IBM Research

Class 4: Tracking

Andrew Senior

aws@andrewsenior.com
http://www.andrewsenior.com/technical
Exploratory Computer Vision Group
IBM T. J. Watson Research Center
Yorktown Heights, NY

Overview

- Why tracking?
- 2D Tracking
  - Tracking types
  - Tracking by data association
    - Occlusions
    - Fragmentation
  - Model update
    - Localisation
- Conclusions

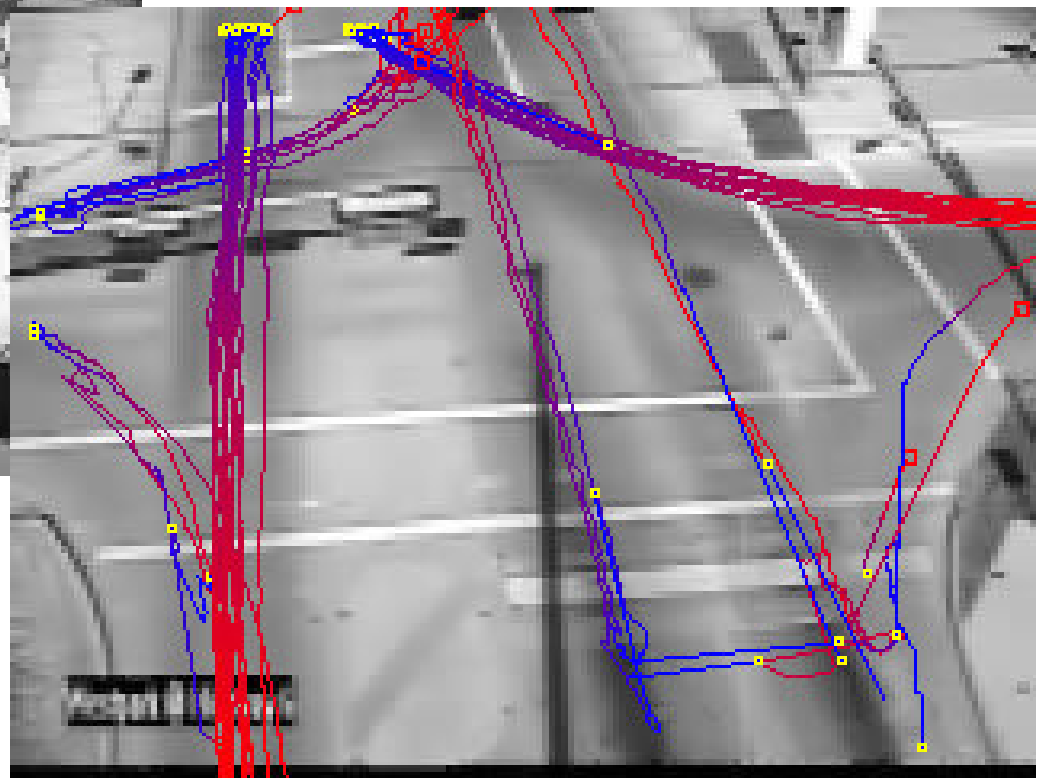
What is tracking?

- Locating an object over time
- Tracking has a long history e.g. in radar

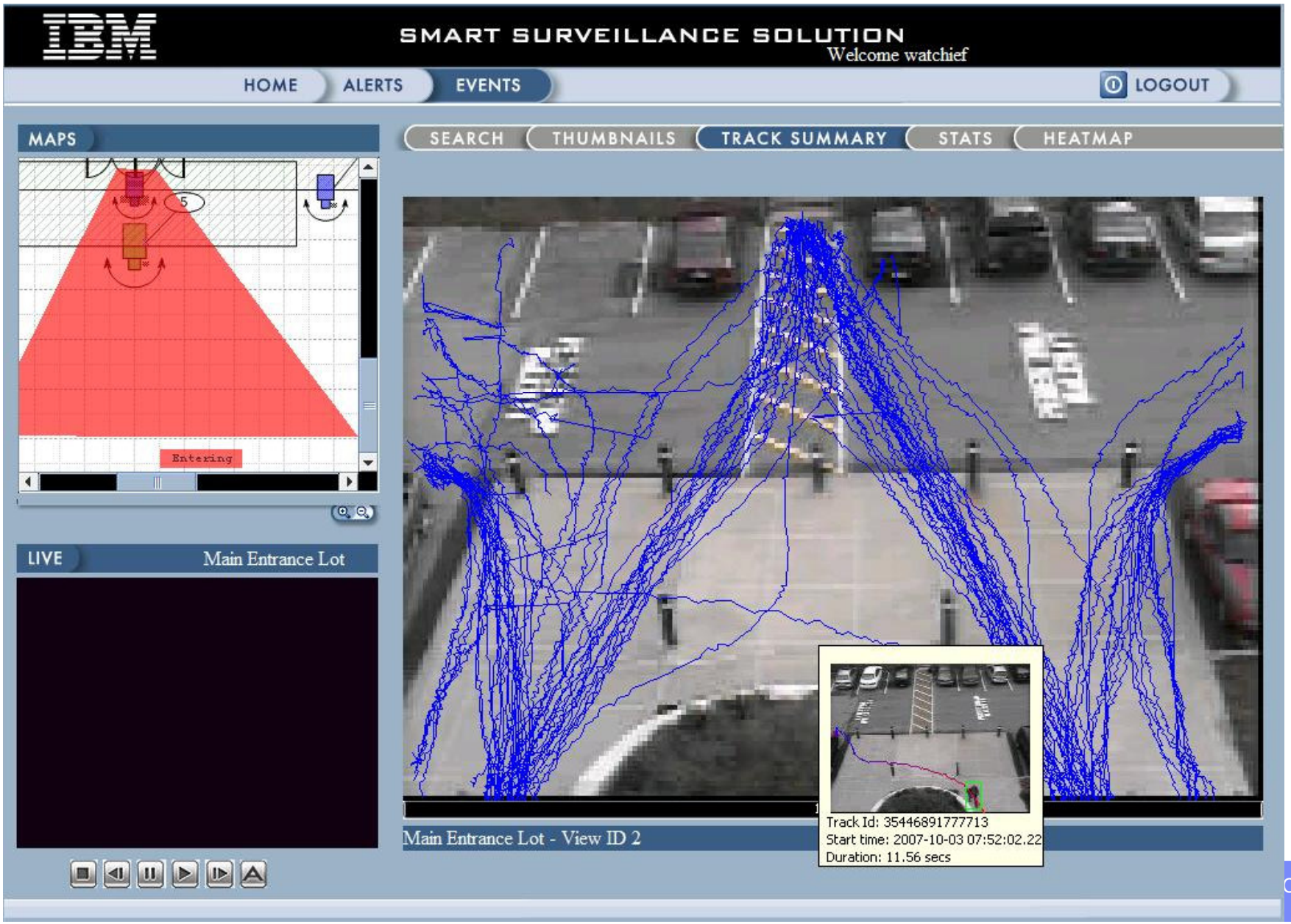
Why tracking?

- BGS is sufficient to detect moving objects
  - BGS contains no temporal information, results are generated for every frame.
    - lots of data
  - Sufficient for detection of motion
    - For video compression
    - e.g. alerts for perimeter protection
  - Need “higher level” information for search
- Tracking associates multiple observations of an object and treats them as a unitary whole.
  - Representation & visualisation
  - Search
  - Behavioural understanding & alert triggering
    - speed, trajectory
  - Compression

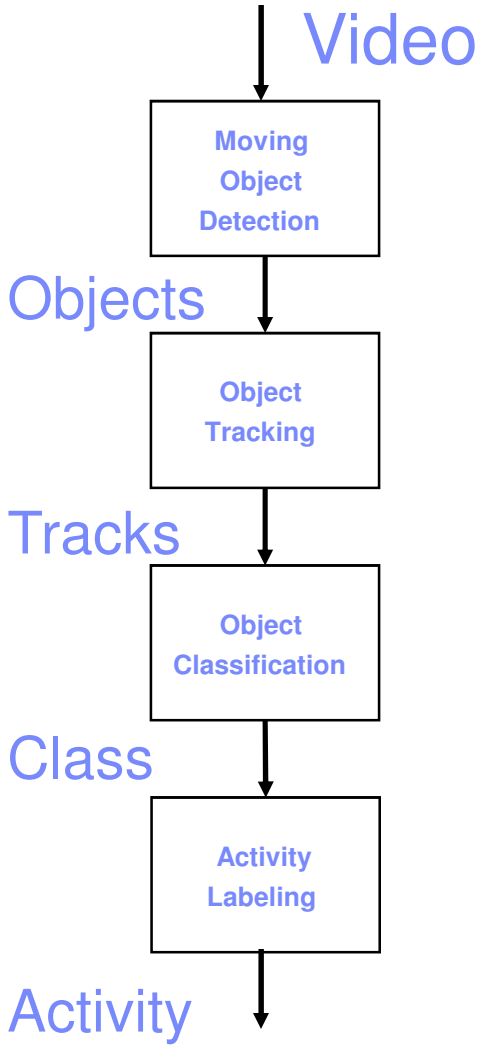
Trajectories



Track-based interface



Smart Surveillance Engine



Tracking types

- Clustering contour features
- FG blob assignment
  - Assignment problem
  - Splits, merges, occlusions
  - Occlusion bridging
  - Tracking through occlusions
    - Appearance models

Tracking by clustering contour features

- Pingali et al. 98 Tracking tennis players and people in stores
- Uses simple frame differencing (no background model) with morphology to join regions.
- Find curvature extrema on countours
- Match features with distance measure

$$k_r \delta r^2 + k_\theta \delta\theta^2 + k_\kappa \delta\kappa^2$$

  - Weighting location, angle, curvature
- Features that move similarly over a period are grouped into clusters

Features (black) & cluster trajectories (grey)

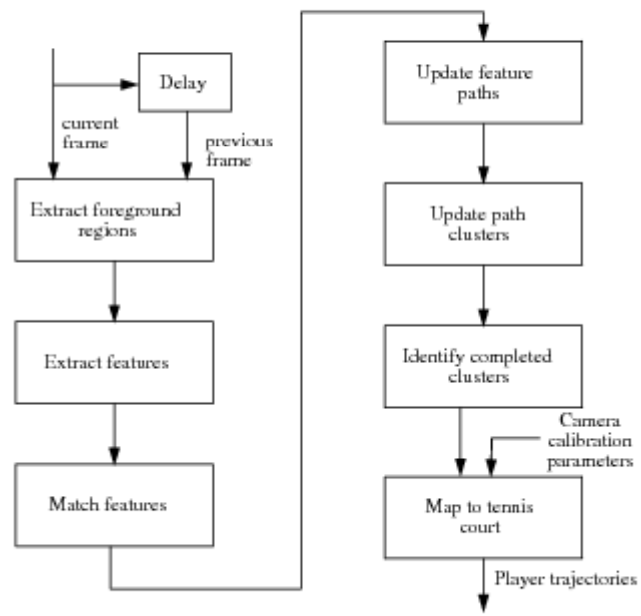
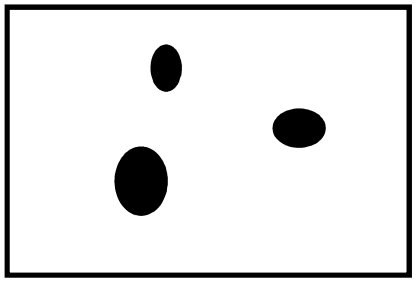


Figure 2. Steps in tracking player motion from video

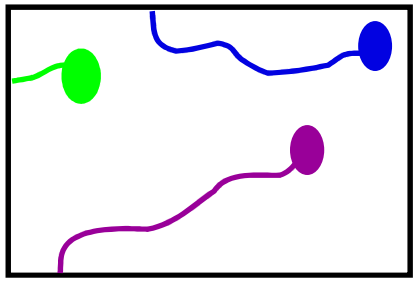
Tracking foreground regions

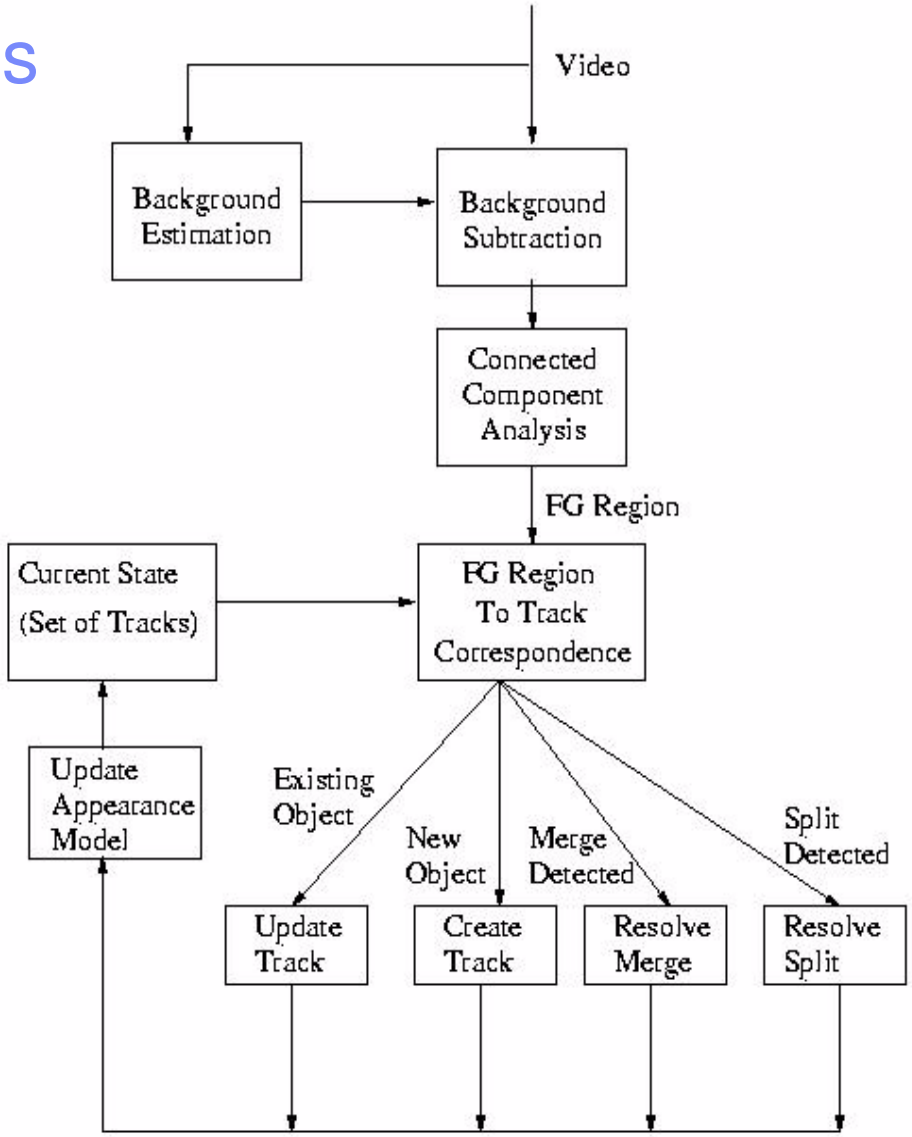
- 2D Tracking associates foreground regions to form tracks
- An *Assignment* problem

Foreground objects in current frame



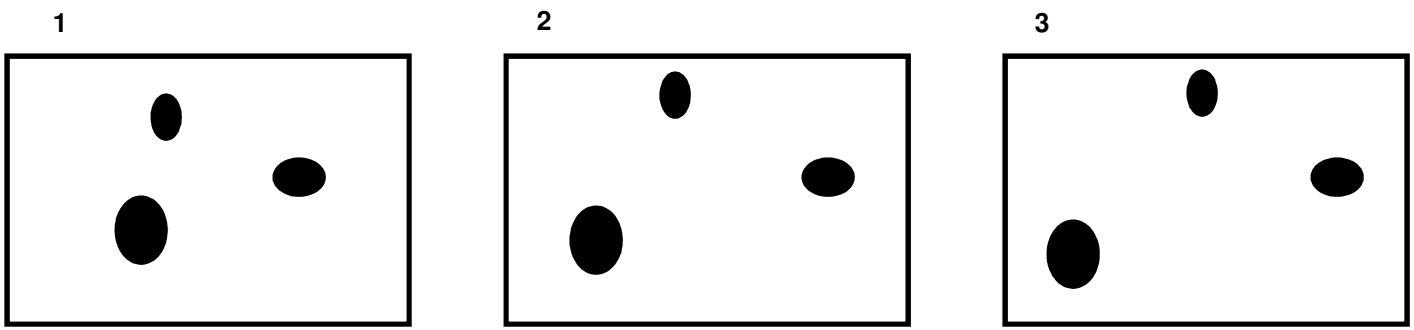
Tracks from previous frame



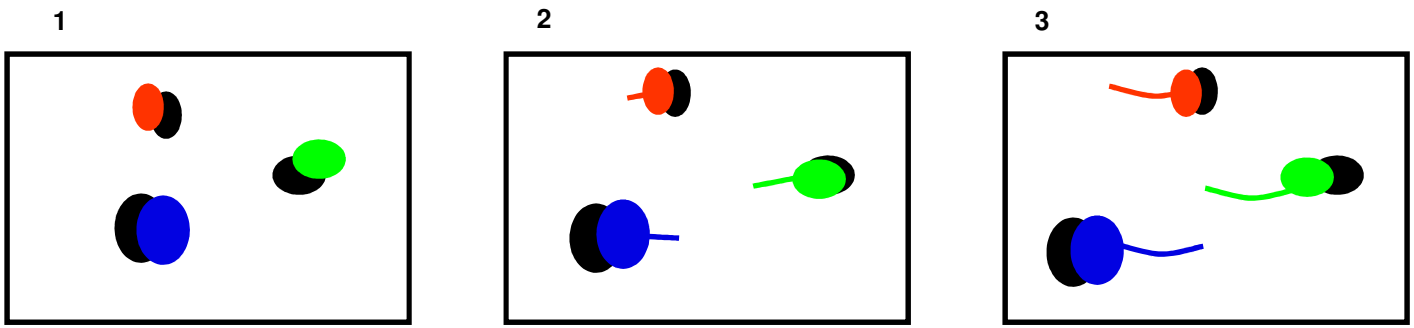


BGS Masks

- All “foreground” pixels in each frame



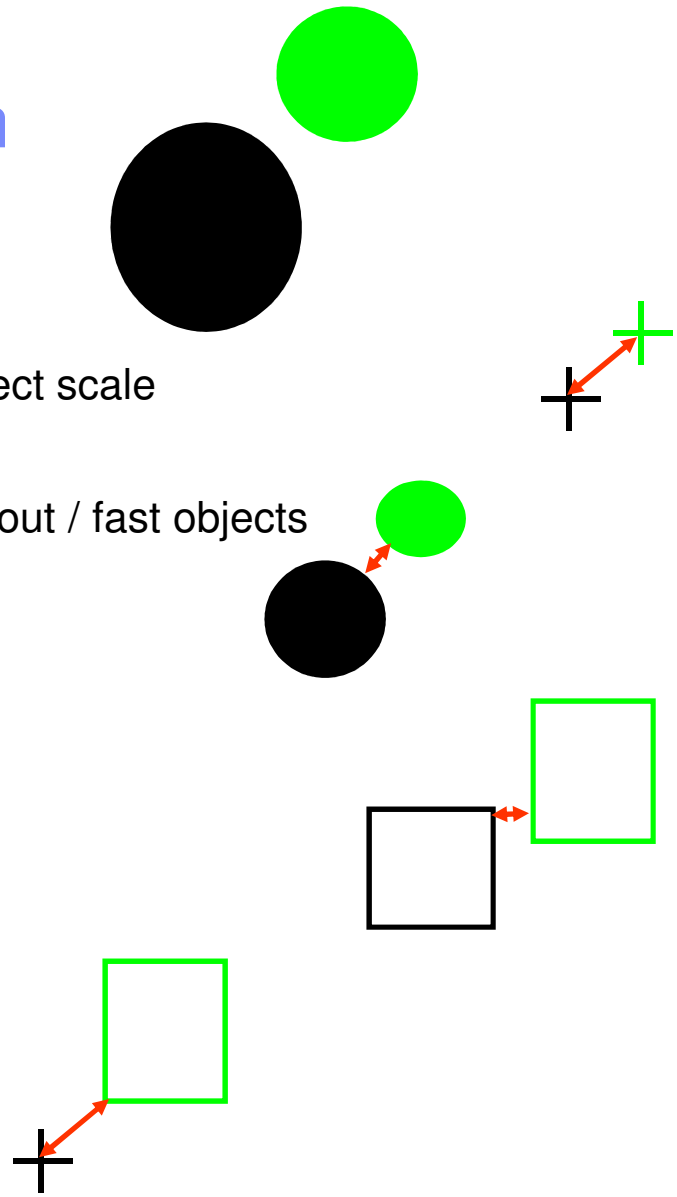
Association by proximity



- Associate foreground regions with current tracked objects

Object-track association

- Association metric choice:
  - Proximity of centroids
    - very dependent on object scale
  - Overlap
    - Can fail with BGS dropout / fast objects
  - Boundary distance
    - Expensive to calculate
  - Bounding box distance
  - Bounding box to centroid



Track-object association

- Compute matrix of object-track distances

Foreground regions

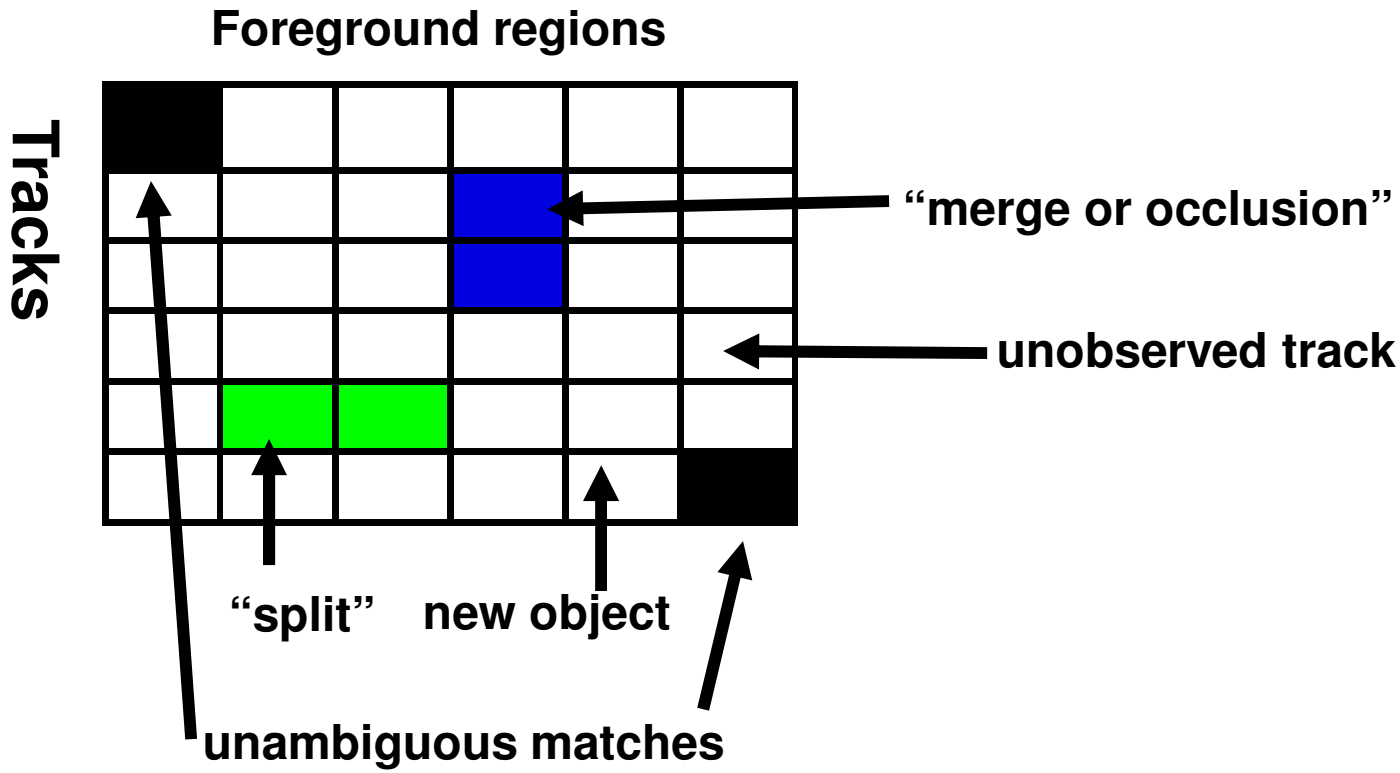
Tracks

0	50	100	90	74	12
60	60	20	2	120	15
60	60	20	2	120	15
58	85	49	49	47	0
12	0	0	67	26	24
37	45	84	85	36	37

- Threshold to keep matches

Track-object association

- Thresholded distance matrix



Unambiguous matches

- If there is a clear correspondance between track and object
- Build up a track history by adding new observation to object
  - Centroid
  - Bounding box
  - Mask
  - Appearance
  - Velocity
  - All observations are noisy
- For sparse scenes, matches may all be unambiguous (or fragmentation)

Track creation / destruction

- Create track when (sufficiently) detected
  - Discard short tracks as noise
- Consider track concluded when it hasn't been observed for N frames
  - N depends on scene and algorithm- false negative characteristics

Prediction

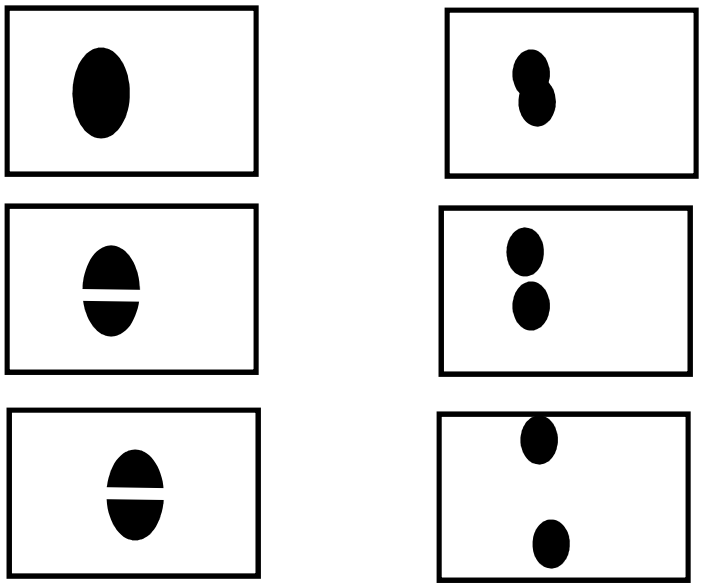
- The history can be used to predict the object's location
  - Predicted location should give smaller, less ambiguous distances
- First order model (constant velocity) using smoothed velocity history
  - $x_{t+1}=x_t+v.\Delta t$
- Higher order models unlikely to fit motion better. Don't account for "innovation", particularly in noisy data
- Kalman filter can be used for prediction
  - Stauffer & Grimson
- Learned behaviour might also make predictions
  - Known turns, decelerations etc.

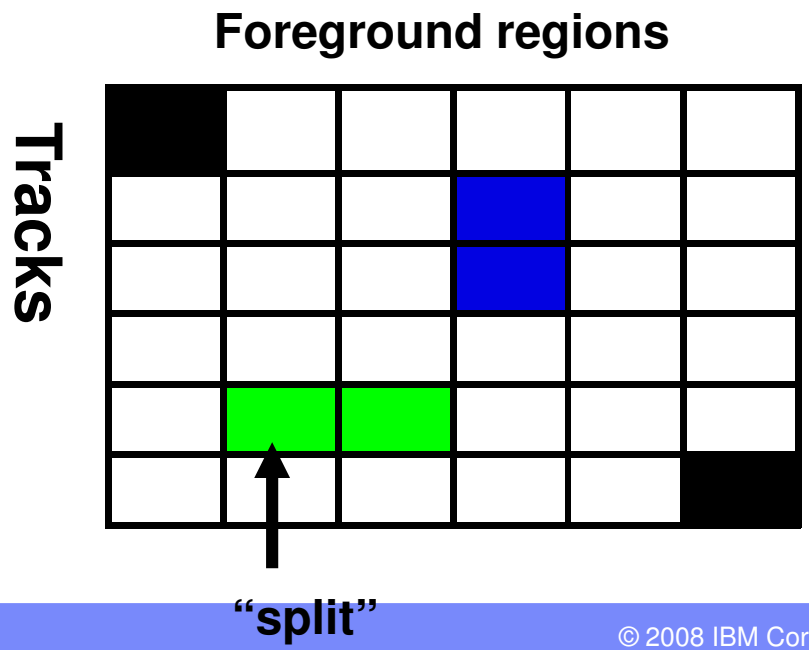
Prediction

- Displacement of an object is inversely proportional to frame rate
- At sufficiently high frame rates (depends on object speed and dimensions), object will overlap with its previous location
  - e.g. 400mm wide person at 30fps will overlap at speeds <12m/s (27mph)
  - 5m vehicle at 15fps will overlap at speeds <75m/s (168mph)
- Opinion
  - prediction isn't so important- it will only disambiguate targets if they are likely to occlude one another anyway
  - Tracking becomes easier (and faster) at high frame rates so track fast and often
- Systems are mostly designed for live video (albeit with processing constraints)
  - Few systems would track well on stored video at <=4fps

Splitting

- One tracked object corresponds to multiple FG fragments
  - 1) Object fragmented (BGS dropout)
    - Need to associate all fragments with same object
    - Boult et al. 2001
  - 2) Two objects were being tracked as one object and have separated
    - Need to create new object(s)

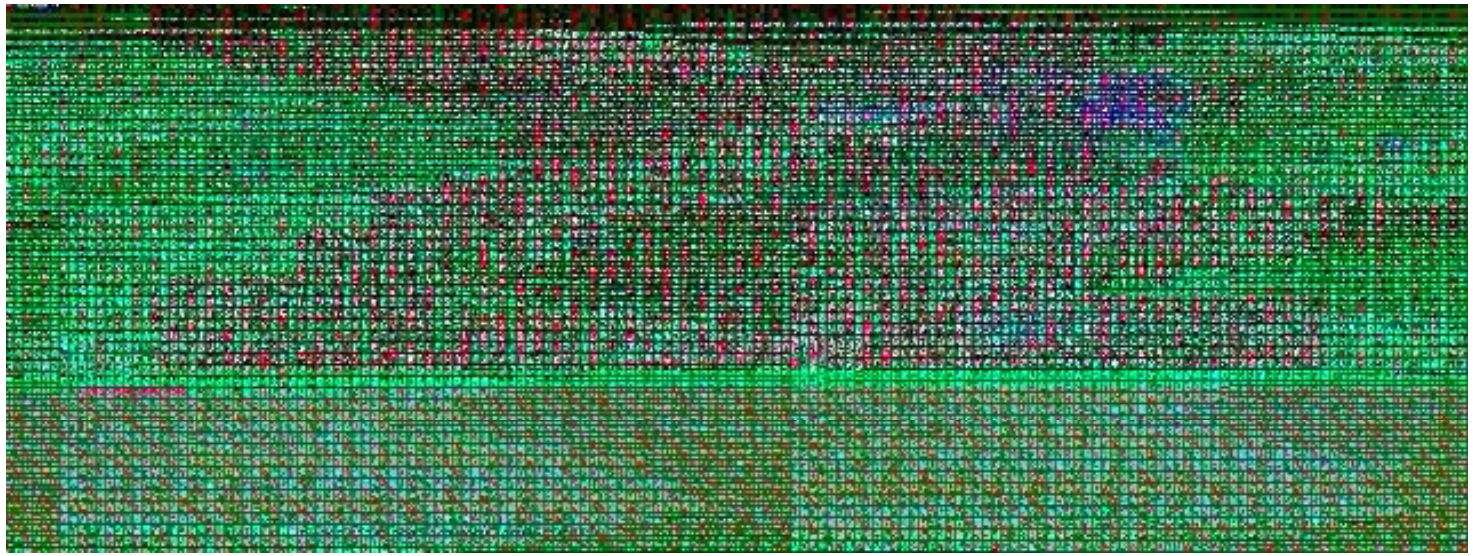


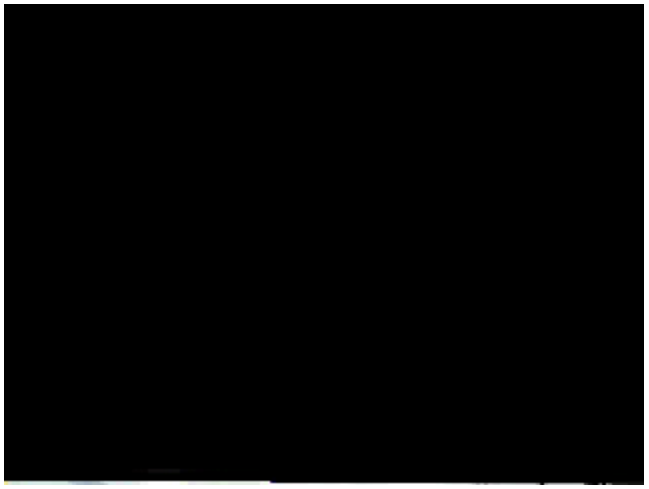


Splitting

- Need to distinguish type 1 & type 2 motions
- Accumulate evidence in “fission” data structure
- e.g. measure relative positions of blobs, separation, velocities
  - Consistently separating motion indicates type 2
  - Unsustained fragmentation, random motion (different fragmentation) or similar motion indicate BGS failures
- Wait until evidence accumulates before splitting object
  - Representation?
    - Need to infer the motion of the two objects now we know there are two
    - Copy trajectory
    - Copy trajectory with shift (assuming one was always offset)

PETS 2002 tracking in shopping mall



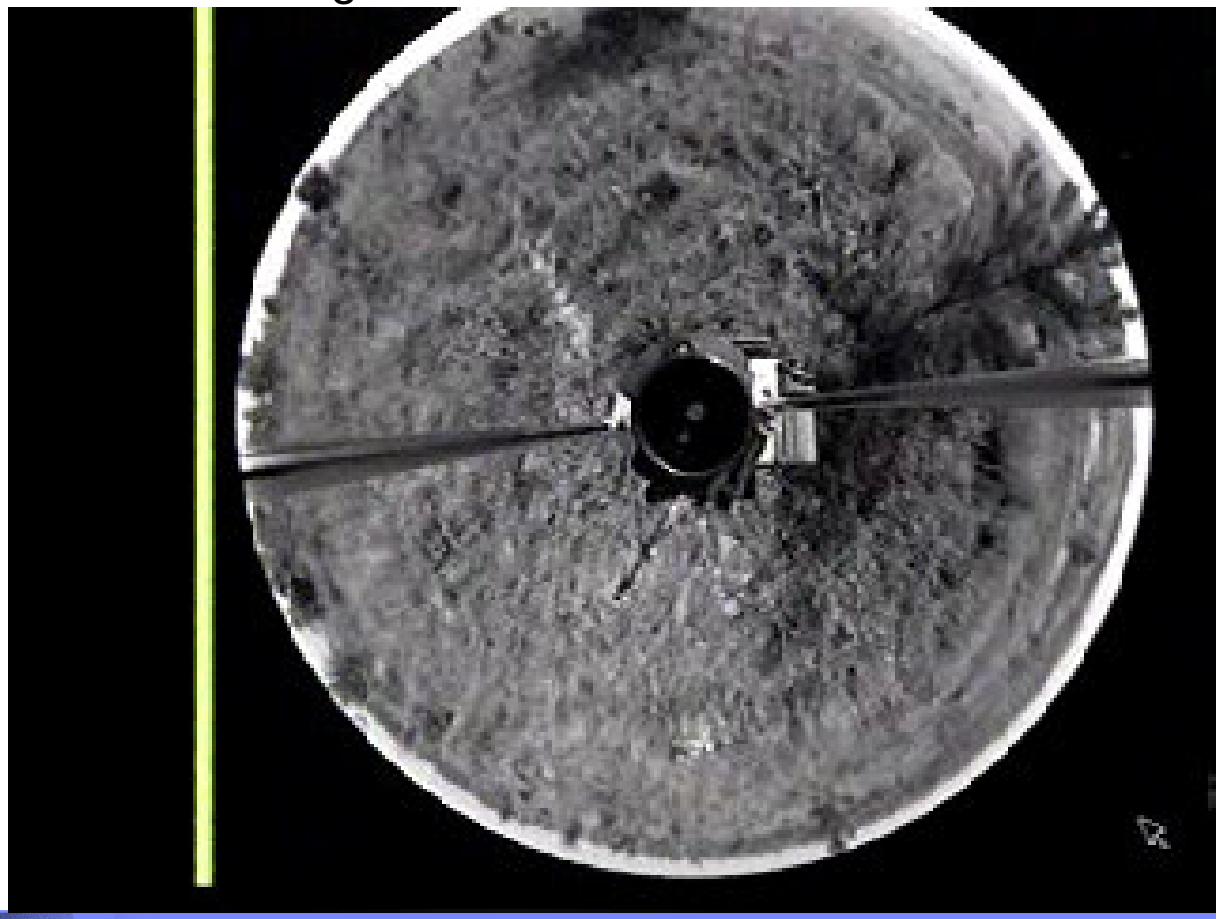


Stauffer & Grimson

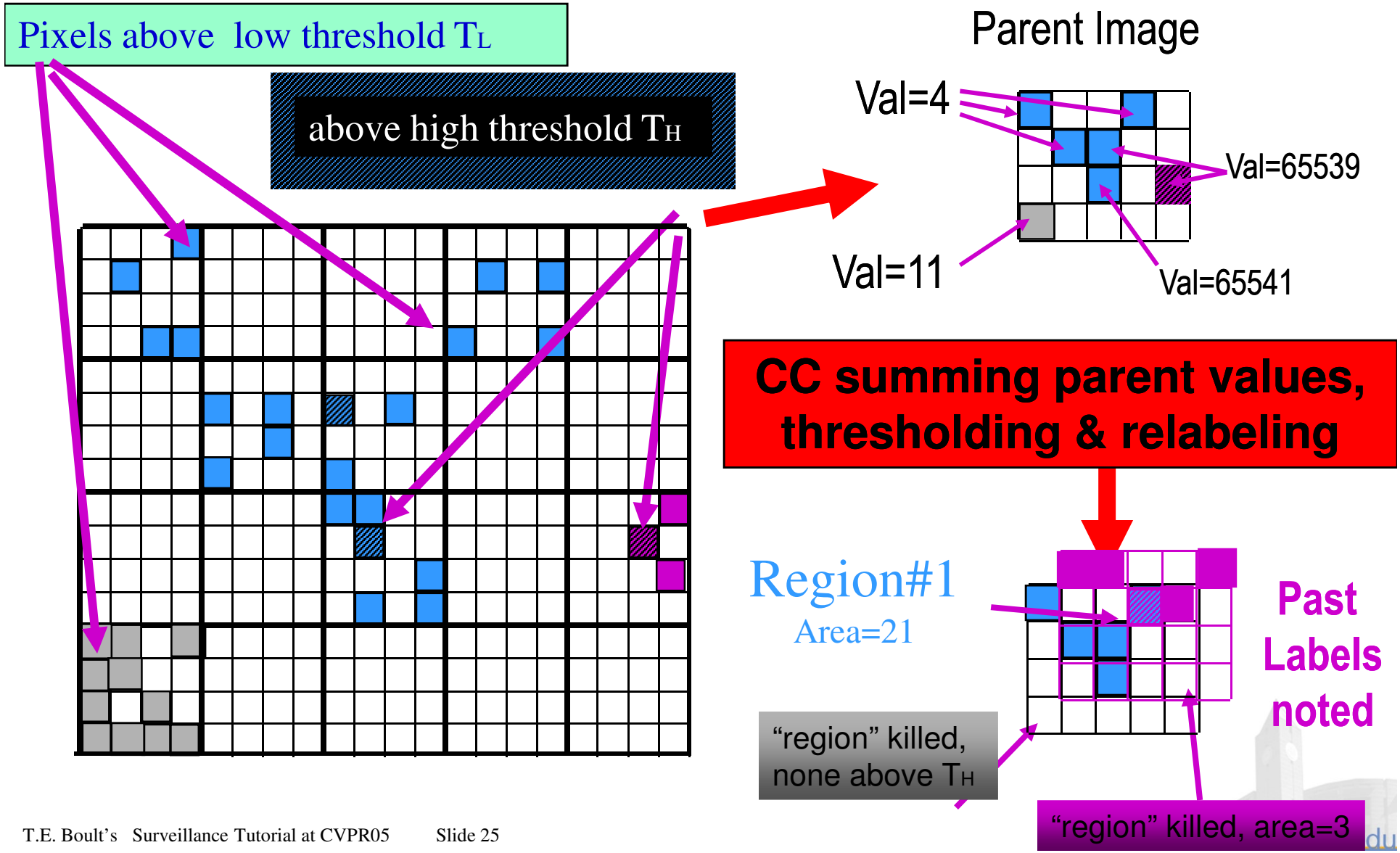
- Hypothesize object based on pairs of components from consecutive frames
- Prediction by Kalman filter
- Probabilistic assignment of FG regions to tracks
- <u>Does not attempt to track through occlusions</u>

Boult et al. “Into the woods” led to “Guardian Solutions” surveillance product

- Tracking snipers
- Uses Thresholding with hysteresis on background differences
- Analysis with Quasi-Connected components
- Tracking seems to be by simple overlap

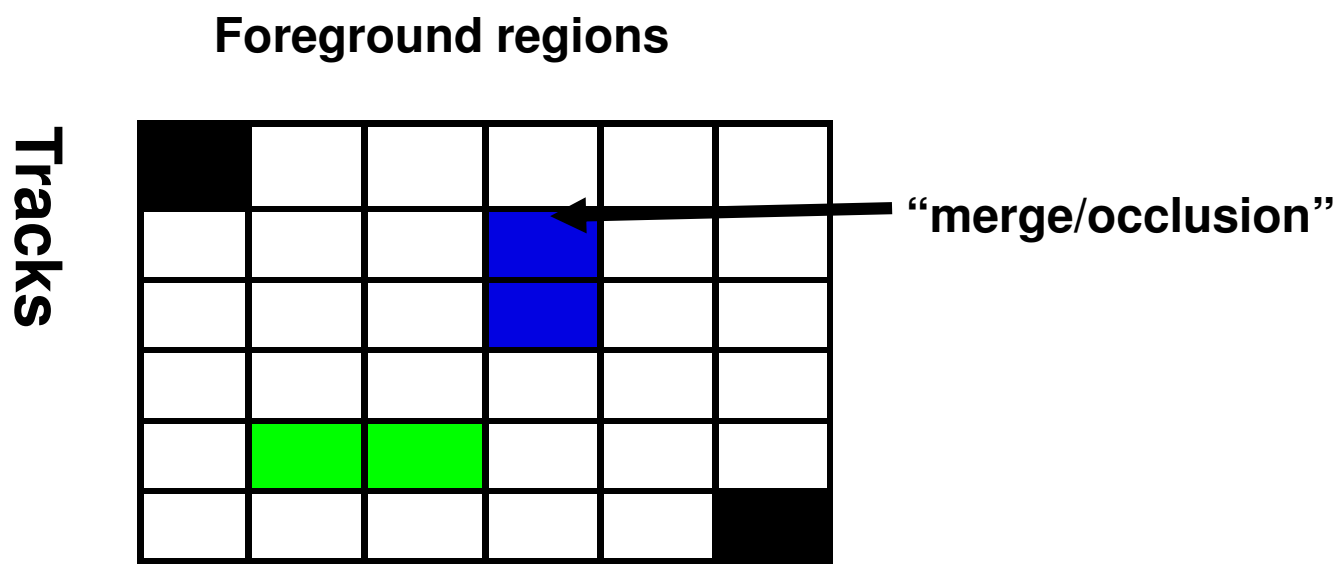


Quasi-Connected Components



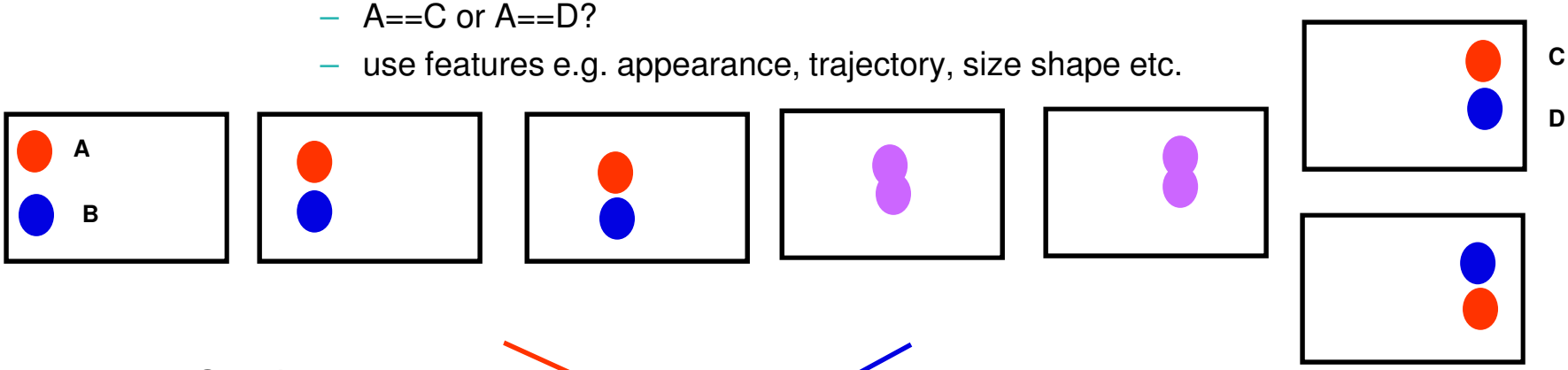
Occlusions

- Two or more tracked objects overlap.
- Trivial association no longer works- 2 or more objects correspond to one or more FG regions.

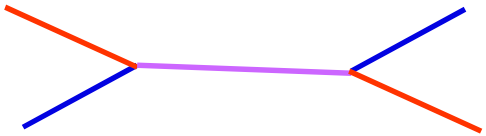


Occlusion bridging

- Simple approach to dealing with occlusions
  - Track the occlusion as a new object
  - When the object splits, try to work out correspondence
    - A==C or A==D?
    - use features e.g. appearance, trajectory, size shape etc.



- Graph structure

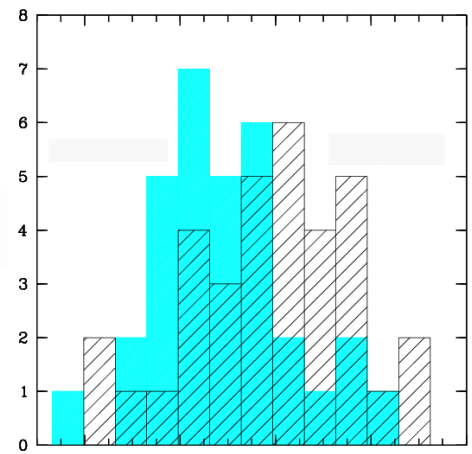


- Problems:
  - multiple occlusions
  - Lighting or appearance changes during occlusion
  - Fragmentation, merges & splits during occlusion

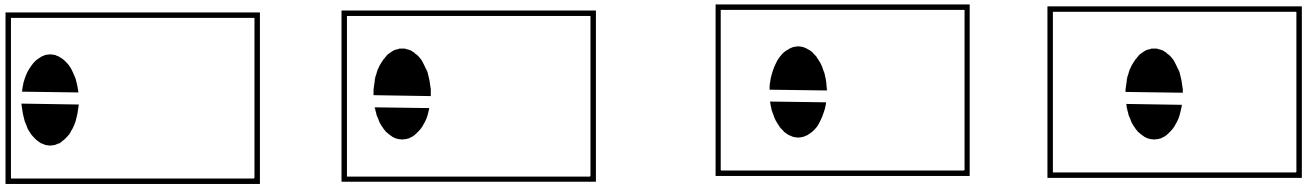
Similarity metrics

- Most common probably histogram intersection
- Find histograms H_A, H_B of A & B at every frame
- Store previous values when entering an occlusion

- When objects leave occlusion calculate histograms of C &D
- Calculate similarity $s(H_A, H_C)=sum_i(min(H_A[i], H_c[i]))$
  - $0<= s(H_A, H_C) <=1$
- Choose assignment according to sign of
- $s(H_A, H_C)+s(H_B, H_D)-s(H_A, H_D)-s(H_B, H_C)$
  - <0 => A==D, B==C
  - >0 => A==C & B==D
- Other histogram distances (Euclidean, cross …)
- Multiple histograms to account for spatial distributi



Merging



- When two objects move together consistently, then perhaps they're the same object.
- e.g. a person's head and foot enter the scene first and are detected as two well-separated FG blobs
- Solution
  - detect consistent motion of distinctly tracked objects in a "Fusion" data structure and detect consistency- e.g. scaled variance in separation of centroids, maximum distance between objects.

Object modelling

- Improve tracking by modelling the object
- Use model to improve localization & track through occlusions
- e.g. W4 Haritaoglu et al.
  - Grey-scale only
  - Prediction using second order model (constant acceleration)
  - Assignment with overlapping regions
  - Align model median pixel with FG region median pixel
    - Then find maximum correlation of model/FG silhouette
  - Occlusions tracked separately, then bridged
  - Does handle true / false splitting

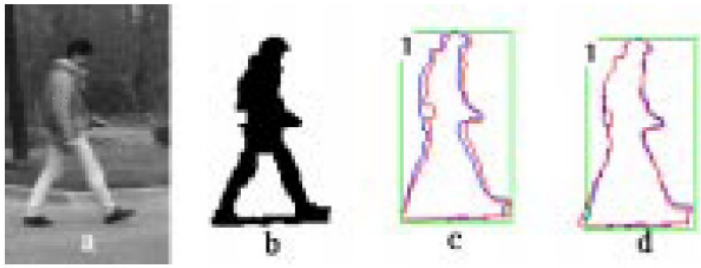


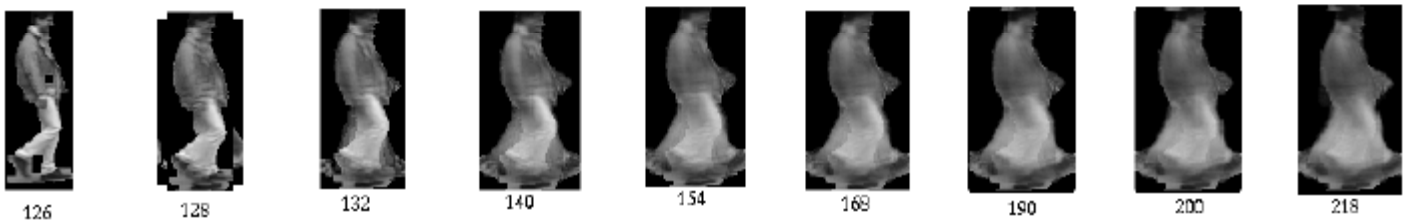
Figure 1: Motion estimation of body using Silhouette Edge Matching between two successive frame a: input image; b: detected foreground regions; c: allignment of silhoutthe edges based on difference in median; d: final allignment after silhouette corelation

W4 : Occlusion Bridging

- Temporal texture template

$$\Psi^t(x,y) = \frac{I(x,y) + w^{t-1}(x,y) \times \Psi^{t-1}(x,y)}{w^{t-1}(x,y) + 1}$$

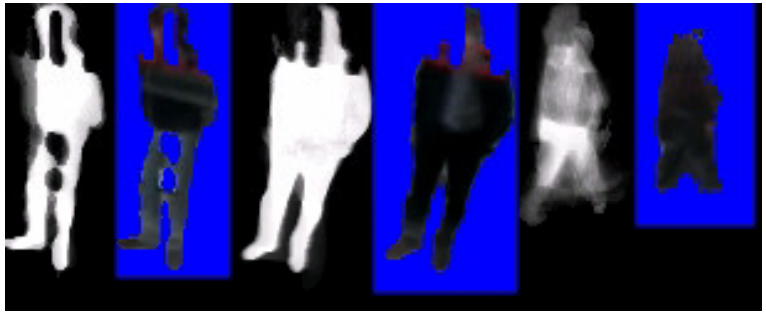
- Update over time



Appearance models

- **Model foreground regions with an appearance model+probability mask**

$$P_c(\mathbf{x},t),\ \ M_{RGB}(\mathbf{x},t)$$



- **Update each by blending:**

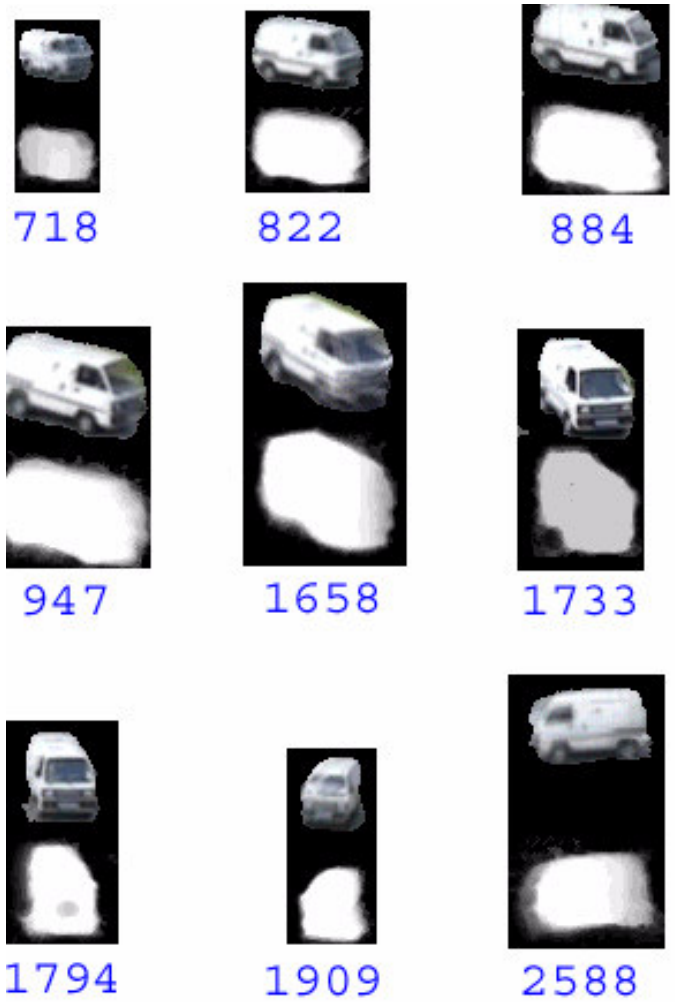
$$M_{RGB}(\mathbf{x},t) = M_{RGB}(\mathbf{x},t-1)\alpha + (1-\alpha)\mathbf{I}(\mathbf{x},t) \ \text{ if } \mathbf{x} \in \mathrm{F}$$

$$P_c(\mathbf{x},t) = \begin{cases} P_c(\mathbf{x},t-1)\lambda & \text{if } \mathbf{x} \notin \mathrm{F} \\ P_c(\mathbf{x},t-1)\lambda + (1-\lambda) & \text{if } \mathbf{x} \in \mathrm{F} \end{cases}$$

- **Trim borders when unlikely for speed/compactness**
  - Remove low probability pixels
  - Trim low probability edge rows/columns (when object shrinks)

Model evolution over time

- Model adapts to changes in shape, size pose, lighting etc.
- Does blur colours
  - Could use a multi-modal distribution



Locating the objects

- **Find best fit location by searching over x (correlation)**

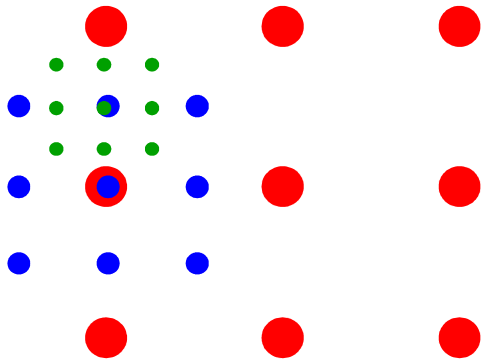
$$p(I,\mathbf{x},M)=\prod_{\mathbf{y}} p_{RGB}(\mathbf{x}+\mathbf{y})P_c(\mathbf{x}+\mathbf{y})P_{BG}(\mathbf{x}+\mathbf{y})$$

$$p_{RGB}(\mathbf{x})=(2\pi\sigma^2)^{-\frac{3}{2}}e^{\frac{-\left\|I(\mathbf{x})-M(\mathbf{x})\right\|^2}{2\sigma^2}}$$

- **Bias against fitting "background" pixels**

$$P_{BG}(\mathbf{x})=\begin{matrix}1 & \text{if } \mathbf{x}\in \mathrm{F}\\ P_{\text{penalty}}=0.05 & \text{if } \mathbf{x}\notin \mathrm{F}\end{matrix}$$

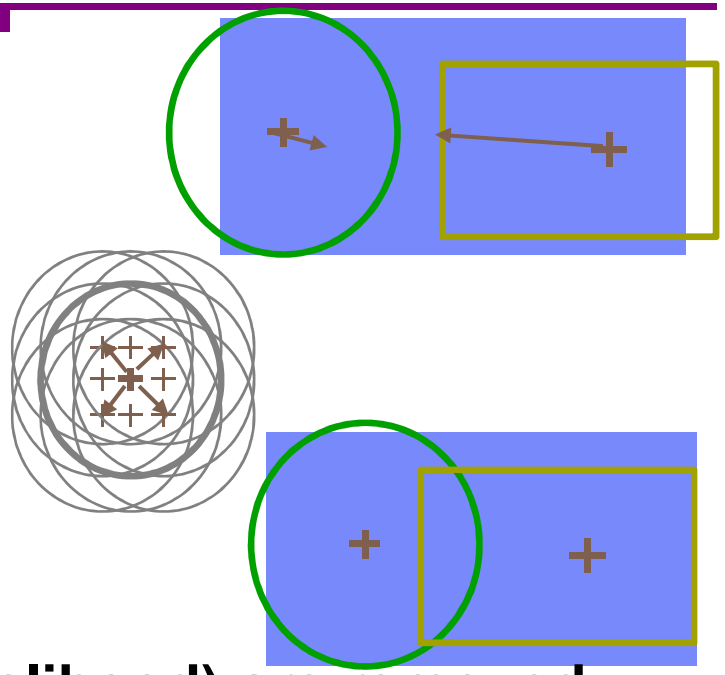
- **Coarse-to-fine search**



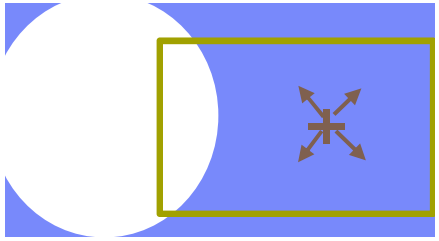
Occlusion Resolution

Proceeding in depth order:

- **Using 1st order model, centroid locations are predicted**
- **Correlate each object's visible pixels with the image near the predicted position, to find MaxLikelihood location**
  (ignoring previously explained pixels)



- **Pixels explained by model (with high likelihood) are removed**
  - **Fit "deeper" models**
  - **No penalty for fitting "explained" pixels**

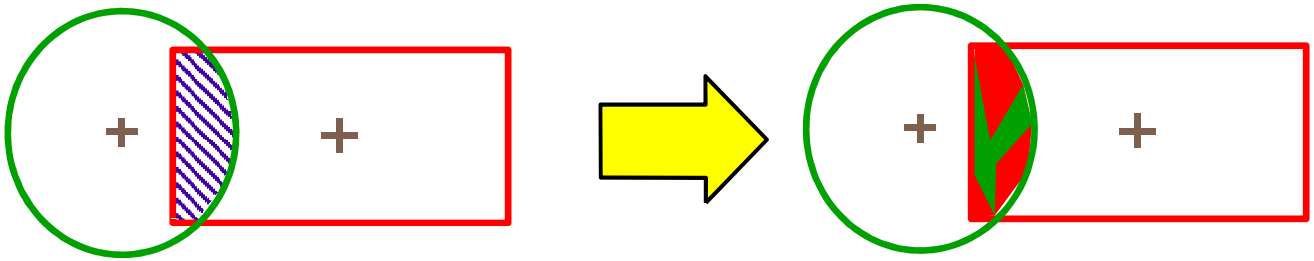


Depth ordering

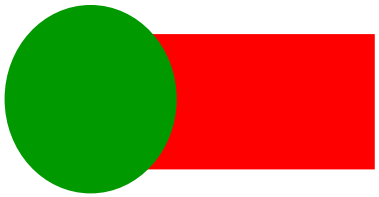
- Disputed pixels are classified using ML

$$\text{class}(\mathbf{x}) = \text{argmax}_i \; p_{RGB_i}(\mathbf{I}(\mathbf{x}))p_{c_i}(\mathbf{x})P_{NO_i}$$

- Ambiguous pixels are labelled as such



- In regions of overlap, choose model that explains most pixels in the disputed regions
- Consider that object the frontmost, and assign all disputed pixels to it



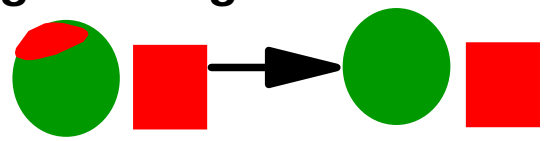
- In successive frames, fit front-most object first.
  - But recalculate depth ordering

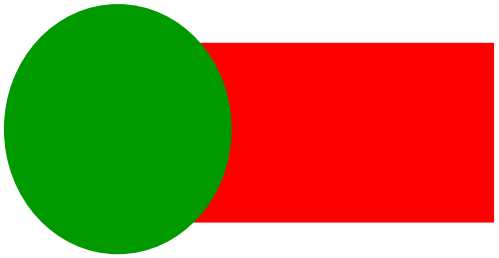
Appearance models for tracking

- **Each pixel in foreground region must be explained**
- **Hypothesize:**
  - **predicted existing object**
    - **including occlusion resolution**
  - **new part of existing object**
  - **new object**

Occlusion resolution

- **Objects are ordered so that those which are assigned fewer disputed pixels are given greater depth. Those with few visible pixels are marked as occluded.**
- **Pixels overlapped by two objects are assigned to the frontmost object which overlapped them.**
- **Unclassified pixels (novel) are 'filled' from neighbouring regions, or assigned to the nearest object**
- **Simple rules for defragmentation**



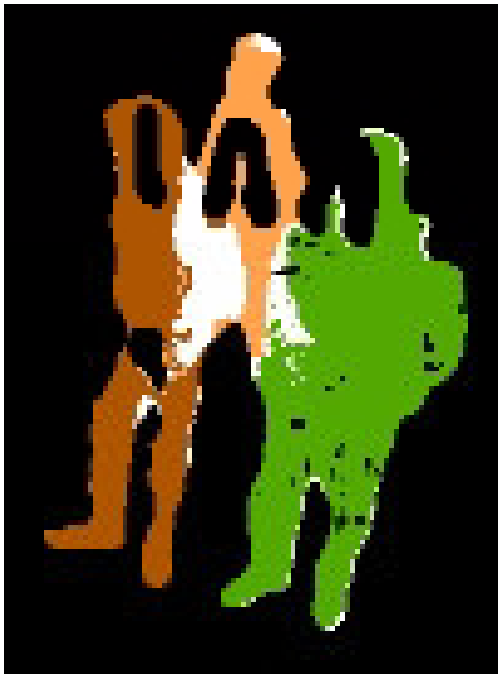


Model update

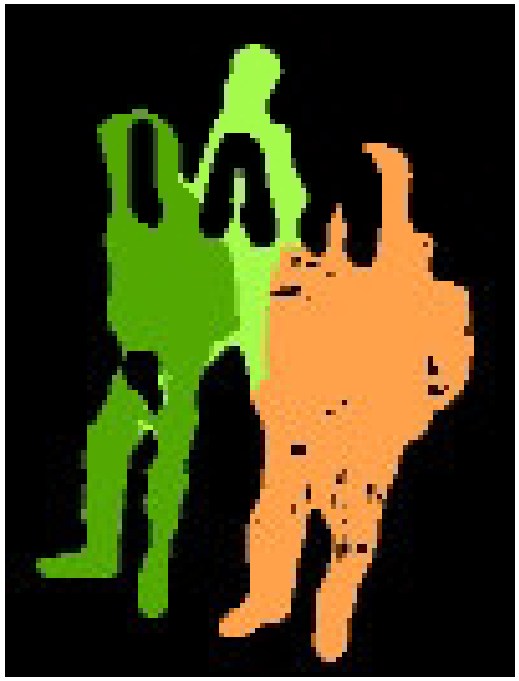
- During occlusions model update is riskier
- Assignment is inaccurate
- Model drift problems – learn wrong appearance
  - Reduce update,
  - Ignore some regions
  - Turn off update altogether

Handling occlusions

BGS



Initial



Final

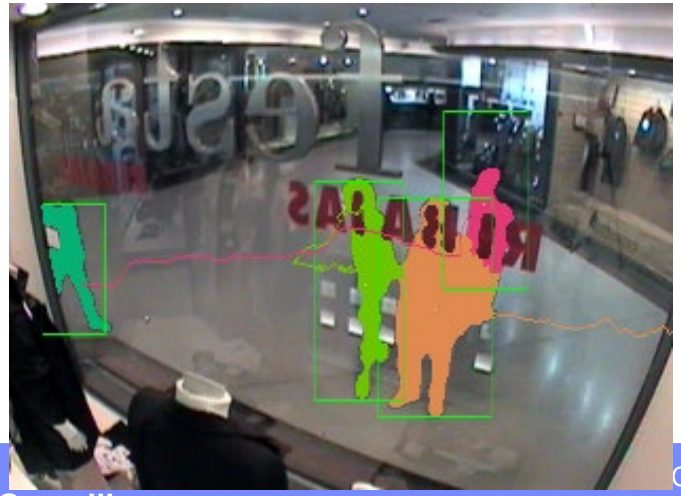
Tracking through occlusions

Sequence 3

554

575

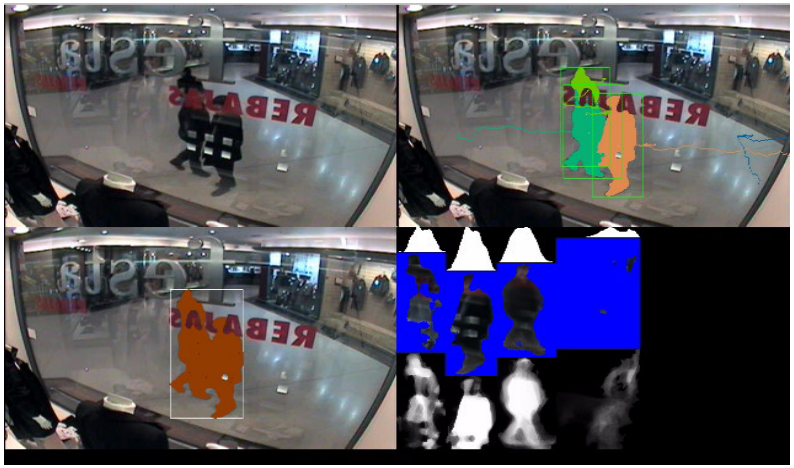
590



616

Tracking Through Occlusions II

Sequence 3

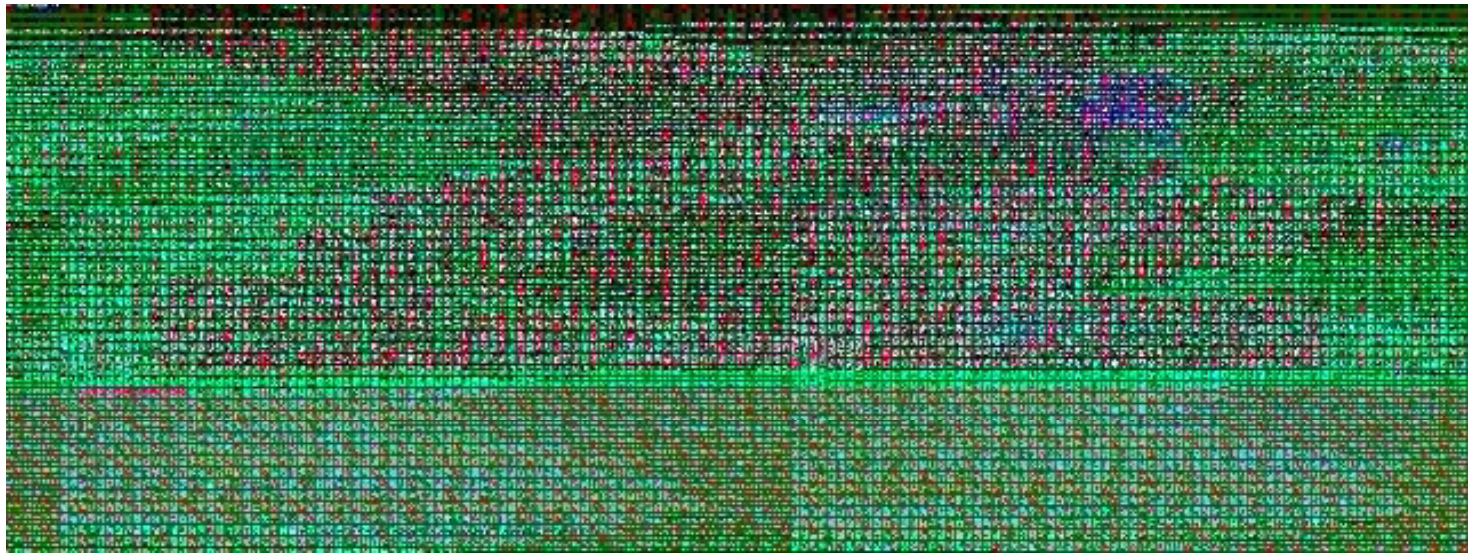


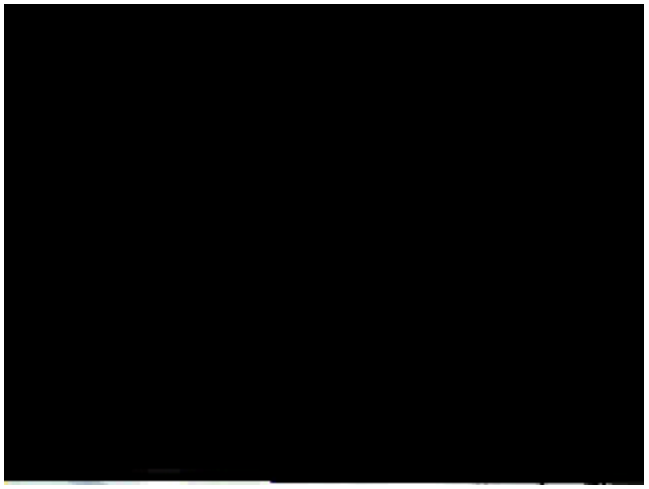
719

742

783

PETS 2002 tracking in shopping mall



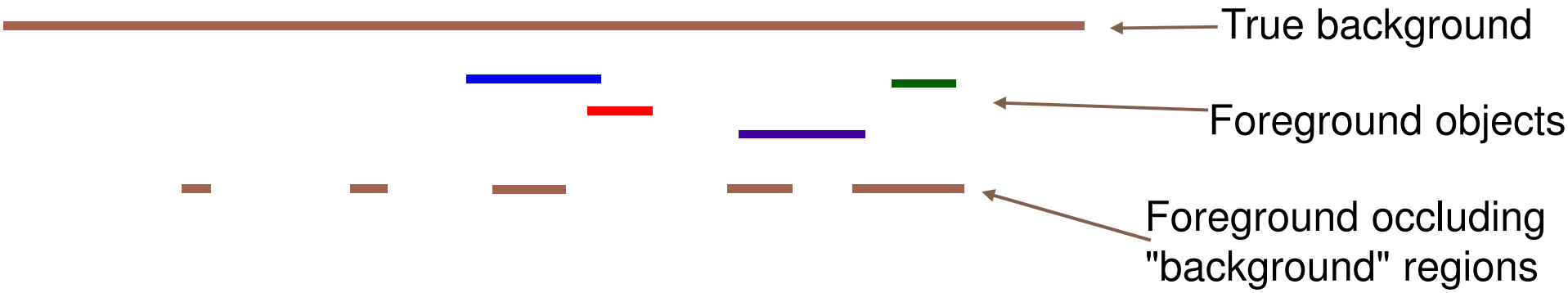


Region of uninterest

- High noise regions give many false positives
  - Waving trees
  - water,
  - flashing lights
  - CRT displays
  - areas where perspective displays many small objects
- Hand marked or (ideally) learned
- Reduce false positive detections and consequent tracking errors by masking out misleading areas
- Tracking needs to know where these regions are – lost tracking or missed observations

Foreground occlusion model

- **Foreground objects may be occluded by pixels in the "background model"**
  - **Reflections, window text, window frame**
  - **Results in erosion and poor fitting of foreground models**
- **Learn the areas where this happens**

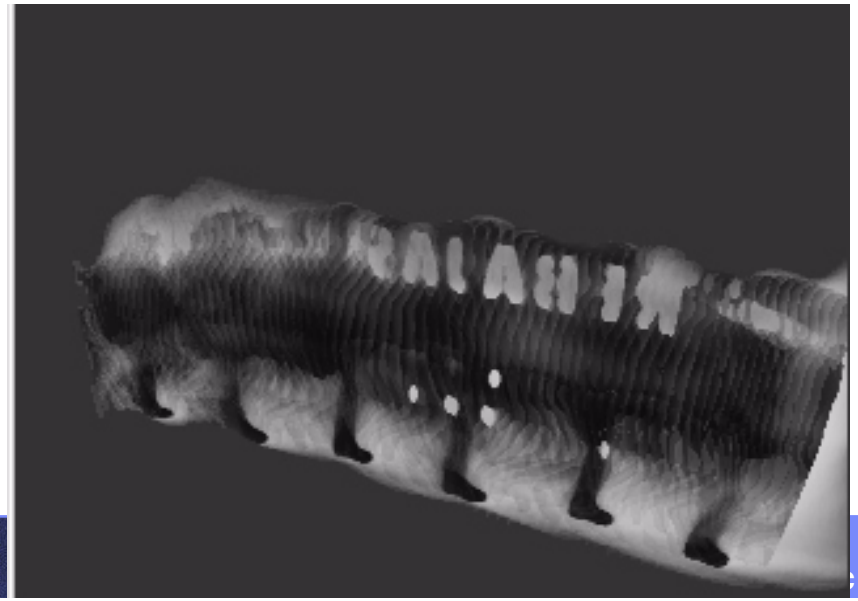


Foreground occlusion model

- **Whenever a pixel is classified as background but is overlapped by a foreground object, increment its likelihood of being "foreground occluding"**
- **Probability mask update is now:**

$$P_f(\mathbf{x}) = \frac{k_n + \sum_{t:\mathbf{x}\notin \mathrm{F}} P_c(\mathbf{x})}{k_d + \sum_{\forall t} P_c(\mathbf{x})}$$

$$P_c(\mathbf{x},t) = \begin{cases} P_c(\mathbf{x},t-1)(1-\lambda(1-P_f(\mathbf{x}))) & \text{if } \mathbf{x}\notin \mathrm{F} \\ P_c(\mathbf{x},t-1)\lambda + (1-\lambda) & \text{if } \mathbf{x}\in \mathrm{F} \end{cases}$$

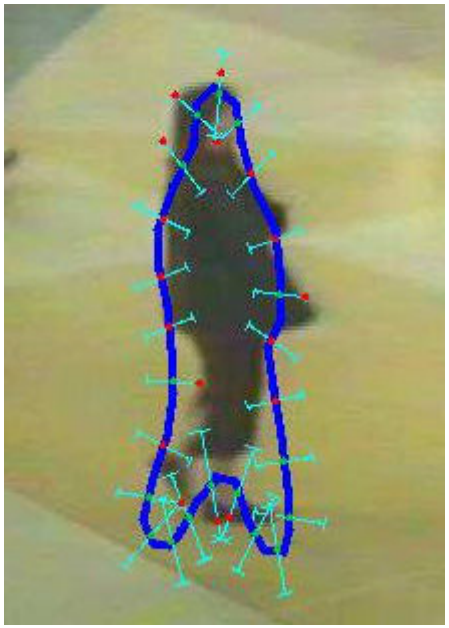


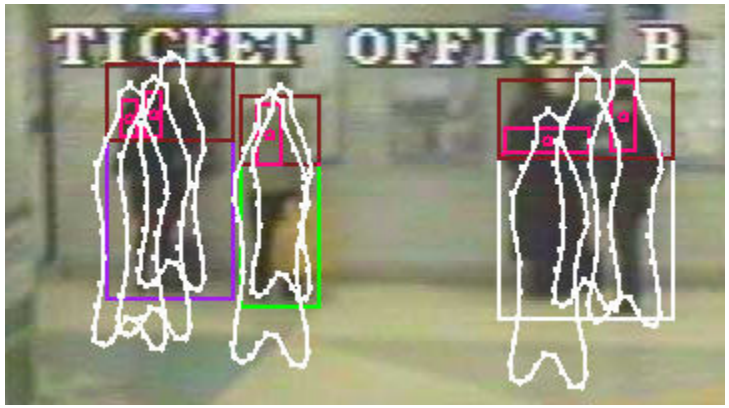
Use the foreground occlusion model

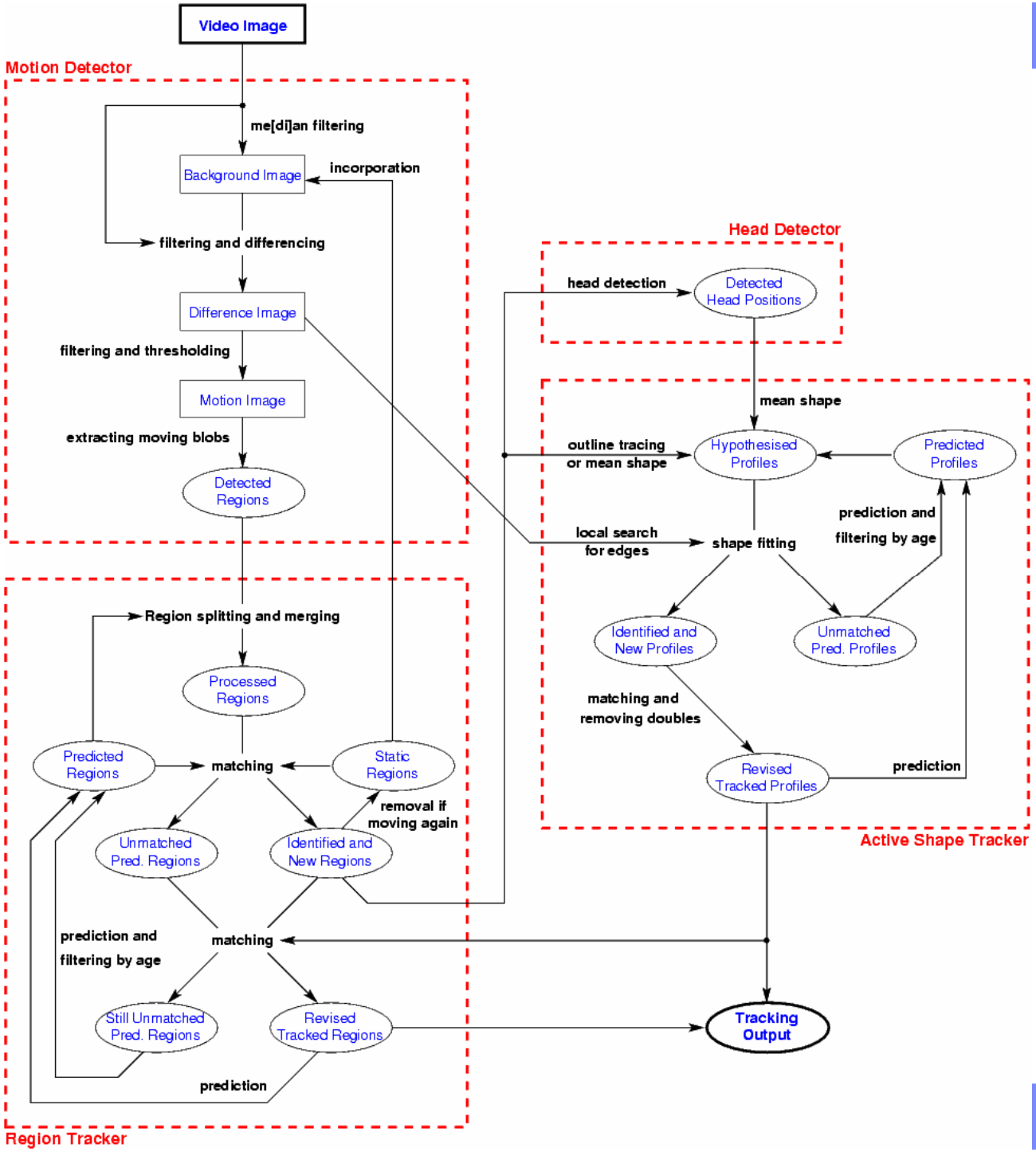
- Discount observation failures at known foreground occlusions
  - Don’t reduce probability
- Record track disappearance when it enters FGO regions

Baumberg “ADVISOR” tracker

- Baumberg 1995
- Detection by BGS.
- Modelling people by snakes

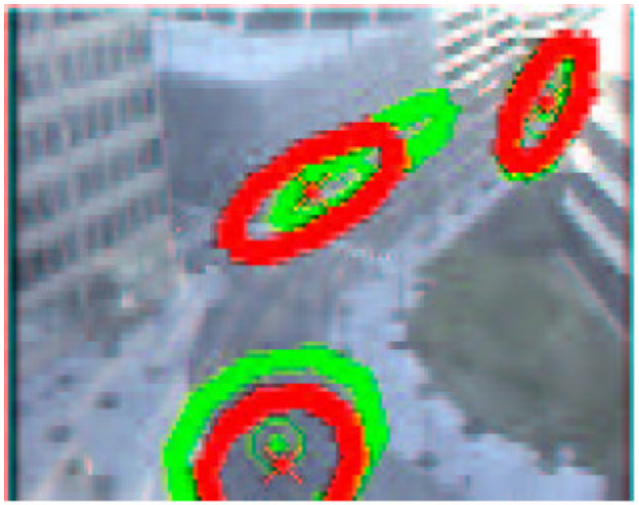


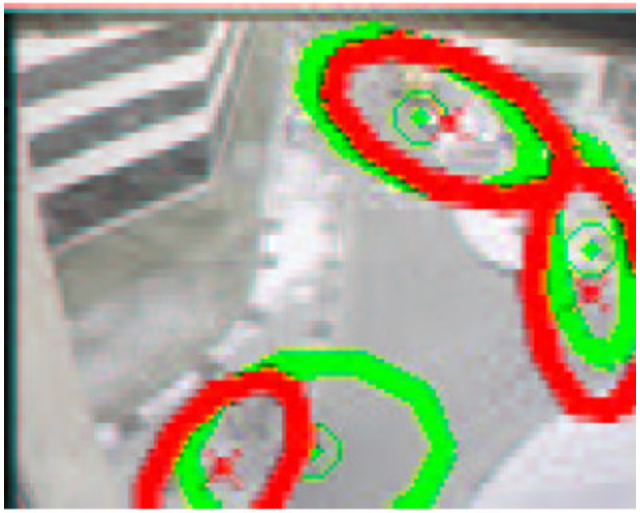




Track Sources and Sinks

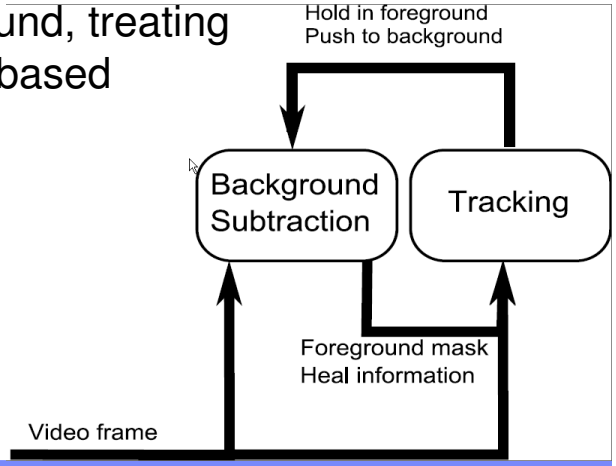
- Hand mark / learn where objects appear and disappear (see behaviour analysis class)
  - Stauffer “Estimating Sources and Sinks”
- Information can be used to distinguish between noise and true observations
  - A new object shouldn’t appear except at a source
  - Objects reaching a sink are likely to disappear





Interaction of tracking and background subtraction

- Often constructed as a modular, feed-forward system
  - Simpler analysis
- Tracking can inform background subtraction
- Object detection
  - BGS is a one-class classification problem
  - With a known object, 2-class classification is easier
    - Like Boult's threshold-with-hysteresis
- Tracker "understands" "objects"
  - Knows that an object is stopped or moving
  - Tracker can control when objects become part of background, treating them as unitary regions, whereas BGS must rely on pixel-based methods or region heuristics.



Tracking difficulties

- Many other tracking problems:
  - Fragmentation- BGS often fails. An object becomes two regions
    - new fragments are absorbed into nearby tracks untils split by fission
    - “Fusion” class accumulates evidence for nearby objects merging
  - Two objects may enter together and be indistinguishable until later
    - “Fission” class accumulates evidence for splitting object
  - One object leaves as another enters
    - Detect “Relay” tracks
  - One object occludes another for a long period
  - Objects stop and are “learned” by the background model
    - Tracker control over the BGS inhibits adaptation of tracked objects
    - Tracker forces push/pop to background model for truly static objects

References

- Into the woods: visual surveillance of non-cooperative and camouflaged targets in complex outdoor settings Terrance E. Boult□□ Ross J. Micheals Xiang Gao Michael Eckmann Vision And Software Technology (VAST) Laboratory, Lehigh University
- Gopal Pingali, Yves Jean and Ingrid Carlbom, "Real Time Tracking for Enhanced Tennis Broadcasts," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 1998, pp. 260-265.
- Chris Stauffer, Eric Grimson, "Learning Patterns of Activity Using Real-Time Tracking", IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI), 22(8):747-757, 2000.
- W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People Ismail Haritaoglu, David Harwood and Larry S. Davis International Conference on Face and Gesture Recognition, April 14-16, 1998,
- Estimating Tracking Sources and Sinks Chris Stauffer Proceedings of the Second IEEE Workshop on Event Mining, July 17, 2003
- Appearance Models for Occlusion Handling A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, R. Bolle in Journal of Image and Vision Computing Volume 24 Issue 11, pp 1233-1243 November 2006
- Fusion of Multiple Tracking Algorithms for Robust People Tracking Nils T Siebel and Steve Maybank ECCV 2002

Homework

- Read & write a short summary of
- **Real-Time Tracking of Non-Rigid Objects using Mean Shift (2000)** Comaniciu, Ramesh and Meer