Medical Imaging Inspired Vertex Reconstruction at the Large Hadron Collider

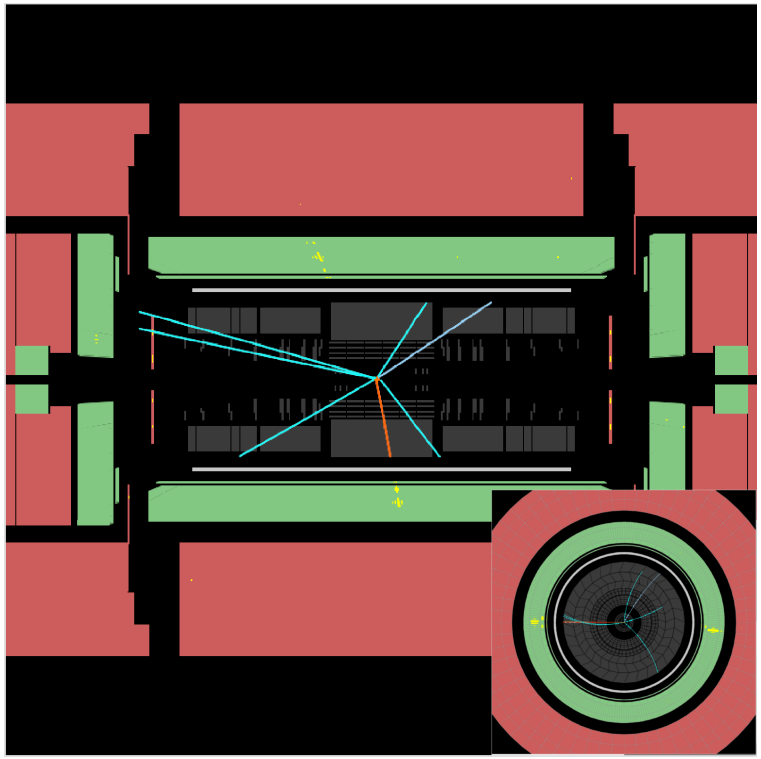
S Hageböck E von Toerne

Physikalisches Institut
University of Bonn

05/24/2012

Motivation

- New ansatz: Apply Medical Imaging (MI) to vertex reconstruction
  Suggested by H. Kagan, Ohio State University

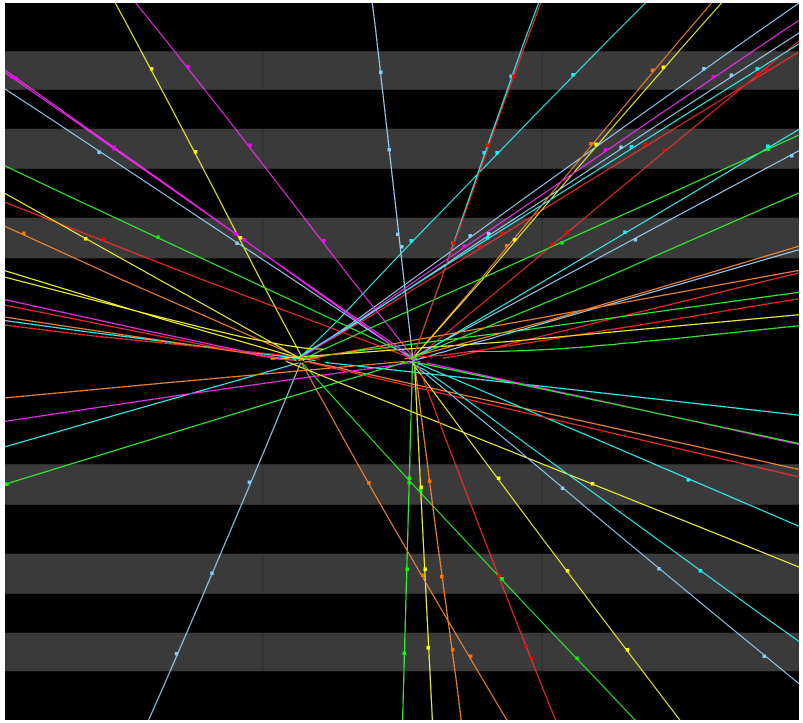


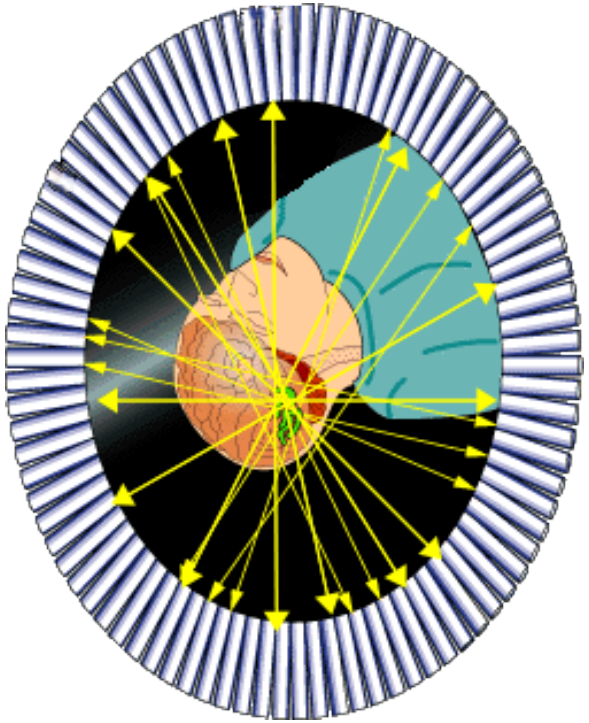
- **Find** primary vertices (hard and minimum bias), seed for fits
  - 4-momenta in $H \to \gamma\gamma$: "Hard vertex" looks like min bias
  - Luminosity: Count vertices
- Based on Radon-transformation, similarities to ZVTop in HEP, but modern MI filtering techniques

- Works with large numbers of tracks
- Fast with many tracks

Comparison: HEP & PET

	HEP	**P**ositron **E**mission **T**omography
Problem	Position of reaction/interaction	Position of tumor
Data	Tracks	Photon pairs from $e^+e^- \to \gamma\gamma$
Amount of data	$< N_{\mathrm{Trk}}/\mathrm{Vtx} > \approx 20$	$> 10^6$, **more is better**
Methods	Adaptive fitting / Topological finding	Filtered Backprojection $\to$ inv. Radon-transformation





Topological Vertex Reconstruction: ZVTop

- D J Jackson. "A Topological Vertex Reconstruction Algorithm for Hadronic Jets". In: *Nucl. Inst. & Meth. A* 388 (1997), pp. 247–253
- Similarity to inverse Radon-transformation

Steps of topological reconstruction:

1. Create track density distribution
2. Apply "filter"
3. Find vertex candidates (high track density)
4. Associate tracks to candidates
5. Fit the vertices

The problems of vertex reconstruction

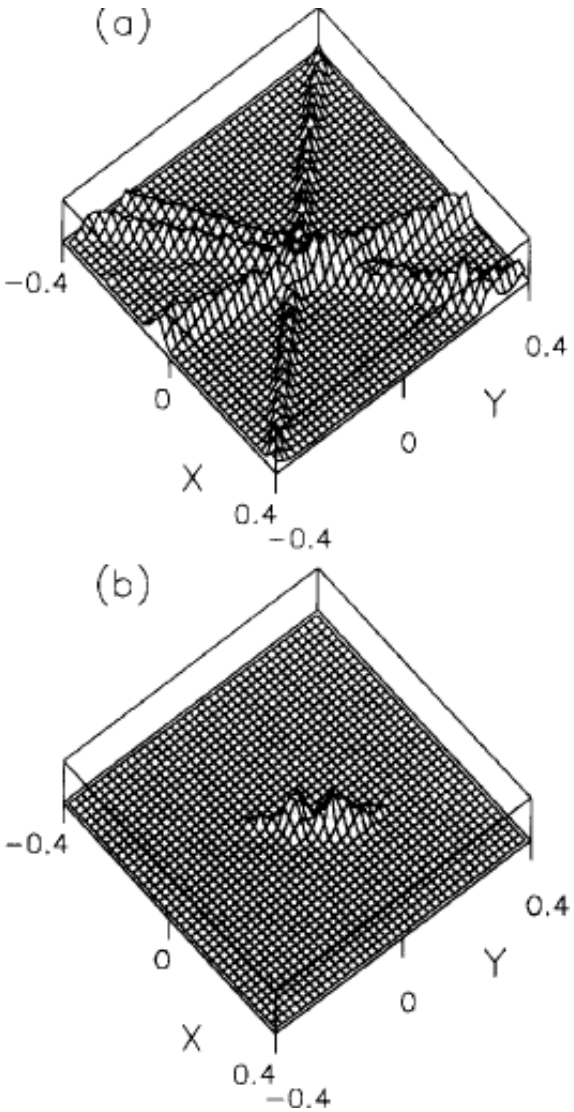
1. Vertex finding
2. Track association
3. Vertex fitting

Topological Vertex Reconstruction: ZVTop

- D J Jackson. "A Topological Vertex Reconstruction Algorithm for Hadronic Jets". In: *Nucl. Inst. & Meth. A* 388 (1997), pp. 247–253
- Similarity to inverse Radon-transformation

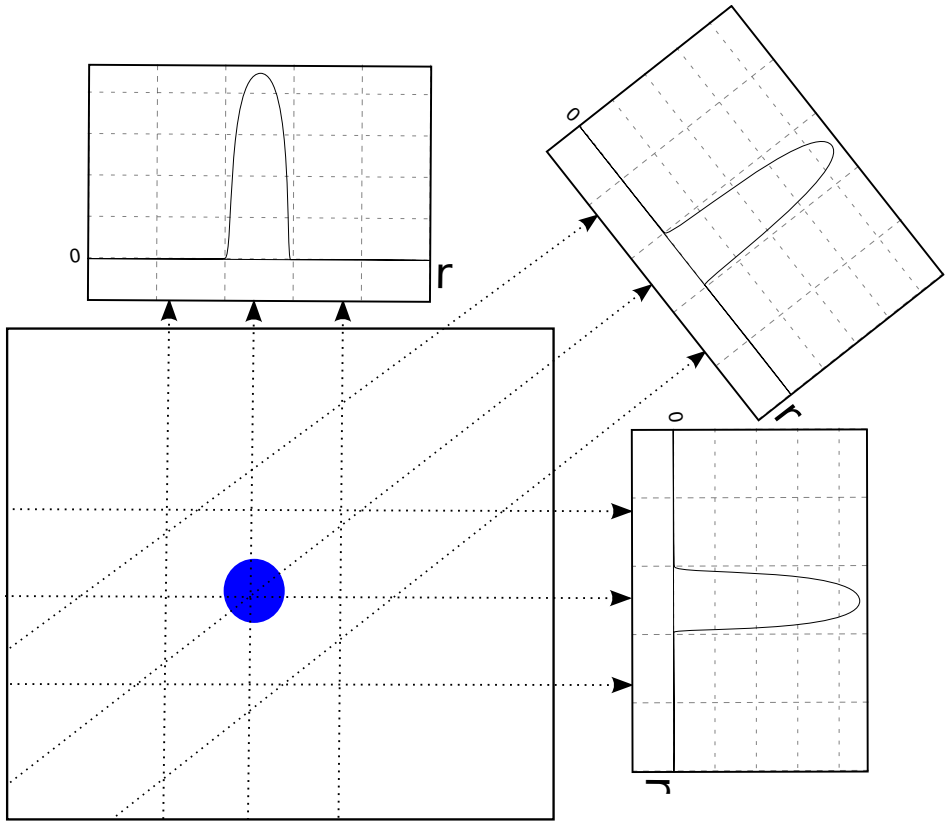
Steps of topological reconstruction:

1. Create track density distribution
2. Apply "filter"
3. Find vertex candidates (high track density)
4. Associate tracks to candidates
5. Fit the vertices



Radon Transformation

- Most medical imaging algorithms 2D
- Tomography: Imaging technique on sections
- Sections divided into multitude of projections

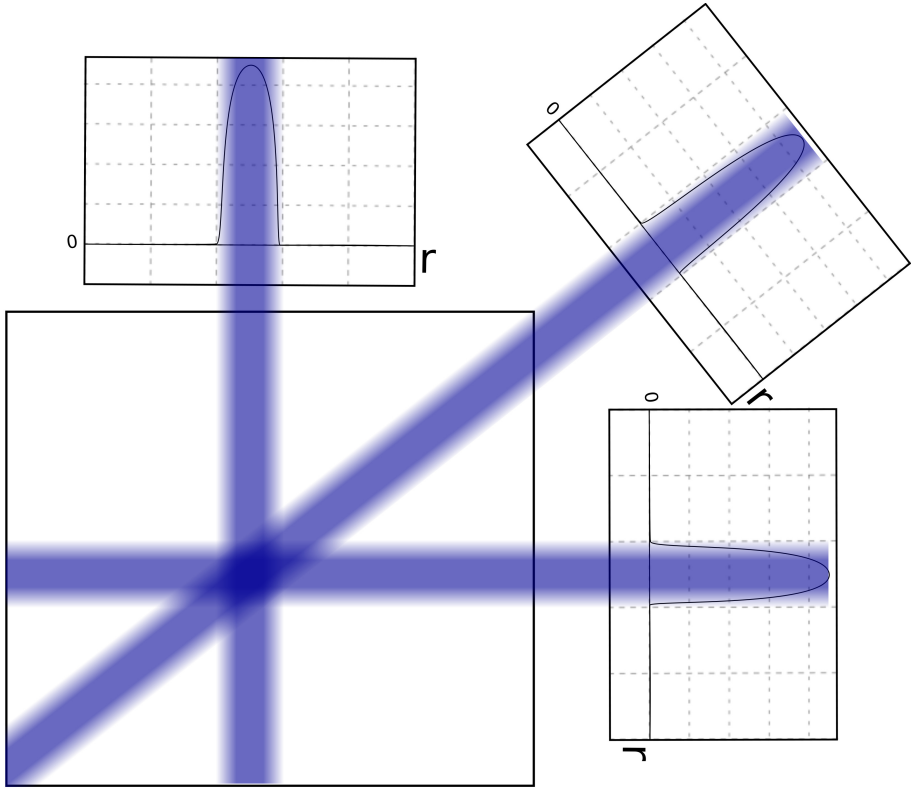


- Problem in MI: Only line-*integrals* known
- HEP: Position of Vertex along the track unknown
- Mathematics: **Radon-transformation**

$$p_\alpha(r) = \int_L f(x,y)\ \mathrm{d}l$$

Towards Medical Imaging Filters: Backprojection

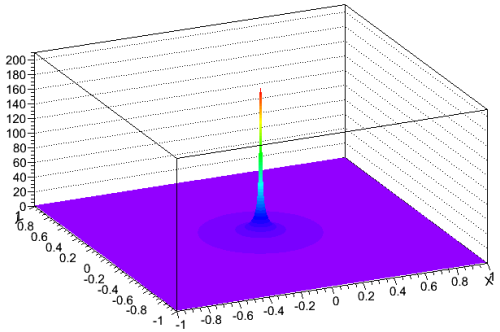
- Image reconstruction: Inverse Radon-transformation
- **Backproject** all projections onto grid



- Problem of line-*integrals*: Star artifacts
- Many projections: Blurring
- Counteract blurring by Filtering

Filtered Backprojection: Counteract the Blurring

- Backprojection: Convolution of true image with the Point Spread Function



$$f(x,y)_{\mathrm{BP}} = \left(f_{\mathrm{true}} * s\right)(x,y)$$

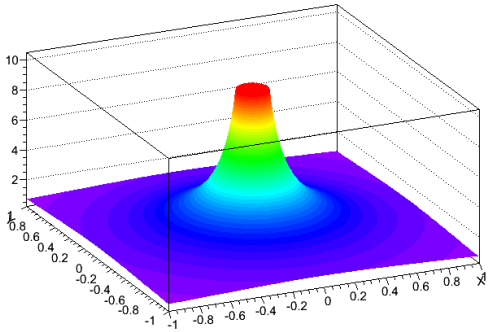
$$= \iint f(x-a, y-b)_{\mathrm{true}} \cdot s(a,b)\,\mathrm{d}a\,\mathrm{d}b$$

$$s = \frac{1}{r}$$

a, b: Space coordinates used for integration

Filtered Backprojection: Counteract the Blurring

- Backprojection: Convolution of true image with the Point Spread Function

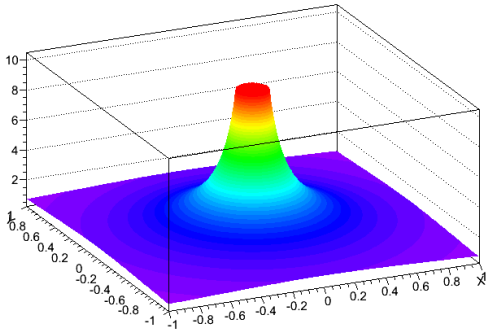


$$f(x,y)_{\mathrm{BP}} = \left(f_{\mathrm{true}} * s\right)(x,y)$$
$$= \iint f(x-a, y-b)_{\mathrm{true}} \cdot s(a,b)\,\mathrm{d}a\,\mathrm{d}b$$
$$s = \frac{1}{r}$$

a, b: Space coordinates used for integration

Filtered Backprojection: Counteract the Blurring

- Backprojection: Convolution of true image with the Point Spread Function



$$f(x,y)_{\mathrm{BP}} = \left(f_{\mathrm{true}} * s\right)(x,y)$$
$$= \iint f(x-a, y-b)_{\mathrm{true}} \cdot s(a,b)\,\mathrm{d}a\,\mathrm{d}b$$
$$s = \frac{1}{r}$$

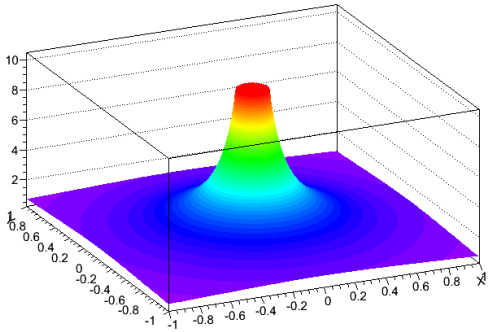
a, b: Space coordinates used for integration

- Deconvolution $\rightarrow$ Fourier space:

$$F(u,v)_{\mathrm{BP}} = F_{\mathrm{true}}(u,v) \cdot S(u,v)$$
$$S(u,v) = \frac{1}{k} \qquad k = \left|\begin{pmatrix} u \\ v \end{pmatrix}\right|$$

Filtered Backprojection: Counteract the Blurring

- Backprojection: Convolution of true image with the Point Spread Function



$$f(x,y)_{\mathrm{BP}} = \left(f_{\mathrm{true}} * s\right)(x,y)$$
$$= \iint f(x-a, y-b)_{\mathrm{true}} \cdot s(a,b)\,\mathrm{d}a\,\mathrm{d}b$$
$$s = \frac{1}{r}$$

a, b: Space coordinates used for integration

- Deconvolution $\rightarrow$ Fourier space:

$$F(u,v)_{\mathrm{BP}} = F_{\mathrm{true}}(u,v) \cdot S(u,v)$$
$$S(u,v) = \frac{1}{k} \qquad k = \left|\begin{pmatrix} u \\ v \end{pmatrix}\right|$$

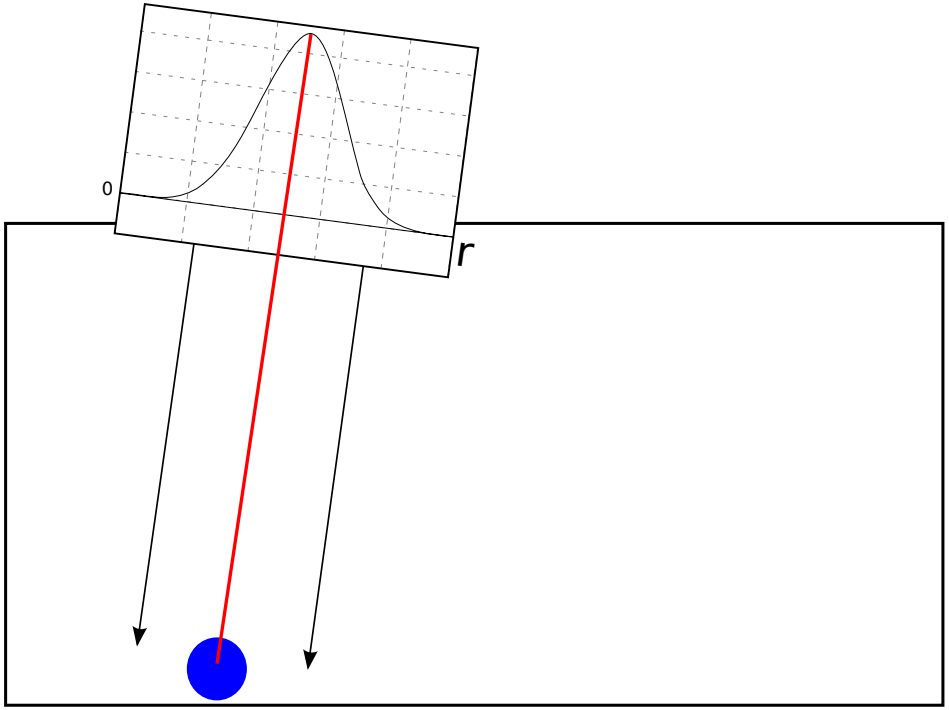
Filtered image in Fourier space ("k filter, Ramp filter")

$$F_{\mathrm{true}}(u,v) = F(u,v)_{\mathrm{BP}} \cdot k$$

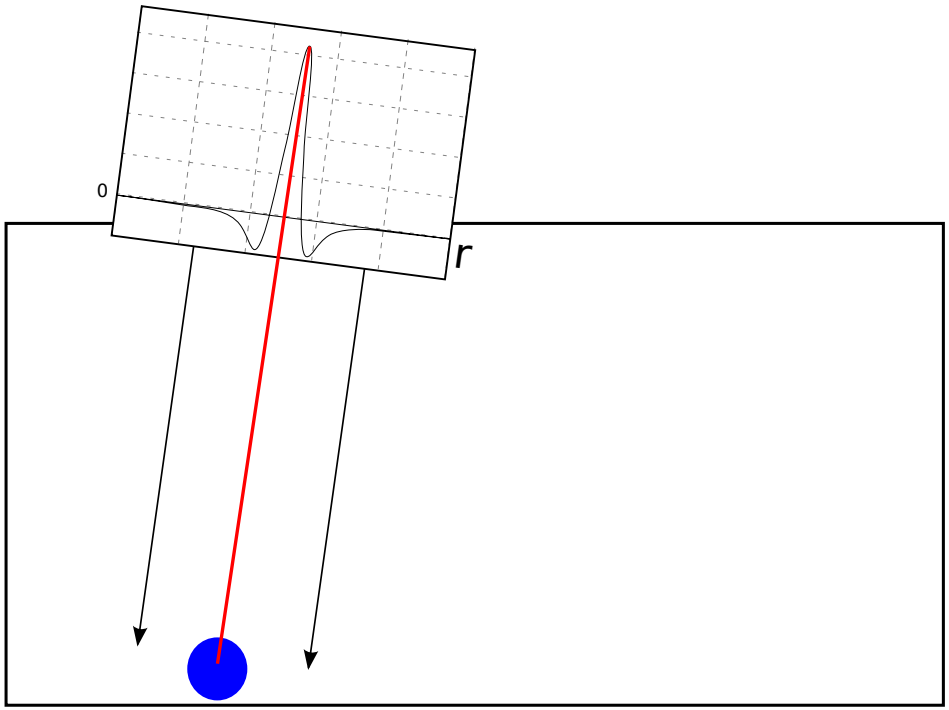
Example with 2D-only tracks

- Gaussian profile for each track
- Project track after track into ROOT TH2D

No filter:



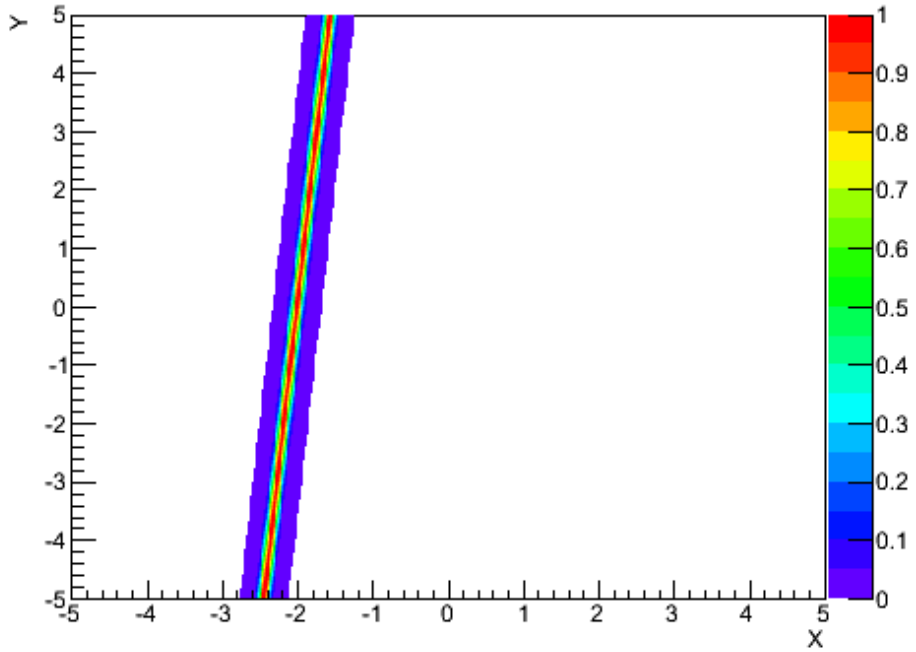
Ramp filter:



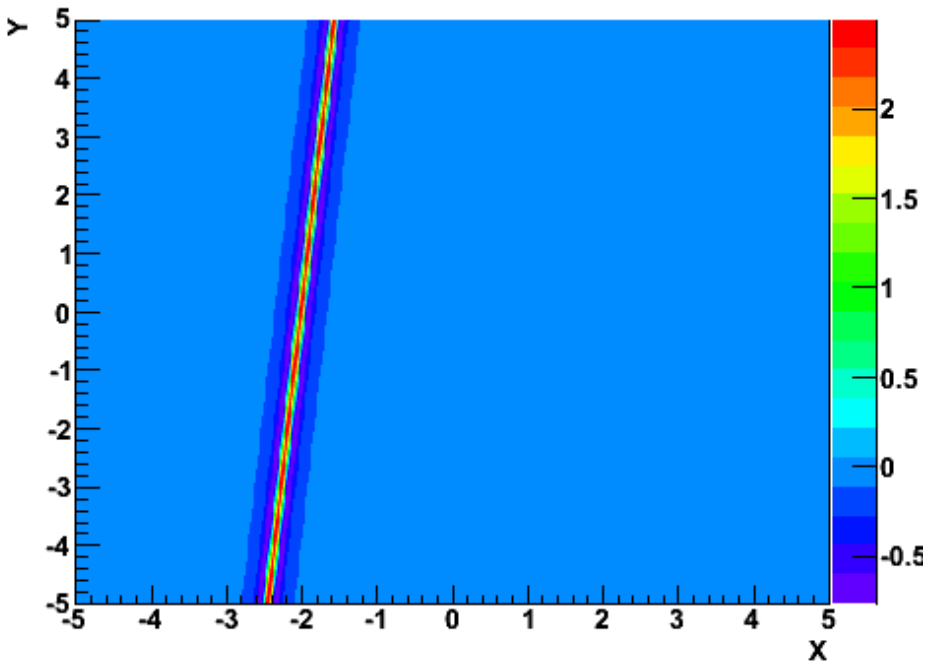
Example with 2D-only tracks

- Gaussian profile for each track
- Project track after track into ROOT TH2D

No filter:

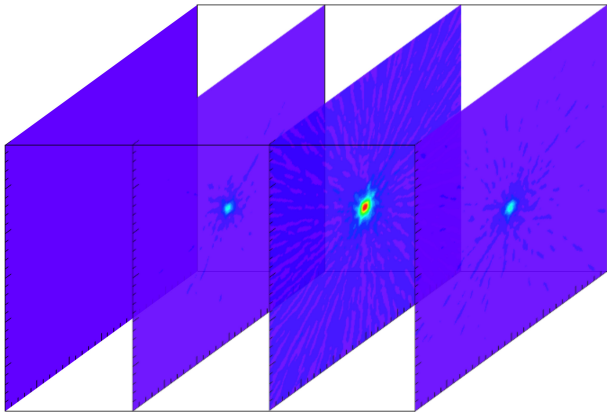


Ramp filter:

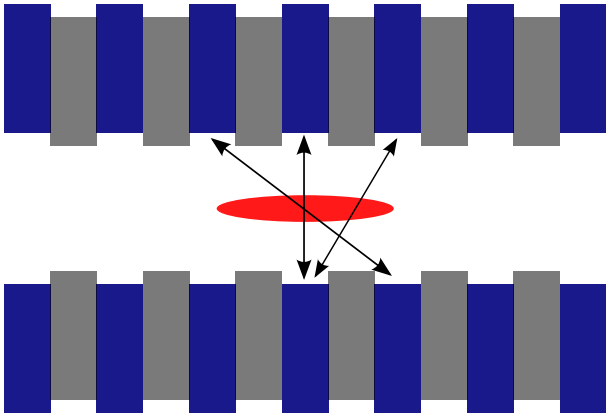


Medical Imaging in Three Dimensions

Stack transverse planes:



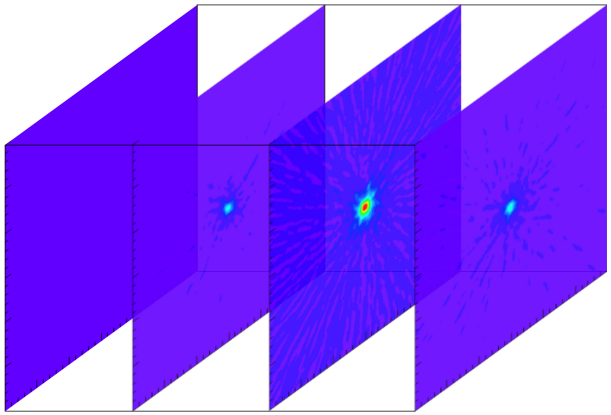
3D PET:



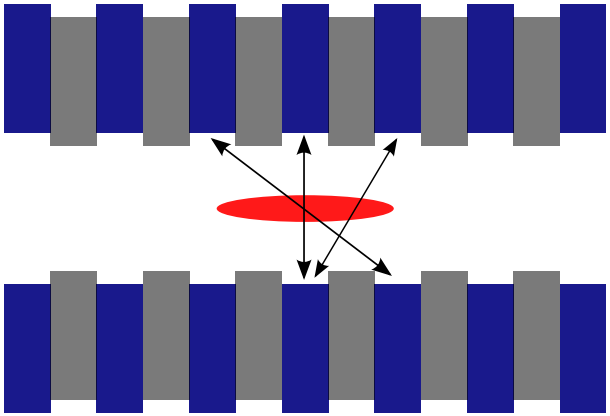
- Conventional approach
  - Reuse 2D algorithms on multiple planes
  - Not suitable for vertex finding

Medical Imaging in Three Dimensions

Stack transverse planes:



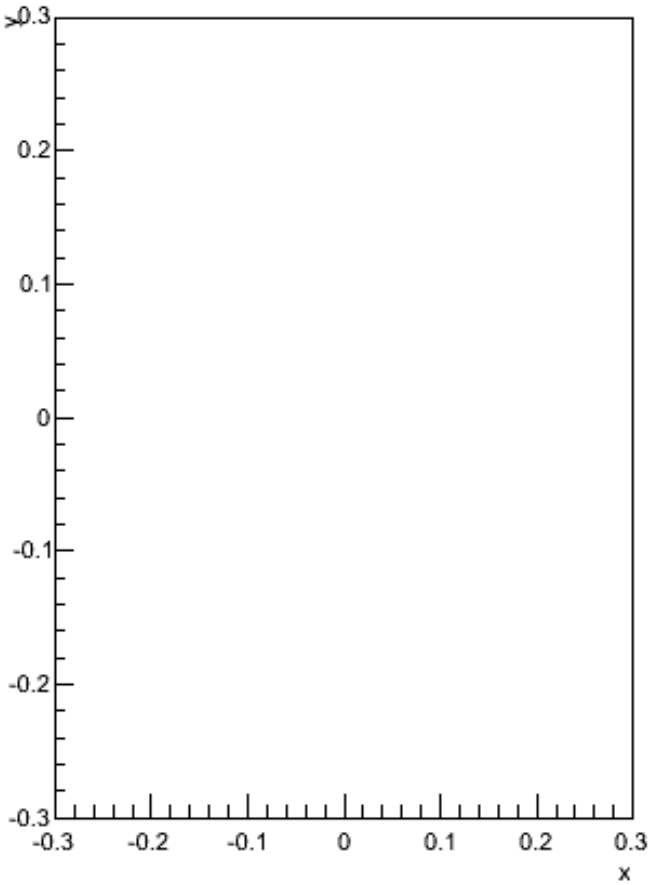
3D PET:

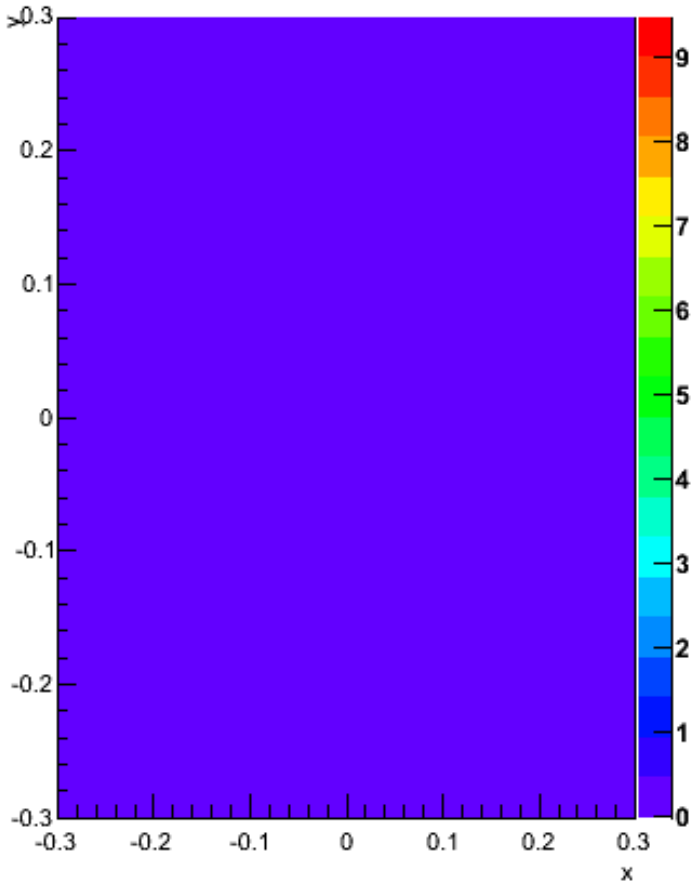


- Conventional approach
  - Reuse 2D algorithms on multiple planes
  - Not suitable for vertex finding
- We use **full 3D tomography**:
  - Acceptance corrected Ramp-Filter* ($\eta = 2.5 \rightarrow 80^\circ$)
  - 3D window function
  - Math more complex: 3D FFTs
  - All measurements usable

* M Defrise, D W Townsend, and R Clack. "Three-dimensional image reconstruction from complete projections". In: *Physics in Medicine and Biology* 34.5 (1989), p. 573

Vertex Density for Simulated LHC Events





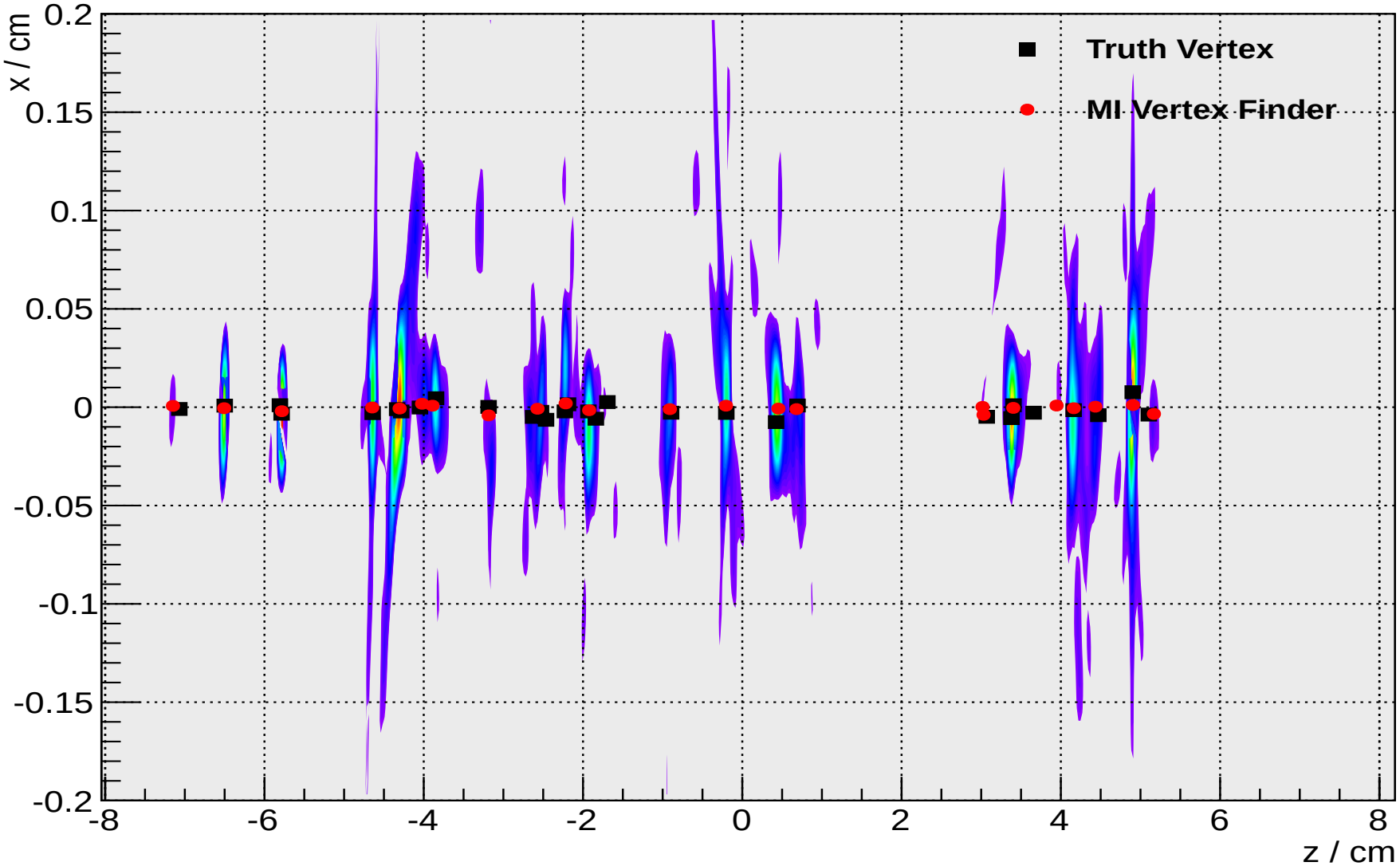
Left Simple backprojection Right 3D-filtered backprojection

"Flight" through typical 2011 LHC beamspot

$\sigma_{\mathrm{xy}} = 32\,\mu\mathrm{m} \quad \sigma_{\mathrm{z}} = 4.2\,\mathrm{cm}$

Vertex Density for Simulated LHC Events

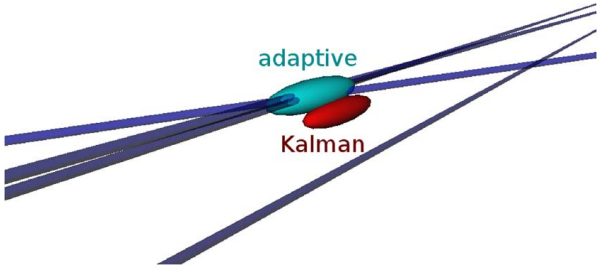
- Tune sharpness of filter → window function
  Best Results: XY: soft Z: sharp



Comparison to vertex finders used in HEP: LHC

RAVE 0.6.6

- Open source vertexing toolkit
- W Waltenberger et al. "The RAVE/VERTIGO vertex reconstruction toolkit and framework". In: *Journal of Physics: Conference Series* 119.3 (2008), p. 032037
- Provides modern adaptive algorithms in use at LHC: AVR (Adaptive Vertex Reconstructor)

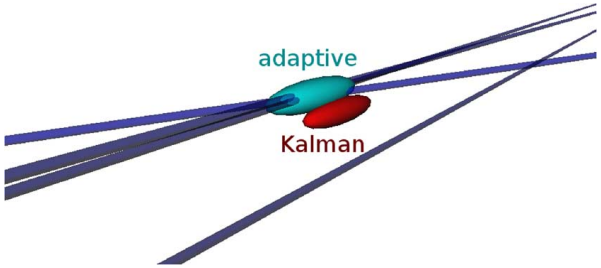


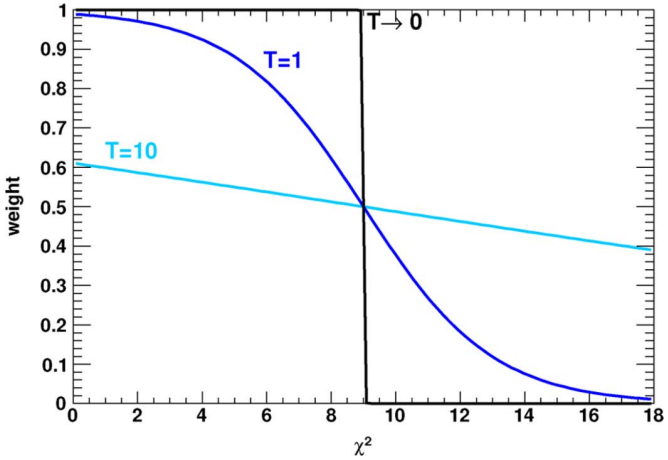
- Adaptive: Robustification of conventional χ^2-fits ("Kalman")

Comparison to vertex finders used in HEP: LHC

RAVE 0.6.6

- Open source vertexing toolkit
- W Waltenberger et al. "The RAVE/VERTIGO vertex reconstruction toolkit and framework". In: *Journal of Physics: Conference Series* 119.3 (2008), p. 032037
- Provides modern adaptive algorithms in use at LHC: AVR (Adaptive Vertex Reconstructor)



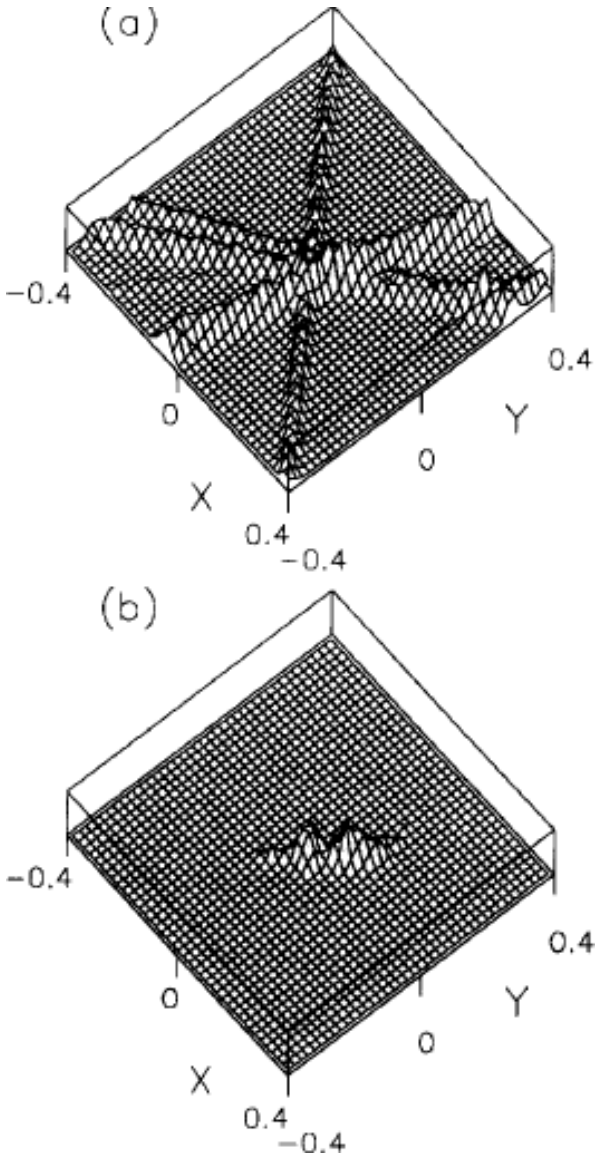


- Adaptive: Robustification of conventional χ^2-fits ("Kalman")
- How: Introduce weights for each track i:

$$w_i = \frac{e^{-\chi_i^2/2T}}{e^{-\chi^2_{\mathrm{Cutoff}}/2T} + e^{-\chi_i^2/2T}}$$

- Automatic track association
  $\rightarrow$ Can be used with tracks from multiple vertices

Comparison to vertex finders used in HEP: ZVTop



- C++ implementation of ZVTop for ILC integrated in RAVE (experimental)
- Very slow even when used with only a few vertices
- Tailored to secondary vertex finding in jets
- Still usable for primary vertex finding

Setup for Benchmarks

Collision:	
Min N_{trk}	4
Min p_{t}	50 GeV
E_{CM}	7 TeV
Beamspot:	
$\sigma_{xy,\mathrm{BS}}$	34 µm
$\sigma_{z,\mathrm{BS}}$	4.6 cm
Tracks:	
Min p_{t}	250 MeV
Max η	2.5
Uncertainties ($\lvert\eta\rvert \approx 0.5$):	
$\sigma(\phi_0)$	70 µrad
$\sigma(\cot\theta)$	0.7×10^{-3}
$\sigma(d_0)$	10 µm
$\sigma(z_0 \times \sin\theta)$	91 µm

- LHC-Collisions Pythia 6.416, default settings. 1 hard vtx, $\mathrm{Poisson}(\mu)$ min bias
- Vertex distribution 3D-Gaussian
- Tracking detector Gaussian smearing to simulate resolution, inspired by ATLAS/CMS

Setup for Benchmarks

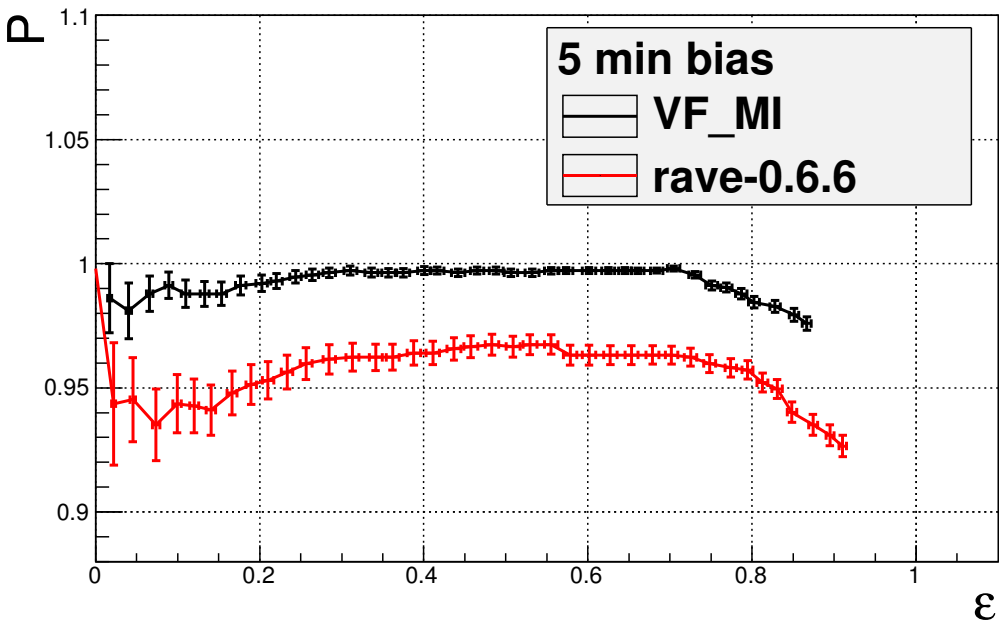
Collision:	
Min N_{trk}	4
Min p_{t}	50 GeV
E_{CM}	7 TeV
Beamspot:	
$\sigma_{xy,\mathrm{BS}}$	34 µm
$\sigma_{z,\mathrm{BS}}$	4.6 cm
Tracks:	
Min p_{t}	250 MeV
Max η	2.5
Uncertainties ($\lvert\eta\rvert \approx 0.5$):	
$\sigma(\phi_0)$	70 µrad
$\sigma(\cot\theta)$	0.7×10^{-3}
$\sigma(d_0)$	10 µm
$\sigma(z_0 \times \sin\theta)$	91 µm

- LHC-Collisions Pythia 6.416, default settings. 1 hard vtx, $\mathrm{Poisson}(\mu)$ min bias
- Vertex distribution 3D-Gaussian
- Tracking detector Gaussian smearing to simulate resolution, inspired by ATLAS/CMS
- Evaluation
  - Assign tracks to closest reconstructed vertex (Exception: $d > 4\sigma$)
  - Require minimum of 3 tracks per reco vertex
  - $> 50\,\%$ of tracks from one truth vertex

$\Rightarrow$ **Vertex accepted**

universitätbonn

Purity and Efficiency

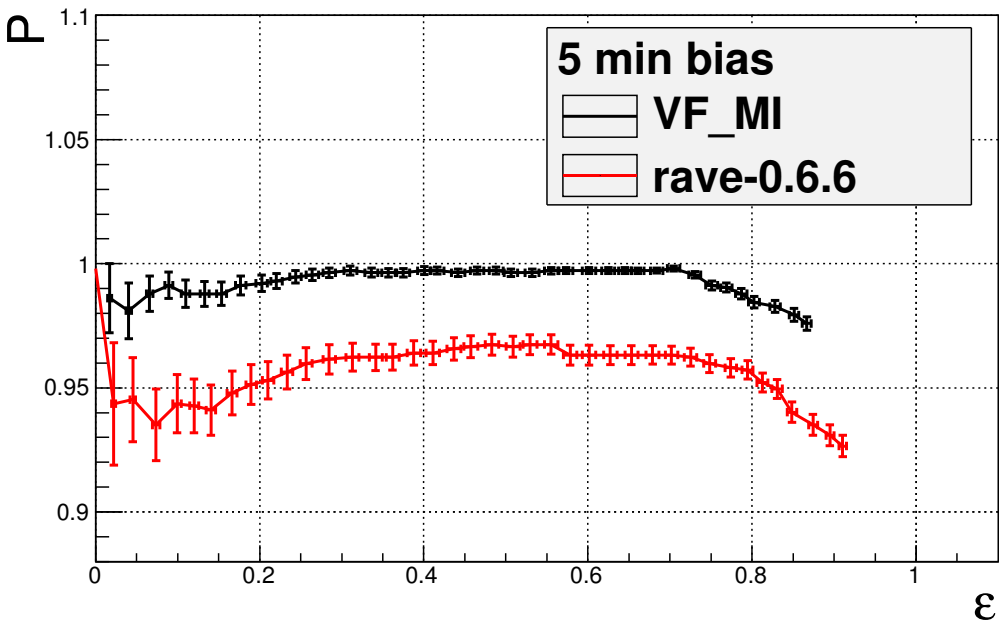


- Evaluation using hard **and** min bias vertices
- Compare multiple operating points (vertex quality cut): Purity vs. Efficiency curves
- $\varepsilon = \frac{N_{\text{correct}}}{N_{\text{truth}}}$ $\mathrm{P} = \frac{N_{\text{correct}}}{N_{\text{reco}}}$
- Vertex quality:

  RAVE $\frac{\chi^2}{\text{n.d.f.}}$

  MI VF Vertex density
- Cannot find all vertices (merging) $\Rightarrow \varepsilon < 1$

Purity and Efficiency



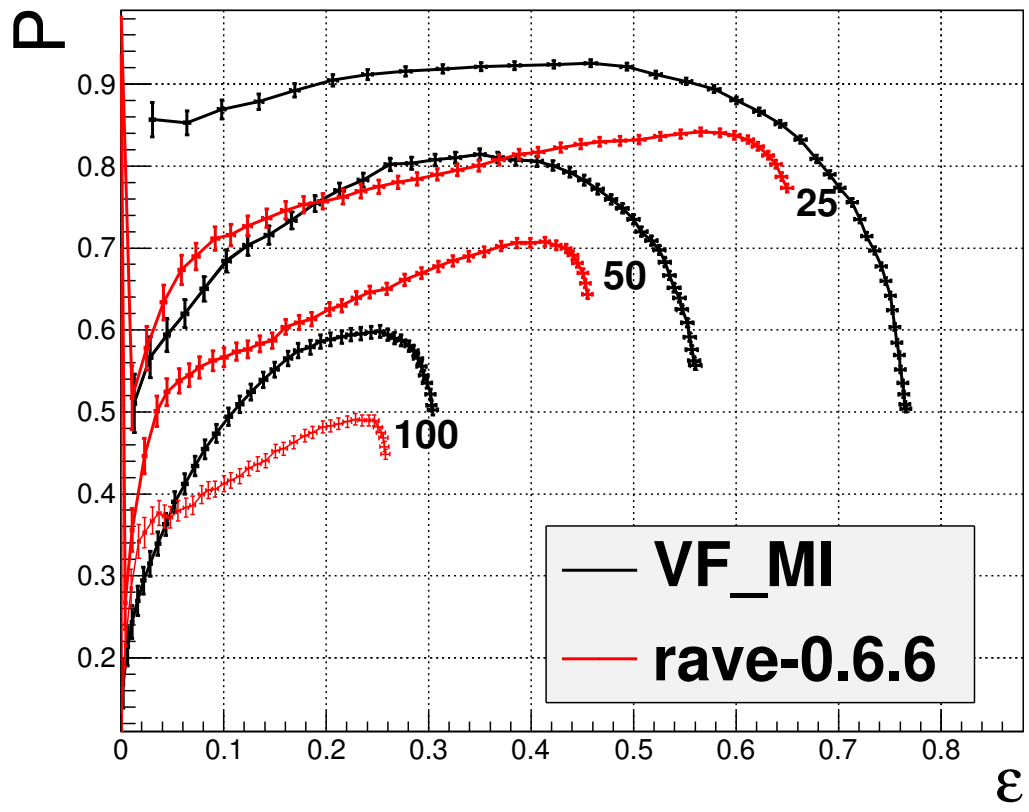
- Evaluation using hard **and** min bias vertices
- Compare multiple operating points (vertex quality cut): Purity vs. Efficiency curves
- $\varepsilon = \frac{N_{\text{correct}}}{N_{\text{truth}}}$ $\mathrm{P} = \frac{N_{\text{correct}}}{N_{\text{reco}}}$
- Vertex quality:

  RAVE $\frac{\chi^2}{\text{n.d.f.}}$

  MI VF Vertex density
- Cannot find all vertices (merging) $\Rightarrow \varepsilon < 1$

- LHC at low luminosity ($<\mu>\approx 5$): RAVE more efficient, MI more pure

Purity and Efficiency



- Evaluation using hard **and** min bias vertices
- Compare multiple operating points (vertex quality cut): Purity vs. Efficiency curves
- $\varepsilon = \frac{N_{\text{correct}}}{N_{\text{truth}}}$ $\mathrm{P} = \frac{N_{\text{correct}}}{N_{\text{reco}}}$
- Vertex quality:

  RAVE $\frac{\chi^2}{\text{n.d.f.}}$

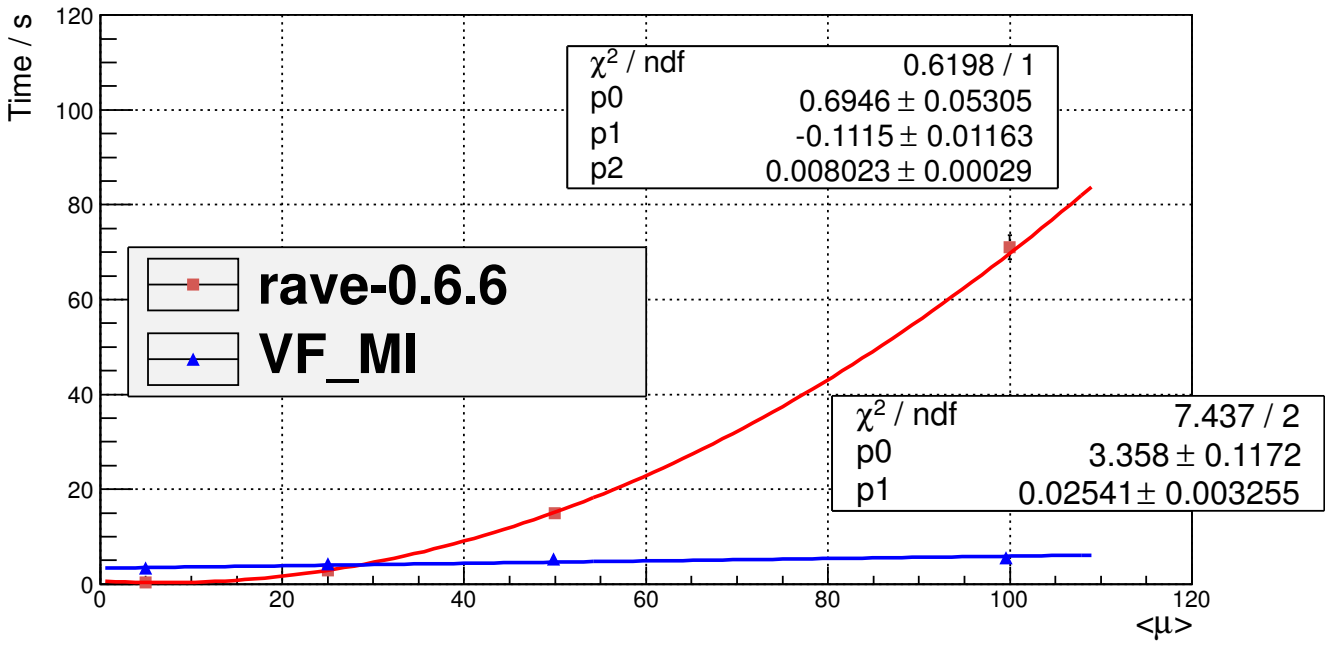
  MI VF Vertex density
- Cannot find all vertices (merging) $\Rightarrow \varepsilon < 1$

- LHC at low luminosity ($<\mu>\approx 5$): RAVE more efficient, MI more pure
- LHC at high luminosity ($<\mu>\approx 25$) and beyond: MI more pure and efficient

CPU Time

- Times for MI

5 Vtx	100 Vtx
Backprojection	
0.01 s	0.11 s
FFTs, filtering, copying	
3 s	3 s
Clustering	
0.36 s	2.4 s



- RAVE: Number of two-track-combinations: $\sum_{i=1}^{n} i - 1 = \frac{n^2 - n}{2}$
- MI: Execution time approximately linear in N_{Vtx}, N_{Track}
- $\Rightarrow$ MI much faster for high numbers of tracks / vertices

Summary

Results

- MI is a promising new ansatz for vertex finding at high luminosities / with many tracks
- Comparison to adaptive vertex fitter (conditions similar to ATLAS/CMS)
  - Higher efficiency and purity at LHC design lumi + beyond
  - Much faster for high numbers of tracks / vertices (linear)
  - No advantage at low numbers of vertices

Outlook:

- Fit found vertices to improve resolution
- Study effects of non-Gaussian track errors
- Applicable to b-tagging / boosted Higgs?

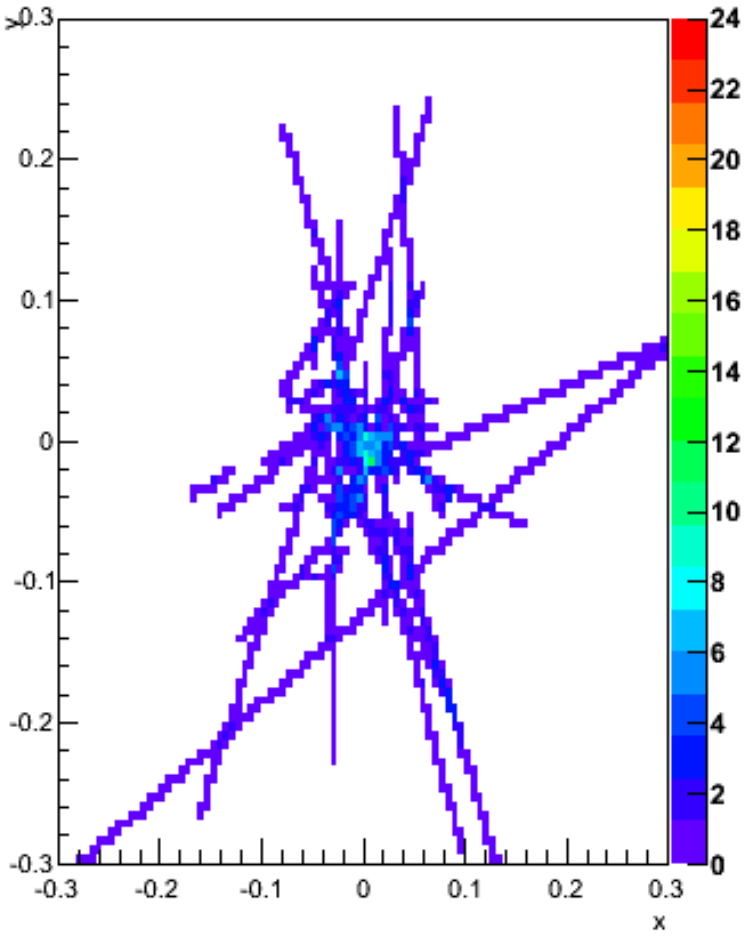
universitätbonn

Appendix

- Filtering 3D
- Track Density 2D
- Benchmarking 3D

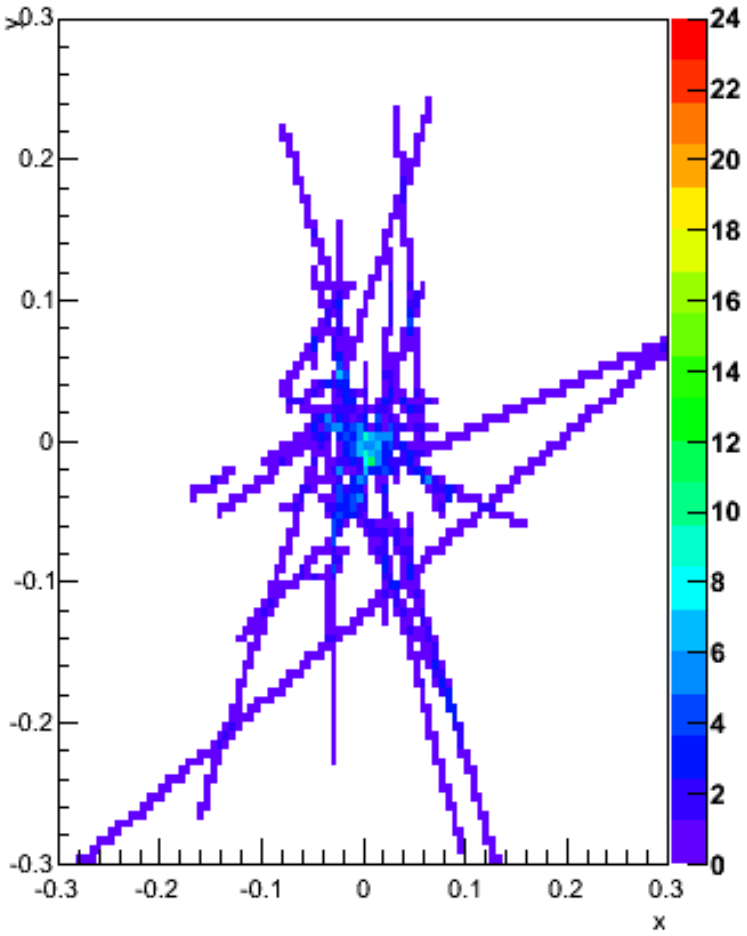
Steps for Vertex Reconstruction Using 3D MI

1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D



Steps for Vertex Reconstruction Using 3D MI

1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D
2. 3D Fourier transform:
   Transform using FFTW*

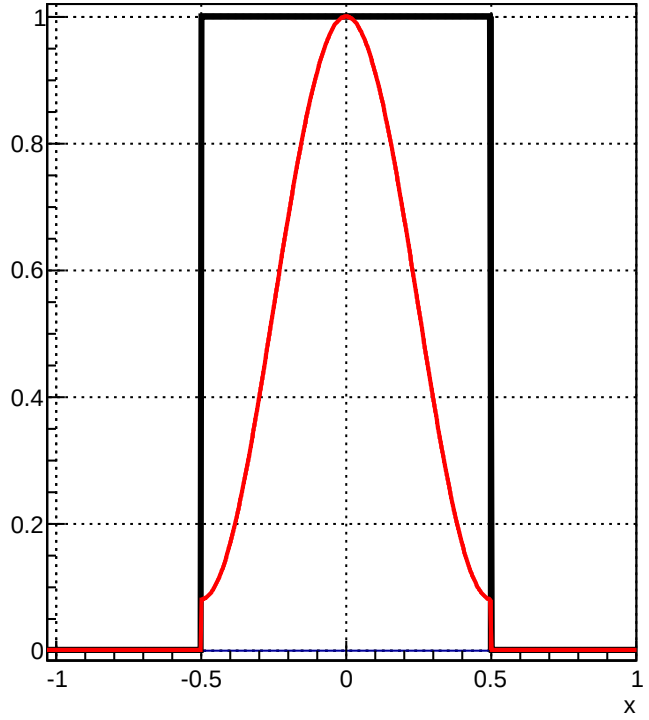


* M Frigo and S G Johnson. "The design and implementation of FFTW3". In: *Proceedings of the IEEE*. 2005, pp. 216–231

Steps for Vertex Reconstruction Using 3D MI

1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D
2. 3D Fourier transform:
   Transform using FFTW*
3. Acceptance corrected 3D k-filter,
   3D window function (smooth image)
   $w_0(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$

Window functions:

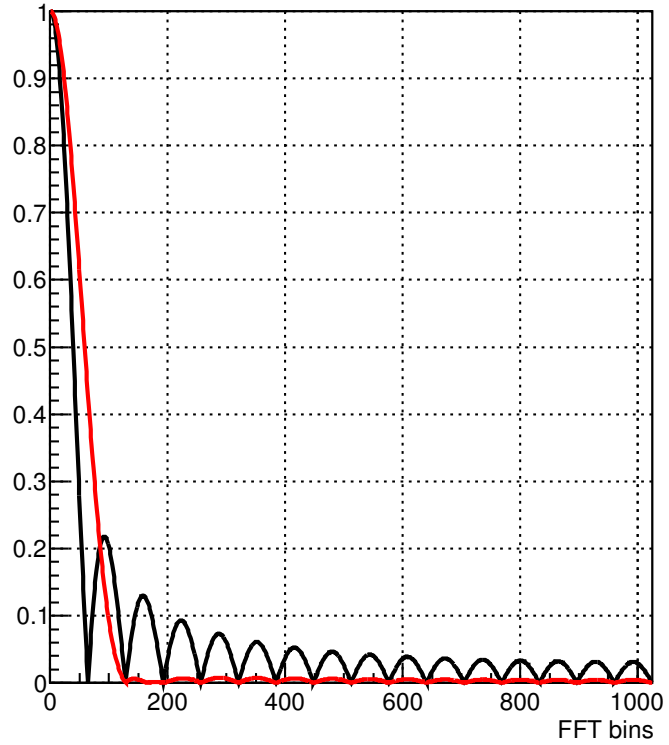


* M Frigo and S G Johnson. "The design and implementation of FFTW3". In: *Proceedings of the IEEE*. 2005, pp. 216–231

Steps for Vertex Reconstruction Using 3D MI

1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D
2. 3D Fourier transform:
   Transform using FFTW*
3. Acceptance corrected 3D k-filter,
   3D window function (smooth image)
   $w_0(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$

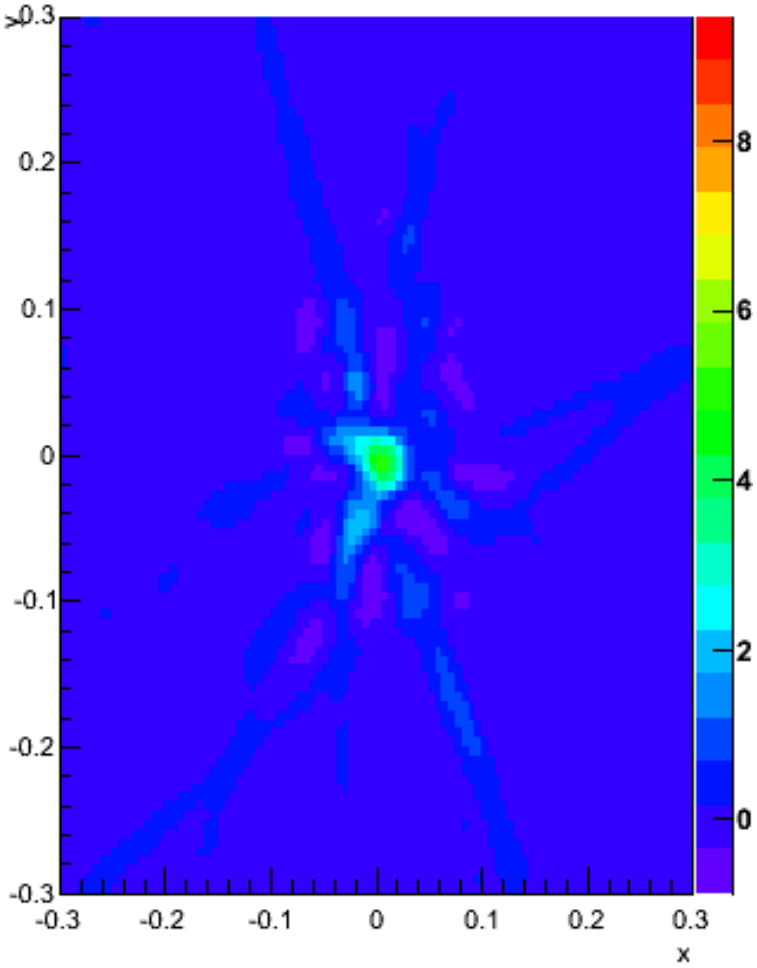
Fourier transforms of window functions:



* M Frigo and S G Johnson. "The design and implementation of FFTW3". In: *Proceedings of the IEEE*. 2005, pp. 216–231

Steps for Vertex Reconstruction Using 3D MI

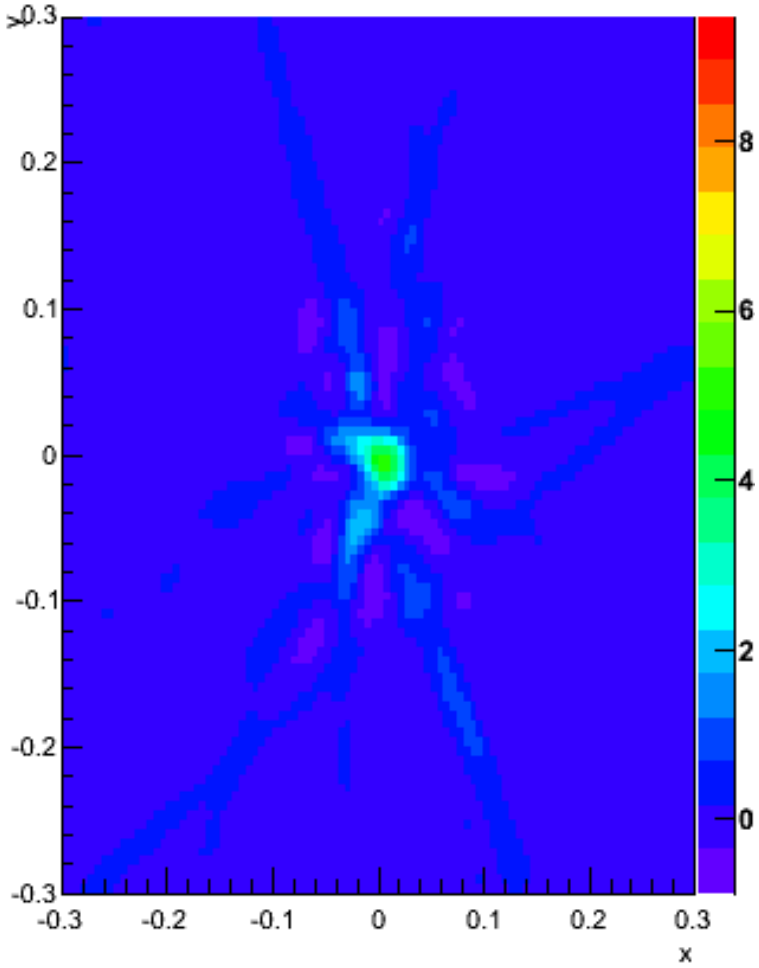
1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D
2. 3D Fourier transform:
   Transform using FFTW*
3. Acceptance corrected 3D k-filter,
   3D window function (smooth image)
   $w_0(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$
4. 3D Fourier transform (backw.)



* M Frigo and S G Johnson. "The design and implementation of FFTW3". In: *Proceedings of the IEEE*. 2005, pp. 216–231

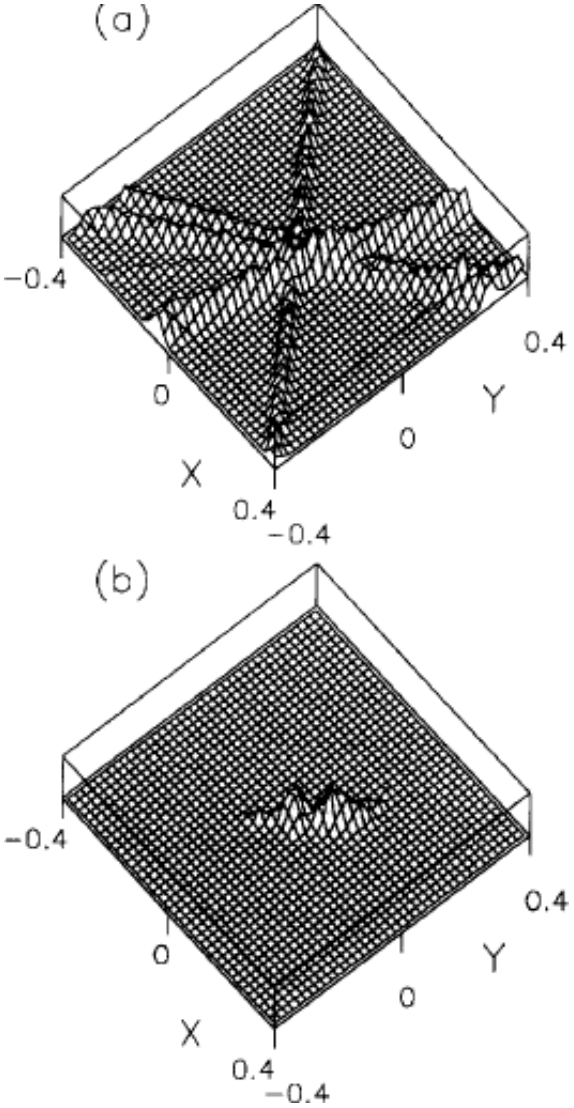
Steps for Vertex Reconstruction Using 3D MI

1. Create unfiltered track density:
   Backproject tracks into ROOT TH3D
2. 3D Fourier transform:
   Transform using FFTW*
3. Acceptance corrected 3D k-filter,
   3D window function (smooth image)
   $w_0(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$
4. 3D Fourier transform (backw.)
5. Find vertices:
   Clustering similar to D Jackson's ZVTop



* M Frigo and S G Johnson. "The design and implementation of FFTW3". In: *Proceedings of the IEEE*. 2005, pp. 216–231

ZVTop



1. Track density: Tracks represented by Gaussian tubes f_i
2. Filter: Remove contribution of single tracks

$$V(\mathbf{r}) = \sum_{i=0}^{N} f_i(\mathbf{r}) - \frac{\sum_{i=0}^{N} f_i^2(\mathbf{r})}{\sum_{i=0}^{N} f_i(\mathbf{r})}$$

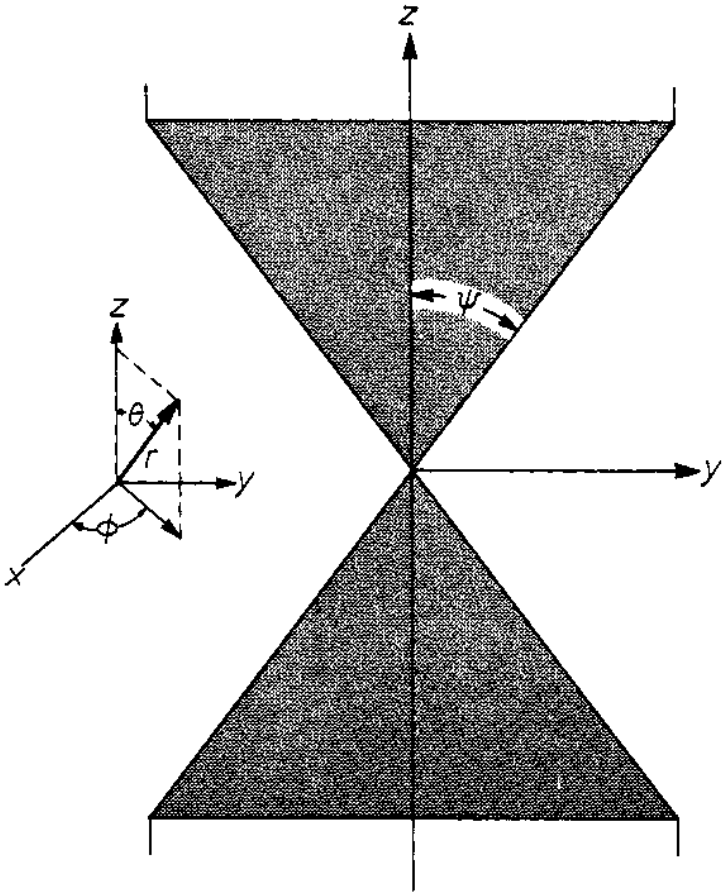
3. Vertex candidates: Cluster filtered vertex density using resolver criterion

$$\frac{\min\{V(\mathbf{r}) : \mathbf{r} \text{ on line connecting } \mathbf{r}_1, \mathbf{r}_2\}}{\min\{V(\mathbf{r}_1), V(\mathbf{r}_2)\}} < R_0$$

D J Jackson. "A Topological Vertex Reconstruction Algorithm for Hadronic Jets". In: *Nucl. Inst. & Meth. A* 388 (1997), pp. 247–253

3D Filtered Backprojection

Problem: 3D detectors usually don't cover the full solid angle.

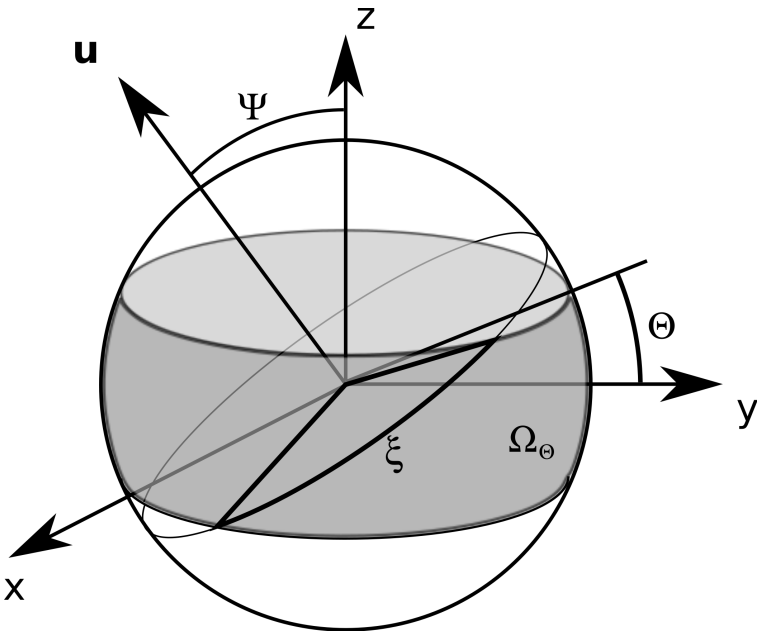


- Filters need to deal with a truncated Point Spread Function (Ψ: acceptance angle):

$$h(r,\theta,\phi) = \frac{1}{r^2}\mathrm{rect}_\Psi\left(\theta - \frac{\pi}{2}\right)$$

3D Filtered Backprojection

Problem: 3D detectors usually don't cover the full solid angle.



Frequency space

- Filters need to deal with a truncated Point Spread Function (Ψ: acceptance angle):

$$h(r,\theta,\phi) = \frac{1}{r^2}\mathrm{rect}_\Psi\left(\theta - \frac{\pi}{2}\right)$$

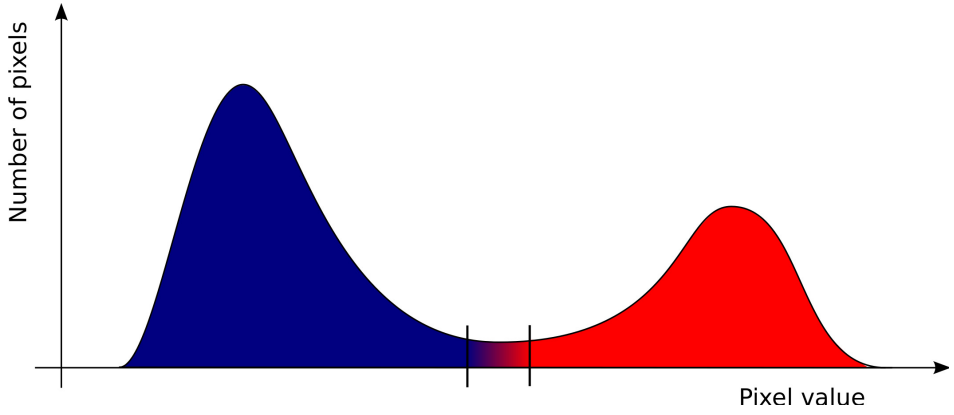
- Analytical filter derived by Colsher:

$$H_\mathrm{C}(\mathbf{u}) = \begin{cases} \frac{u}{2\pi} & |\cos\Psi| \geq \cos\Theta \\ \frac{u}{4\sin^{-1}\left(\frac{\sin\Theta}{|\sin\Psi|}\right)} & |\cos\Psi| < \cos\Theta \end{cases}$$

J G Colsher. "Fully-three-dimensional positron emission tomography". In: *Physics in Medicine and Biology* 25.1 (1980), p. 103

Analyzing the track density: Image Segmentation

Thresholding: The image is divided into foreground and background regions.

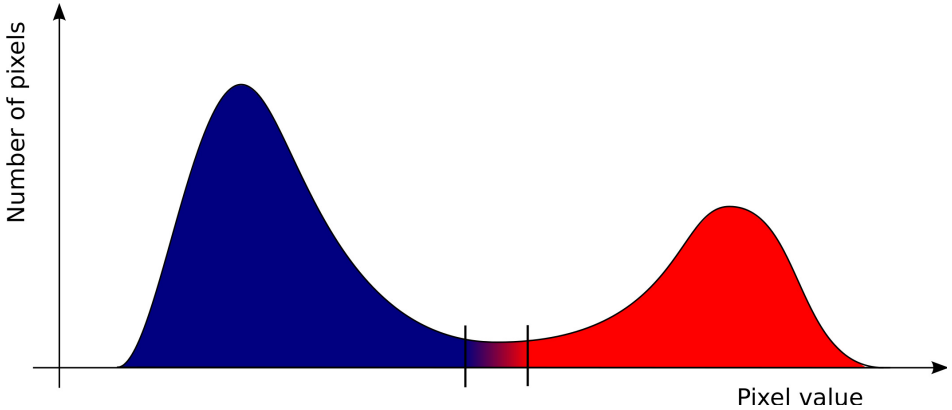


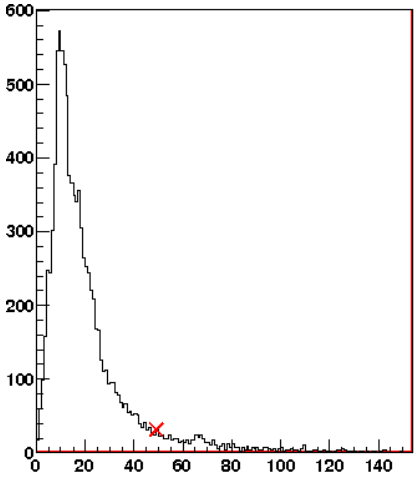
Watershed segmentation: Find the watersheds in an image.

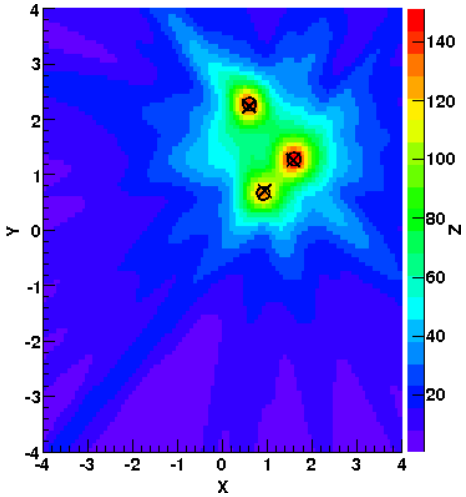
Analyzing the track density: Image Segmentation

Thresholding: The image is divided into foreground and background regions.

Watershed segmentation: Find the watersheds in an image.



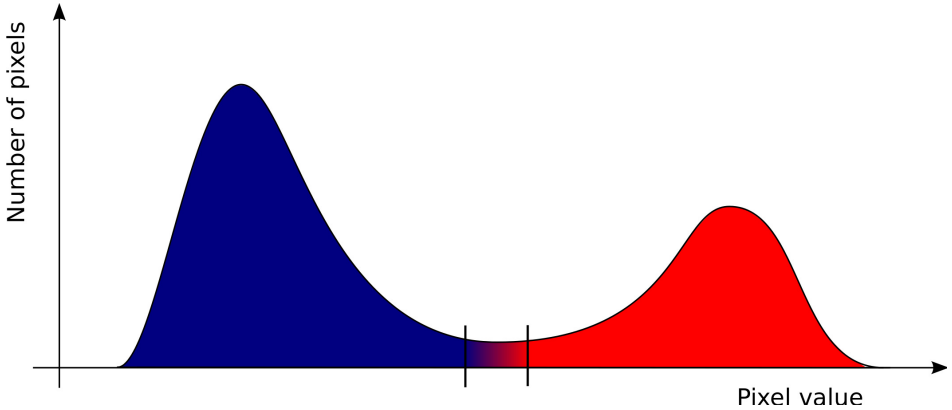


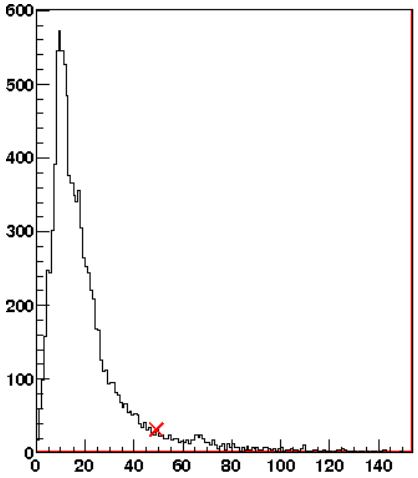


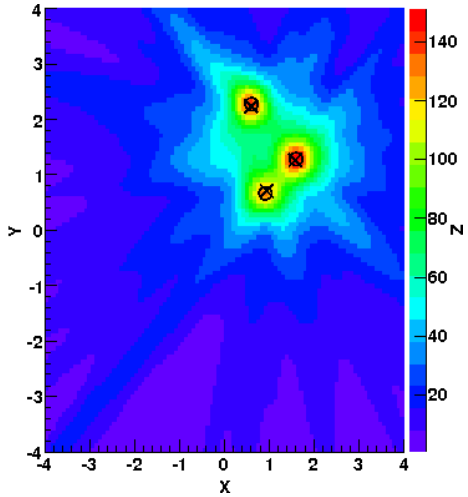
Analyzing the track density: Image Segmentation

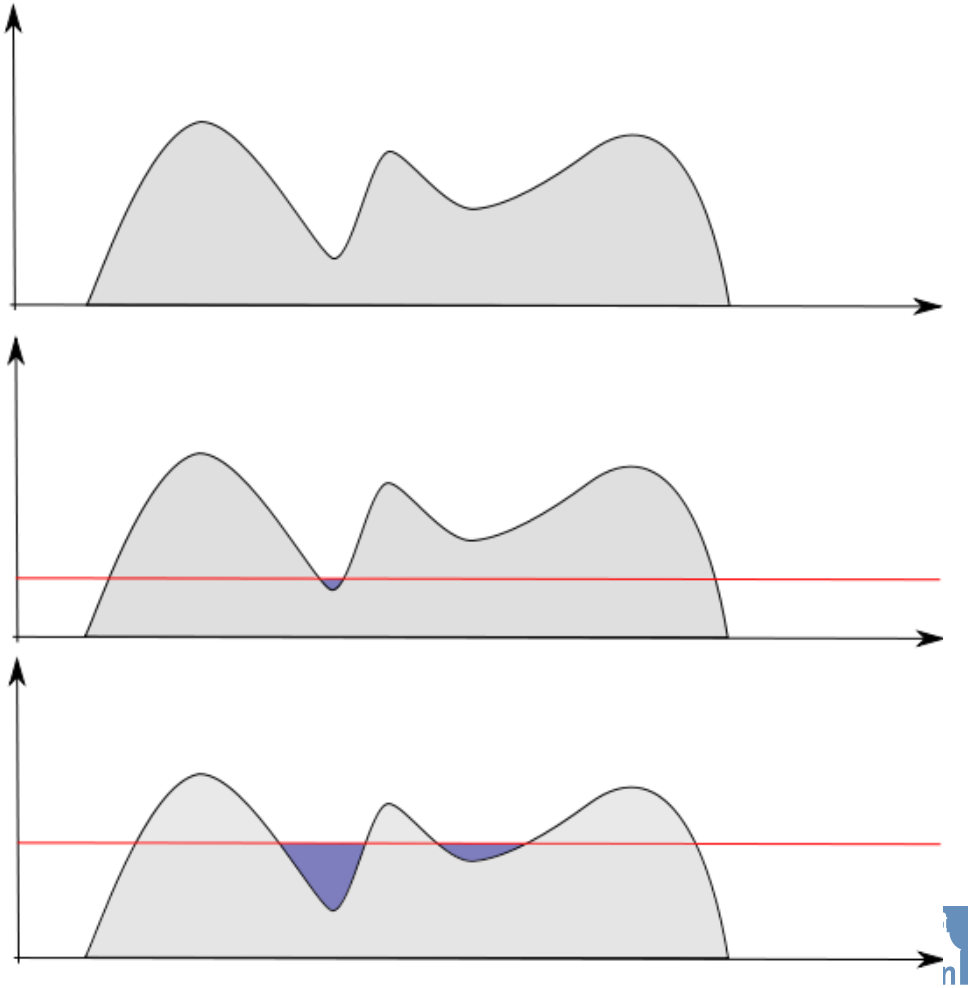
Thresholding: The image is divided into foreground and background regions.

Watershed segmentation: Find the watersheds in an image.









Pythia settings

Hard event:

```
============================================================
I                                       I                  I
I  ISUB  Subprocess name                I  Maximum value   I
I                                       I                  I
============================================================
I                                       I                  I
I   11   f + f' -> f + f' (QCD)         I    9.3780D-03    I
I   12   f + fbar -> f' + fbar'         I    5.1417D-05    I
I   13   f + fbar -> g + g              I    6.4778D-05    I
I   28   f + g -> f + g                 I    8.1586D-02    I
I   53   g + g -> f + fbar              I    1.6266D-03    I
I   68   g + g -> g + g                 I    8.3124D-02    I
I   96   Semihard QCD 2 -> 2            I    1.8899D+04    I
============================================================
```

Min bias:

```
============================================================
I                                       I                  I
I  ISUB  Subprocess name                I  Maximum value   I
I                                       I                  I
============================================================
I                                       I                  I
I   91   Elastic scattering             I    2.2209D+01    I
I   92   Single diffractive (XB)        I    7.1513D+00    I
I   93   Single diffractive (AX)        I    7.1513D+00    I
I   94   Double  diffractive            I    1.0287D+01    I
I   95   Low-pT scattering              I    5.4714D+01    I
I   96   Semihard QCD 2 -> 2            I    1.8899D+04    I
============================================================
```