Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density

Chris Kreucher, Keith Kastella, Alfred Hero

This work was supported by United States Air Force contract #F33615-02-C-119, Air Force Research Laboratory contract #SPO900-96-D-0080 and by ARO-DARPA MURI Grant #DAAD19-02-1-0262.

Overview

- We present a method of tracking multiple targets based on recursive estimation of the Joint Multitarget Probability Density (JMPD).
  - This is different from traditional multitarget tracking algorithms such as MHT, JPDA, etc. in that we are interested in estimating the full joint multitarget density.
- We give a particle filter algorithm for recursively estimating the JMPD
  - The particle filter algorithm uses an adaptive sampling scheme that exploits the multitarget nature of the problem.
  - We show that the particle filter implementation provides computational tractability.
- We detail the inherent permutation symmetry associated with JMPD
  - Permutation symmetry is inherent in any multitarget tracker
  - This symmetry manifests itself in the particle filter implementation as *partition swapping*.
  - We show that the partition swapping is automatically removed through repeated resampling

Single Target Bayesian Filtering Paradigm & Notation

- The state of an individual target is modeled by $\mathbf{x}$, e.g. $\mathbf{x} = [x\ \dot{x}\ y\ \dot{y}]'$
- We model the state at time *k* probabilistically using $p\left(\mathbf{x}^k \mid \mathbf{Z}^k\right)$

  the state to be estimated based on a sequence of noisy measurements taken over *k* time steps, $\mathbf{Z}^k = \{\mathbf{z}^1 \cup \mathbf{z}^2 \ldots \cup \mathbf{z}^k\}$
- The target motion is modeled as Markov using a Kinematic prior

$$p(\mathbf{x}^k \mid \mathbf{x}^{k-1})$$

- The sensor output is modeled using

$$p(\mathbf{z}^k \mid \mathbf{x}^k)$$

We allow for the target motion to be non-linear, the measurement to state coupling to be non-linear, and that posterior density to be non-Gaussian.

Single Target Bayesian Filtering Paradigm & Notation

- Bayes' rule and the Chapman-Kolmogorov equation give the procedure for incorporating measurements and evolving the density through time:

<u>Prediction</u> (generating the Kinematic prior)

$$p\left(\mathbf{x}^k \mid \mathbf{Z}^{k-1}\right) = \int p\left(\mathbf{x}^k \mid \mathbf{x}^{k-1}\right) p\left(\mathbf{x}^{k-1} \mid \mathbf{Z}^{k-1}\right) d\mathbf{x}^{k-1}$$

<u>Update</u> (Bayes' rule to Incorporate Measurements)

$$p\left(\mathbf{x}^k \mid \mathbf{Z}^k\right) = \frac{p\left(\mathbf{z}^k \mid \mathbf{x}^k\right) p\left(\mathbf{x}^k \mid \mathbf{Z}^{k-1}\right)}{p\left(\mathbf{z}^k \mid \mathbf{Z}^{k-1}\right)}$$

$$\text{where } p\left(\mathbf{z}^k \mid \mathbf{Z}^{k-1}\right) = \int p\left(\mathbf{z}^k \mid \mathbf{x}^k\right) p\left(\mathbf{x}^k \mid \mathbf{Z}^{k-1}\right) d\mathbf{x}^k$$

- In the general setting of non-linear target kinematics, non-linear measurements and non-Gaussian densities, an analytic solution for these recursions does not exist and so approximate techniques are required

Preliminaries : Tracking a Single Target Using a Particle Filter

- Approximate the density by a set of weighted samples (particles)

$$p(\mathbf{x} \mid \mathbf{Z}) \approx \sum_{p=1}^{N} w_p d(\mathbf{x} - \mathbf{x}_p)$$

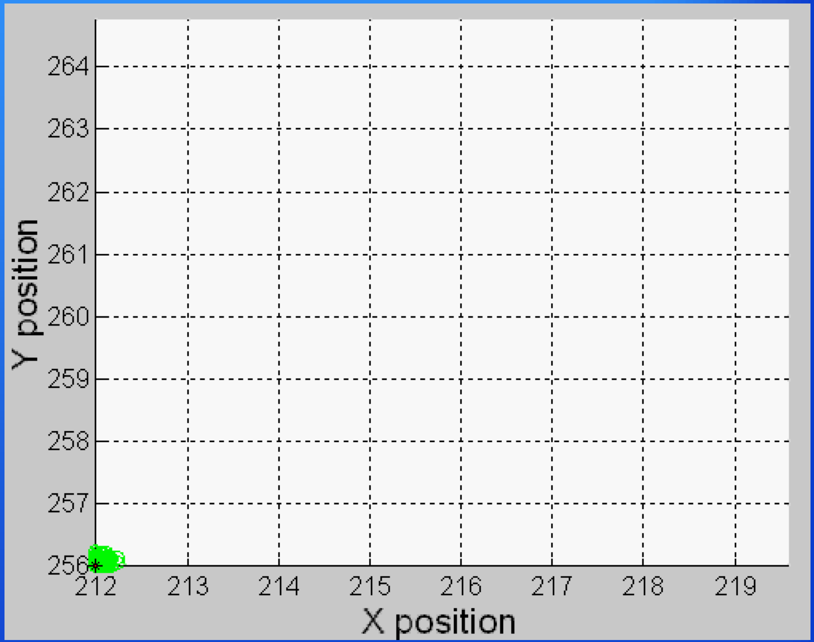
- At each time step, propose new particles from the existing particles based on the importance density

$$q\left(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{Z}^k\right) \approx p\left(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{Z}^k\right)$$

- Weight the particles based on the measurement likelihood

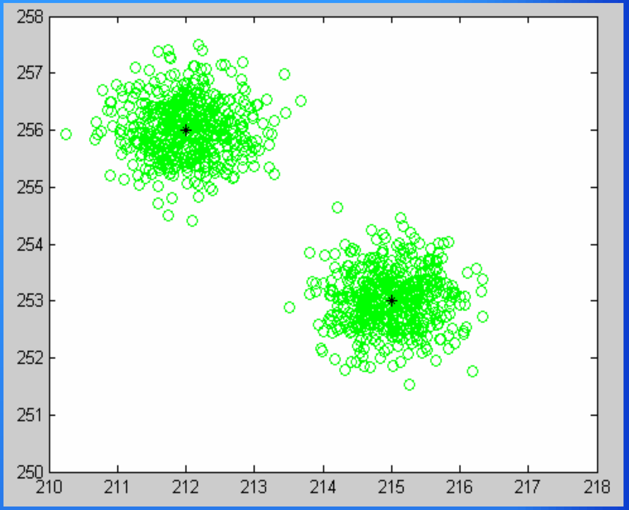
$$w_p^k \propto w_p^{k-1} \frac{p\left(\mathbf{z} \mid \mathbf{x}_p^k\right) p\left(\mathbf{x}_p^k \mid \mathbf{x}_p^{k-1}\right)}{q\left(\mathbf{x}_p^k \mid \mathbf{x}_p^{k-1}, \mathbf{z}\right)}$$

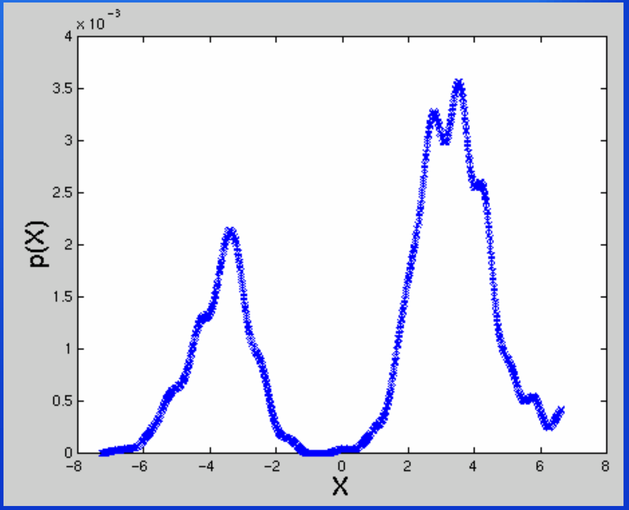
- Resample the particles



The Wrong Way to do Multitarget Tracking

- A naïve approach to tracking multiple targets is to maintain the same single target state space (e.g. $[x \; \dot{x} \; y \; \dot{y}]'$) and expect the density to be multimodal
  - This does not model the density correctly – as a joint multitarget density. Coupling between the targets is ignored.
  - Furthermore, in this paradigm a multimodal density may represent multiple targets or merely an uncertainty in a single target that has two peaks.
- A multitarget filter constructed in this manner is guaranteed to lose targets
  - These consequences appear irrespective of the manner of implementation (i.e. particle filter, Gaussian sum, grid-based method).

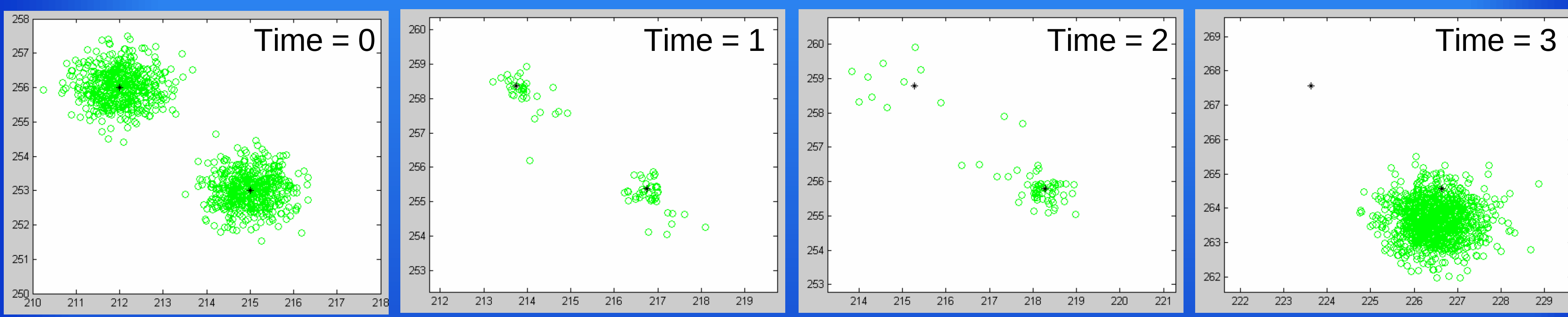




One target or two?

The Wrong Way to do Multitarget Tracking

- Specifically, in a particle filter implementation of the naïve approach,
  - Measurements are made and the particles are weighted accordingly
  - Resampling is performed, eliminating particles with low weight in favor of those with high weight
  - Targets corresponding to low-likelihood measurements (e.g. missed detections) are resampled away.



- An artificial way to compensate for this is to introduce target birth and death
  - Targets are continually reinitiated (born) to overcome the fact that the filter incorrectly kills off targets due to its flawed formulation.
  - This birth/death has no relation to the actual physics behind what is going on.

The Correct Formulation: The Joint Multitarget Probability Density (JMPD)

- As before, the state of an individual target is modeled by $\mathbf{x}$.
- To adequately model the joint multitarget density, the state vector of the system (where perhaps the number of targets *T* is unknown) is defined as

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_{T-1} \quad \mathbf{x}_T]'$$

- The central element that summarizes our knowledge of the system at time *k* is the *joint multitarget probability density (JMPD),*

$$p(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots \mathbf{x}_{T-1}^k, \mathbf{x}_T^k \mid \mathbf{Z}^k) = p(\mathbf{X}^k \mid \mathbf{Z}^k), \qquad T = 1 \ldots \infty$$

which is to be estimated based on a sequence of noisy measurements taken over *k* time steps,

$$\mathbf{Z}^k = \{\mathbf{z}^1 \cup \mathbf{z}^2 \ldots \cup \mathbf{z}^k\}$$

The Joint Multitarget Probability Density (JMPD)

- As examples, the sample space of $p(\mathbf{X}^k|\mathbf{Z}^k)$ contains

  $p(\varnothing \mid \mathbf{Z}^k)$, The posterior probability density for no targets in the surveillance region

  $p(\mathbf{x}_1,\mathbf{x}_2 \mid \mathbf{Z}^k)$, The posterior probability density for two targets in states $\mathbf{x}_1$ and $\mathbf{x}_2$
  Notice the permutation symmetry inherent in JMPD

- State evolution is modeled as Markov using a Kinematic prior $p(\mathbf{X}^k \mid \mathbf{X}^{k-1})$
  - This includes target motion models which may be class dependent
  - Constraints such as roadways, terrain maps, and hospitability enter into this prior.
  - If there really is birth and death (such as targets entering/leaving the surveillance region along the boundaries), that probability enters in here.

- The sensor output is modeled using $p(\mathbf{z}^k \mid \mathbf{X}^k)$

The Joint Multitarget Probability Density (JMPD), cont'd

- The JMPD obeys the usual rules of Bayesian filtering : the two- step recursion of prediction and update:

Prediction (generating the Kinematic prior)

$$p\left(\mathbf{X}^k \mid \mathbf{Z}^{k-1}\right) = \int p\left(\mathbf{X}^k \mid \mathbf{X}^{k-1}\right) p\left(\mathbf{X}^{k-1} \mid \mathbf{Z}^{k-1}\right) d\mathbf{X}^{k-1}$$

Update (Bayes' rule to Incorporate Measurements)

$$p\left(\mathbf{X}^k \mid \mathbf{Z}^k\right) = \frac{p\left(\mathbf{z}^k \mid \mathbf{X}^k\right) p\left(\mathbf{X}^k \mid \mathbf{Z}^{k-1}\right)}{p\left(\mathbf{z}^k \mid \mathbf{Z}^{k-1}\right)}$$

$$\text{where } \; p\left(\mathbf{z}^k \mid \mathbf{Z}^{k-1}\right) = \int p\left(\mathbf{z}^k \mid \mathbf{X}^k\right) p\left(\mathbf{X}^k \mid \mathbf{Z}^{k-1}\right) d\mathbf{X}^k$$

Particle Filter Implementation of JMPD

- As in the single target case, there are several methods of solving the prediction and update equations : e.g. Gaussian sum and discretization onto a fixed grid.
- The strategy we employ here is to avoid a fixed grid and Gaussian approximations of any sort with particle filtering.
- Let the Joint Multitarget Probability Density (JMPD)

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{T-1}, \mathbf{x}_T \mid \mathbf{Z}) = p(\mathbf{X} \mid \mathbf{Z}), \qquad T = 1 \ldots \infty$$

be approximated by *N* weighted samples (particles) as

$$p(\mathbf{X} \mid \mathbf{Z}) \approx \sum_{p=1}^{N} w_p d(\mathbf{X} - \mathbf{X}_p)$$

Particle Filter Implementation of JMPD

- Each of the particles $\mathbf{X}_p$ is a sample drawn from the JMPD $p(\mathbf{X}^k|\mathbf{Z}^k)$
  - Therefore, a particle will contain an estimate of both the number of targets in the surveillance region and the states of each.
- A particle $\mathbf{X}_p$ will be written as

$$\mathbf{X}_p = [\mathbf{x}_{p,1} \quad \mathbf{x}_{p,2} \quad \dots \quad \mathbf{x}_{p,T-1} \quad \mathbf{x}_{p,T}]'$$

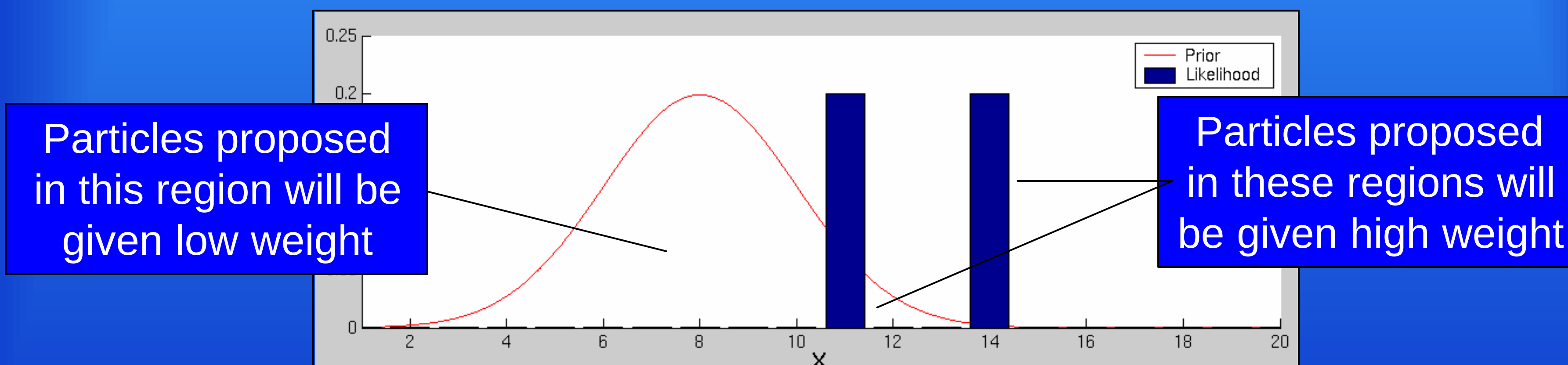
  - Each $\mathbf{x}_{p,i}$ in the particle $\mathbf{X}_p$ is the state vector of a particular target, and will be called a partition of the state vector.
  - A particle may have *0, 1, …* ¥ partitions, each partition corresponding to a different target.
  - The number of partitions in a particle is the particles estimate of the number of targets in the surveillance region.
- We want to generate a set of samples (particles) that approximate the joint multitarget probability density $p(\mathbf{X}^k|\mathbf{Z}^k)$.

Design of Importance Density

- The key to computational tractability in a particle filtering algorithm is designing a good importance (sampling) density

$$q\left(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{z}^k\right) \approx p\left(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{z}^k\right)$$

- It is important to ensure that proposed particles end up in the correct part of state space – consistent with both the kinematic prior and the likelihood



The Multitarget Proposal Density

- Recall that the posterior density is approximated by a set of *N* particles

$$p(\mathbf{X}\,|\,\mathbf{Z}) \approx \sum_{p=1}^{N} w_p d\,(\mathbf{X}-\mathbf{X}_p)$$

and each particle $\mathbf{X}_p$ is partitioned as

$$\mathbf{X}_p = \begin{bmatrix} \mathbf{X}_{p,1} \\ \vdots \\ \mathbf{X}_{p,T} \end{bmatrix} \quad \text{e.g.} \quad \mathbf{X}_p = \begin{bmatrix} 13.25 \\ -0.05 \\ 3.13 \\ 0.03 \\ 1.44 \\ 0.07 \\ 3.05 \\ 0.00 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} \mathbf{x}_{p,1} \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} \mathbf{x}_{p,2} \end{matrix}$$

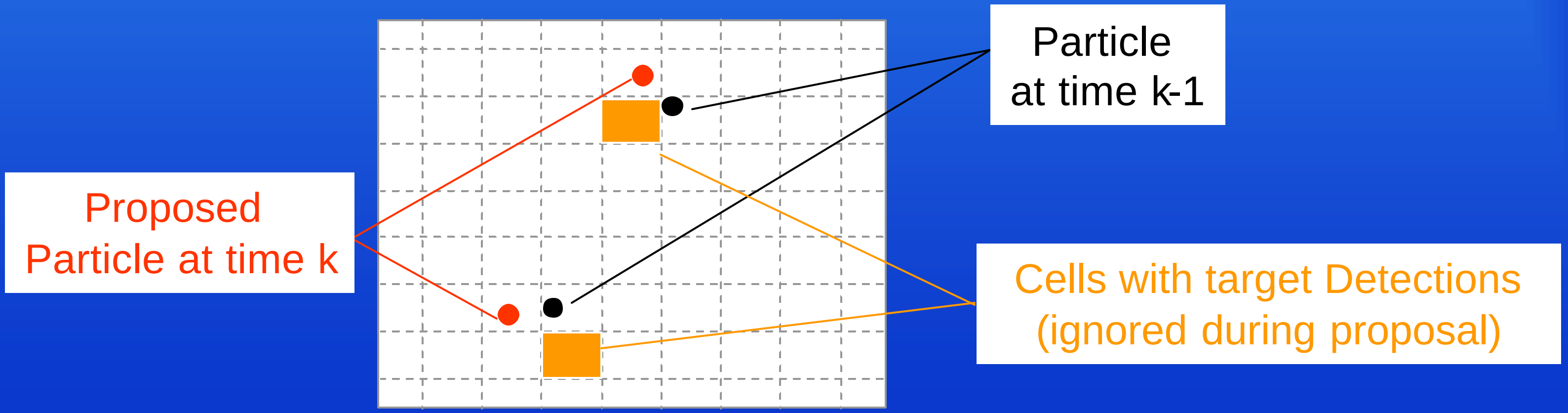
where each partition corresponds to a target $\mathbf{x}_{p,i}$ =[x_i $?_i$ y_i $?_i$]’

- We focus on designing a multitarget importance density that uses the fact that each particle contains multiple partitions corresponding to multiple targets.

The Standard Importance Density: The Kinematic Prior

Proposing via the Kinematic Prior

- In this traditional method of proposing particles, each particle at time *k-1* generates a new particle at time *k* via the repeated application of the single target kinematic (motion) model $p(\mathbf{x}^k|\mathbf{x}^{k-1})$
- Distinguishing Feature : Measurements are not used when proposing particles



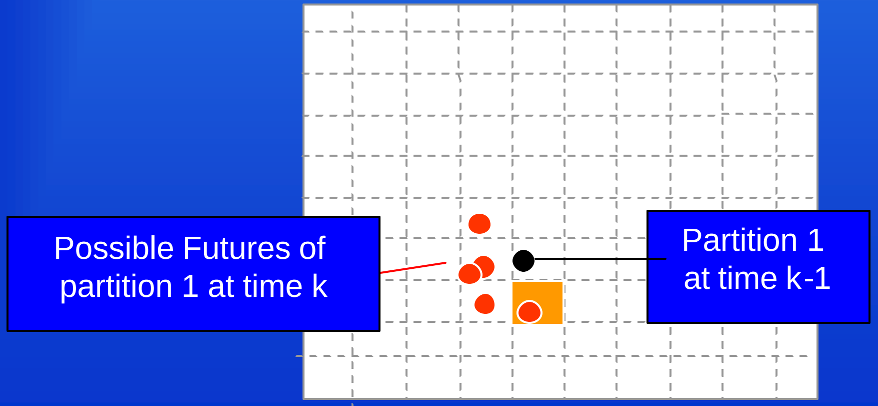
Importance Densities that Explicitly Model Multitarget Nature of the Problem

Coupled Partition (CP)

- Particles at time *k* built partition-by-partition. For each of the *N* samples in a partition, we propose *R* possible futures via the Kinematic prior, weight each using the measurements, and select one.

- CP appropriate under the most general of conditions, but computationally costly. It is in fact an approximation to the OID.

Partition 1 of a particular particle

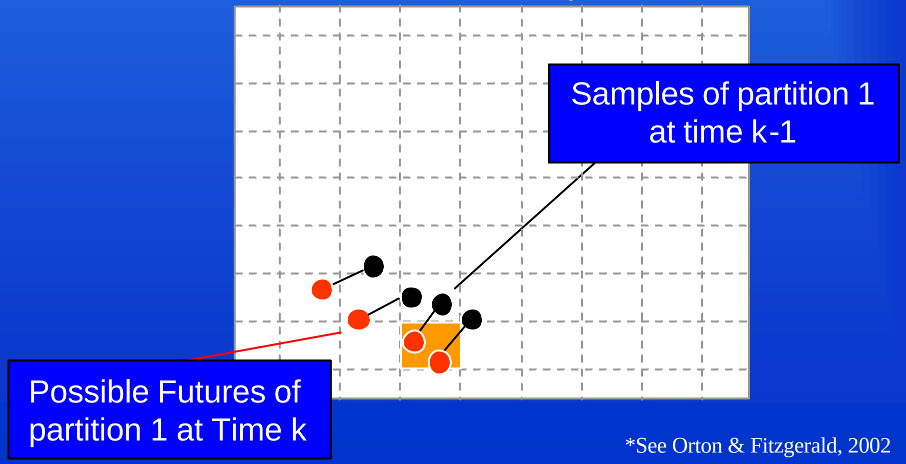


Independent Partition (IP)*

- Particles at time *k* built partition-by-partition. For each of the *N* samples in a partition, propose one new sample via Kinematic prior and weight via measurements. Select with replacement *N* samples from the group.

- IP only appropriate when targets (partitions) are independent; Significantly lightens computational burden when applicable.

Partition 1 of all particles



*See Orton & Fitzgerald, 2002

Under What Circumstances is the IP Method Applicable?

- The JMPD is permutation symmetric : If $\mathbf{x}_1$ and $\mathbf{x}_2$ are states of two targets, then $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2]$ and $\mathbf{X} = [\mathbf{x}_2,\mathbf{x}_1]$ refer to the same event.
- The particle filter manifestation of this permutation symmetry is *partition swapping*.
  - A particle contains an estimate of both the number of targets and their states, e.g. when target state is modeled $[x_i\ ?_i\ y_i\ ?_i]'$, a 2-target particle may be

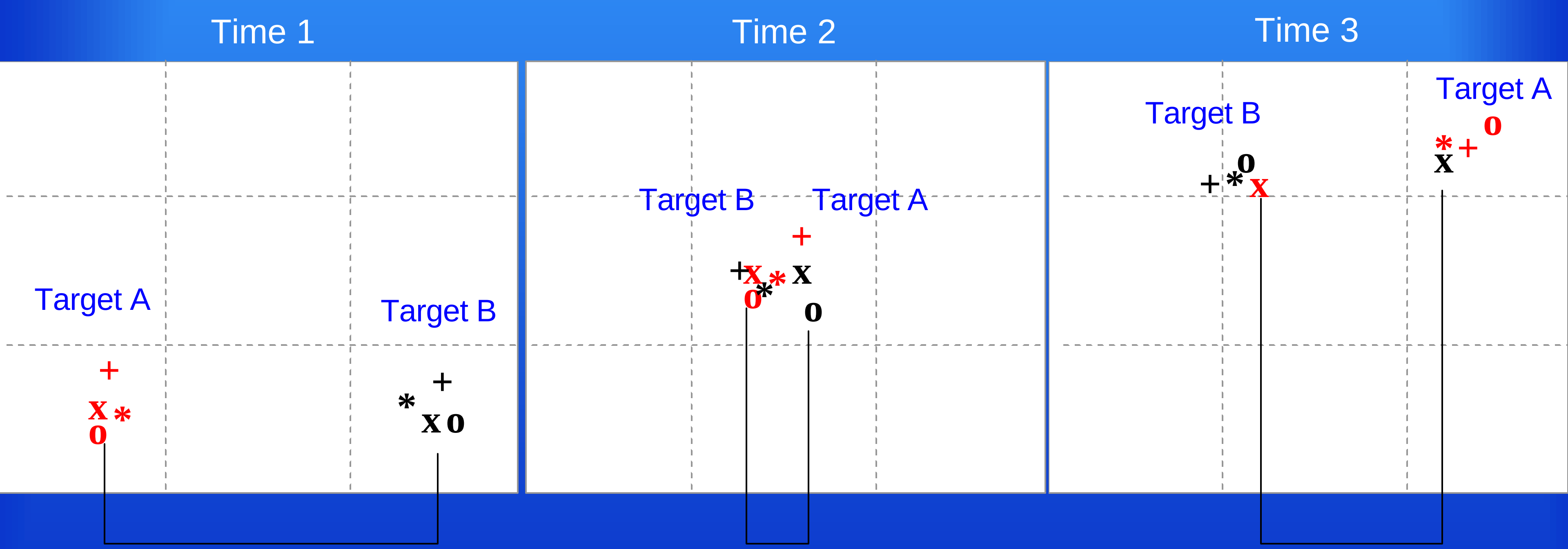
$$\mathbf{X}_p = \begin{bmatrix} 13.25 \\ -0.05 \\ 3.13 \\ 0.03 \\ 1.44 \\ 0.07 \\ 3.05 \\ 0.00 \end{bmatrix}$$

  - This symmetry manifests itself directly in the particles used to approximate the density. The two particles $\mathbf{X}_1$ and $\mathbf{X}_2$ represent the same event:

$$\mathbf{X}_1 = \begin{bmatrix} 13.25 \\ -0.05 \\ 3.13 \\ 0.03 \\ 1.44 \\ 0.07 \\ 3.05 \\ 0.00 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} 1.44 \\ 0.07 \\ 3.05 \\ 0.00 \\ 13.25 \\ -0.05 \\ 3.13 \\ 0.03 \end{bmatrix} \qquad\qquad \bar{\mathbf{X}} = \begin{bmatrix} 7.35 \\ 0.01 \\ 3.09 \\ 0.02 \\ 7.35 \\ 0.01 \\ 3.09 \\ 0.02 \end{bmatrix}$$

Partition Swapping

- Consider 4 particles (denoted by "o","x","+" and "*") that are each tracking two targets (Target A and Target B)
- Each particle has two partitions – color coded black and red
- When proposing according to the Kinematic prior, partition swapping may occur when targets cross – this is completely acceptable.



Each particle has an estimate of both target A and target B.

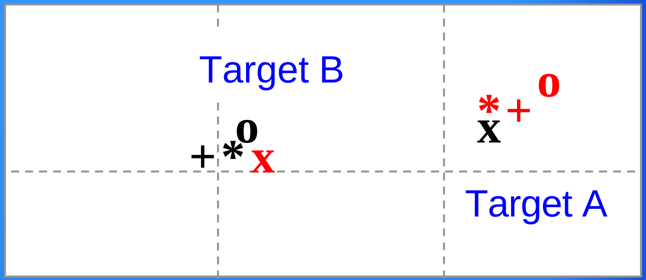
When targets "cross" partition swapping is possible.

The ordering of target partitions in particle "x" is opposite of the others.

Partition Swapping

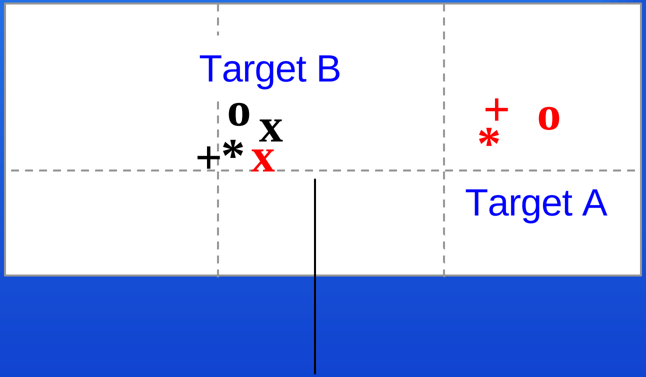
- Using the IP Method in scenarios where swapping has occurred is unacceptable
  - IP assumes that a particular partition is associated with one target
  - e.g. IP assumes all of the red partitions are tracking the same target.

Time 3



- Using IP at Time 3 may lead to some particles that have both partitions associated with the same target
  - To build a new particle, IP proposes a new partition 1 by sampling from the set *, o, +, x and a new partition 2 by sampling from the set *, o, +, x
  - This may lead to a particle which is constructed using x and o

Time 4



This particle (x) now has both partitions tracking target B – i.e. it (incorrectly & artificially) contributes probability mass to the state “two targets at location B”

Partition Swapping, cont'd

- The CP Method does not mix particles – lineage is maintained.
  - New particles will be proposed with the same ordering as particles from the previous time step.
  - Permutation symmetry is respected and probability mass is not artificially transferred to incorrect states.
- CP applicable in all scenarios.
  - Significantly less efficient then IP method
  - When IP appropriate, it should be used.
- IP applicable when targets are 'well separated' (acting independently) and the partitions are ordered identically.

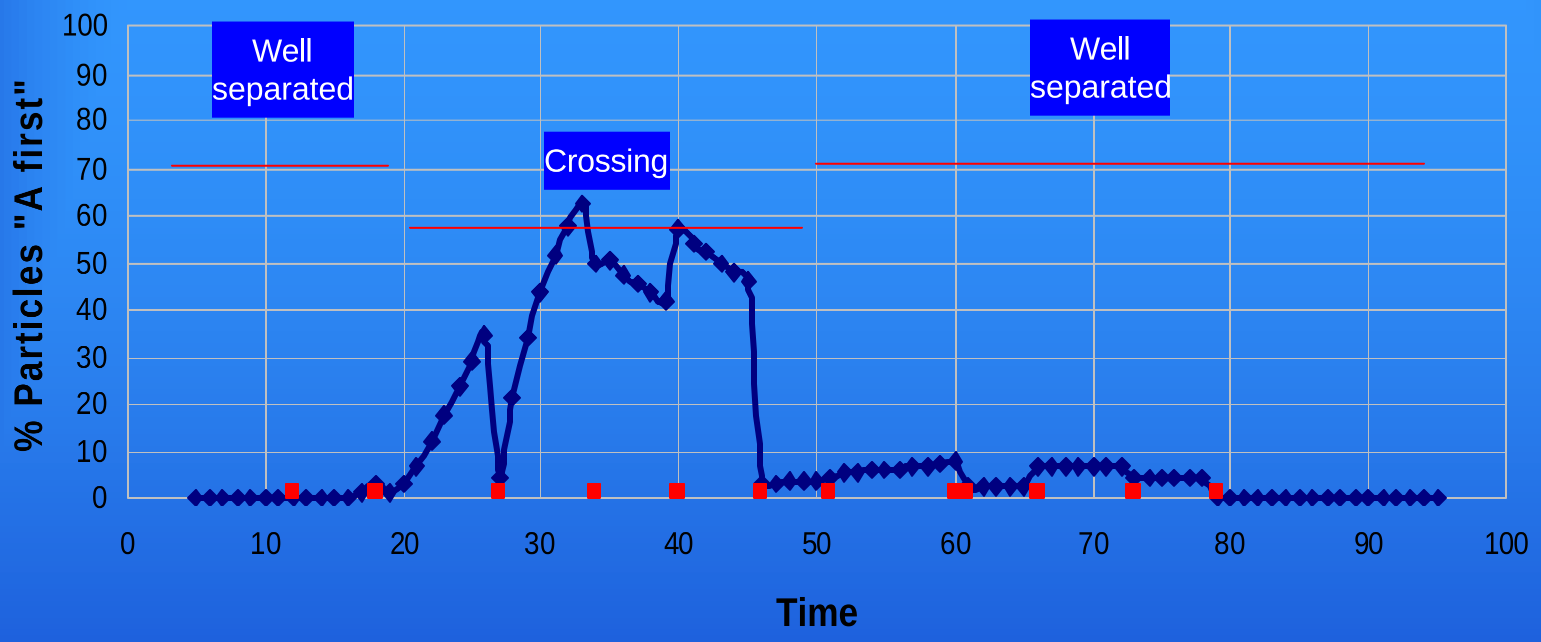
Reordering Partitions

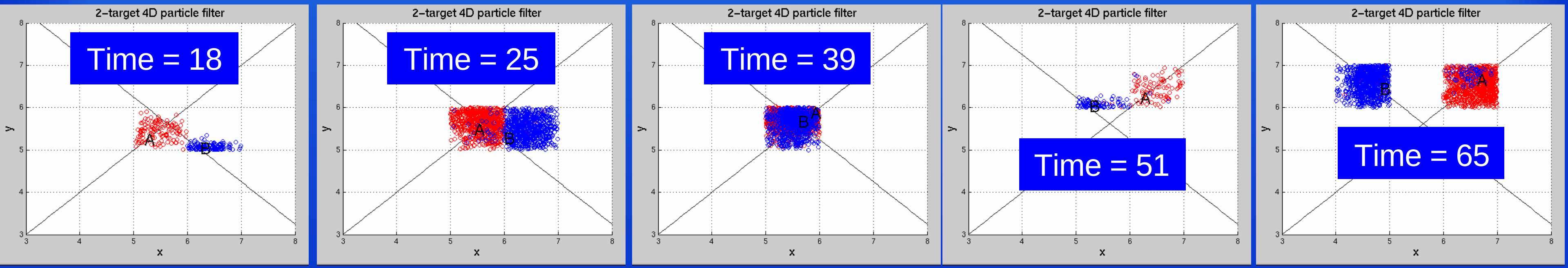
- Assume now that the actual targets are well separated, but different particles have different orderings of targets

$$\mathbf{x}_1 = \begin{matrix} A \\ B \end{matrix} \quad \mathbf{x}_2 = \begin{matrix} B \\ A \end{matrix} \quad \mathbf{x}_3 = \begin{matrix} A \\ B \end{matrix} \quad \mathbf{x}_4 = \begin{matrix} B \\ A \end{matrix} \quad \mathbf{x}_5 = \begin{matrix} A \\ B \end{matrix} \quad \cdots$$

- We call the [A B] particles “A-first” particles and the [B A] “B-first” particles.
- We find that the particles automatically order themselves identically and allow for the IP method to be applied
  - In general, each resampling results in a new set of particles with different distribution of A first and B first particles.
  - The only stable state is for 100% to be A-first of 100% to be B-first.
  - In practice, resampling quickly moves the distribution to a stable state.
  - It can be shown analytically that the automatic ordering will happen with a time constant on the order of *N*, the *effective* number of particles (in practice, the time steps until automatic ordering << number of particles)

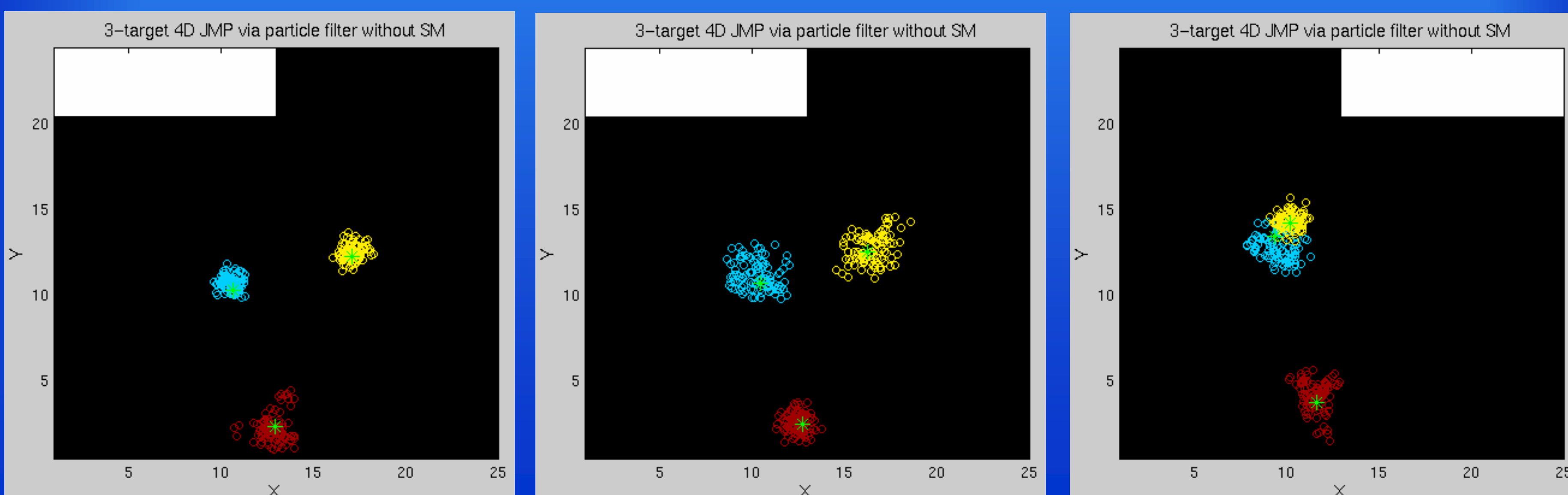
Reordering Partitions





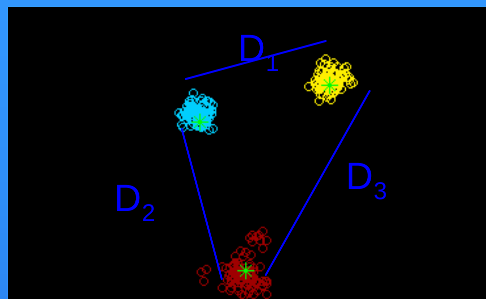
Multitarget Proposal Densities

- When targets are well separated (in the measurement space), each sample is associated with a particular target. IP is appropriate here.
- When targets become “close” samples commingle and measurements of one target may effect samples associated with other targets. IP is not appropriate.
- The importance density should be Independent Partitions (IP) when targets are well separated and Coupled Partitions (CP) when they are not.

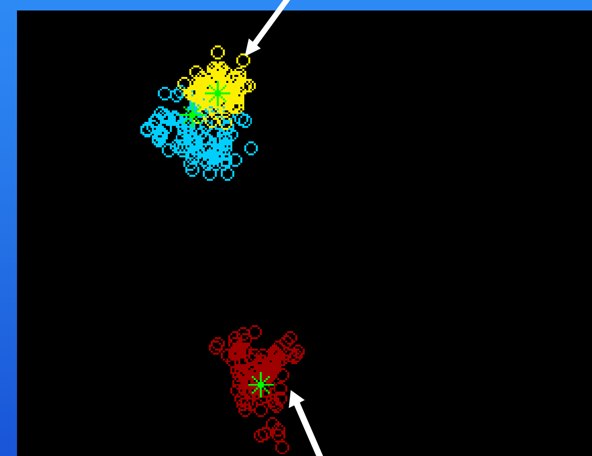


Adaptive Proposal Method Switching

When are partitions ‘well separated’?



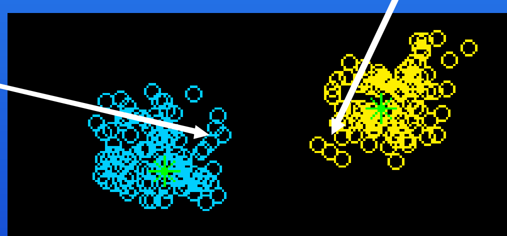
Use CP on these



Use IP on this

Sample from partition *i* closest to mean of partition *j*

Sample from partition *j* farthest from mean of partition *j*



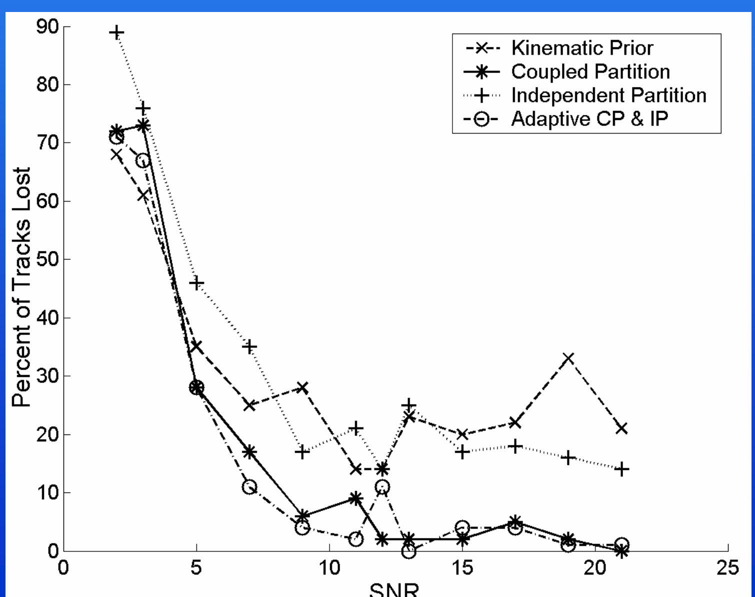
Mahalanobis Distance

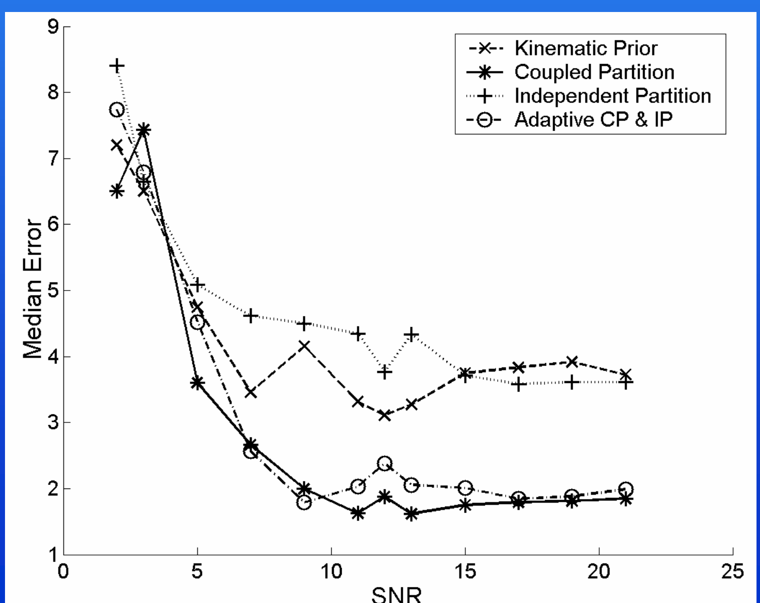
$$r_{i,j}^2 = (\mathbf{x}_i - \mathbf{m}_j)' \mathbf{S}_j^{-1} (\mathbf{x}_i - \mathbf{m}_j)$$

Multitarget Proposal Densities

- Simulation: Three targets moving on a grid.
- Targets spend approximately 50% of the time 'near' each other (when only CP is appropriate) and 50% of the time well separated (where IP is appropriate)
- Adaptive method achieves similar performance as CP at half the FLOPS.

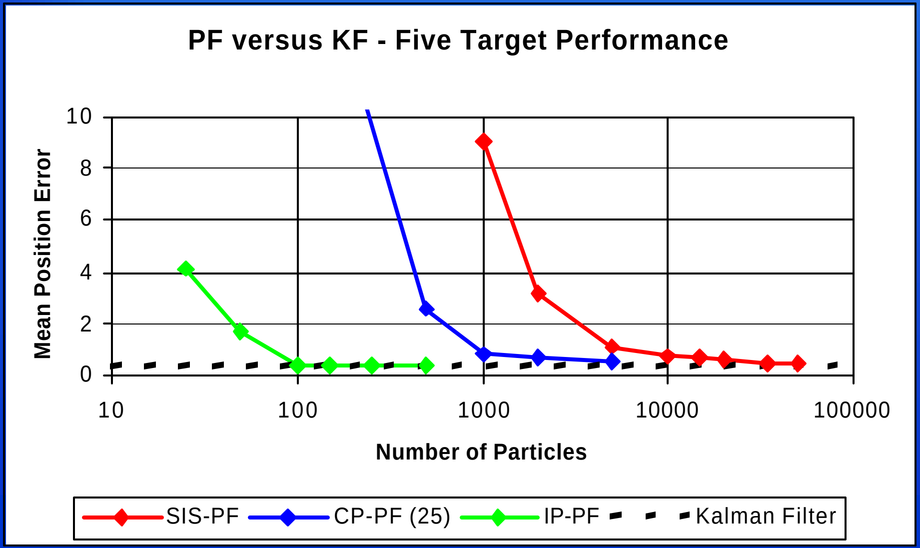
Method	Flops
Kinematic Prior	6.32E+06
Independent Partition	6.74E+06
Adapative CP/IP	5.48E+07
Coupled Partition	1.25E+08

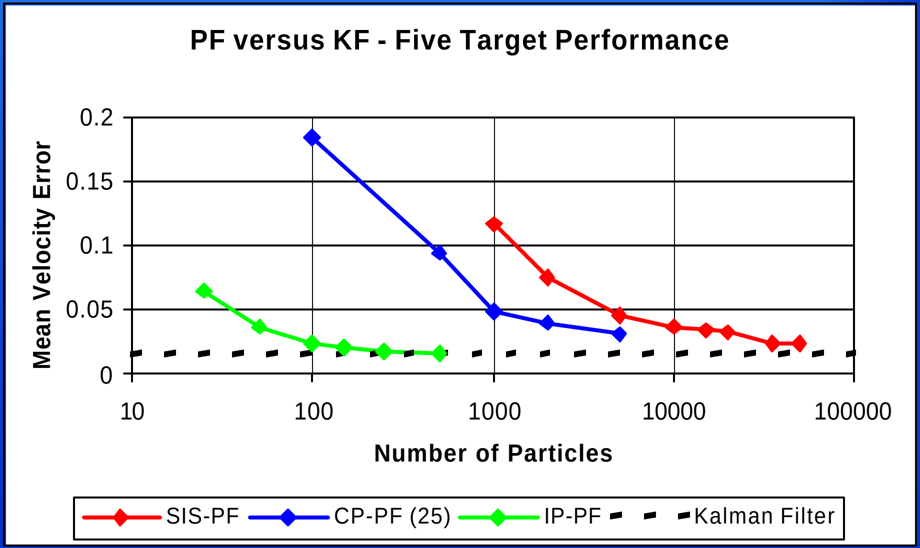




How much Effort does the adaptive strategy save?

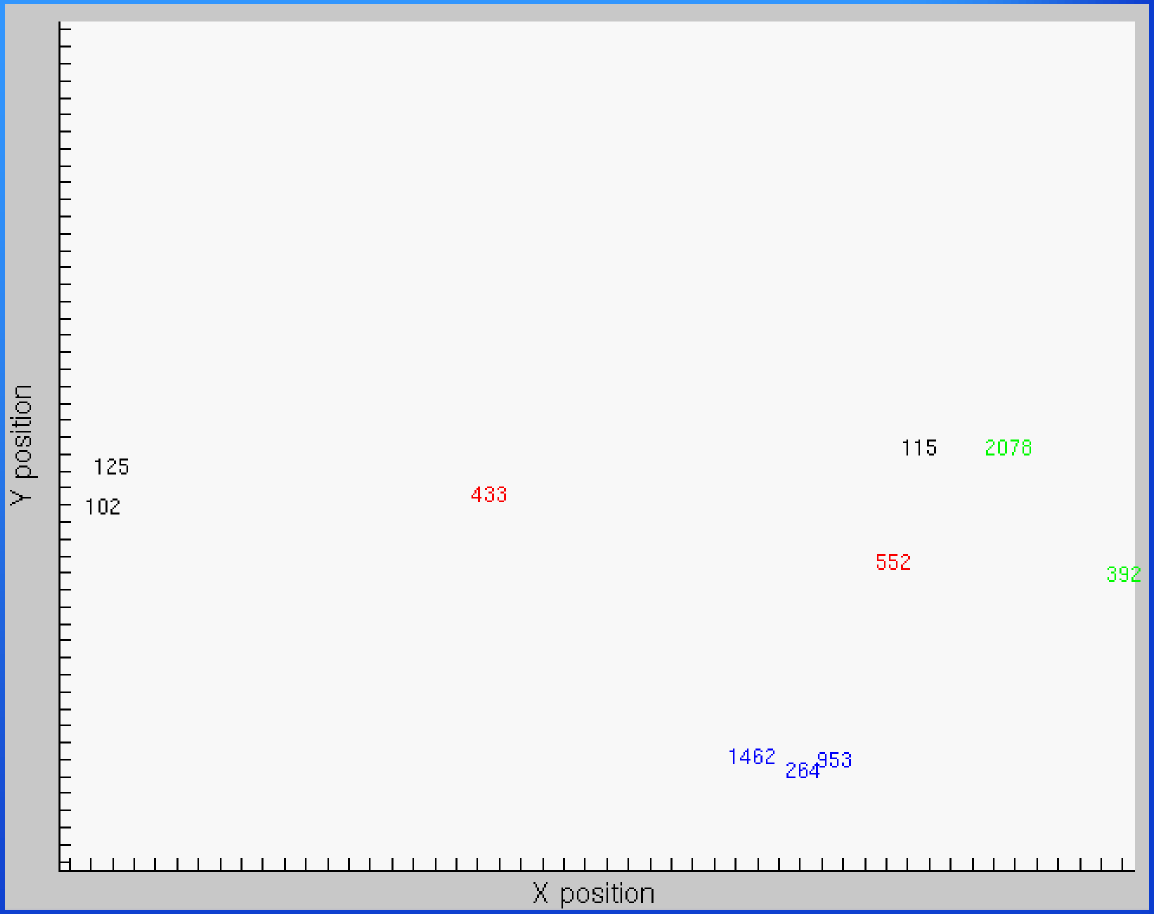
- We compare a PF using the Kinematic Prior with one using the adaptive strategy.
- Particle Filtering allows for
  - Non-linear Measurement to State Coupling
  - Non-linear State Evolution (Target Motion)
  - Non-Gaussian Densities
- **We ignore all these benefits for a moment**
- How well does the multi-target PF perform in comparison to a Kalman Filter in the regime where a Kalman Filter is applicable (and optimal)?
  - Simulation: Linear motion, linear measurements, Gaussian pdf.
  - Five (well separated) targets with state vectors [x ? y ?]





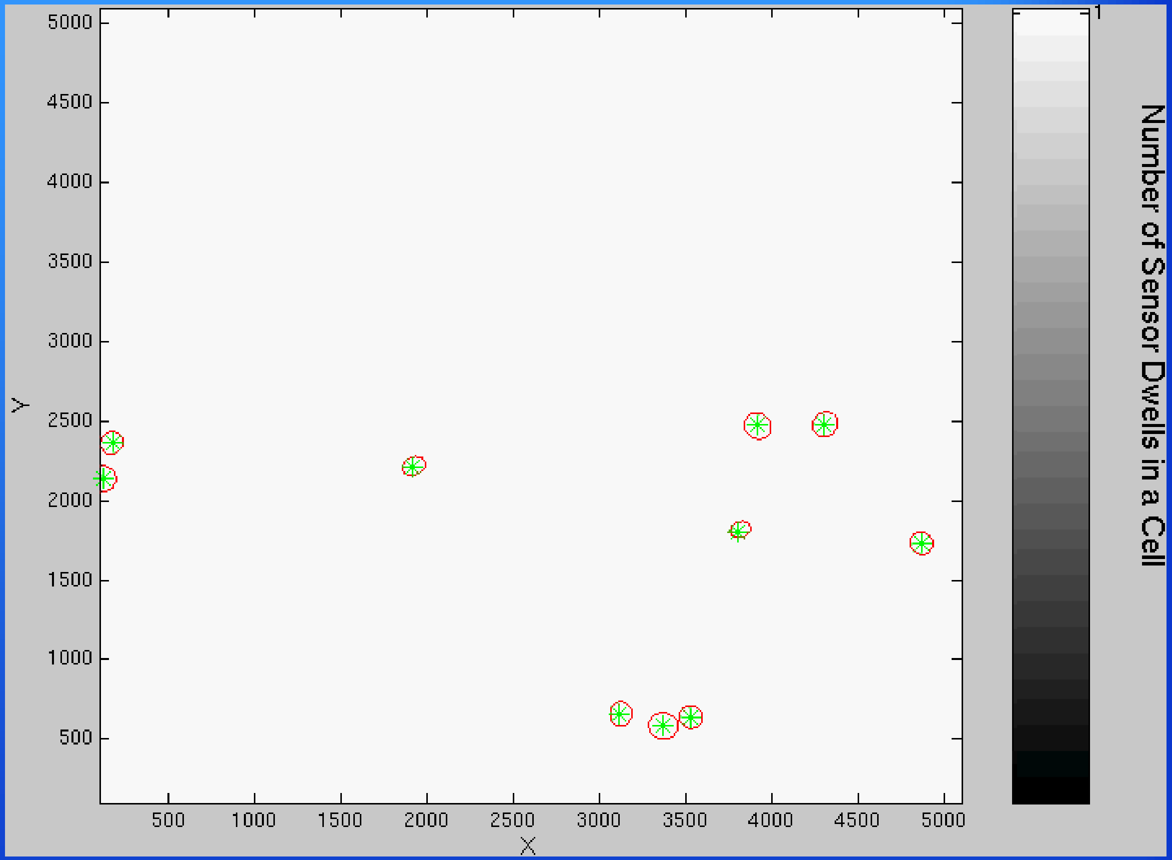
Is it Tractable? 10 ***Real*** Targets

- Vehicle Trajectories
  - 10 Real targets culled from the NTC Sensor Strike Track Files
    - #433, #552 Cross
    - #392, #2078 travel together sometimes
    - #264, #953, #1462 travel together a lot
    - #102, #115, #125 added to bring the total to 10
  - 1000 time steps, 1 second apart
  - Vehicles are time & space shifted to be in the same region at the same time



Is it Tractable? 10 ***Real*** Targets

- Sensor Simulation
  - A quasi-GMTI simulation where sensor measures 10x1 grid cell and gets 10 returns
  - The sensor grid is 50 cells x 50 cells. Each cell is 100m x 100m.
  - SNR = 12
- JMPD - Particle Filter
  - Nparts = 500
  - Fully adaptive switching between CP and IP based on sample distance



Runtime ~ 1 Hour on Off the shelf Linux Box
3 times "real time"

Conclusion

- We've presented a method of tracking multiple targets based on recursive estimation of their Joint Multitarget Probability Density (JMPD).
- Computational tractability is provided by Particle Filter-based implementation.
  - Adaptive sampling schemes exploit multitarget nature of the problem.
  - Permutation symmetry manifests itself as partition swapping
- Natural framework to do sensor management where the JMPD is used to compute the area of maximal expected information gain.

- Backup slides: Information based sensor management using the particle filter implementation of JMPD.

Information Based Sensor Management

- The problem of Sensor Management is to determine the best way to task a sensor where the sensor may have many modes and be directed in many manners.

- We take an *information-based* sensor management route and rephrase the problem in terms of tasking the sensor to make the measurement that maximizes the *expected* amount of information gained.

- We require a measure of information gained by making a sensing action, where a sensing action may be a sensing modality (e.g. SAR or GMTI), a sensing direction (e.g. pointing angle) or a combination of the two.

$$D_a\left(p(\mathbf{X}^k \mid \mathbf{Z}^{k-1}), p(\mathbf{X}^k \mid \mathbf{Z}^k)\right)$$

And seek to make the sensing action, m, that maximizes the expected information gain.

$$\left\langle D_a\left(p(\mathbf{X}^k \mid \mathbf{Z}^k), p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\right)\right\rangle_m = \int d\mathbf{z}_m\, p(\mathbf{z}_m \mid \mathbf{Z}^{k-1}) D_a\left(p(\mathbf{X}^k \mid \mathbf{?}^{k-1}, \mathbf{z}_m), p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\right)$$

Information Based Senor Management: Measures of Gain in Information

- The Kullback-Leibler (KL) Divergence between two densities *p* and *q* is

$$D(p,q) \equiv \int_x p(x) \ln\left(\frac{p(x)}{q(x)}\right) dx$$

- In the JMPD setting:

$$D\left(p(\mathbf{X}^k \mid \mathbf{Z}^k), p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\right) \equiv \sum_X p(\mathbf{X}^k \mid \mathbf{Z}^k) \ln\left(\frac{p(\mathbf{X}^k \mid \mathbf{Z}^k)}{p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})}\right)$$

- More generally, the Rényi (α-) Divergence between *p* and *q* is

$$D_a(p,q) \equiv \frac{1}{a-1} \ln \int_x p^a(x)^{1-a} q(x) dx$$

- In the JMPD setting:

$$D_a\left(p(\mathbf{X}^k \mid \mathbf{Z}^{k-1}), p(\mathbf{X}^k \mid \mathbf{Z}^k)\right) \equiv \frac{1}{a-1} \ln \sum_{\mathbf{X}} p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})^{1-a} p(\mathbf{X}^k \mid \mathbf{Z}^k)^a$$

- When $\alpha \rightarrow 1$ the α-divergence converges to the KL divergence.

Expected Gain
Derivation and Calculation

- We wish to make the measurement m (dwell point / mode, etc.) that has the best *expected* information gain, i.e that *m* which maximizes

$$\left\langle D_a\left(p(\mathbf{X}^k \mid \mathbf{Z}^k), p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\right)\right\rangle_m = \int d\mathbf{z}_m \, p(\mathbf{z}_m \mid \mathbf{Z}^{k-1}) D_a\left(p(\mathbf{X}^k \mid \mathbf{?}^{k-1}, \mathbf{z}_m), p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\right)$$

- Assume for the moment that we've chosen a particular dwell point *m* and made a measurement $\mathbf{z}$

- Using Bayes' rule $p(\mathbf{X}^k \mid \mathbf{Z}^k) = \dfrac{p(\mathbf{z} \mid \mathbf{X}^k)\, p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})}{p(\mathbf{z} \mid \mathbf{Z}^{k-1})}$

- The Rényi Divergence simplifies to

$$D_a = \frac{1}{a-1} \ln \frac{1}{p(\mathbf{z} \mid \mathbf{Z}^{k-1})^a} \sum_{\mathbf{X}} p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\, p(\mathbf{z} \mid \mathbf{X}^k)^a$$

Expected Gain Derivation and Calculation

- The Particle Filter approximation represents the posterior as a sum of weighted delta functions

$$p(\mathbf{X}^k \mid \mathbf{Z}^{k-1}) \approx \sum_{p=1}^{N_{part}} w_p d(\mathbf{X} - \mathbf{X}_p)$$

- Which yields

$$D_a = \frac{1}{a-1} \ln \frac{1}{p(\mathbf{z})^a} \sum_{i=1}^{N_{part}} w_p p(\mathbf{z} \mid \mathbf{X}_p)^a$$

- Where

$$p(\mathbf{z}) = \sum_{i=1}^{N_{part}} w_p p(\mathbf{z} \mid \mathbf{X}_p)$$

Expected Gain Derivation and Calculation

- If the measurement $\mathbf{z}$ has not yet been made, we can calculate instead the *expected* Rényi Divergence for a sensing action *m* as

$$\langle D_a \rangle = \int_{\mathbf{z}_m \in \mathbf{Z}} \left(\frac{1}{a-1} \ln \frac{1}{p(\mathbf{z}_m)^a} \sum_{i=1}^{N_{part}} w_p \, p(\mathbf{z}_m \mid \mathbf{X}_p)^a \right) p(\mathbf{z}_m) d\mathbf{z}_m$$

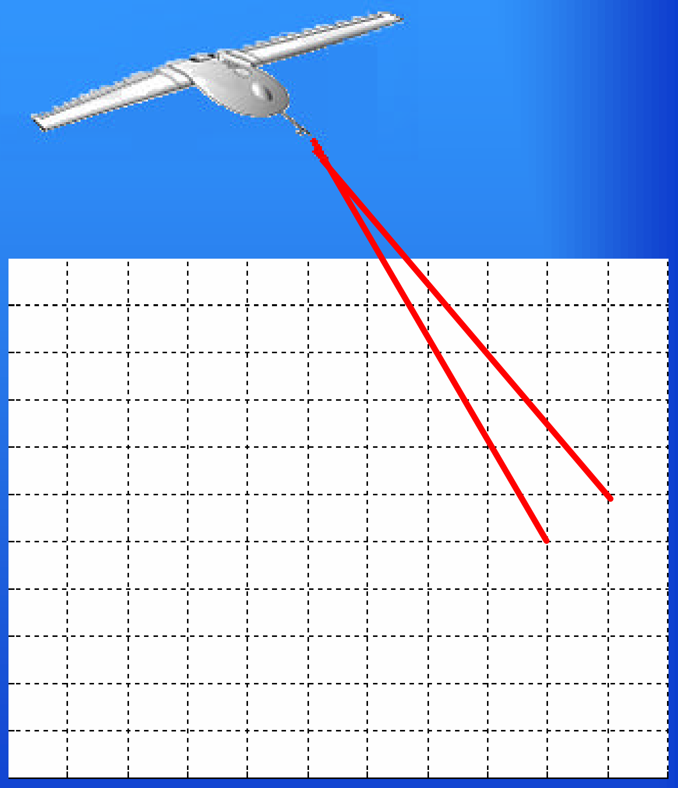
- Which in the thresholded case ($\mathbf{z}$ ♑ {0, 1}) becomes simply

$$\langle D_a \rangle = \sum_{z=0}^{1} \frac{p(z)}{a-1} \ln \frac{1}{p(z)^a} \sum_{i=1}^{N_{part}} w_p \, p(z \mid \mathbf{X}_p)^a$$

- The quantity $\langle D_a \rangle$ can be calculated for many different sensing actions simultaneously, with computational burden dominated by the number of particles
  - e.g. Calculate $\langle D_a \rangle$ for measuring cell *n*, *n=1…N*, with modality *m*, *m=1…M*.
  - Each different sensing action will of course have a different $p(z)$, and $p(z|\mathbf{X}_p)$

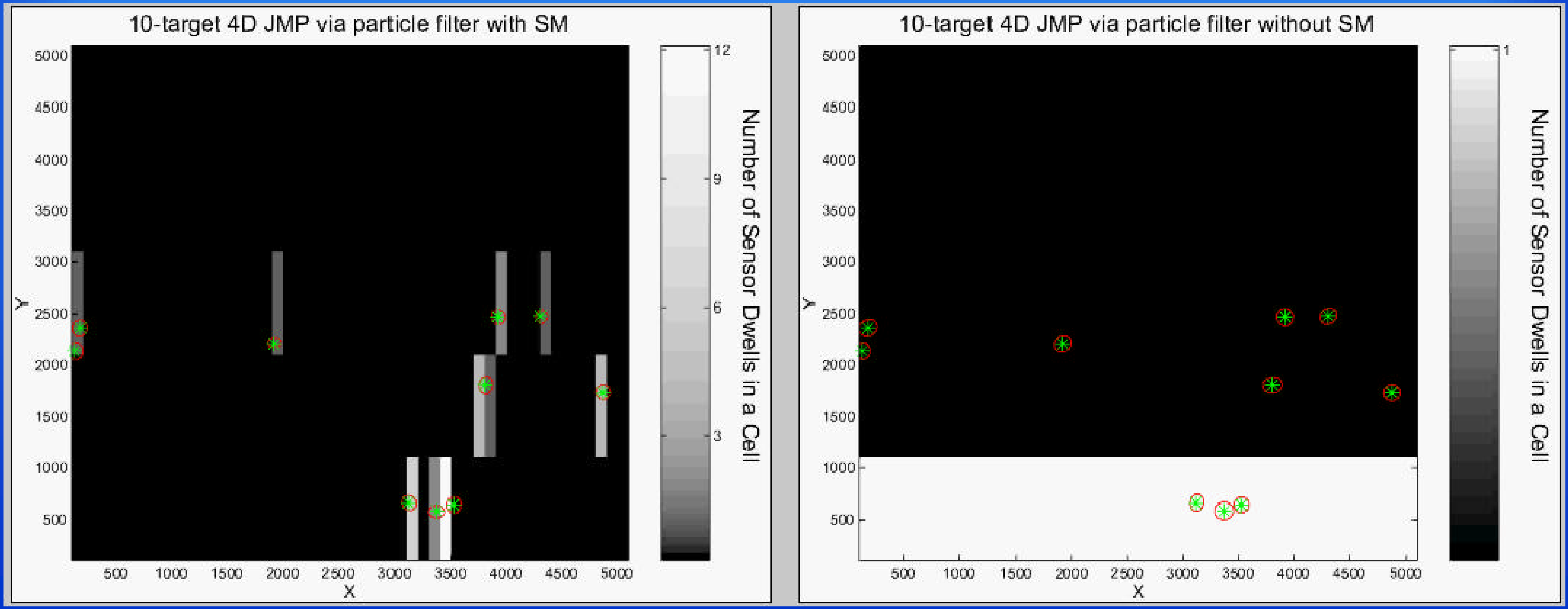
Sensor Management Algorithm – A Summary

- In summary, our sensor management algorithm proceeds as follows.
  - At each occasion where a sensing action is to be made, we evaluate the expected information gain for each possible sensing action *m* using the Rényi Information Divergence measure.
  - We then select and make the sensing action that gives maximal expected information gain.
- This is a greedy (myopic) approach that maximizes information gain on only the current scan



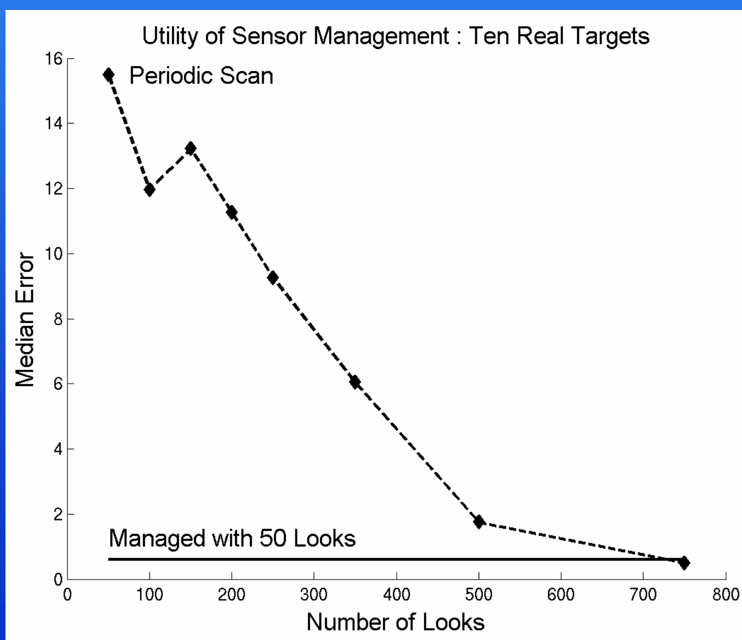
Tracker Comparison
Managed vs. Non-Managed

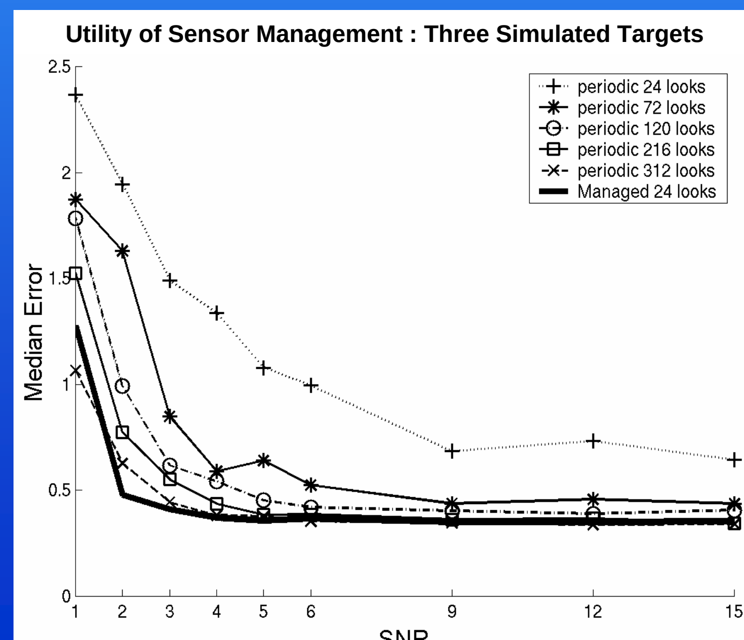
- We illustrate the benefit of this sensor management scheme given in this talk using a particle filter implementation of JMPD tracking 10 moving targets.
- Below, we contrast the performance when the sensor is directed using the sensor management algorithm with the performance when a periodic (non-managed) scan of the surveillance region is employed.



Tracker Comparison Managed vs. Non-Managed

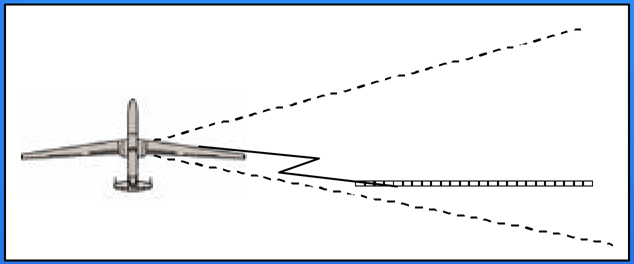
- Monte Carlo tests (left) show performance with SM using 50 looks similar to periodic scan with 700 looks
  - SM makes the tracker 12 times as efficient in terms of sensor resources needed.
- More extensive runs in similar scenario (right) with 3 targets show performance with SM using 24 looks similar to periodic (non-managed) performance with 312 looks
  - SM makes the tracker approximately 13 times as efficient in this scenario.
  - Performance of managed scenario with 24 looks at SNR = 2 (3dB) similar to performance of periodic management at SNR = 9 (9.5dB) – approximately a 6.5dB performance gain.



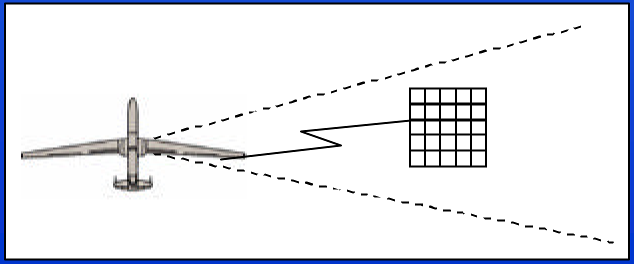


The Information Based Method Automatically Optimizes Across Modes

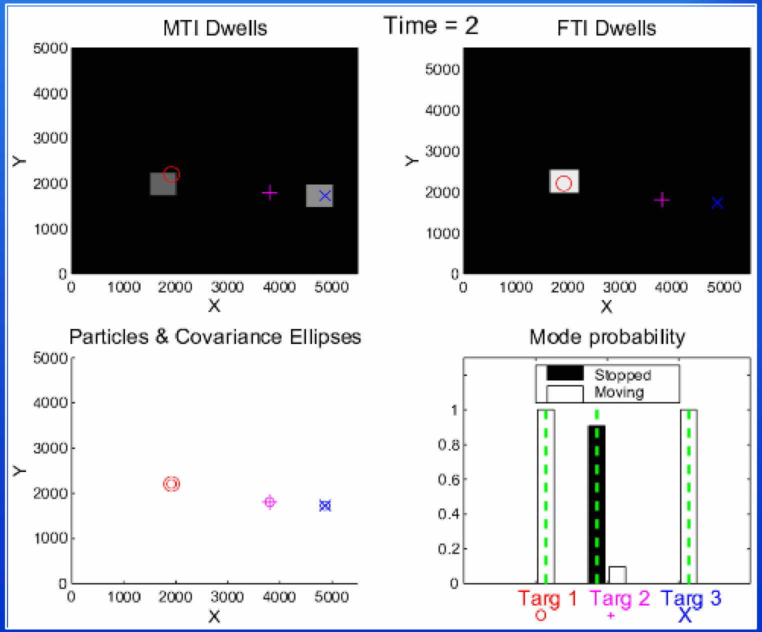
- MTI Mode
  - Measures cells that are 500m x 20m
  - Measures strips 1x25 cells long
  - Pd = 0.9, Pfa = .001
  - Detects targets with velocity > MDV



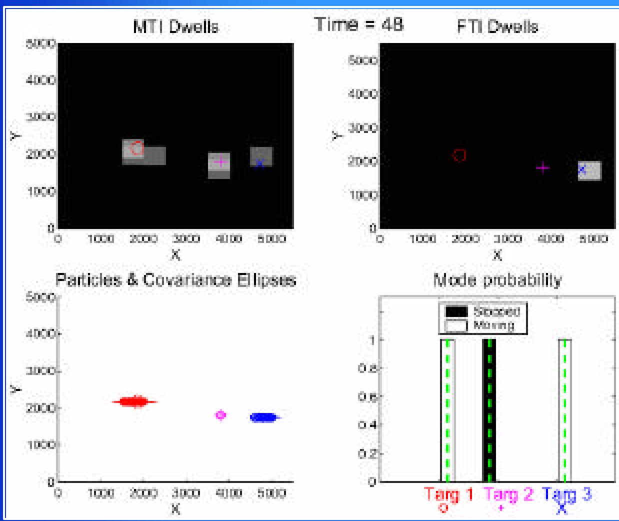
- FTI Mode
  - Measures cells that are 100m x 100m
  - Measures spots 5x5 cells on the ground
  - Pd = 0.5, Pfa = 1e-12
  - Detects stopped targets



- Particle Filter
  - Multiple model (stopped and moving)
  - Adaptive Proposal Method
  - 500 Particles, 3 Targets
- Sensor Management
  - Expected gain for each modality and pointing angle before each measurement.
  - 12 Looks/time step each of 250km^2 (total approximately 10% of surveillance area)



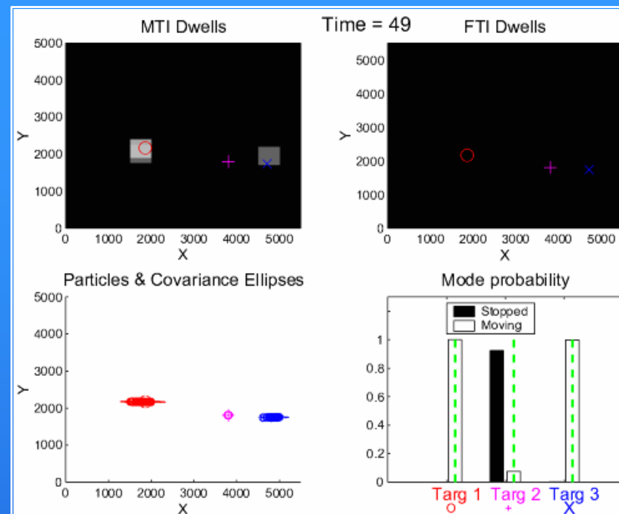
Optimizing Collection Strategy Across Modes



Up to Time 48,
Target 2 is stopped.

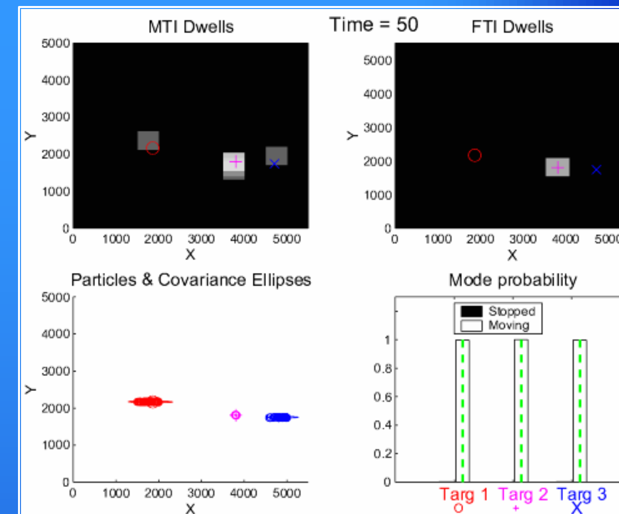
The filter accurately has 100% of particles in "stopped" mode for Target 2.

MTI Dwells are used periodically to detect if it has transitioned to "moving" mode.



At Time 49, Target 2 Starts Moving.

All sensor dwells are given to Targets 1 and 3 at this time, so the state change is not detected.



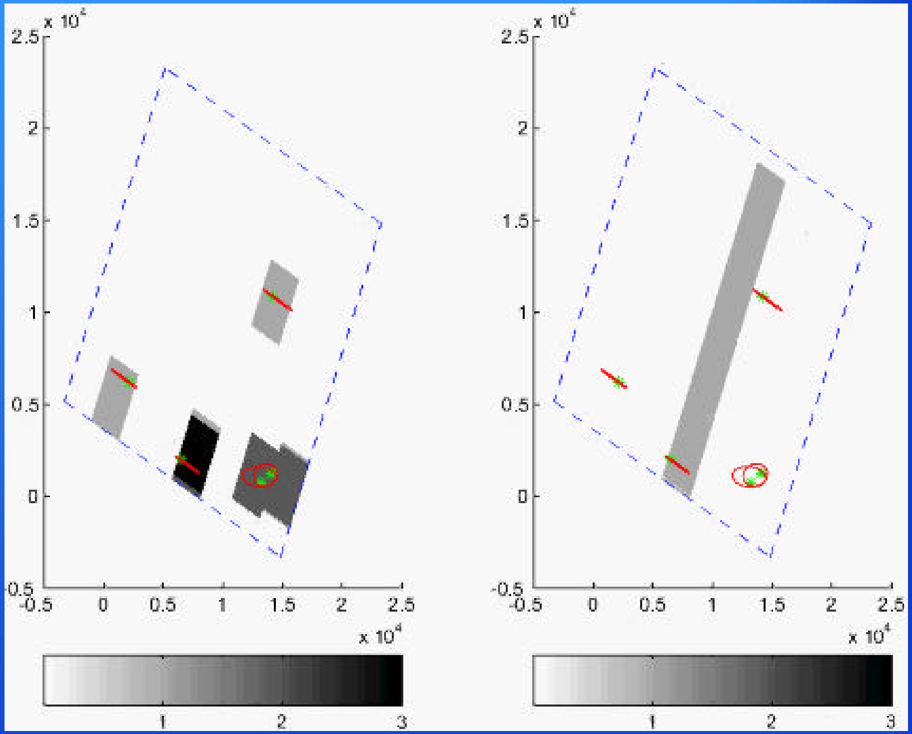
At Time 50, an MTI dwell is made on Target 2 to see if has started moving.

The target is detected, and so another MTI dwell is made, and then an FTI dwell to ensure that the target has actually started moving.

The filter is then certain that the target moving.

The Information Based Method Automatically Optimizes Across Modes

- A surevillance region of 20km x 20km which contains 5 targets is considered.
- The sensor has two modes : Coarse mode and Fine mode.
  - "Fine mode" uses 200 x 8 grid (resolution = 100m x 2500m)
  - "Coarse mode" uses 40 x 8 grid (resolution = 500m x 2500m)
- Each scan contains 39 cells in the range direction.
- The sensor platform rotates counter clockwise around surveillance region 1 degree per second, always focusing on patch center.



Conclusions and Future Work

- We've presented a method of tracking multiple targets and sensor management based on recursive estimation of the Joint Multitarget Probability Density (JMPD).
  - Computational tractability is provided by Particle Filter-based implementation with adaptive sampling schemes that exploit multitarget nature of the problem.

- We have demonstrated a method of sensor management (SM) which uses the JMPD and tasks the sensor to make measurements that yield the maximum expected information gain.
  - In the application of tracking multiple moving ground targets, SM of this type is able to use the sensor more than 10 times as efficiently as a simple periodic scan.
  - The method automatically captures the tradeoff between sensor modalities (e.g. fine resolution mode vs. coarse resolution mode, GMTI mode vs. SAR mode) without any ad hoc adjustments.

- The SM algorithm presented is myopic (greedy).
  - Sensor platform motion (not modeled here) makes it important to plan ahead and choose optimal measurement sequences rather than just individual measurements
  - Future work includes extending these principles to non-myopic sensor management, perhaps using a MDP formulation and Monte Carlo or rollout type solutions.