

Outline

- Data Association Scenarios
 - Track Filtering and Gating
 - Global Nearest Neighbor (GNN)
 - Review: Linear Assignment Problem
 - Murthy's k-best Assignments Algorithm
- Probabilistic Data Association (PDAF)
 - Joint Probabilistic Data Assoc (JPDAF)
 - Multi-Hypothesis Tracking (MHT)
 - Markov Chain Monte Carlo DA (MCMCDA)

TODAY

Multi Target Tracking (MTT)

Scenario: Detect and Track

Given: N tracked target trajectories and M new sensor observations

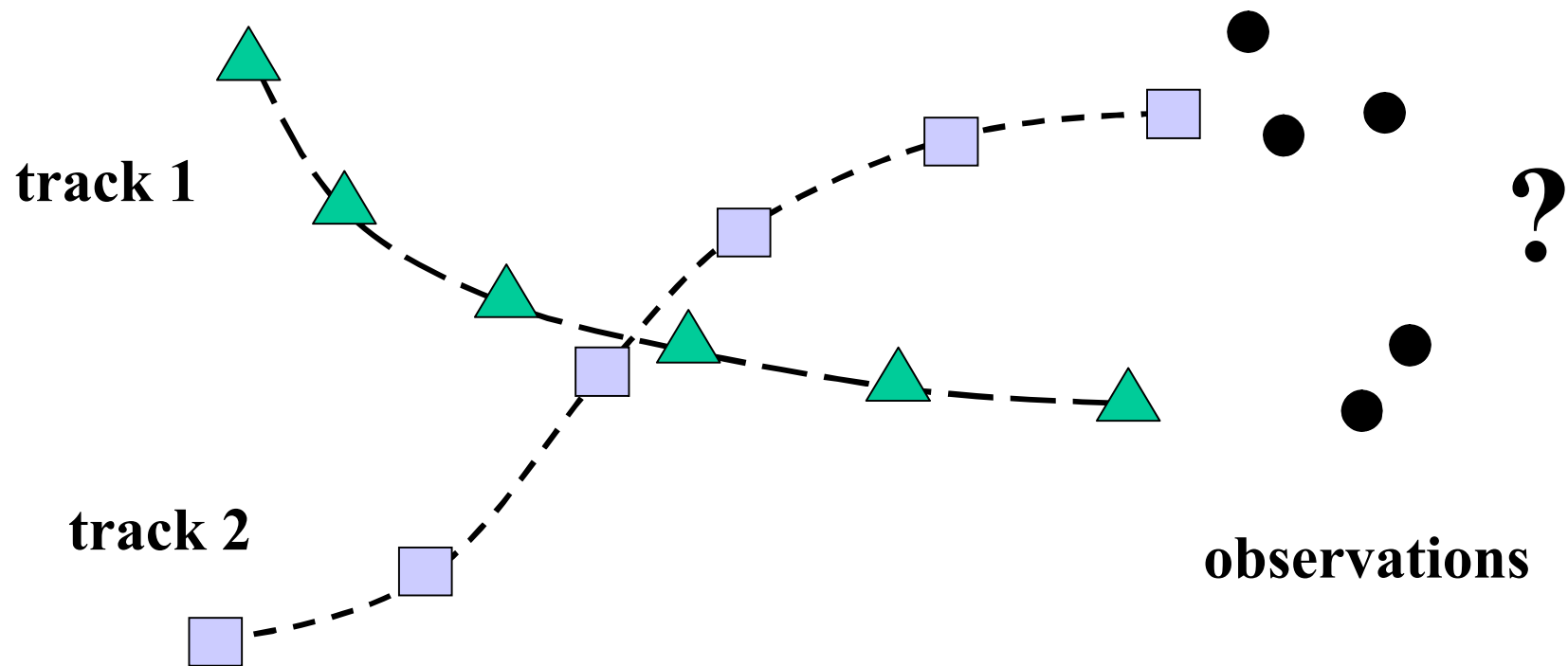
Problem: deciding which target generated which observation

Data Association = assigning sensor responses to trajectories

Note: if assignment was always unambiguous, we could turn MTT into independent single-target tracking problems. However, when the association is ambiguous (multiple observations in one gating region; common observations in multiple gating regions), then the assignment decisions become coupled and much harder to solve.

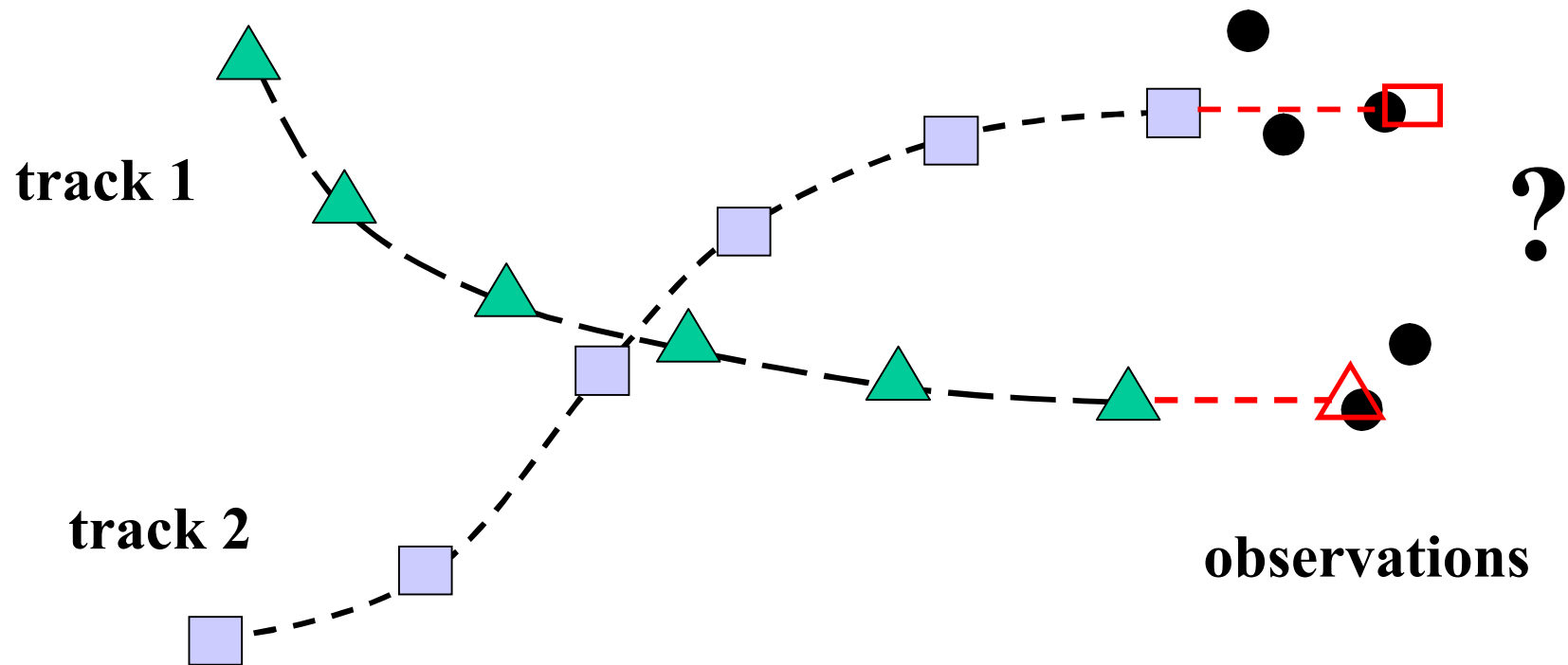
Recall: Data Assoc Scenario

Multi-frame Matching (matching observations in a new frame to a set of tracked trajectories)



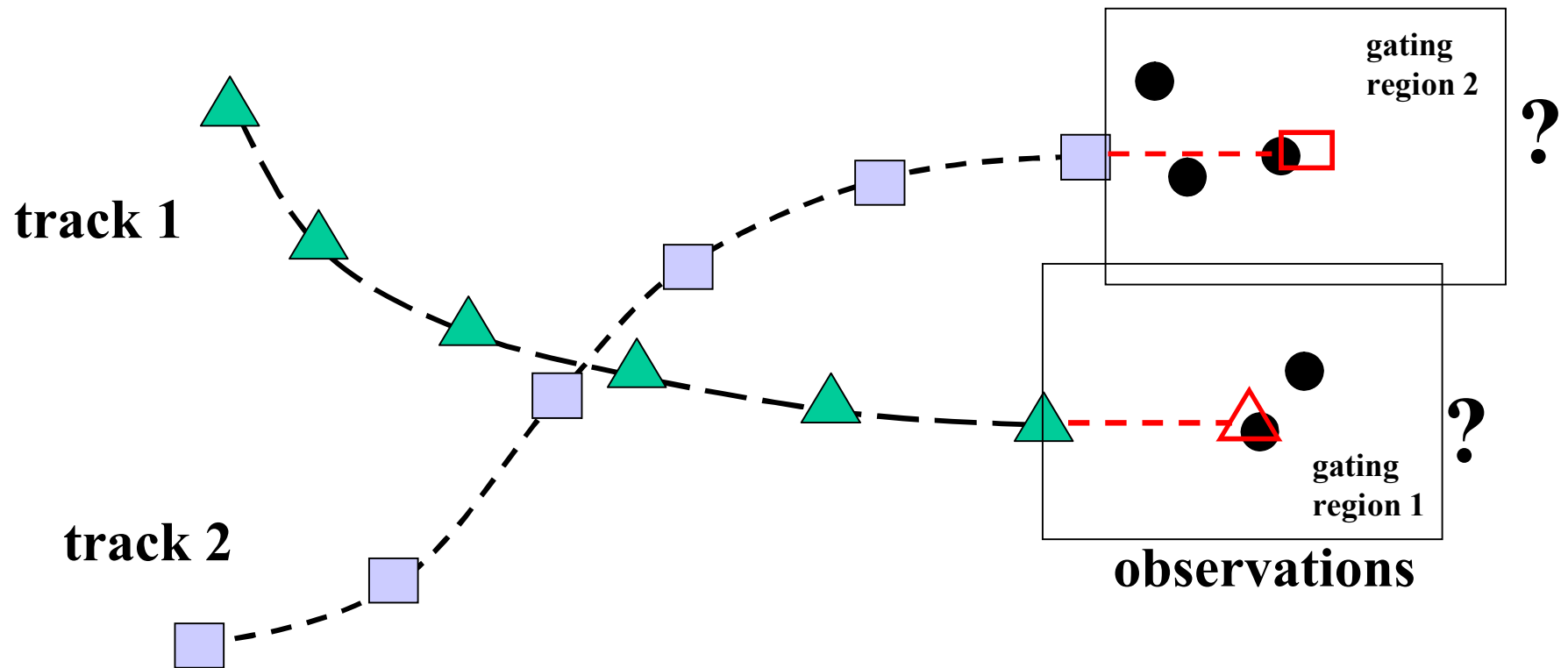
Recall: Data Assoc Scenario

Predict next target position along each track.



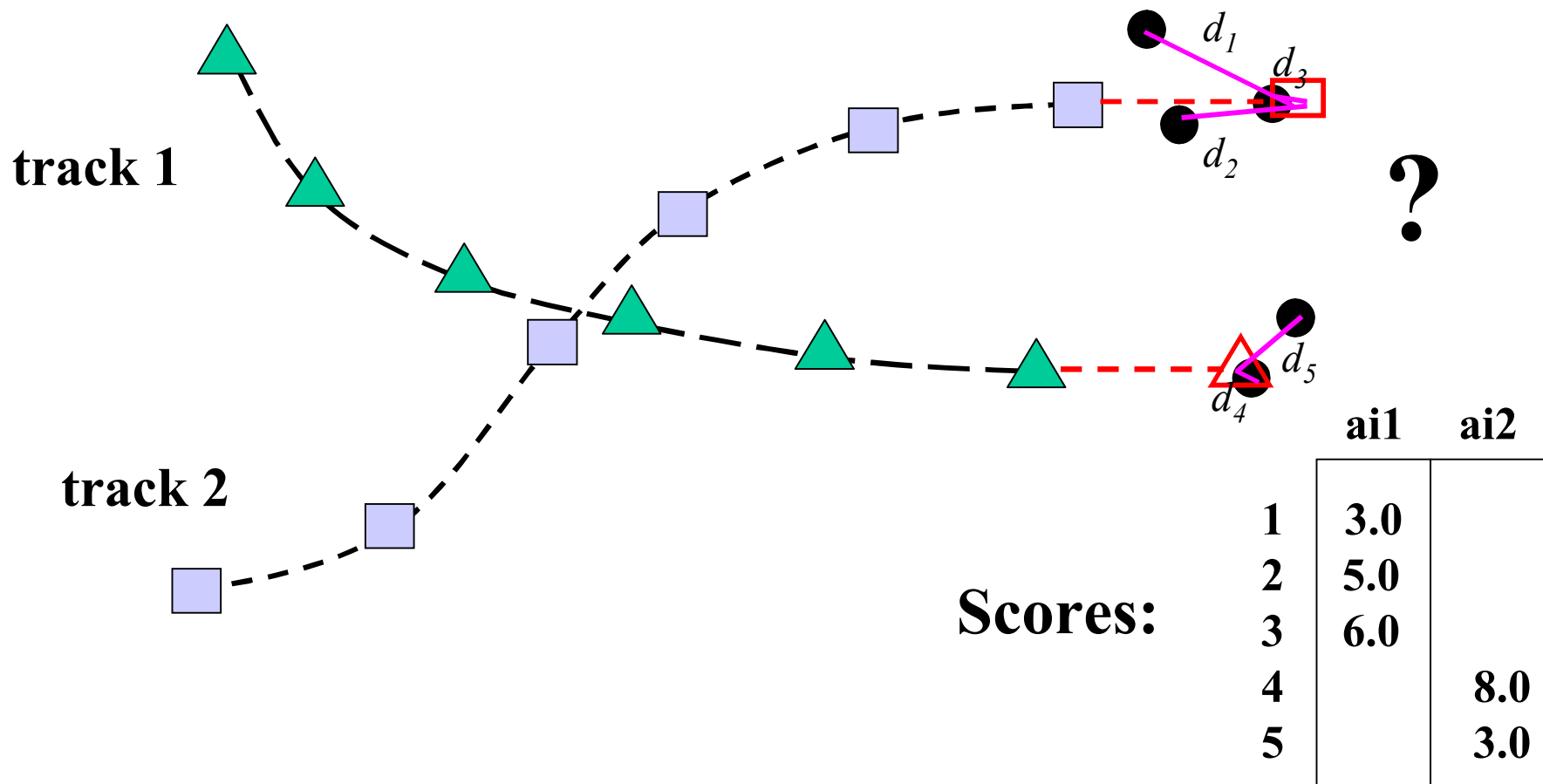
Recall: Data Assoc Scenario

Form a gating region around each predicted target location. This is a divide and conquer approach.



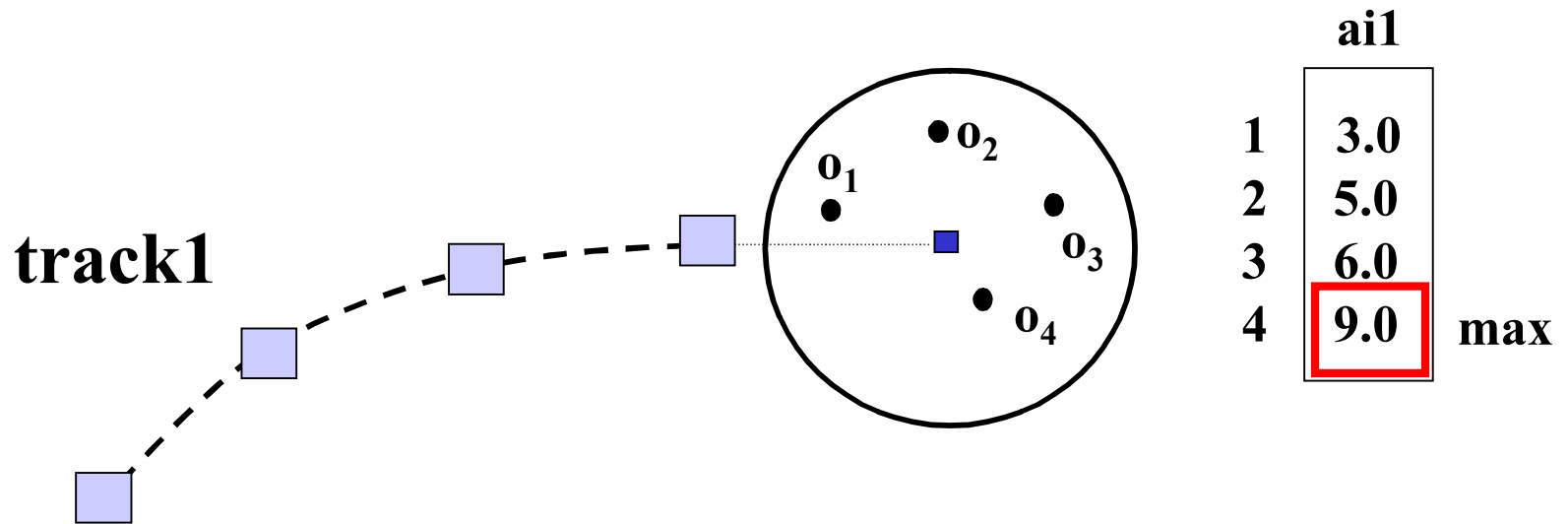
Recall: Data Assoc Scenario

For each candidate matches (a track-to-data pairing), compute match score based on likelihood of the data given the track.



Recall: Global Nearest Neighbor

Evaluate each observation in track gating region.
Choose “best” one to incorporate into track.



a_{i1} = score for matching observation i to track 1

Choose best match $a_{m1} = \max\{a_{11}, a_{21}, a_{31}, a_{41}\}$

PDAF

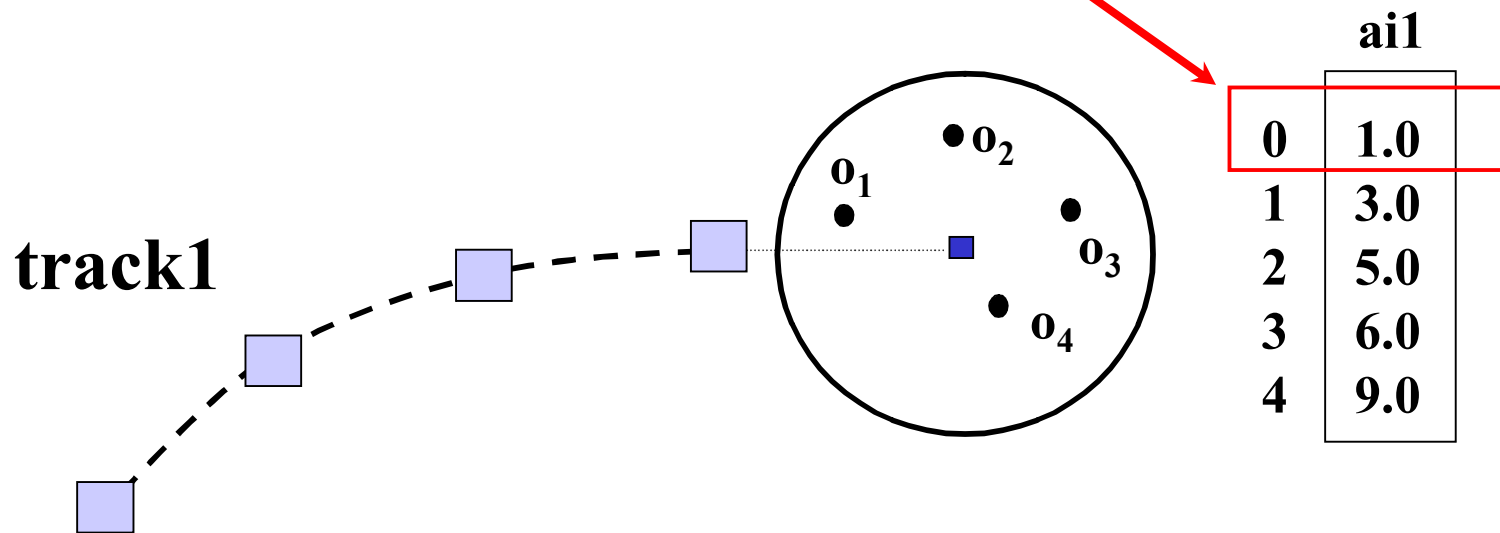
Probabilistic Data Association Filter

Updating single track based on new observations.

General idea: Instead of matching a single best observation to the track, we update based on all observations (in gating window), weighted by their likelihoods.

PDAF

Consider all points in gating window. Also consider the additional possibility that no observations match.



p_{i1} = “probability” of matching observation i to track 1

$$p_{i1} = \frac{a_{i1}}{\sum_{i=0}^n a_{i1}}$$

PDAF

Kalman filter update is based on residual vector (diff between predicted location and observed location)

When using single best observation

$$\tilde{y}_k = o_{\max} - H_k \hat{x}_{k|k-1}$$

PDAF uses weighted combination of observations

$$\tilde{y}_k = \sum_{i=1}^n p_{i1} (o_i - H_k \hat{x}_{k|k-1})$$

note: if we weren't consider the possibility of no match, this would exactly be the diff between the weighted center of mass of observations and the predicted location

PDAF

Computation of Kalman posterior covariance must change too, to incorporate weighted matches and possibility of no match.

Typical computation when single match is used:

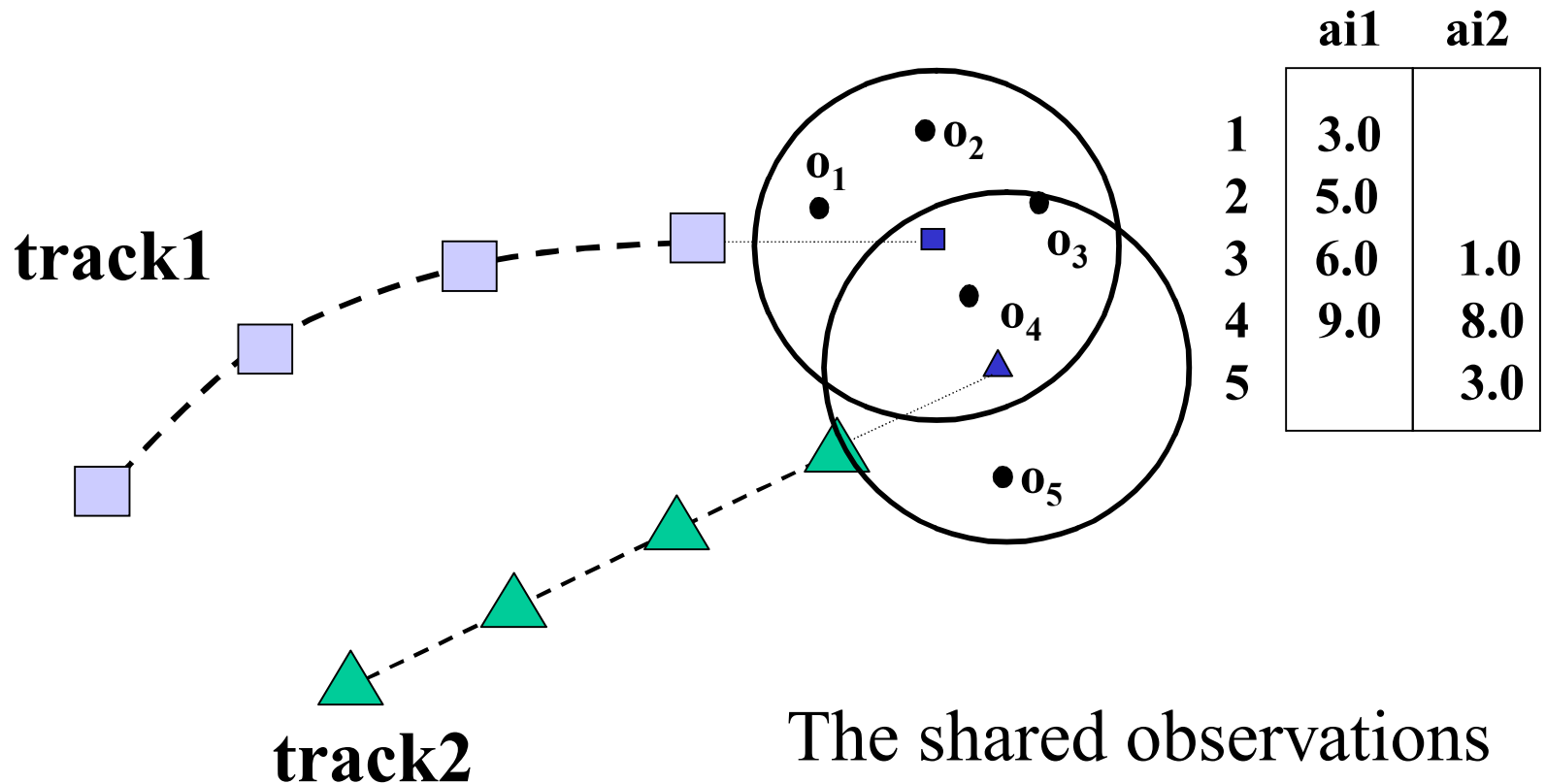
$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

PDAF computation:

$$\begin{aligned} P_{k|k} = & p_{01} P_{k|k-1} && \text{no match, no update} \\ & + (1 - p_{01}) [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] P_{k|k-1} && \text{update if any match} \\ & + \mathbf{K}_k \text{Cov}(\mathbf{o}_i - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \mathbf{K}_k' && \text{correction term to} \\ & && \text{reflect uncertain} \\ & && \text{association (spread} \\ & && \text{of possible matches} \\ & && \text{or no match)} \end{aligned}$$

Problem

When gating regions overlap, the same observations can contribute to updating both trajectories.



The shared observations introduce a coupling into the decision process.

JPDAF

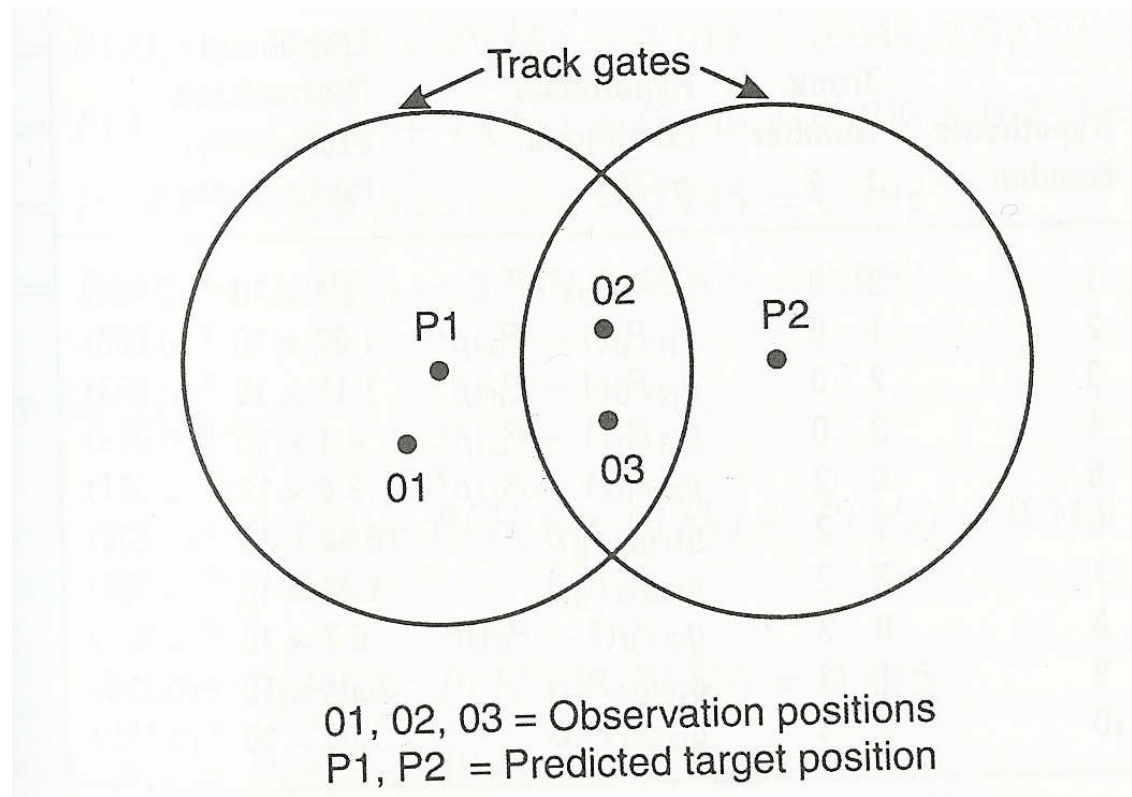
Joint Probabilistic Data Association Filter

If maintaining multiple tracks, doing PDAF on each one independently is nonoptimal, since observations in overlapping gate regions will be counted more than once (contribute to more than one track).

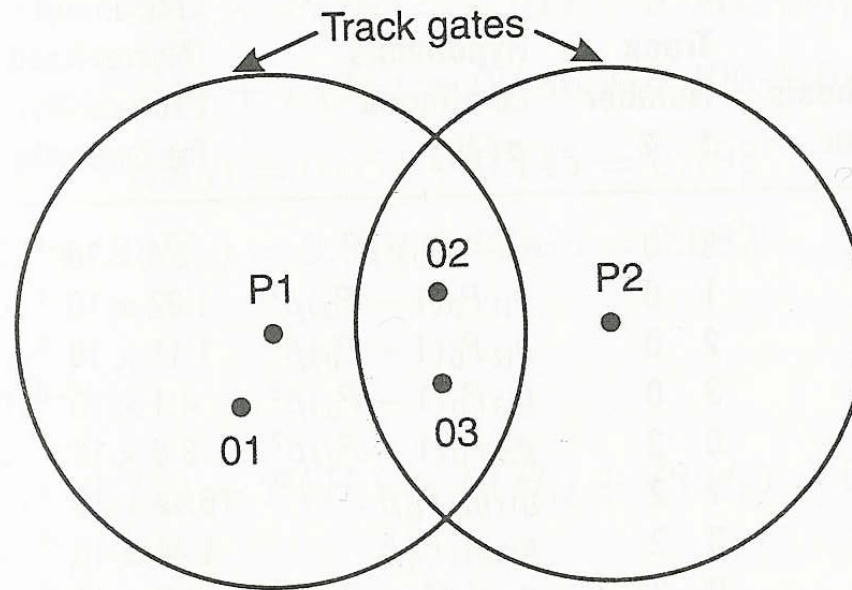
JPDAF reasons over possible combinations of matches, in a principled way.

JPDAF

Example (from Blackman and Popoli).



JPDAF



Candidates:

track1: 0 1 2 3

track2: 0 2 3

01, 02, 03 = Observation positions + Oo = no match
P1, P2 = Predicted target position

Possible assignments: (i,j) = assign i to track1, j to track2

(0, 0) (1, 0) (2, 0) (3, 0)

(0, 2) (1, 2) ~~(2, 2)~~ (3, 2)

(0, 3) (1, 3) (2, 3) ~~(3, 3)~~

**don't assign same
observation twice**

JPDAF

Each possible (non-conflicting) assignment becomes a hypothesis with an associated probability.

Table 6.8
Hypothesis Matrix for Example of Figure 6.3

Hypothesis Number	Track Number		Hypothesis Likelihood $p(H_i)$	Likelihood (Normalized Probability) for Example
	1	2		
1	0	0	$(1 - P_0)^2 \beta^3$	2.4×10^{-6} (0.011)
2	1	0	$g_{11} P_0 (1 - P_0) \beta^2$	1.82×10^{-5} (0.086)
3	2	0	$g_{12} P_0 (1 - P_0) \beta^2$	1.11×10^{-5} (0.053)
4	3	0	$g_{13} P_0 (1 - P_0) \beta^2$	4.1×10^{-6} (0.019)
5	0	2	$g_{22} P_0 (1 - P_0) \beta^2$	8.6×10^{-6} (0.041)
6	1	2	$g_{11} g_{22} P_0^2 \beta$	6.47×10^{-5} (0.306)
7	3	2	$g_{13} g_{22} P_0^2 \beta$	1.44×10^{-5} (0.068)
8	0	3	$g_{23} P_0 (1 - P_0) \beta^2$	6.7×10^{-6} (0.032)
9	1	3	$g_{11} g_{23} P_0^2 \beta$	5.04×10^{-5} (0.239)
10	2	3	$g_{12} g_{23} P_0^2 \beta$	3.06×10^{-5} (0.145)

we will talk more about these in a moment

JPDAF

Now compute probability p_{ij} that each observation i should be assigned to track j , by adding probabilities of assignments where that is so. Example: p_{11} = prob that observation should be assigned to track 1.

Table 6.8
Hypothesis Matrix for Example of Figure 6.3

Hypothesis Number	Track Number		Hypothesis Likelihood $p(H_i)$	Likelihood (Normalized Probability) for Example
	1	2		
1	0	0	$(1 - P_D)^2 \beta^3$	2.4×10^{-6} (0.011)
2	1	0	$g_{11} P_D (1 - P_D) \beta^2$	1.82×10^{-5} (0.086)
3	2	0	$g_{12} P_D (1 - P_D) \beta^2$	1.11×10^{-5} (0.053)
4	3	0	$g_{13} P_D (1 - P_D) \beta^2$	4.1×10^{-6} (0.019)
5	0	2	$g_{22} P_D (1 - P_D) \beta^2$	8.6×10^{-6} (0.041)
6	1	2	$g_{11} g_{22} P_D^2 \beta$	6.47×10^{-5} (0.306)
7	3	2	$g_{13} g_{22} P_D^2 \beta$	1.44×10^{-5} (0.068)
8	0	3	$g_{23} P_D (1 - P_D) \beta^2$	6.7×10^{-6} (0.032)
9	1	3	$g_{11} g_{23} P_D^2 \beta$	5.04×10^{-5} (0.239)
10	2	3	$g_{12} g_{23} P_D^2 \beta$	3.06×10^{-5} (0.145)

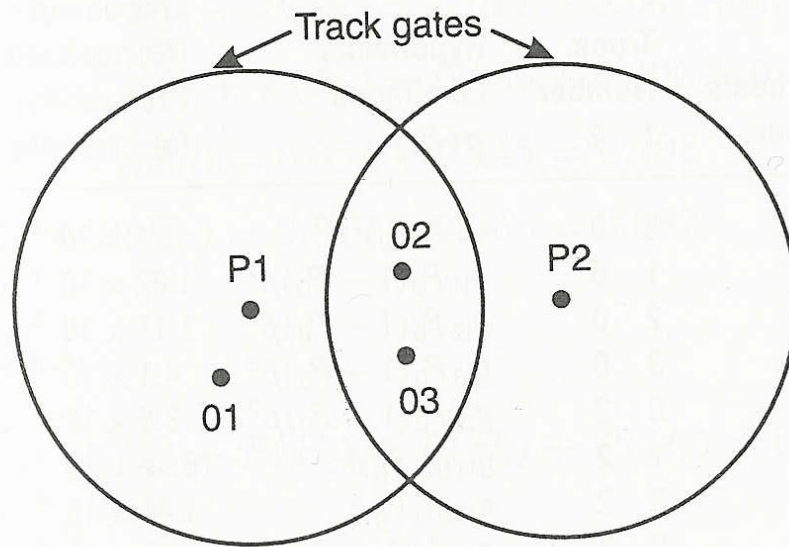
.086

+ .306

+ .239

.631

JPDAF



01, 02, 03 = Observation positions
P1, P2 = Predicted target position

Combined probabilities

Track1:

$$p_{10} = .084$$

$$p_{11} = .631$$

$$p_{12} = .198$$

$$p_{13} = .087$$

Track2:

$$p_{20} = .169$$

$$p_{21} = .0$$

$$p_{22} = .415$$

$$p_{23} = .416$$

JPDAF

Continuing the example:

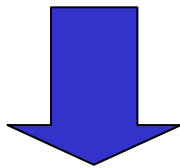
Track 1 probabilities

$$p_{01} = .084$$

$$p_{11} = .631$$

$$p_{21} = .198$$

$$p_{31} = .087$$



**PDAF filter
for track 1**

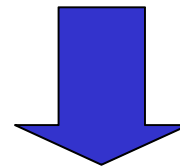
Track 2 probabilities

$$p_{02} = .169$$

$$p_{12} = 0.0$$

$$p_{22} = .415$$

$$p_{32} = .416$$



**PDAF filter
for track 2**

Note: running PDAF filters on each track independently is now OK because any inconsistency (double counting) has been removed.

Hypothesis Likelihoods

Table 6.8

Hypothesis Matrix for Example of Figure 6.3

Hypothesis Number	Track Number 1 2		Hypothesis Likelihood $p(H_i)$	Likelihood (Normalized Probability) for Example
1	0	0	$(1 - P_D)^2 \beta^3$	2.4×10^{-6} (0.011)
2	1	0	$g_{11} P_D (1 - P_D) \beta^2$	1.82×10^{-5} (0.086)
3	2	0	$g_{12} P_D (1 - P_D) \beta^2$	1.11×10^{-5} (0.053)
4	3	0	$g_{13} P_D (1 - P_D) \beta^2$	4.1×10^{-6} (0.019)
5	0	2	$g_{22} P_D (1 - P_D) \beta^2$	8.6×10^{-6} (0.041)
6	1	2	$g_{11} g_{22} P_D^2 \beta$	6.47×10^{-5} (0.306)
7	3	2	$g_{13} g_{22} P_D^2 \beta$	1.44×10^{-5} (0.068)
8	0	3	$g_{23} P_D (1 - P_D) \beta^2$	6.7×10^{-6} (0.032)
9	1	3	$g_{11} g_{23} P_D^2 \beta$	5.04×10^{-5} (0.239)
10	2	3	$g_{12} g_{23} P_D^2 \beta$	3.06×10^{-5} (0.145)

P_D is prob of detect
 B is prob of false alarm
 g_{ij} likelihood of observe j
 given track i

$$P(H) = \prod_{\text{Track } i \text{ assigned to observation } j} g_{ij} P_D \prod_{\text{Tracks assigned to no match (0)}} (1 - P_D) \prod_{\text{Unassigned observations}} B$$

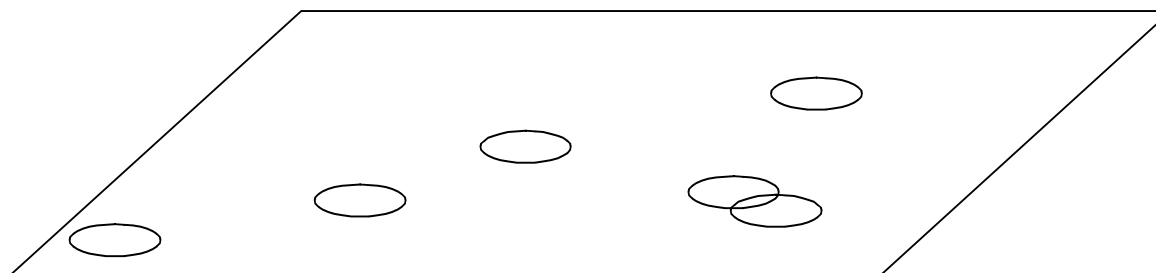
Multi-Hypothesis Tracking

Basic idea: instead of collapsing the 10 hypotheses from the last example into two trajectory updates, maintain and propagate a subset of them, as each is a possible explanation for the current state of the world.

This is a delayed decision approach. The hope is that future data will disambiguate difficult decisions at this time step.

Multi-Hypothesis Tracking

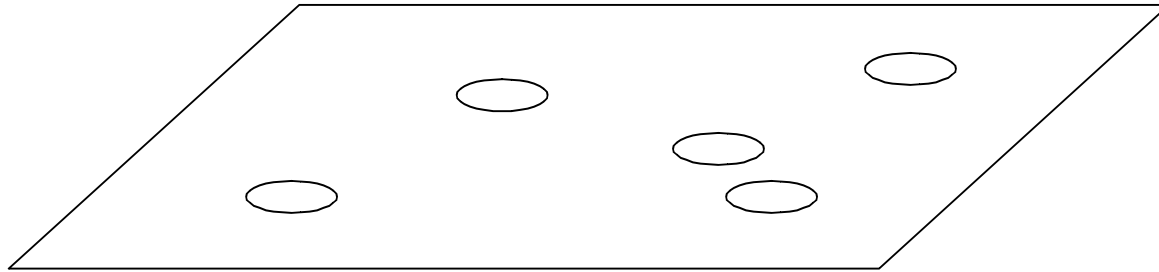
scan t



6 contacts

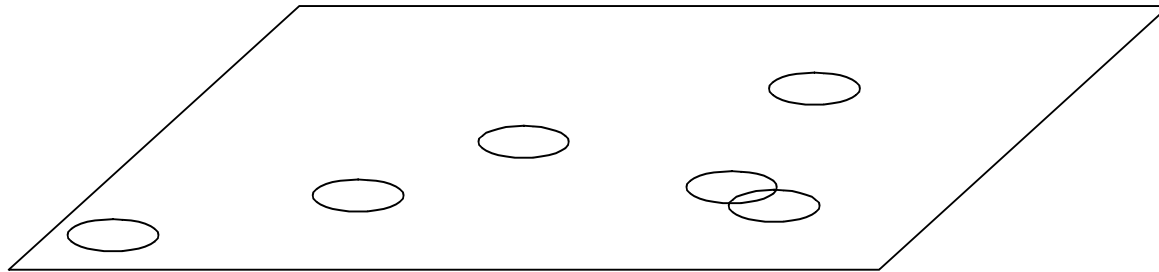
Multi-Hypothesis Tracking

scan t-1



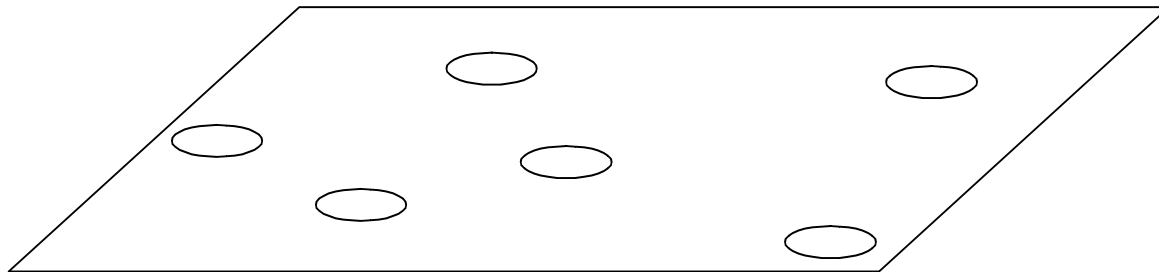
5 contacts

scan t



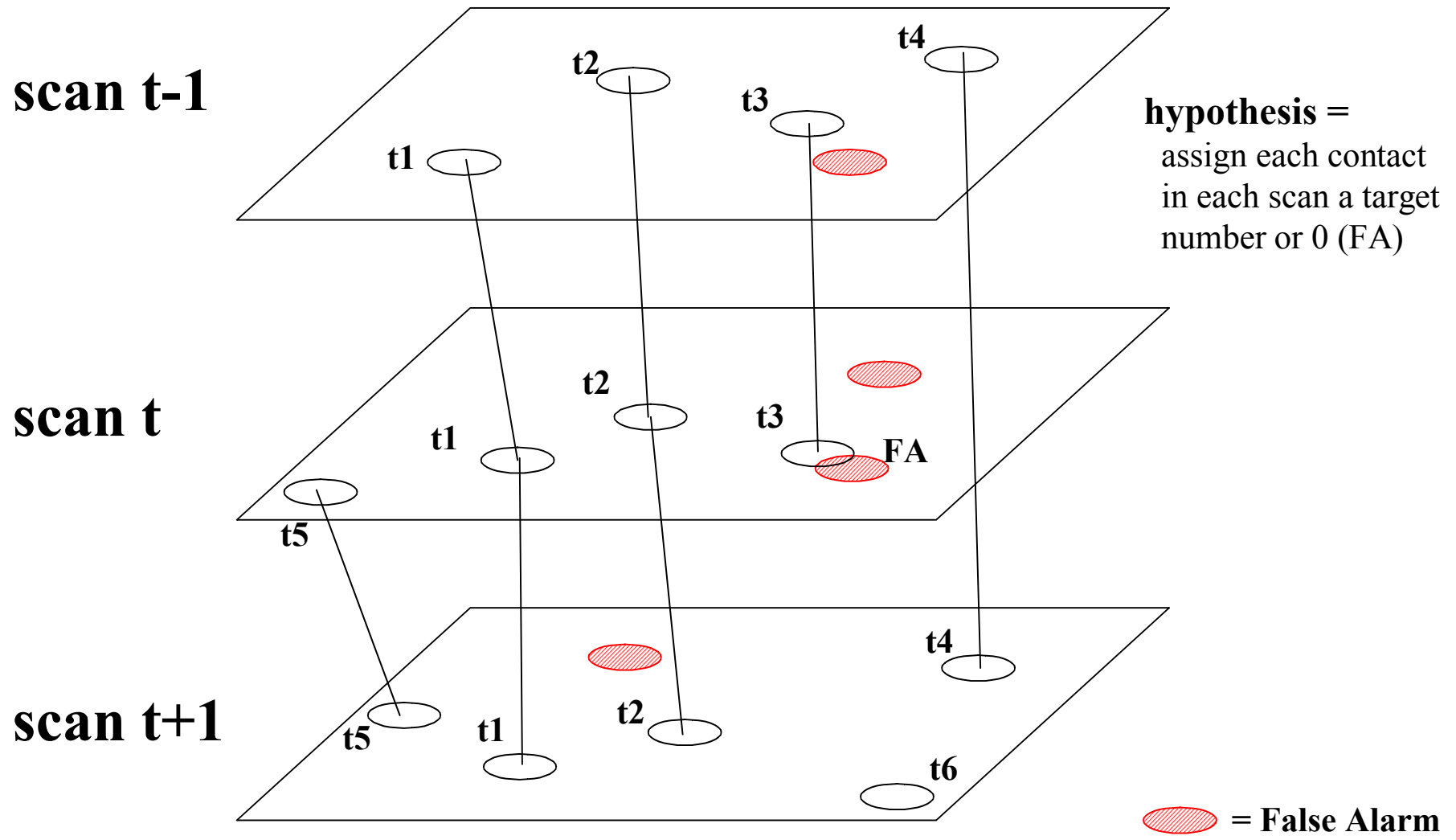
6 contacts

scan t+1

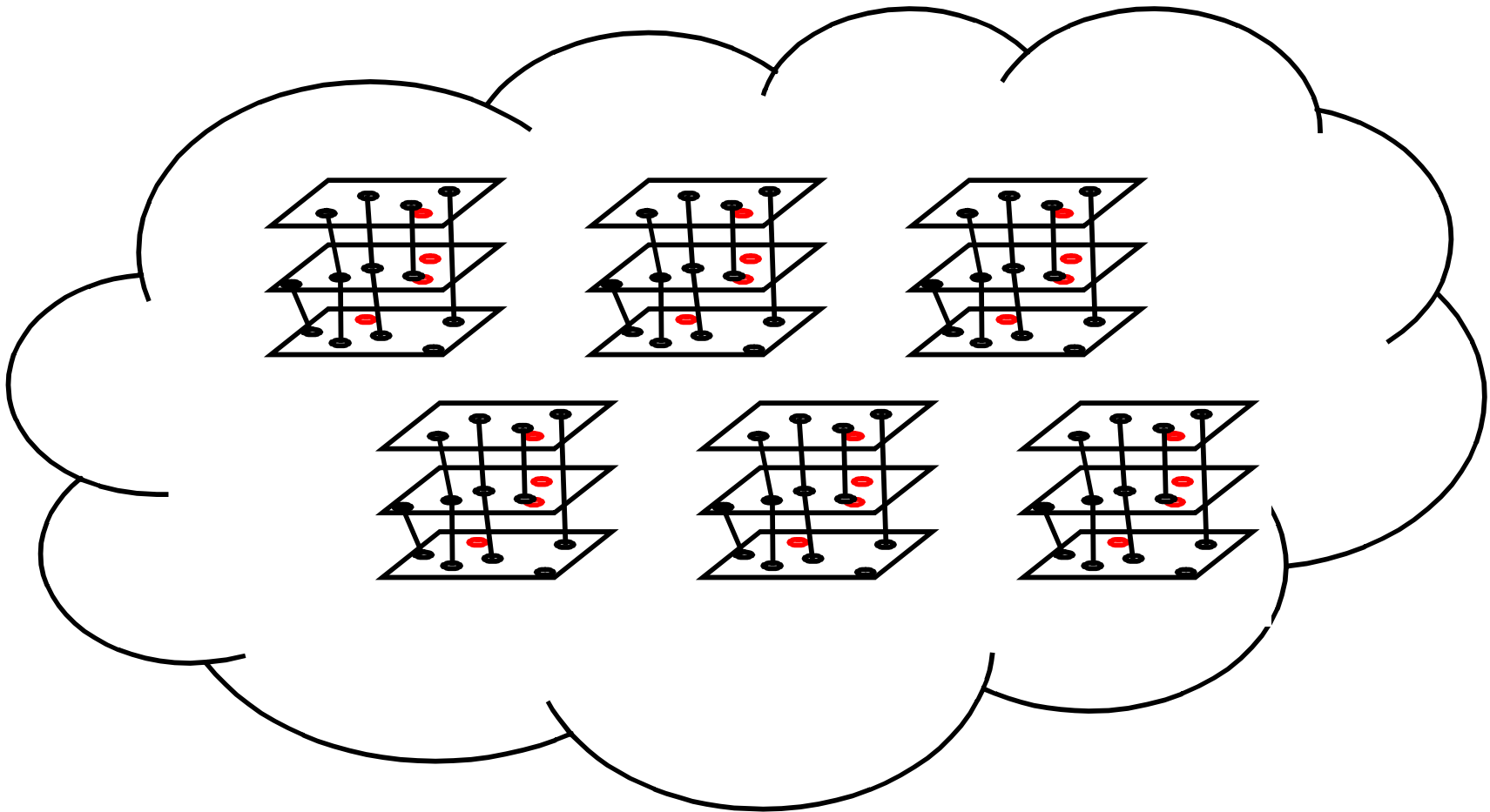


6 contacts

Multi-Hypothesis Tracking



Multi-Hypothesis Tracking

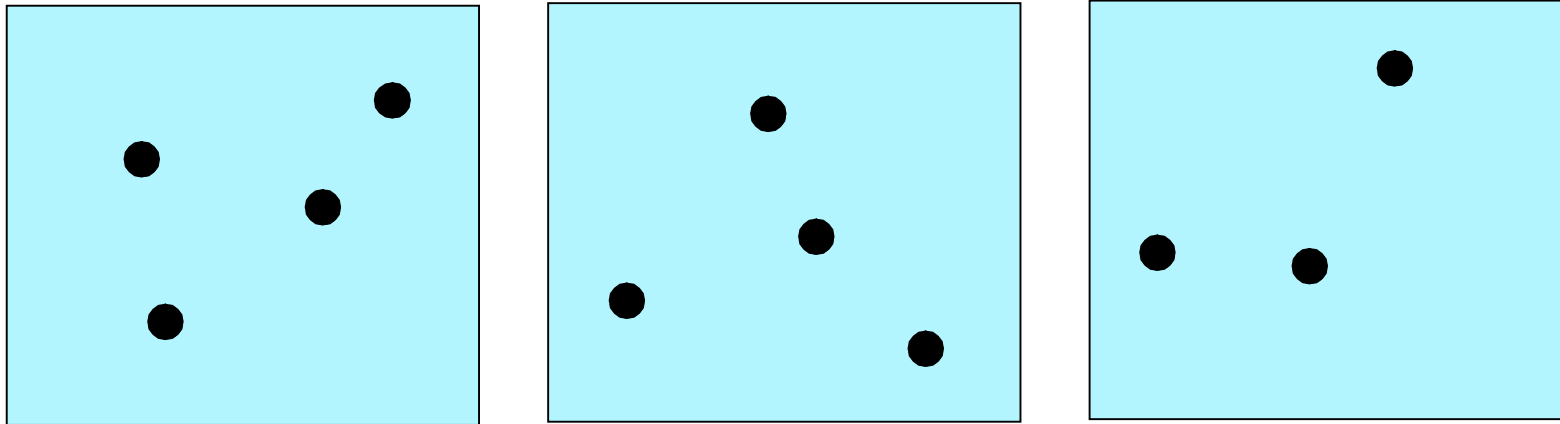


We maintain a set of such hypotheses. Each is one possible set of assignments of observations to targets or false alarms.

Combinatorial Explosion

Rough order of magnitude on number of hypotheses:

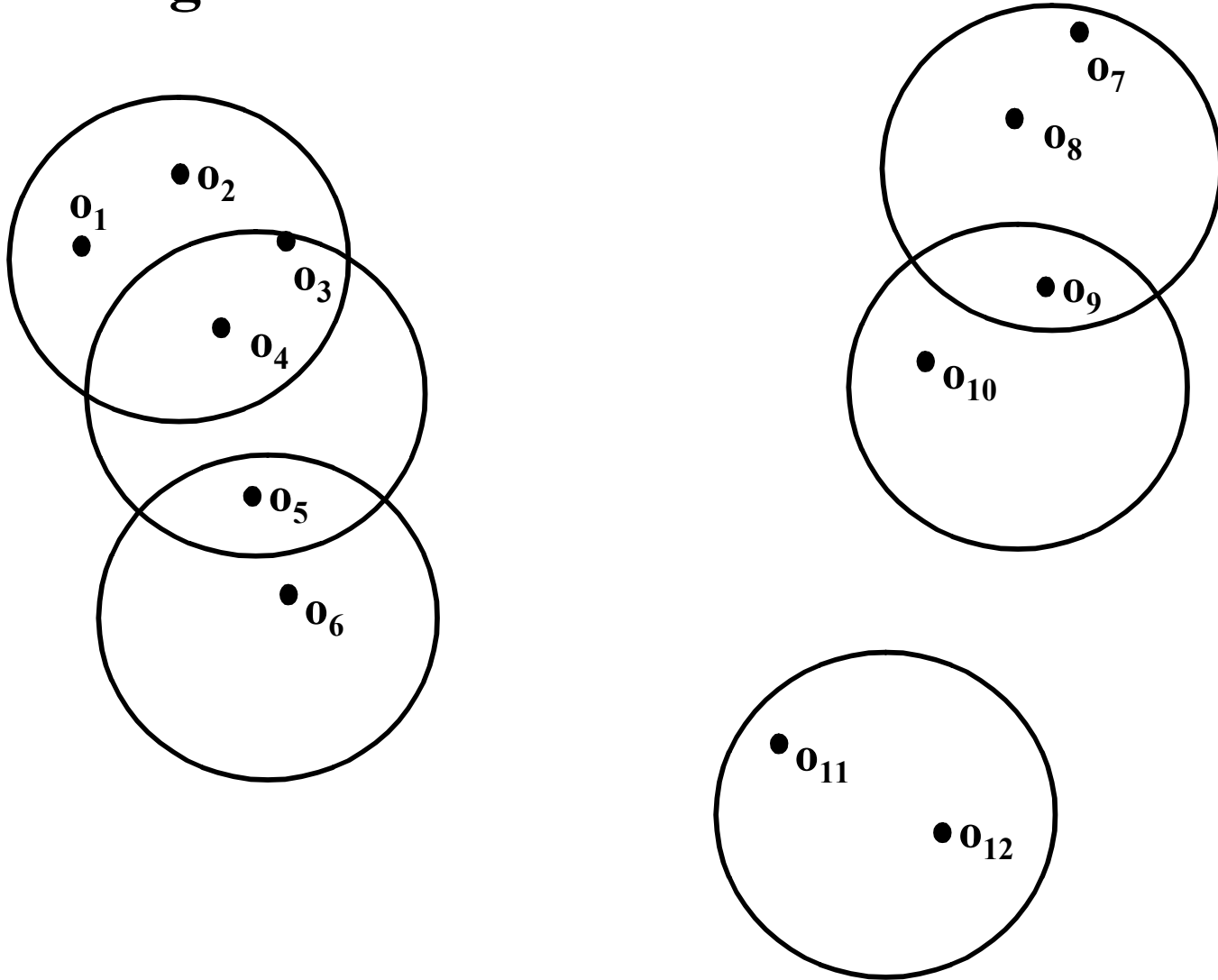
Let's say we have an upper bound N on number of targets and we can associate each contact in each scan a number from 1 to N. (we are ignoring false alarms at the moment)



$$\frac{N!}{(N-4)!} * \frac{N!}{(N-5)!} * \frac{N!}{(N-3)!}$$

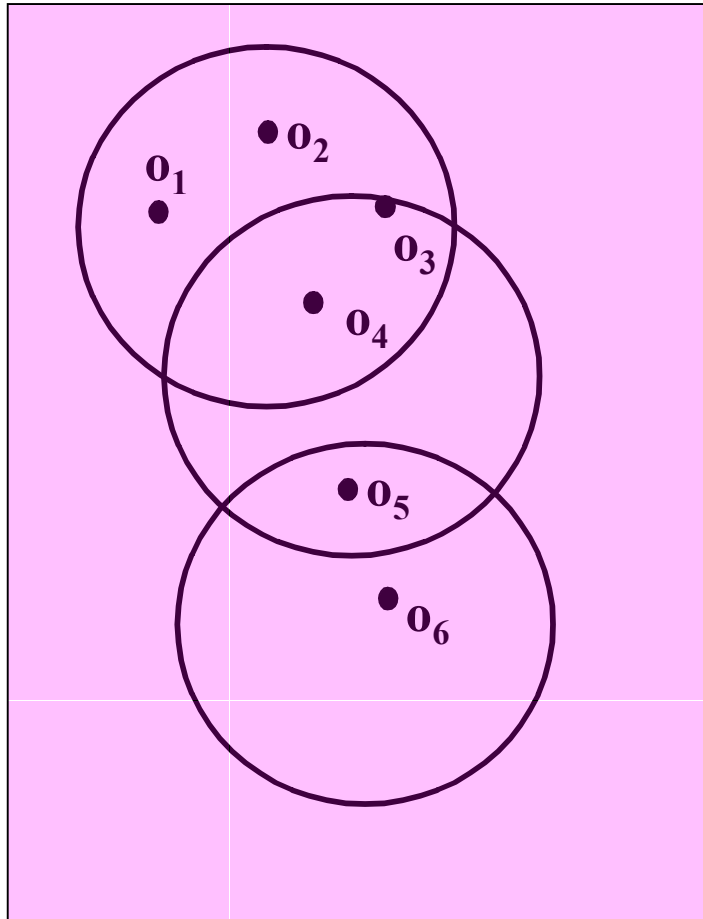
Mitigation Strategies

Clustering

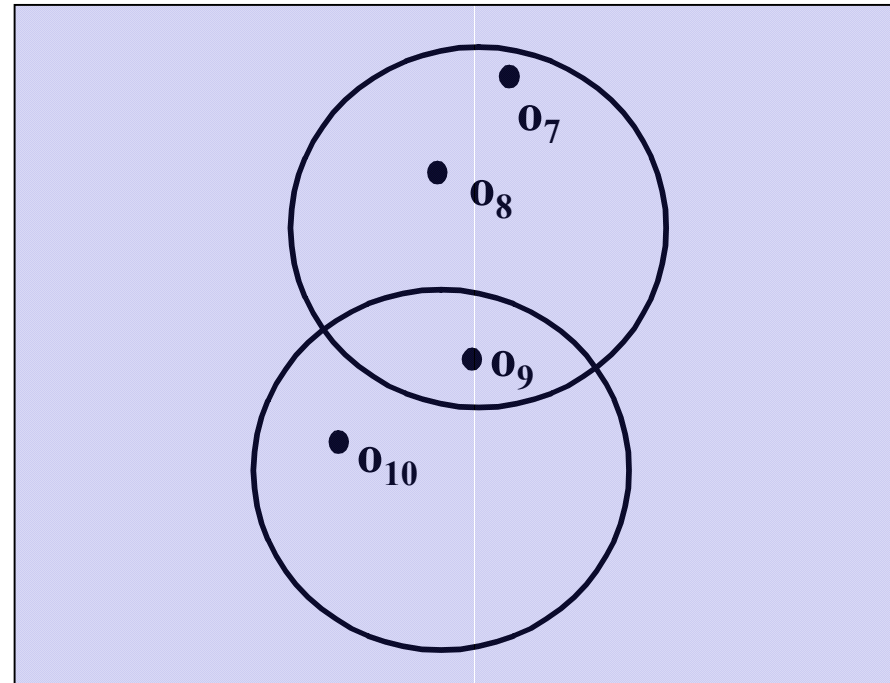


Mitigation Strategies

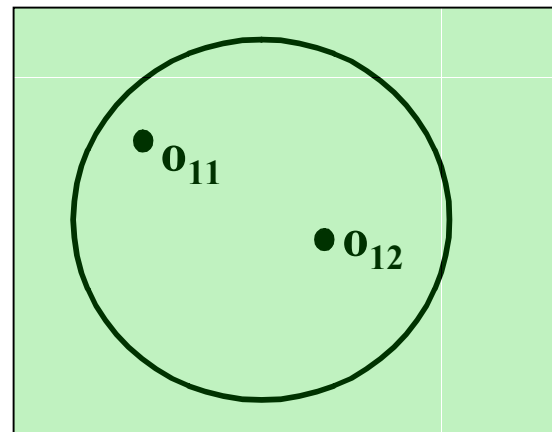
Clustering: can analyze each cluster independently (e.g. on a separate processor)



cluster1



cluster2



cluster3

Mitigation Strategies

Track Merging

merge similar trajectories (because this might allow you to merge hypotheses)

- **common observation history**
(e.g two tracks having the last N observations in common)
- **similar current state estimates**
(e.g same location and velocity in Kalman Filter)

Mitigation Strategies

Pruning: Discard low probability hypotheses

For example, one hypothesis that is always available is that every contact ever observed has been a false alarm! However, that is typically a very low probability event.

One of the most principled approaches to this is by Cox and Hingorani (PAMI'96). They combine MHT with Murty's k-best assignment algorithm to maintain a fixed set of k best hypotheses at each scan.

MHT with Murty's k-best

One way to avoid combinatorial explosion is to fix the number of hypotheses maintained at each frame, and use Murty's method for k-best assignments to find the k best hypotheses.

Example: let's say we want 5 hypotheses at each stage:

Given the 5 old hypotheses from time $t-1$, perform Hungarian algorithm to find best assignment of the observations in the current frame to each of them, forming new hypotheses at time t . Rank order them by hypothesis likelihood and put them in a priority queue: $\{H1, H2, H3, H4, H5\}$.

MHT with Murty's k-best

Now perform Murty's method using H_1 , to find the k highest variants of H_1 . Let's say $k=3$, and those variants are H_{11} , H_{12} , H_{13} .

By insertion sort, put them in with the original list of hypotheses, bumping out any hypotheses as necessary to keep a total list length of 5.

For example, we might now have $\{H_1, H_{11}, H_2, H_3, H_{12}\}$, where H_4 , H_5 and H_{13} have been discarded

Now perform Murty's method on the next item in the list, which is H_{11} , and so on. If H_{11} had been less than H_2 in score, then Murty's would have been performed on H_2 instead.

MCMCDA

Idea: use Markov Chain Monte Carlo (MCMC) to sample from / explore the huge combinatorial space of hypotheses.

S. Oh, S. Russell, and S. Sastry, 2004. Markov Chain Monte Carlo data association for general multiple-target tracking problems. In Proc. IEEE Int. Conf. on Decision and Control, pages 735–742, 2004.

Yu, G. Medioni, and I. Cohen, 2007. Multiple target tracking using spatio-temporal Markov Chain Monte Carlo data association. In Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pages 1–8, 2007.

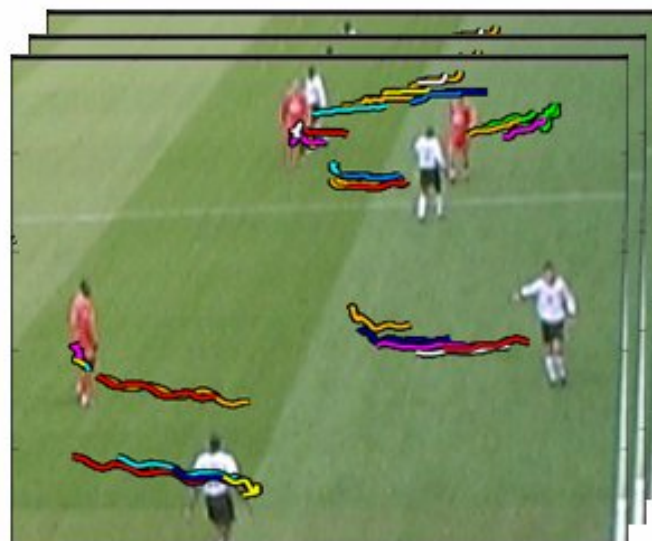


W.Ge and R.Collins, 2008, "Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation," British Machine Vision Conference (BMVC'08), University of Leeds, September 2008, pp. 935-944.

MCMCDA

Rather than use detections in each frame, first extract a set of “tracklets” by tracking detections through short subsequences of the original video.

For example, detection “seeds” at every 10th frame are tracked through the next 30 frames (1 second) of video.



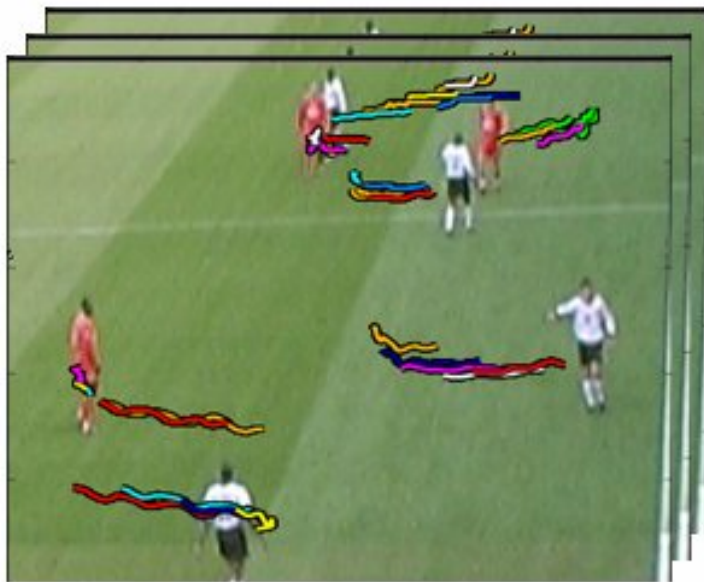
raw tracklets

Why?

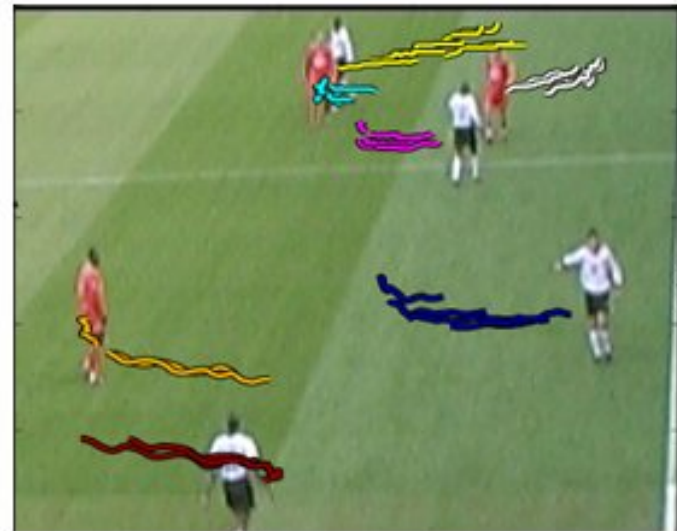
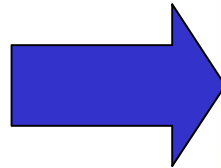
- tracklets provide more spatial/temporal context than raw detections
- short tracklets can be generated by simple (fast) trackers
- less prone to drift/occlusion than longer tracks

MCMCDA

Problem we are trying to solve: Find a partition of the set of overlapping tracklets such that tracklets belonging to the same object are grouped together. They could obviously be merged after that by a postprocessing stage.



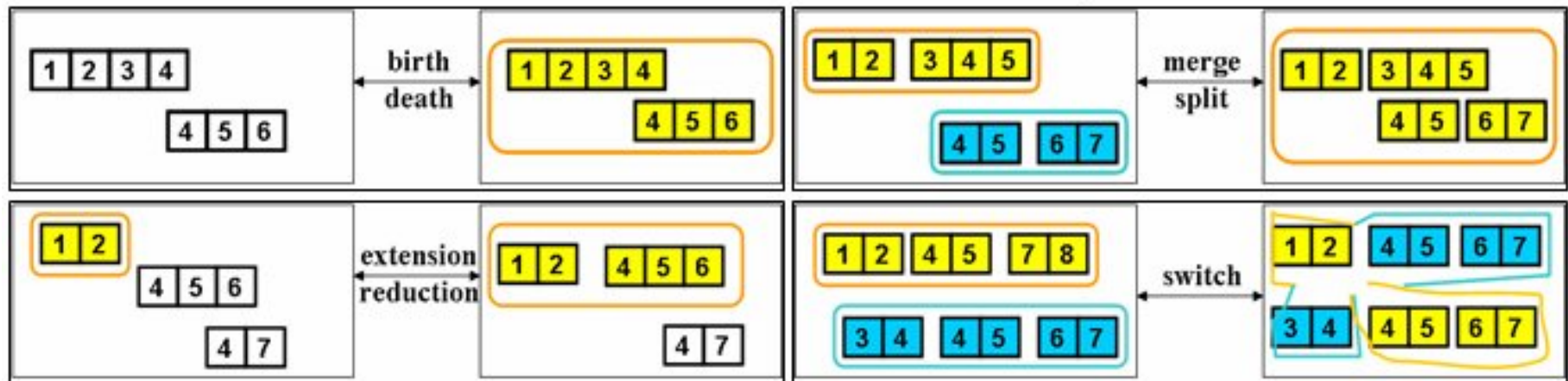
raw tracklets



estimated tracklet partition

MCMCDA

Recall that MCMC stochastically explores the search space (of tracklet partitions, in this example) by proposing a set of “moves” from the current state to a new state.



MCMC moves

MCMCDA

MCMC then decides whether or not to accept the proposal based, in part, on the ratio of likelihoods of the current state and the proposed state.

Z=observed tracklets
w = partition

$$\omega^* = \arg \max_{\omega} (p(\omega|Z)) \xleftrightarrow{\text{Bayes' rule}} \arg \max_{\omega} (\underbrace{p(Z|\omega)}_{\text{likelihood}} \underbrace{p(\omega)}_{\text{prior}})$$

The likelihood and prior are calculated as functions of 8 features:

Likelihood features

Color Appearance

Object Size

Spatial Proximity

Velocity Coherence

Prior features

False Alarm Rate

Trajectory Length

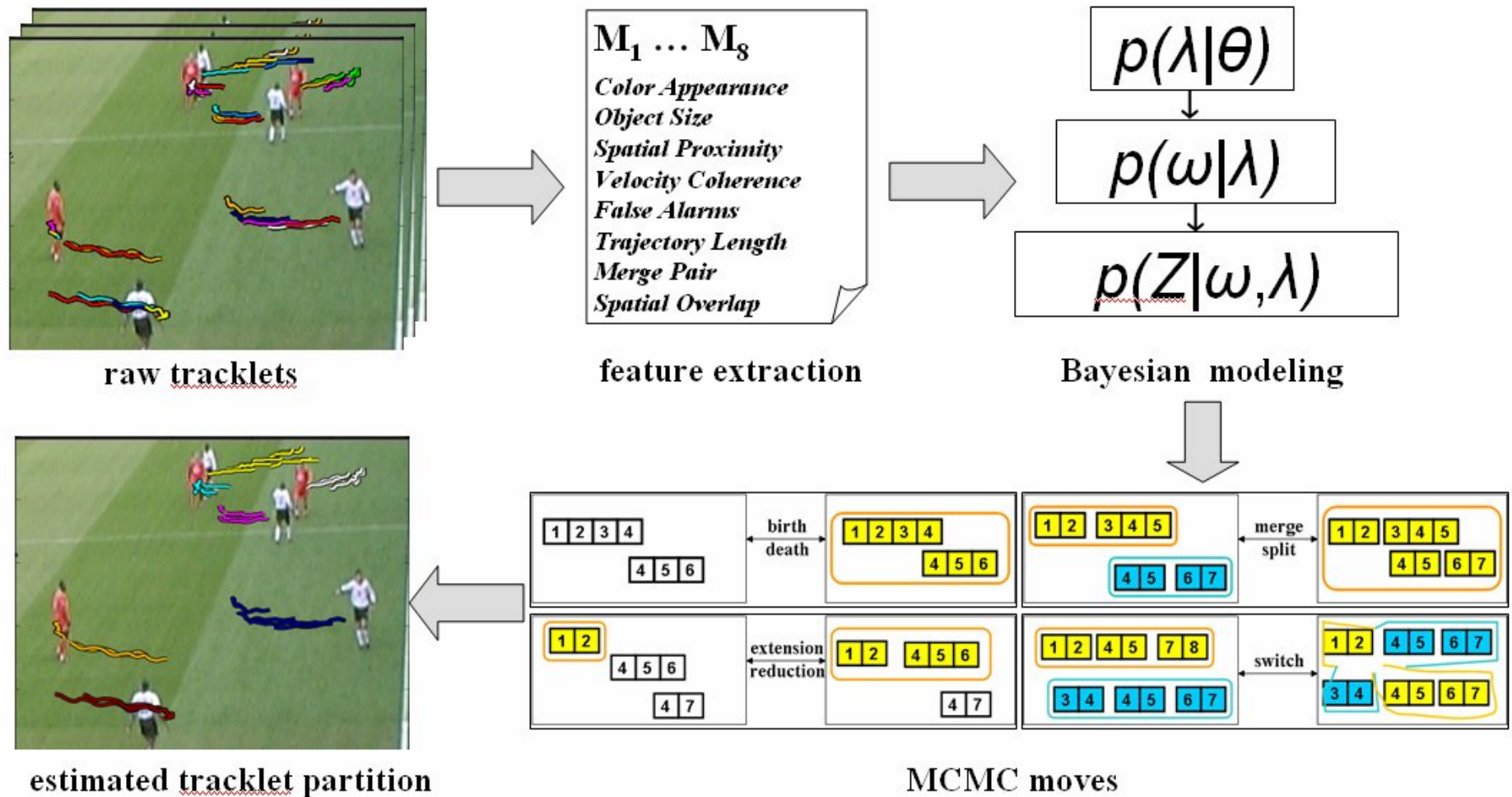
Merge Pair (encourage merging rather than starting new tracks)

Spatial Overlap (discourage overlaps between diff tracks)

note, she actually solved a harder problem of also inferring hyperparameters from the data

MCMCDA

Putting it all together:



MCMCDA

input tracklets

hypothesized tracks (at some time)

