Detecting Special Nuclear Material With Muon Scattering Tomography

Christian Thomay[1], Jaap Velthuis[1], Paolo Baesso[1], David Cussans[1]
Chris Steer[2], Jonathan Burns[2], Steve Quillin[2], M. Stapleton[2]

[1]University of Bristol, UK
[2]Atomic Weapons Establishment, Reading, UK

Muon Scattering Tomography

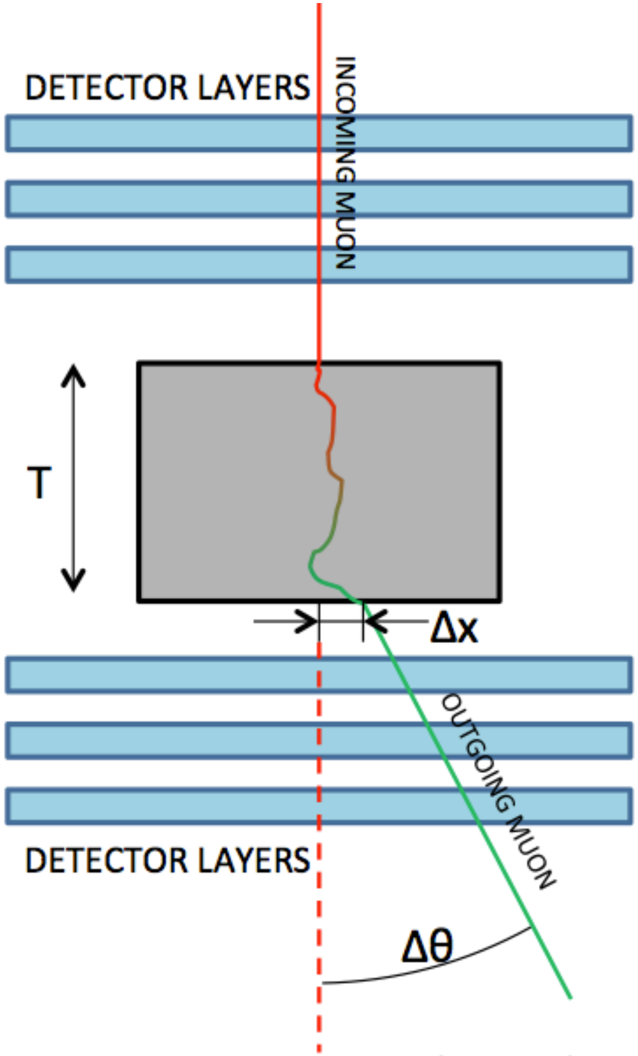
- Homeland security a hot topic: need to scan shipping containers for nuclear material to stop potential threats.
- University of Bristol (UoB) and the UK Atomic Weapons Establishment (AWE) in a partnership to build and study a prototype scanner for Muon Scattering Tomography (MST).
- The prototype is used to scan a target volume in search of high-Z material.
- Muons are excellent probes:
  - readily available, with flux of ~100 Hz/m^2 and energies from 0.1 GeV upwards.
  - Virtually impossible to screen against, since for 1 GeV muons $dE/dx \sim 2\ \mathrm{MeV \cdot g^{-1} \cdot cm^2}$.
  - Charged; can be easily detected.
  - No radiation hazard for the scanner operators.
  - MST is undetectable by the scanned object, since no extra radiation is introduced.

Muon Scattering Tomography

- Muons undergo multiple Coulomb scattering within the detector volume.
- The angular distribution is approximately Gaussian, with σ_0 depending on the radiation length X_0 (and ultimately on Z^2).
- Due to the Z^2, the method is very sensitive for high-Z material.

$$X_0 \approx \frac{A \cdot 716.4 g / cm^2}{\rho \cdot Z \cdot (Z+1) \ln(287/Z)}$$

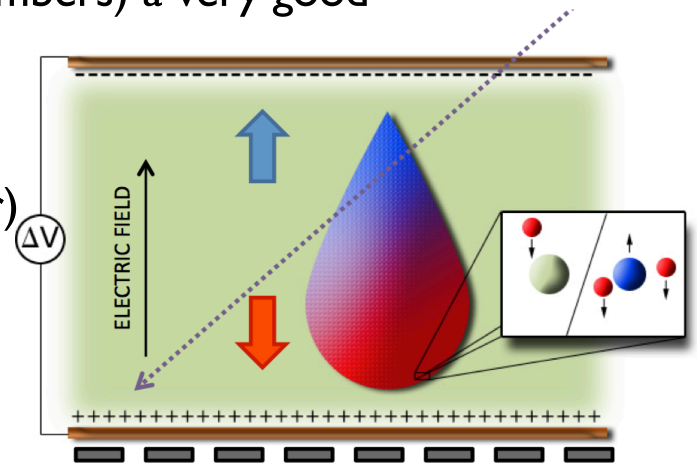
$$\sigma = \frac{13.6 \text{ MeV}}{\beta c p} \sqrt{T/X_0} \left(1 + 0.038 \ln(T/X_0) \right)$$



Muon scattering tomography principle.

RPCs for MST

- Applying MST to homeland security introduces certain requirements for the detector:
  - Large area (shipping containers, trucks)
  - Scalability
  - Low cost per unit area
- RPCs (Resistive Plate Chambers) a very good option for MST:
  - Good resolution (timing / spatial / angular)
  - Efficient
  - Robust
  - Cheap to build



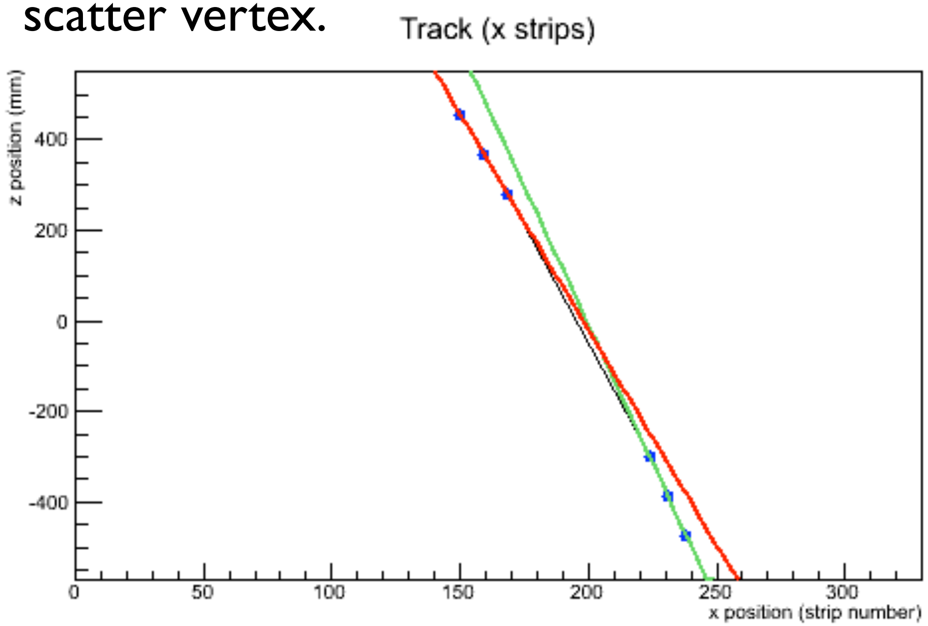
RPC operating principle.


Prototype detector setup, 6 layers of X/Y readout above and below the target.

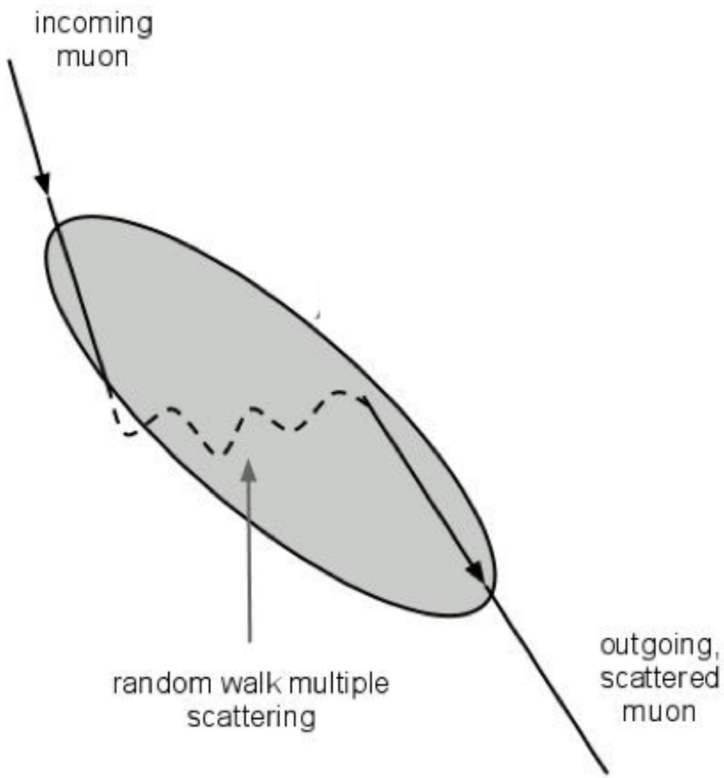
Data Analysis – Tracks

- Tracks and scatter vertices are fitted simultaneously. 12 data points (6 x/y hits in the upper/lower detectors) are fitted using MINUIT as two straight lines with one common point.

- Reconstruct incoming/outgoing tracks and scatter vertex.

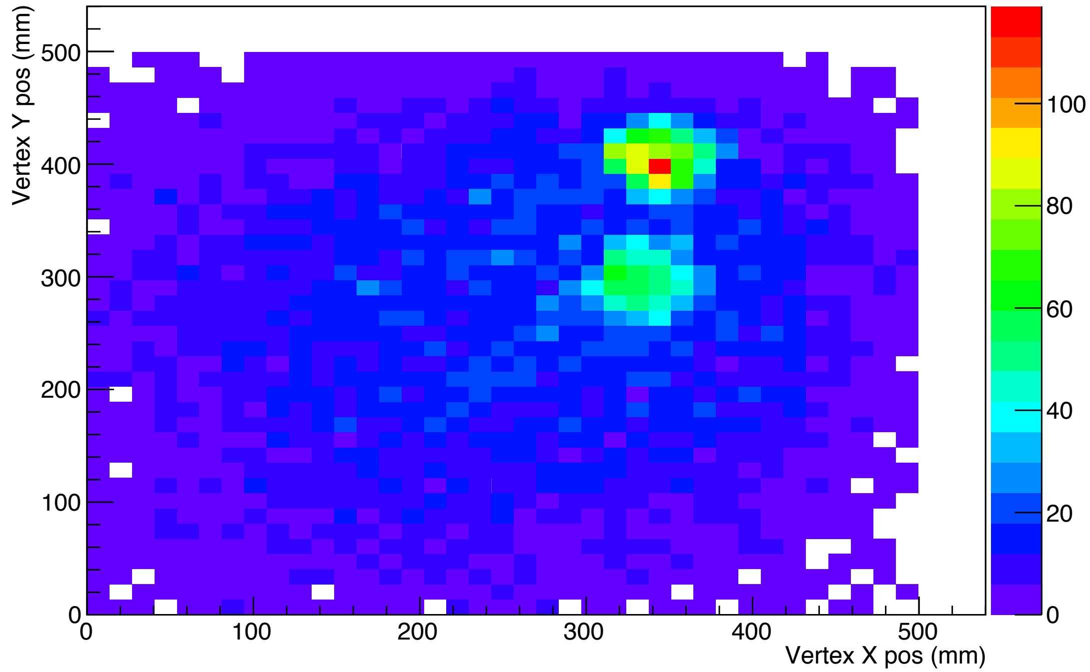


Muon track fit in the XZ plane.



Data Analysis

- Get kinematic variables from fit:
  - fit chi squares
  - scatter angle
  - track offset
  - vertex uncertainty ellipsoid shapes



Scatter vertex distribution in the XY plane above scatter angle cut.

- To illustrate: two blocks ($4.5x4.5x4.5cm^3$), one tungsten, one iron, in target volume on real data. Plot vertices with scatter angle above a cut-off.
- Blocks clearly visible.

Analysis Methods

- In real life application, need to make decision in ~1min
- Different analysis methods in employ
- Main method in the field: Maximum Likelihood Expectation-Maximization (L. Schultz *et al*, [3])
- Have developed complementary methods: calculate single discriminator value to differentiate between *investigate*, *OK*, and *scan longer*
    - Large sub-volumes metric method ← focus of talk
    - Graph-based clustering algorithm
- Metric method also useable for long time exposure imaging (applicable e.g. in context of nuclear waste)

Metric Method

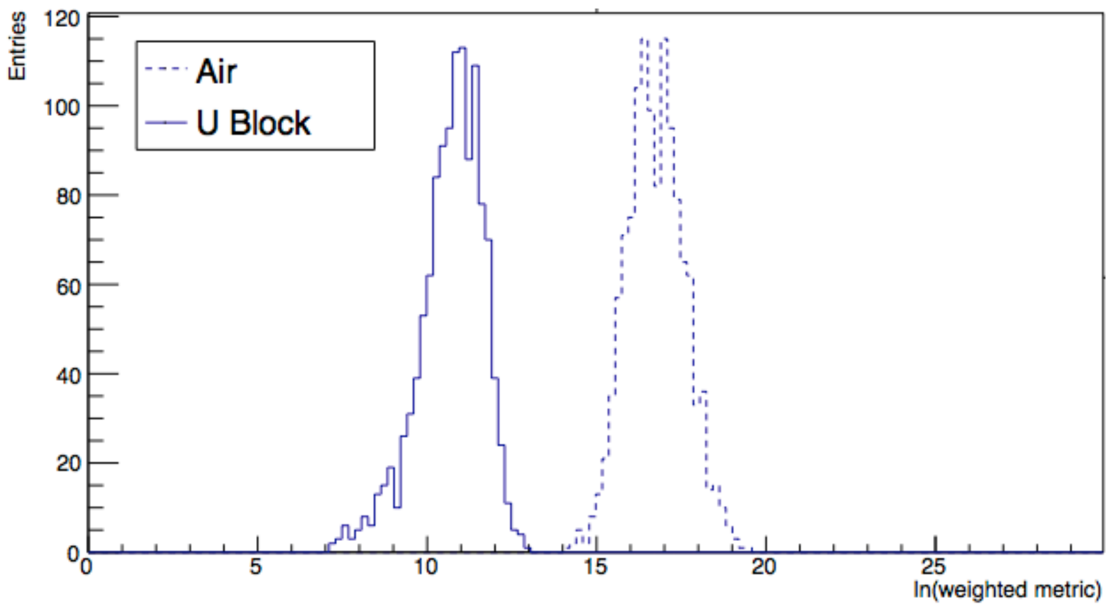
- Exploit that scatter events in signal material will be clustered more closely together by defining a *vertex metric*:

$$m_{ij} = \|\mathbf{v_i} - \mathbf{v_j}\|$$

- Weight metric by scatter angle, normalized with momentum:

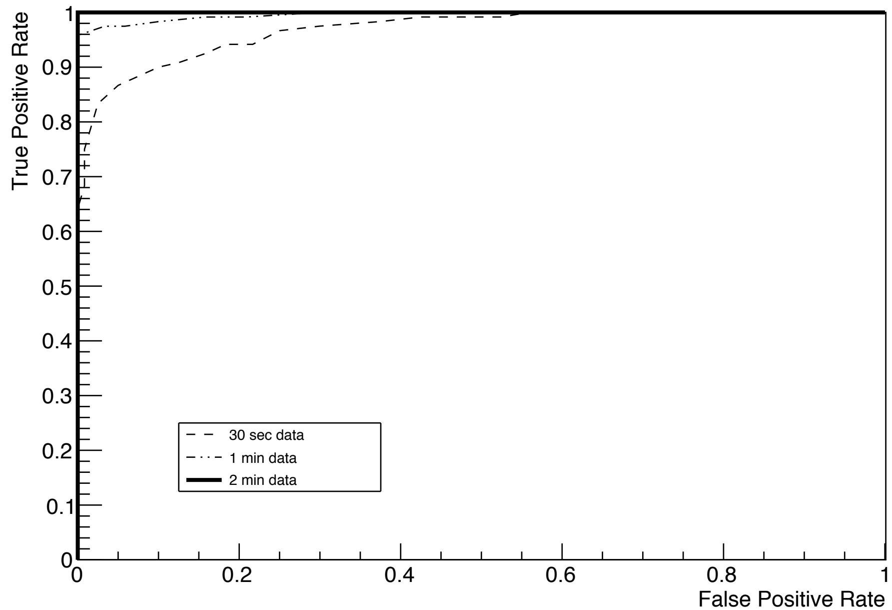
$$\widetilde{m}_{ij} = \frac{m_{ij}}{(\theta_i \tilde{p}_i) \cdot (\theta_j \tilde{p}_j)}$$

- Define median of this distribution as discriminating value.



Weighted vertex metric on MC.

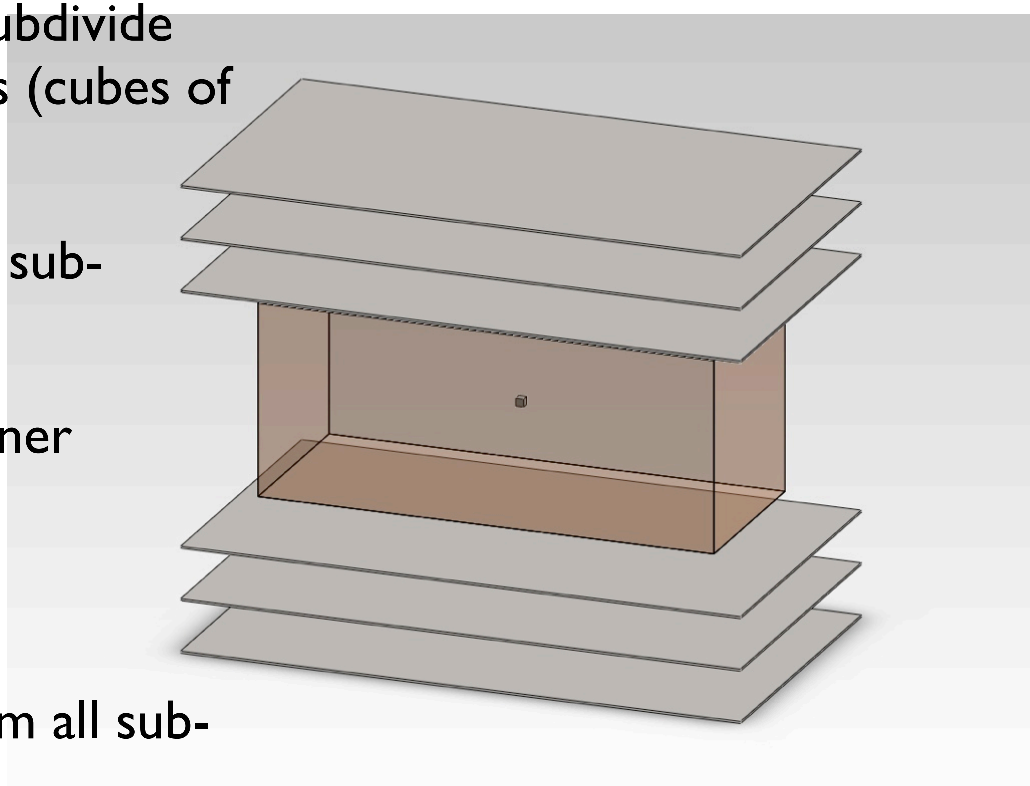
Performance On Real Data



- Investigate performance in ROC analysis
- Every data point on curve corresponds to choice of threshold for classifier
- Discriminator from 'target' data set below threshold ⟹ true positive
- Discriminator from 'background' set below threshold ⟹ false positive
- ROC curve has optimum at (0,1) ⟹ threshold with perfect separation
- 'Target' data set contains 7.5x7.5x7.5cm^3 tungsten block, background is empty cabinet
- Performance mainly constrained by lack of momentum measurement in prototype

Large Scale Case

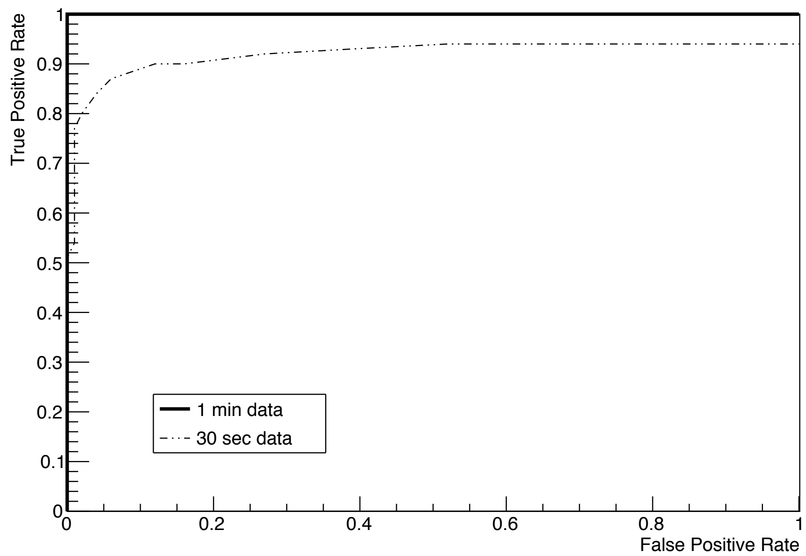
- For the large-scale application, subdivide volume into smaller sub-volumes (cubes of 25 cm side length)
- Perform metric analysis in every sub-volume
- Target could be on the edge/corner between multiple sub-volumes
  - Run on multiple shifted grids
- Output lowest discriminator from all sub-volumes as final discriminator



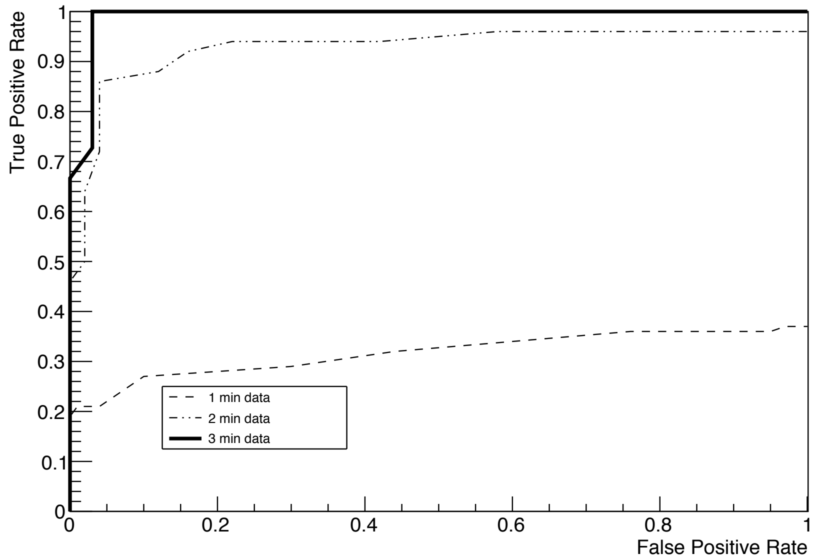
Schematic of 20ft container in detector.

Large Scale -- Performance

- 20ft container has hidden uranium block inside ($10 \times 10 \times 10 cm^3$)
- Uranium block is hidden in different (homogeneous) materials
- Container is filled up to weight limit (~26t)



ROC curves for container with rock.

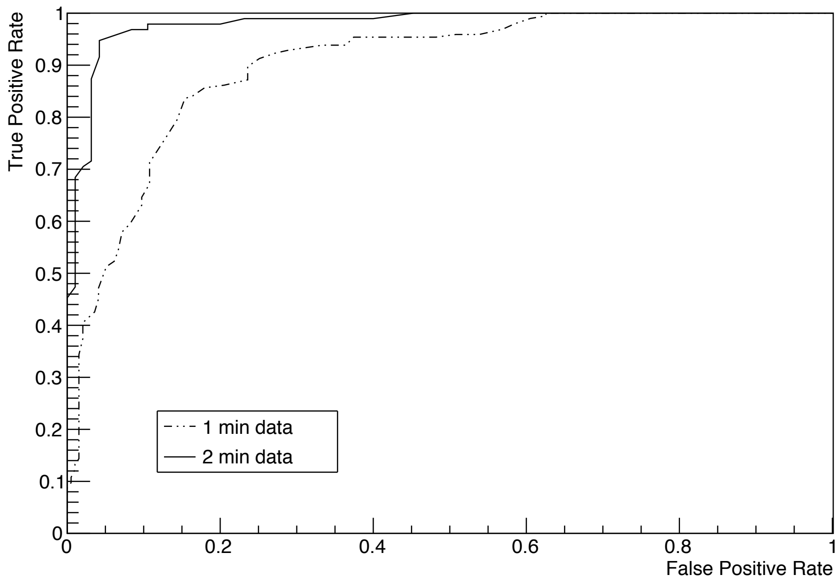


ROC curves for container with scrap iron.

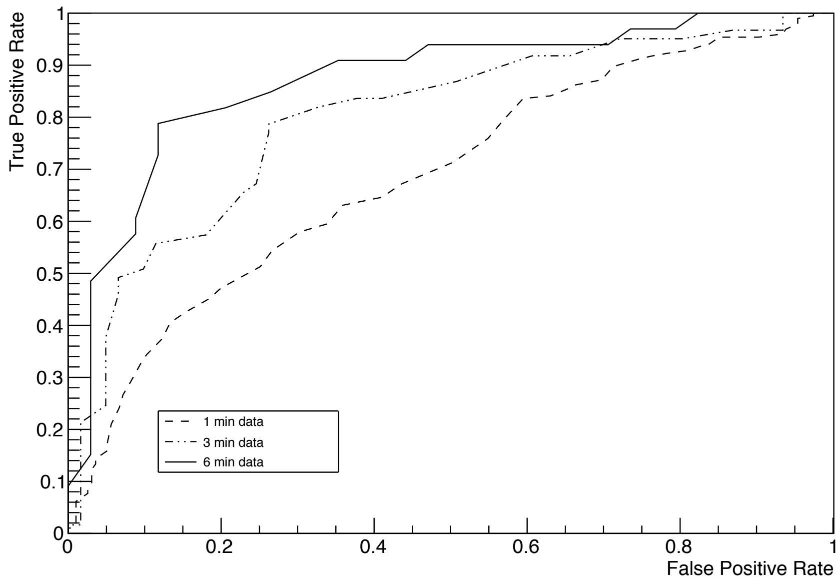
2013 JINST 8 P10013

Material ID

- MST in general is only sensitive to radiation length
- 'Harmless' material with short radiation length can lead to false positives
- Test material ID: $10\times10\times10\,cm^3$ U block vs $10\times10\times10\,cm^3$ blocks of W and Pb



ROC curves for U vs Pb.



ROC curves for U vs W.

Summary & Outlook

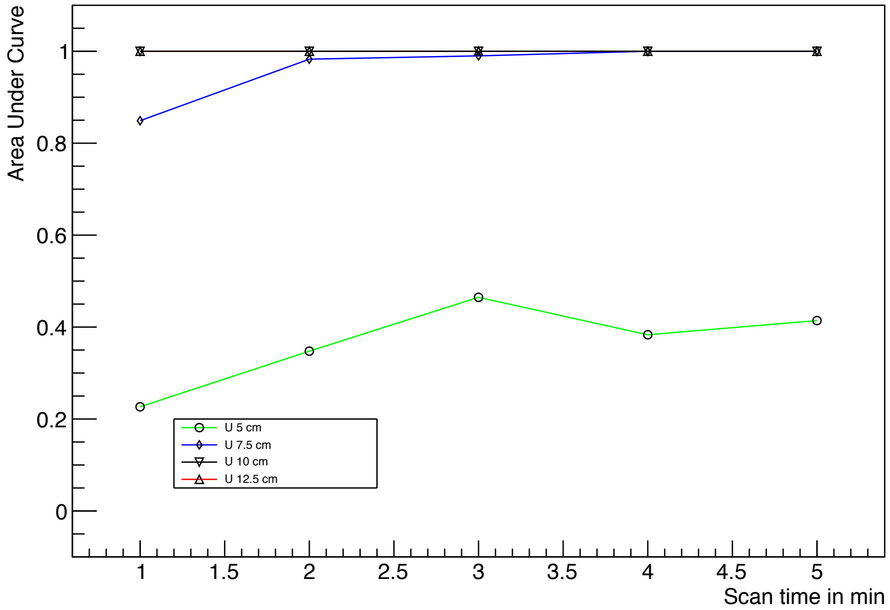
- Have developed a metric-based method to classify data sets into high-Z signal and background.
- Method successful on real data in our prototype and large-scale container simulations: can clear most scenarios in 1 min, problematic scenarios in ~3min
- Method robust against Pb, W in large quantities still somewhat problematic
- Method also applicable for long exposure scanning of objects (e.g. waste drums), work in progress

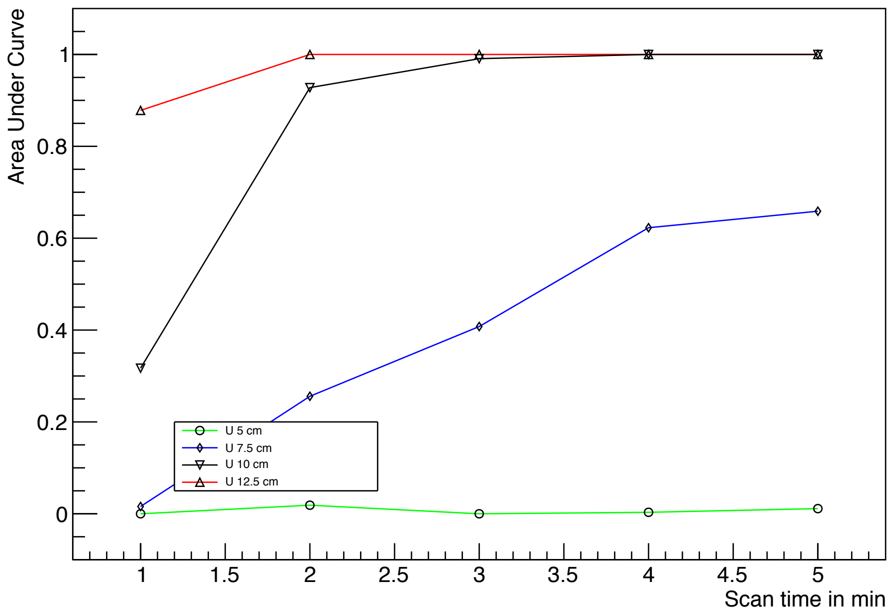
References

- [1] C. Thomay et al, *A binned clustering algorithm to detect high-Z material using cosmic muons*, 2013 JINST 8 P10013
- [2] P. Baesso et al., *A high resolution resistive plate chamber tracking system developed for cosmic ray muon tomography*, 2013 JINST 8 P08006.
- [3] L. Schultz et al, *Statistical Reconstruction for Cosmic Ray Muon Tomography*, IEEE T. Image Process. 16 (2007) 1985.
- [3] F. James, M. Roos (CERN), *Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations*, CERN-DD- 75-20 (1975), Comput.Phys.Commun.10:343- 367,1975.
- [4] R. Brun and F. Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.
- [5] GEANT4 collaboration, S. Agostinelli et al., *GEANT4: A Simulation*

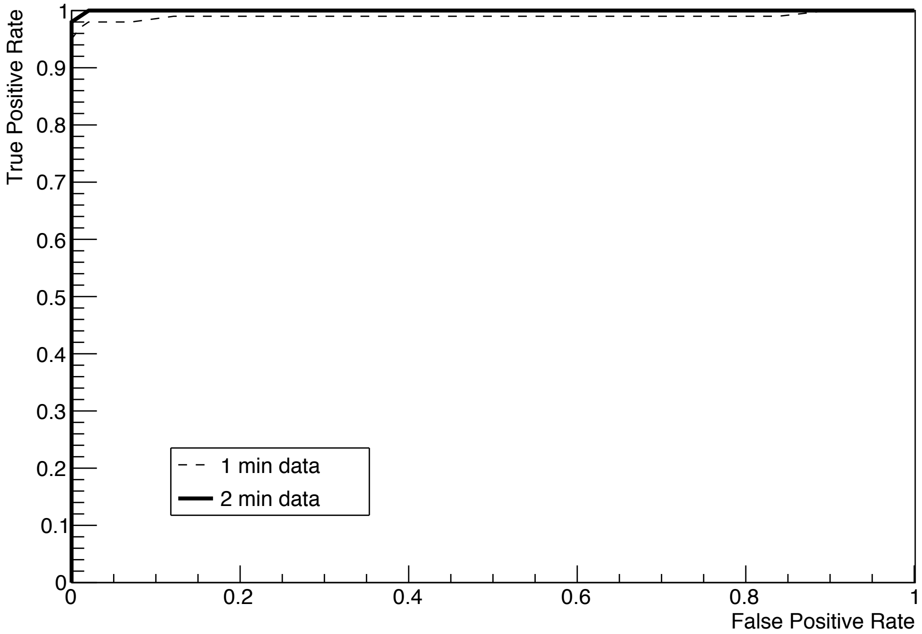
Backup

Size Test

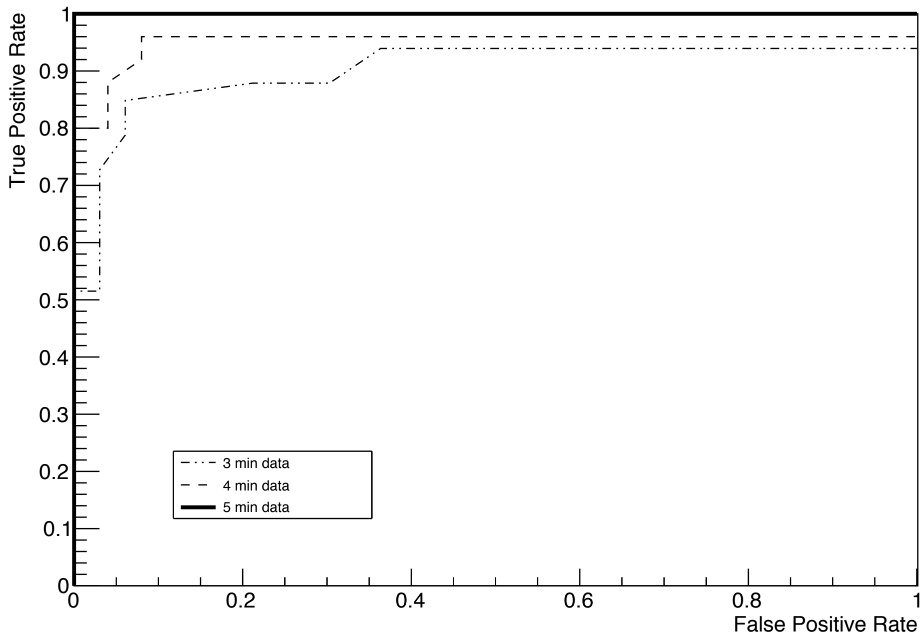

Area Under Curve
1
0.8
0.6
0.4
0.2
0
U 5 cm
U 7.5 cm
U 10 cm
U 12.5 cm
1
1.5
2
2.5
3
3.5
4
4.5
5
Scan time in min


Area Under Curve
1
0.8
0.6
0.4
0.2
0
U 5 cm
U 7.5 cm
U 10 cm
U 12.5 cm
1
1.5
2
2.5
3
3.5
4
4.5
5
Scan time in min

Momentum Uncertainty



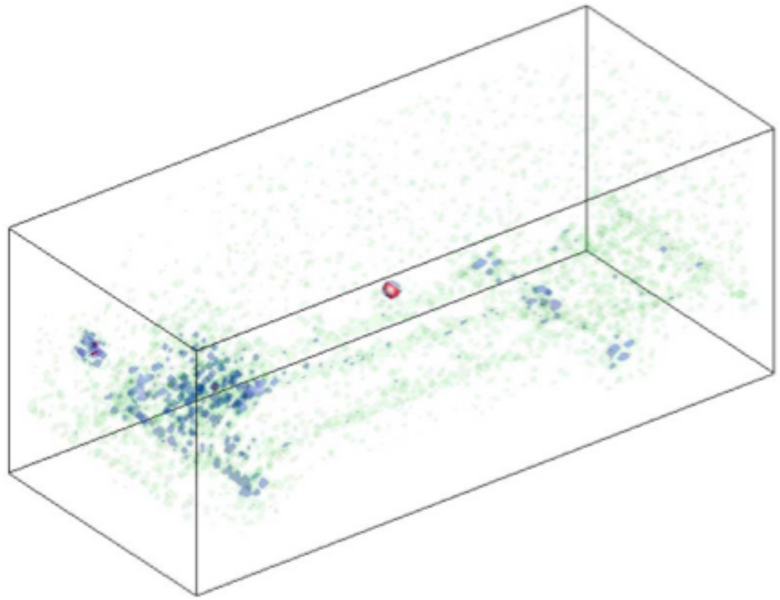
ROC curves for container with rock.



ROC curves for container with scrap iron.

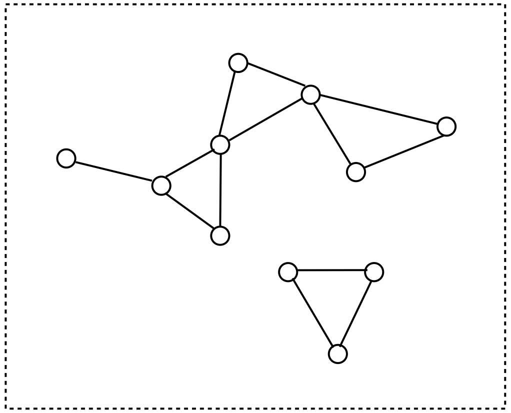
Data Analysis

- Aim is to scan the target for special nuclear material in ~1 min.
- AWE working on an EM (Expectation-Maximization) algorithm to perform tomography on the scanned data set.
- Shows 10x10x10cm tungsten block in a passenger van.
- Different approach being developed at UoB: instead of doing tomography, condense data into a single decision variable (*threat*, *ok*, or *scan longer*).



MRF Method

- Metric method fairly powerful despite its simplicity.
- However, not the final answer: not well-equipped to deal with multiple target blocks or more complicated background geometry.
- Also, not optimal for large-scale applications: due to the combinatorics, the algorithm runs in $O(n^2)$. Ok for 200 tracks, but for ~10^5 tracks (cargo container case) runs for several minutes.
- More refined approach: instead of brute-force summing of all vertex combinations, try to obtain the probability of signal-like regions.
- Basic assumption: the probability of a track being signal-like is conditionally dependent on the signal probability of the tracks in its neighbourhood and independent of everything else.
- With this assumption, can model our system as a Markov random field (MRF).
- If we can establish a MRF for our system, can calculate signal probabilities for any system state.



Schema of a MRF. The circles represent elementary events, the edges represent mutual conditional dependence.

MRF Method

- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.

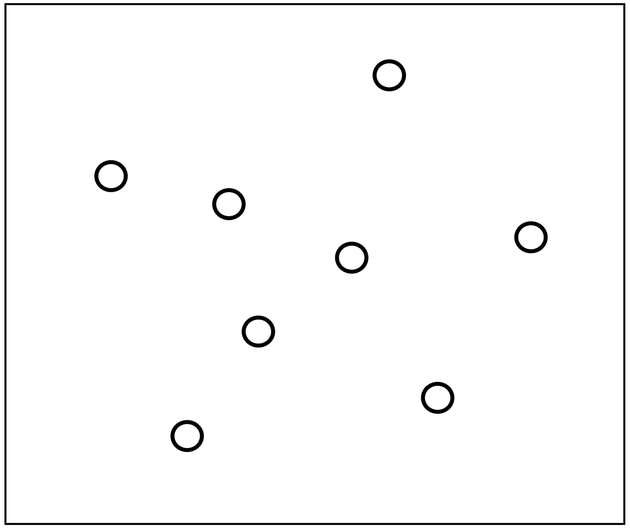
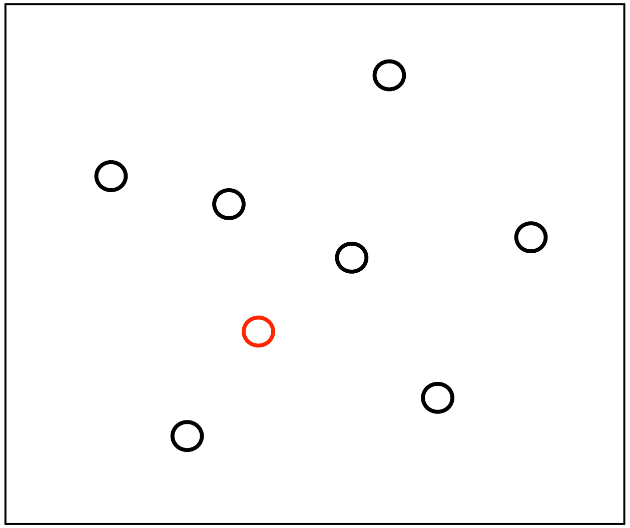


Illustration: starting set.

MRF Method

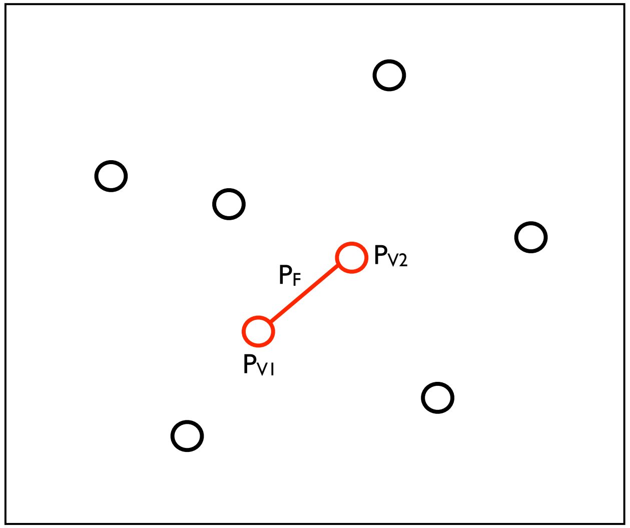
- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



Spawning filter at the red vertex.

MRF Method

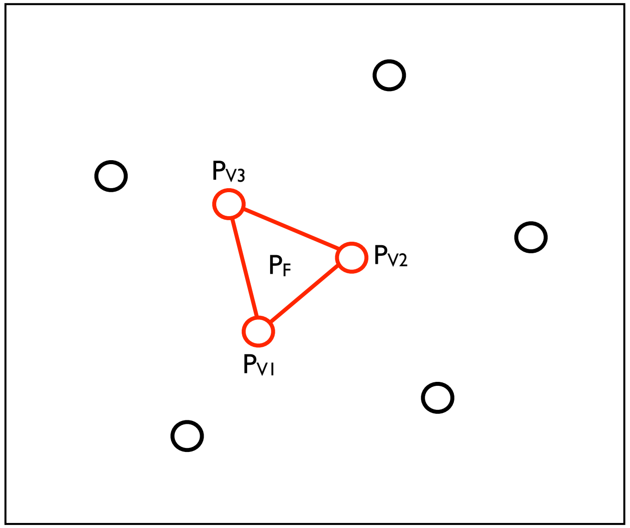
- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



Adding vertex 2 to the filter.

MRF Method

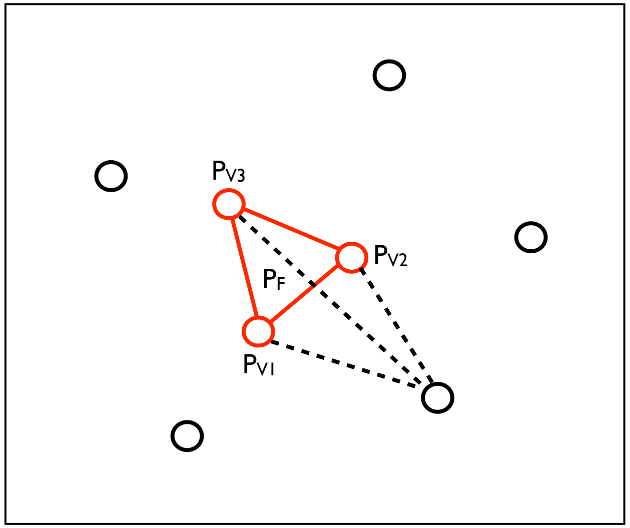
- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



Adding vertex 3. P_{V1} and P_{V2} have increased since there are more edges connected. P_F is at the highest value so far.

MRF Method

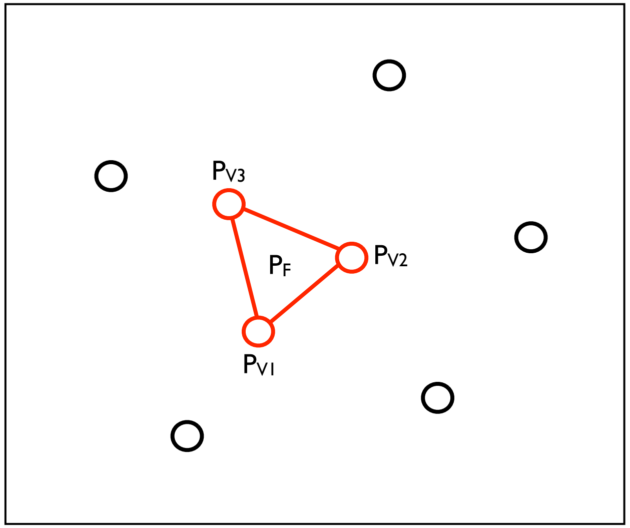
- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



Attempting to add vertex 4. P_F drops below cutoff, so vertex 4 is ignored and the filter restored to the last state.

MRF Method

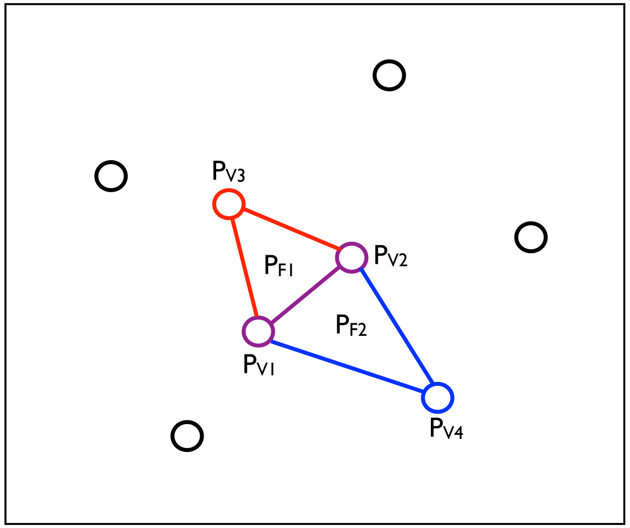
- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



Resized final filter. In this simplified case, the final output would be $-\log(P_F)$.

MRF Method

- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.



In the end, filters may end up sharing vertices (purple); probabilities are still well-defined.

MRF Method

- Establish MRF by using a combinatorial Kalman Filter algorithm.
  - Spawn filter instance for every vertex in set.
  - Define probabilities for individual vertices P_V and the set of vertices currently in the filter P_F, such that P_V increases with the number of attached edges and P_F decreases with the number of vertices → P_F has unique maximum.
  - Add neighbour vertices to filter and update probabilities until P_F drops below a cutoff. Keep P_F for each iteration.
  - Once all filters are done, resize all filters to state of 'maximal conditional dependence' to find the most 'fitting' MRF.
  - Consider the result as a MRF with edges between all vertices of the resized filters.
- Can now ask the MRF for probability that ANY of the resulting filters is completely signal-like. Define -log of that value as the output discriminator.