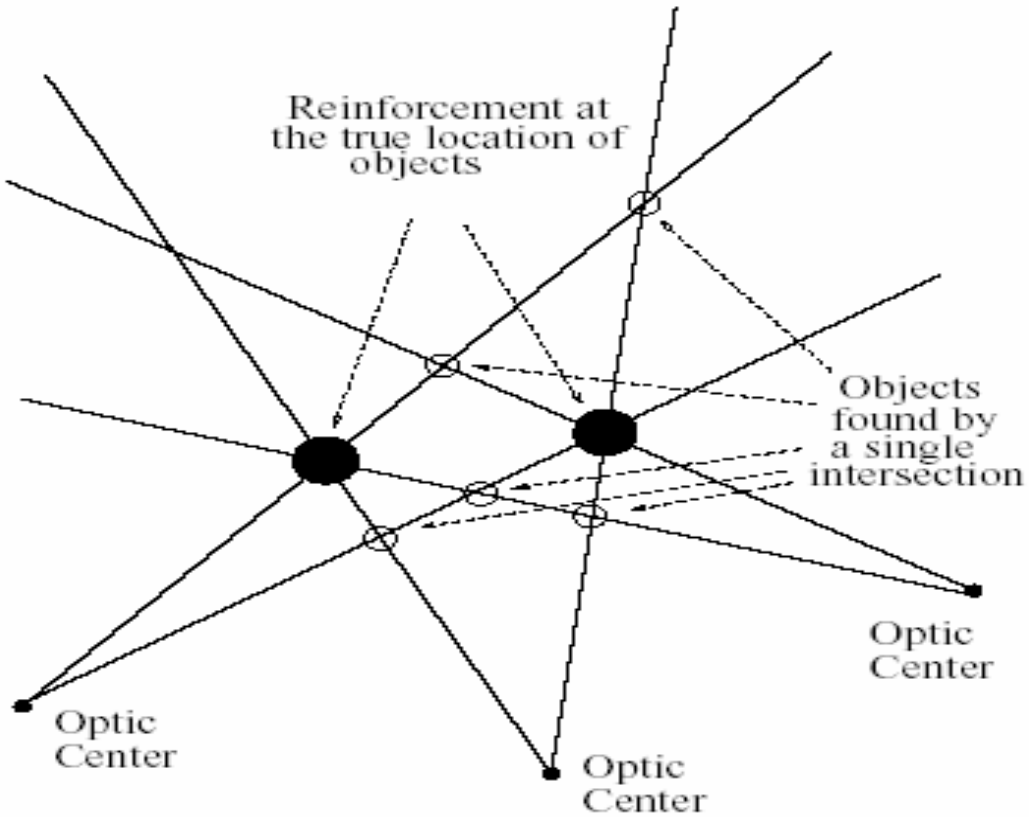


Figure 4: If segment matching fails to distinguish between two objects, the matches would reinforce each other only at the true location of the objects, and the false matches would get eliminated by the weighting scheme.

Tracking in the Ground Plane

- Identify the objects
  - threshold the probabilities
  - determine the connected components and their centroids
- Tracking the objects
  - a quadratic curve is used to fit the centroids of an object from the previous time steps
  - the curve is then used to predict the position of the object at the next time step

Results

No. of people	No. of True Objects	No. of cameras → 4	8	16
3	900	795	0	1
4	1200	1208	135	104
5	1500	209	1	0
6	1800	435	1	0

Table 1: No. of false matches integrated over all frames of the 300-frame sequences

No. of people	No. of True Objects	No. of cameras → 4	8	16
3	900	0	27	35
4	1200	0	6	27
5	1500	200	531	242
6	1800	100	457	312

Table 2: No. of true objects missed

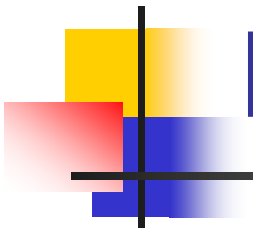
No. of cameras → No. of people	4	8	16
3	0.1281	0.848	0.1019
4	0.1890	0.1259	0.1255
5	0.1682	0.1525	0.1340
6	0.2134	0.1915	0.1584

Table 3: Average non-normalized probability value at Non-object locations. These values can be compared to the threshold value of 5.0 used to identify objects.

Counting Number of Objects

- D.B. Yang, H. González-Baños, and L. Guibas, "*Counting People in Crowds with a Real-Time Network of Image Sensors*", To appear, 2003.

Projecting the Visual Hull



- Cameras are placed pointing horizontally.
- Each measured silhouette sweeps a cone in 3D.
- Project the cone onto the plane parallel to the ground.
- Intersect the resulting projections.

Projecting the Visual Hull (Con't)

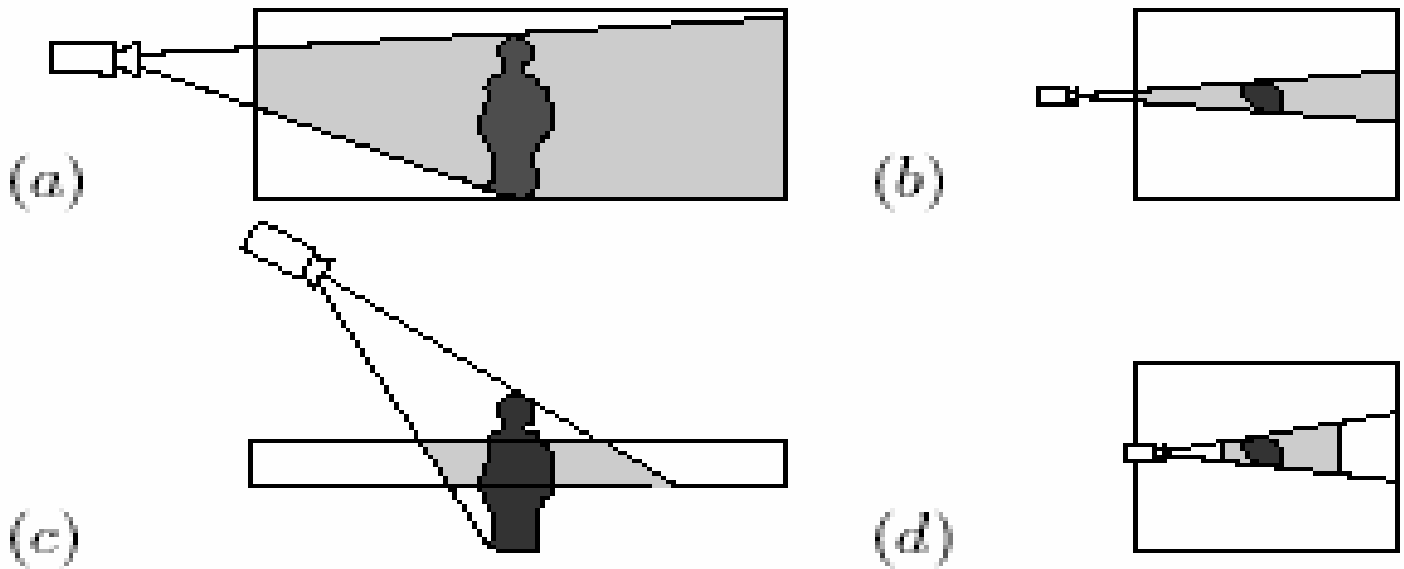


Figure 2: Projection of a silhouette cone. (a) and (c) are side views; (b) and (d) are the corresponding top views showing the projection onto the ground plane.

Pruning the Projection

- The resulting projection is ambiguous

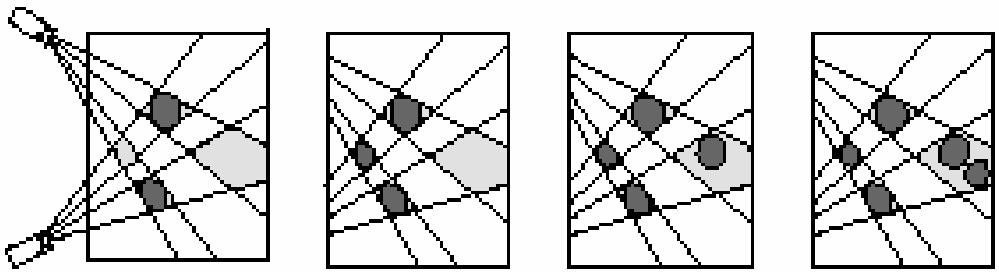
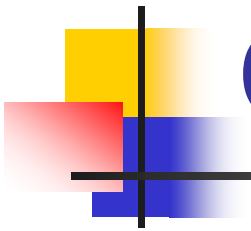


Figure 3: Different object arrangements can be consistent with a given visual hull. Polygons devoid of objects are called phantoms.

- Two pruning strategies
  - polygons smaller than the minimum object size
  - polygons that appear from nowhere



Object Bounds

- Use the history tree to keep track of bounds on the number of objects in each of the remaining polygons as time evolves
  - leaves = current polygons
- Initially, we have two bounds
  - upper bound constraint: area of polygon/smallest object size
  - lower bound constraint: a polygon P contains at least one object if there exists a ray from a camera that intersects only P

Bounds Across the Tree

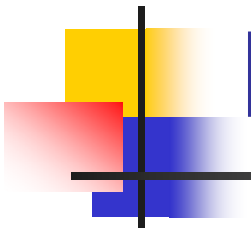
$$l_i = \max\left(l_i, \sum_{\forall j \in children(i)} l_j \;, l_{parent(i)} - \sum_{\forall j \in siblings(i)} u_j \right)$$

$$u_i = \min\left(u_i, \sum_{\forall j \in children(i)} u_j \;, u_{parent(i)} - \sum_{\forall j \in siblings(i)} l_j \right)$$

Updating the Tree

- Let $\mathcal{T}$ be the tree at time *t*. Let Π(t+1) be the set of pruned polygons at time *t+1*.
- Each P∈Π(t+1) will intersect some old polygons $Q_1,\ldots,Q_k$. Objects in P will then have to originate from the Q_is.
- P is added as a leaf to $\mathcal{T}$, and the tree is updated according to how many old polygons P intersects.

Basic Counting Algorithm



- Get new reading from each sensor. Compute the planar projection of each silhouette.
- Compute the intersection of all projections. Store the resultant polygons and remove small polygons and phantoms.
- Update the tree structure.
- Report the new bounds on the number of objects in the workspace.

Results

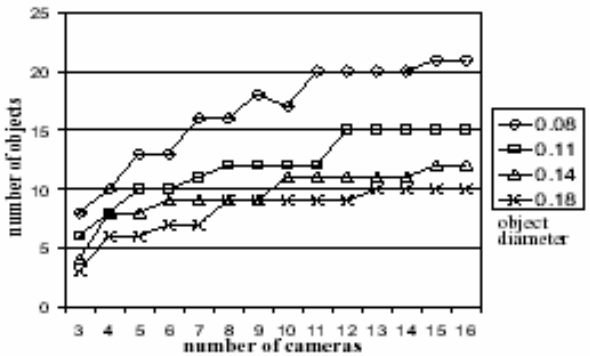


Figure 6: Synthetic objects moving in a square room with width 1. Plotted are the maximum number of objects that a given number of cameras was able to count exactly.

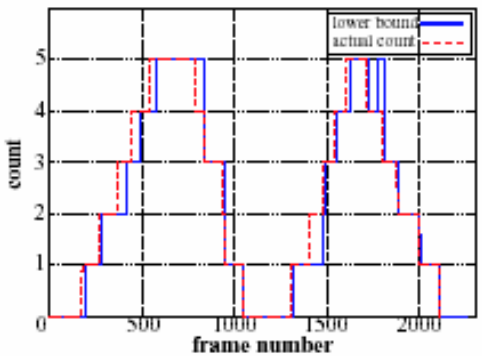


Figure 7: Count of 5 people walking into and out of the workspace.

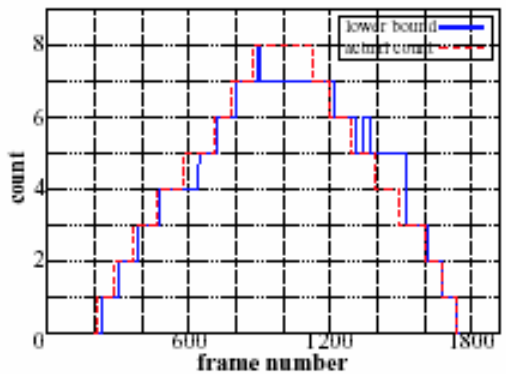


Figure 8: Count of 8 people walking into and out of the workspace.