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 $H \rightarrow \gamma\gamma$: “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

<table>
<thead>
<tr>
<th></th>
<th>HEP</th>
<th>Positron Emission Tomography</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>Position of reaction/interaction</td>
<td>Position of tumor</td>
</tr>
<tr>
<td>Data</td>
<td>Tracks</td>
<td>Photon pairs from $e^+e^- \rightarrow \gamma\gamma$</td>
</tr>
<tr>
<td>Amount of data</td>
<td>$&lt; N_{Trk}/Vtx &gt; \approx 20$</td>
<td>$&gt; 10^6$, <strong>more is better</strong></td>
</tr>
<tr>
<td>Methods</td>
<td>Adaptive fitting / <strong>Topological finding</strong></td>
<td>Filtered Backprojection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\rightarrow \text{inv. Radon-transformation}$</td>
</tr>
</tbody>
</table>
Topological Vertex Reconstruction: ZVTop


- 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


- 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

\[ p_\alpha(r) = \int_L f(x, y) \, dl \]
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

\[ f(x, y)_{BP} = (f_{true} \ast s)(x, y) \]
\[ = \int\int f(x - a, y - b)_{true} \cdot s(a, b) \, da \, db \]
\[ s = \frac{1}{r} \]

a, b: Space coordinates used for integration
Filtered Backprojection: Counteract the Blurring

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

\[ f(x, y)_{\text{BP}} = (f_{\text{true}} * s)(x, y) \]
\[ = \int \int f(x - a, y - b)_{\text{true}} \cdot s(a, b) \, da \, db \]

\[ s = \frac{1}{r} \]

a, b: Space coordinates used for integration
Filtered Backprojection: Counteract the Blurring

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

\[ f(x, y)_{BP} = (f_{true} \ast s)(x, y) \]
\[ = \int \int f(x - a, y - b)_{true} \cdot s(a, b) \, da \, db \]
\[ s = \frac{1}{r} \]

- Deconvolution → Fourier space:

\[ F(u, v)_{BP} = F_{true}(u, v) \cdot S(u, v) \]
\[ S(u, v) = \frac{1}{k} \]
\[ k = \left| \begin{pmatrix} u \\ v \end{pmatrix} \right| \]
Filtered Backprojection: Counteract the Blurring

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

\[
 f(x, y)_{\text{BP}} = \left( f_{\text{true}} * s \right) (x, y) \\
 = \int \int f(x - a, y - b)_{\text{true}} \cdot s(a, b) \, da \, db \\
 s = \frac{1}{r}
\]
a, b: Space coordinates used for integration

- Deconvolution \(\rightarrow\) Fourier space:

\[
 F(u, v)_{\text{BP}} = F_{\text{true}}(u, v) \cdot S(u, v) \\
 S(u, v) = \frac{1}{k} \quad k = \left| \begin{pmatrix} u \\ v \end{pmatrix} \right|
\]

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

\[
 F_{\text{true}}(u, v) = F(u, v)_{\text{BP}} \cdot k
\]
Example with 2D-only tracks

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

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

3D PET:

- Acceptance corrected Ramp-Filter ($\eta = 2.5 \rightarrow 80^\circ$)
- 3D window function
- Math more complex: 3D FFTs
- All measurements usable
Medical Imaging in Three Dimensions

Stack transverse planes:

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

Vertex Density for Simulated LHC Events

Left: Simple backprojection
Right: 3D-filtered backprojection

“Flight” through typical 2011 LHC beamspot

\[ \sigma_{xy} = 32 \ \mu m \quad \sigma_z = 4.2 \text{ cm} \]
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
- Provides modern adaptive algorithms in use at LHC: AVR (Adaptive Vertex Reconstructor)

- Adaptive: Robustification of conventional $\chi^2$-fits (“Kalman”)
Comparison to vertex finders used in HEP: LHC

RAVE 0.6.6

- Open source vertexing toolkit
- Provides modern adaptive algorithms in use at LHC: AVR (Adaptive Vertex Reconstructor)

- **Adaptive**: Robustification of conventional $\chi^2$-fits (“Kalman”)
- **How**: Introduce weights for each track $i$:

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

- Automatic track association
  - 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

<table>
<thead>
<tr>
<th>Collision:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min $N_{\text{trk}}$</td>
<td>4</td>
</tr>
<tr>
<td>Min $p_t$</td>
<td>50 GeV</td>
</tr>
<tr>
<td>$E_{\text{CM}}$</td>
<td>7 TeV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beamspot:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{xy,BS}$</td>
<td>34 $\mu$m</td>
</tr>
<tr>
<td>$\sigma_{z,BS}$</td>
<td>4.6 cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tracks:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min $p_t$</td>
<td>250 MeV</td>
</tr>
<tr>
<td>Max $\eta$</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Uncertainties ($|\eta| \approx 0.5$):
- $\sigma(\phi_0)$: 70 $\mu$rad
- $\sigma(\cot \theta)$: $0.7 \times 10^{-3}$
- $\sigma(d_0)$: 10 $\mu$m
- $\sigma(z_0 \times \sin \theta)$: 91 $\mu$m

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

<table>
<thead>
<tr>
<th>Collision:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min $N_{\text{trk}}$</td>
<td>4</td>
</tr>
<tr>
<td>Min $p_t$</td>
<td>50 GeV</td>
</tr>
<tr>
<td>$E_{\text{CM}}$</td>
<td>7 TeV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beamspot:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{xy,BS}$</td>
<td>34 µm</td>
</tr>
<tr>
<td>$\sigma_{z,BS}$</td>
<td>4.6 cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tracks:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min $p_t$</td>
<td>250 MeV</td>
</tr>
<tr>
<td>Max $\eta$</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Uncertainties ($|\eta| \approx 0.5$):
- $\sigma(\phi_0)$: 70 µrad
- $\sigma(\cot \theta)$: $0.7 \times 10^{-3}$
- $\sigma(d_0)$: 10 µm
- $\sigma(z_0 \times \sin \theta)$: 91 µm

- **LHC-Collisions** Pythia 6.416, default settings. 1 hard vtx, 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
  - $\geq 50\%$ of tracks from one truth vertex

$\Rightarrow$ **Vertex accepted**
Purity and Efficiency

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

Vertex quality:
RAVE \frac{\chi^2}{n.d.f.}
MI VF Vertex density

Cannot find all vertices (merging) \Rightarrow \varepsilon < 1

LHC at low luminosity (\langle \mu \rangle \approx 5): RAVE more efficient, MI more pure

\begin{align*}
\varepsilon &= \frac{N_{\text{correct}}}{N_{\text{truth}}} \\
P &= \frac{N_{\text{correct}}}{N_{\text{reco}}} \\
\end{align*}
Purity and Efficiency

- Evaluation using hard **and** min bias vertices
- Compare multiple operating points (vertex quality cut): Purity vs. Efficiency curves
  \[ \varepsilon = \frac{N_{\text{correct}}}{N_{\text{truth}}} \quad P = \frac{N_{\text{correct}}}{N_{\text{reco}}} \]
- Vertex quality:
  - RAVE \[ \chi^2 \text{ n.d.f.} \]
  - MI VF Vertex density
- Cannot find all vertices (merging) \( \Rightarrow \varepsilon < 1 \)

- LHC at **low** luminosity \( \langle \mu \rangle \approx 5 \): RAVE more efficient, MI more pure
- LHC at **high** luminosity \( \langle \mu \rangle \approx 25 \) and beyond: MI more pure and efficient
CPU Time

- **Times for MI**
  - 5 Vtx: 0.01 s
  - 100 Vtx: 0.11 s
  - Backprojection: 3 s
  - FFTs, filtering, copying: 3 s
  - Clustering: 0.36 s

- **RAVE**: Number of two-track-combinations: \( \sum_{i=1}^{n} i - 1 = \frac{n^2 - n}{2} \)
- **MI**: Execution time approximately linear in \( N_{\text{Vtx}}, N_{\text{Track}} \)
- \( \Rightarrow \) 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?
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

2. 3D Fourier transform:
   Transform using FFTW

3. Acceptance corrected 3D k-filter, 3D window function (smooth image)

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”.
Steps for Vertex Reconstruction Using 3D MI

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

2. 3D Fourier transform:
   Transform using FFTW*

---

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:

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) 
   \( w_0(n) = 0.54 + 0.46 \cos \left( \frac{2\pi n}{N-1} \right) \)

* M Frigo and S G Johnson. “The design and implementation of FFTW3”. 
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.)

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”.
Track density: Tracks represented by Gaussian tubes $f_i$

Filter: Remove contribution of single tracks

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

Vertex candidates: Cluster filtered vertex density using resolver criterion

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

$$\min\{V(\mathbf{r}_1), V(\mathbf{r}_2)\}$$

3D Filtered Backprojection

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

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

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

3D Filtered Backprojection

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

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

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

- Analytical filter derived by Colsher:

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

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:

<table>
<thead>
<tr>
<th>ISUB</th>
<th>Subprocess name</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>$f + f' \rightarrow f + f'$ (QCD)</td>
<td>9.3780D-03</td>
</tr>
<tr>
<td>12</td>
<td>$f + fbar \rightarrow f' + fbar'$</td>
<td>5.1417D-05</td>
</tr>
<tr>
<td>13</td>
<td>$f + fbar \rightarrow g + g$</td>
<td>6.4778D-05</td>
</tr>
<tr>
<td>28</td>
<td>$f + g \rightarrow f + g$</td>
<td>8.1586D-02</td>
</tr>
<tr>
<td>53</td>
<td>$g + g \rightarrow f + fbar$</td>
<td>1.6266D-03</td>
</tr>
<tr>
<td>68</td>
<td>$g + g \rightarrow g + g$</td>
<td>8.3124D-02</td>
</tr>
<tr>
<td>96</td>
<td>Semihard QCD 2 -&gt; 2</td>
<td>1.8899D+04</td>
</tr>
</tbody>
</table>

#### Min bias:

<table>
<thead>
<tr>
<th>ISUB</th>
<th>Subprocess name</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>Elastic scattering</td>
<td>2.2209D+01</td>
</tr>
<tr>
<td>92</td>
<td>Single diffractive (XB)</td>
<td>7.1513D+00</td>
</tr>
<tr>
<td>93</td>
<td>Single diffractive (AX)</td>
<td>7.1513D+00</td>
</tr>
<tr>
<td>94</td>
<td>Double diffractive</td>
<td>1.0287D+01</td>
</tr>
<tr>
<td>95</td>
<td>Low-pT scattering</td>
<td>5.4714D+01</td>
</tr>
<tr>
<td>96</td>
<td>Semihard QCD 2 -&gt; 2</td>
<td>1.8899D+04</td>
</tr>
</tbody>
</table>