Medical Imaging Inspired Vertex Reconstruction at LHC

S Hageböck and E von Toerne
University of Bonn, Physikalisches Institut, Nussallee 12, 53115 Bonn, Germany
E-mail: hageboeck@physik.uni-bonn.de, evt@physik.uni-bonn.de

Abstract. Three-dimensional image reconstruction in medical applications (PET or X-ray CT) utilizes sophisticated filter algorithms to linear trajectories of coincident photon pairs or x-rays. The goal is to reconstruct an image of an emitter density distribution. In a similar manner, tracks in particle physics originate from vertices that need to be distinguished from background track combinations. In this study it is investigated if vertex reconstruction in high energy proton collisions may benefit from medical imaging methods. A new method of vertex finding, the Medical Imaging Vertexer (MIV), is presented based on a three-dimensional filtered backprojection algorithm. It is compared to the open-source RAVE vertexing package. The performance of the vertex finding algorithms is evaluated as a function of instantaneous luminosity using simulated LHC collisions. Tracks in these collisions are described by a simplified detector model which is inspired by the tracking performance of the LHC experiments. At high luminosities (25 pileup vertices and more), the medical imaging approach finds vertices with a higher efficiency and purity than the RAVE “Adaptive Vertex Reconstructor” algorithm. It is also much faster if more than 25 vertices are to be reconstructed because the amount of CPU time rises linearly with the number of tracks whereas it rises quadratically for the adaptive vertex fitter AVR.

1. Introduction
The coming challenges in vertex reconstruction are defined by high-luminosity hadron colliders, like the Large Hadron Collider at CERN or the planned FAIR facility at GSI Darmstadt [1]. Due to the high luminosity at the LHC, each bunch crossing yields several primary vertices from proton-proton collisions. These vertices are mostly caused by soft QCD reactions. The requirement of a trigger constrains the possible event topologies such that each recorded event contains one hard interaction vertex and several1 soft, so-called minimum bias vertices. The reconstruction of all vertices in an event is an important reconstruction task for the LHC experiments. For example there are methods to measure the instantaneous luminosity by counting vertices. Another example is the decay of the Higgs boson to two photons. Here the hard vertex is similar to a minimum bias vertex because the hard event produces no charged tracks. The extrapolation of the photons to the correct vertex is crucial for precise kinematical reconstruction and optimal di-photon mass resolution.

The number of vertices per bunch crossing is expected to increase dramatically in the coming years due to further luminosity upgrades [2]. It is important to note that many vertex detection

1 About twenty proton-proton collisions per bunch crossing at LHC design luminosity.
algorithms are polynomial in processing time per event vs. number of tracks. It is therefore prudent to develop algorithms that are fast, preferably linear with the number of tracks.

On the other hand, medical image reconstruction is routinely performed with millions of individual elements that are quite similar to particle physics tracks. The main reason why large statistics data do not deter the performance of the medical image algorithms is the reduction of data size by pixelization. The event sample is histogrammed and the algorithm is applied to each histogram bin. The processing time thus depends on the number of bins rather than the data size. As will be shown later, our algorithm, the Medical Imaging Vertexer (MIV), largely benefits from the idea of histogrammed data processing.

1.1. Analogies between vertex reconstruction and medical imaging
In this paper we present an algorithm for primary vertex reconstruction that is based on modern methods of medical imaging. Image reconstruction in Positron Emission Tomography (PET) and vertex finding are similar tasks. The main challenge in medical imaging methods such as PET is the reconstruction of a source distribution based on lines that emanate from the sources. One reconstructs a positron emitter density distribution (source distribution) from non-invasive measurements of coincident photon pairs. When a positron annihilates with an electron inside the patient’s body, two photons are emitted back to back. From a single photon pair the following information can be extracted: The positron source is located close to the line connecting the photon detectors that have been hit. The “depth” is unknown.\(^2\)

For vertex finding in high energy physics (HEP) a similar problem has to be solved: a HEP particle detector measures tracks of charged particles non-invasively and close to the interaction region. The goal is to find vertices for sets of tracks inside the interaction region. “Non-invasive” means that there are no tracking devices inside the region of interest. In analogy to PET, a single track carries the following information: the vertex must be close to the track but the vertex position in direction of the track is unknown. In vicinity of the beam spot the tracks may be approximated by straight lines.

To summarize, the two problems are quite similar: reconstruction of medical images using a filtered inverse Radon transformation and vertex detection in high energy physics. The main analogies are

- charged tracks in high energy physics → photon-pair in PET
- primary vertices from a hadron collision → point sources in the source distribution

One approach in medical imaging is the inverse Radon transformation (for a nice review see [3]) which is employed to reconstruct the source distribution from the photon pairs. Most applications of the Radon transformation in medical imaging divide the volume under study into two-dimensional layers and the Radon transformation itself is only two-dimensional. The three-dimensional source distribution is obtained by overlaying the 2D distributions. However there are also genuinely three-dimensional Radon transformation algorithms available that we are going to use in the following.

We should note here that there is a HEP vertexing algorithm similar to the inverse Radon transformation. It is the ZVTOP algorithm developed by D. Jackson [4]. The ZVTOP approach was successfully employed at \(e^+e^-\) colliders in the search for secondary vertices from heavy flavor decay and reliably works for processes with small track numbers [5]. In this algorithm a vertex density is created from the tracks. Each track pair is a possible start point to search for local maxima in the vertex density. The algorithm’s processing time is thus quadratic in the track number. This may be one of the reasons why the success of the algorithm has not yet fully

\(^2\) A similar situation occurs in Computer Tomography (CT) where x-rays are attenuated by the traversed tissue. The x-ray intensity is a measure for the source density on the (linear) x-ray trajectory.
migrated to hadron collider experiments. In addition the ZVTop algorithm employs an effective filtering step that works best for low track densities \(^3\).

We employ an algorithm that differs with respect to ZVTOP in two important aspects: first, the processing is done on a pixelized representation of the track density. The algorithm is thus not quadratic in the track density. Second, we apply a medical imaging filter to the raw vertex density that differs from the effective filtering step in ZVTOP.

2. The Medical Imaging Vertexer algorithm (MIV)

2.1. Vertex detection using a three-dimensional filtered backprojection of track data

One of the main problems in medical imaging is the reconstruction of a source density distribution knowing only lines close to the sources. The source in the case of PET is e.g. a tumor marked by a high density of positron emitters. In PET this problem is frequently solved using “filtered backprojection”. Backprojecting means projecting all measurements (= photon pairs) onto a grid (2D) or into a volume (3D). The regions where the single measurements overlap can be considered regions of interest. In PET this may be the tumor while in high energy physics this may be a vertex – the origin of several tracks. A simple implementation of backprojection produces a blurred version of the original image. This well-known problem in medical imaging can be counteracted or removed completely by applying a filter to the backprojected image \([6, 3]\).

Mathematically speaking, the simple backprojection creates the true image convolved with the Point Spread Function (PSF). The de-convolution step is simplified by a transformation to Fourier space. Here, the filter reduces to a simple multiplication of the Fourier transformed density with the absolute value of the \(k\) frequency vector. This filter is the inverse of the Fourier-transformed PSF and is commonly called “Ramp-filter” or “k-filter”. As it extends to arbitrarily high frequencies it incurs high-frequency distortions that may be avoided by modifying the filter at large frequencies. An estimate of the true image is obtained by backtransforming the filtered image to Euclidean space.

The medical imaging process may be divided into the following steps:

(i) Creation of a three-dimensional source density.

(ii) Filtering the vertex density to remove artifacts and reduce blurring.

(iii) Extraction of sources that match certain templates.

Figure 1 shows the backprojected track data of a two dimensional example. In Figure 1a the unfiltered image is displayed while the Ramp-filter has been used for figure 1b. The usage of the Ramp-filter creates a sharper but also noisier image whereas the unfiltered image shows blurred vertices tending to merge. The noise in the filtered image can be reduced if the Ramp-filter is attenuated by multiplying it with a window function. The resulting image is cleaner but also less sharp \([3]\). For vertex finding, sharpness must be traded off against smoothness. This will be discussed later in this section.

With conditions found at LHC, a three dimensional filtered backprojection needs to be implemented for primary vertex finding. Given a set of tracks the steps of vertex finding are:

(i) Backproject all tracks into a volume to be searched for vertices. In the current implementation this volume is represented by a three dimensional ROOT histogram \([7]\).

For each track, all bins along the “line of response” are incremented by the value 1. In

\[
V(v) = \sum_{i=0}^{N} f_i(v) - \sum_{i=0}^{N} f_i^2(v) / \sum_{i=0}^{N} f_i(v)
\]

We regard this as an effective filtering step that works perfectly for a single track but diminishes as several tracks overlap.
PET, the line of response is the line connecting two photon detectors hit by coincident photons. For primary vertex finding, the analogon for the line of response is a tangent to the track of a charged particle at the point of closest approach (PCA).

After all tracks have been backprojected, the histogram contains a superposition of all tracks suffering from the well-known blurring created by simple backprojection.

(ii) In order to filter the track density it needs to be transformed into Fourier space. Therefore, the contents of the histogram are copied to a three dimensional array that is transformed using a three dimensional discrete Fourier transformation (DFT) provided by the software package FFTW [8].

(iii) In analogy to the two dimensional example in Fig. 1b, the blurring is counteracted by a three dimensional filter. It has to be corrected for detectors with limited acceptance and was derived by Colsher [9] and Defrise [10] for PET applications:

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

$u$ is a point in Fourier space, $\Theta$ is the opening angle of a cylindrical detector (for the tracking detector of ATLAS this is roughly 80°, $\eta = 2.5$) [11]. $\Psi$ is the angle of $u$ with respect to the frequency axis corresponding to the $z$-axis in Euclidean space. The first case in equation (1) is the exact analogon of the Ramp-filter in two dimensions. But this filter also takes into account that a cylindrical detector does not cover the full solid angle and therefore features a special case for frequencies in certain directions which are affected by the limited acceptance.

For each index in the array of frequencies, Colsher’s and Defrise’s filter function (1) is evaluated and the resulting value is multiplied with the frequency at the respective index. Additionally, a three dimensional Blackman-Harris window function is multiplied with all frequencies to smooth the filtered track density [9]. This attenuates high (spatial) frequencies and thus reduces fine and small features like those in the background region.

---

In Defrise’s publication this condition reads $\cos \Psi > \cos \Theta$. Defrise defines $\cos \Psi = |\nu_z|/|\nu|$ with $\nu$ being a vector in frequency space. With the convention $\cos \Psi = \nu_z/|\nu|$ the absolute value has to be used.
of figure 1b. As can be seen there, such a window function is necessary because the filter strongly amplifies high frequencies. The use of a window function with a smooth roll-off at high frequencies suppresses the noise artifacts at the cost of not being able to suppress the blurring completely. The cutoff frequency of this window function must be adjusted in a way that the noise is tolerable and the resolution sufficient.

The typical shape of the interaction region at LHC can be used to reduce noise while retaining much of the sharpness gained from filtering: in $xy$-direction the interaction region is very small, even smaller than the typical track resolution of the LHC experiments (for LHC beamspot parameters: [12], for typical track resolution: [11]). Therefore, it is not likely that vertices can be separated in the $xy$-plane. The cutoff-frequency of the window function can thus be chosen low so that high frequencies are strongly attenuated. This produces a blurred but smooth filtered vertex density when looking in $xy$-direction. In $z$, high frequencies are only slightly attenuated to improve the separation of the vertices in $z$. This specially filtered track density optimized for primary vertex finding is displayed in figure 2.

(iv) The backtransformation to Euclidean space is done using FFTW, the vertex density is extracted from the array and written into a histogram.

(v) The filtered track density is analyzed for the presence of local maxima: A cluster is started at the bin with the highest bin content. If the second highest bin is a neighbor to the previous bin, it is added to the cluster. Otherwise a new cluster is started at the second highest bin. This procedure is repeated with the other bins until the contents of all bins are below an adjustable threshold. Afterwards, clusters are split or merged using a resolution criterion similar to the one used in D. Jackson’s ZVTop algorithm [4]. Clusters are merged if the following inequality is fulfilled:

$$\frac{T_{ij}}{\min\{S_i, S_j\}} > R$$

(2)

$T_{ij}$ denotes the filtered track density at the point where the clusters $i$ and $j$ touch, $S_i$ is the track density at the location where cluster $i$ was started.

(vi) The vertex candidates are obtained from the clusters in the following way: For each cluster the center of gravity is calculated and taken as an estimate for the vertex position. The vertex candidate is accepted if the vertex density at the starting point of the cluster exceeds an adjustable threshold. This threshold can be viewed as a kind of vertex quality cut because vertices with many clearly associated tracks will have a high filtered track density.

3. Performance

3.1. The adaptive vertex reconstructor as reference for comparison

Our vertex finder MIV is compared to the adaptive vertex reconstructor (AVR), a standard vertex reconstruction algorithm currently in use at LHC and available in the open source software package RAVE [13]. RAVE is a detector independent vertex reconstruction toolkit for usage in versatile environments.

3.2. Results

Figure 2 shows the results of the MIV and AVR algorithms on a typical LHC event with one hard and 30 minimum bias events. Shown are the tracks in the $xz$-projection in addition to the vertex position reconstructed by MIV and AVR. Both algorithms are able to reconstruct primary vertices. However, AVR finds vertices outside of the interaction region due to random track combinations although a beamspot constraint is set.

5 The coordinate system is chosen such that the direction of the beams are along the $z$ axis.
Figure 2: $xz$-section through a vertex density created using the medical imaging approach with simulated tracks resembling typical ATLAS resolution. The markers show the position of truth vertices as well as vertices found by RAVE AVR and the MIV vertex finder.

3.3. Efficiency and Purity

For both RAVE AVR and the MIV vertex finder, efficiency and purity are determined using typical LHC collisions simulated with Pythia 6.416 [14]. The efficiency of vertex reconstruction is

$$\varepsilon = \frac{N_{\text{correct reco}}}{N_{\text{truth}}}$$

while purity is

$$P = 1 - \frac{N_{\text{false reco}}}{N_{\text{total reco}}}.$$  

For each vertex finder a vertex quality cut is varied to compare multiple operating points at the same time: For RAVE the operating point is chosen in the $\chi^2_{\text{n.d.f.}}$ distribution where “n.d.f.” stands for number of degrees of freedom. Vertices with bad quality like background events are assumed to coincide with high values of $\chi^2_{\text{n.d.f.}}$. For the MIV algorithm the maximum of the filtered vertex density is taken as a measure for the quality of the vertex as described in section 2. Regions with a low filtered track density are not likely to contain a true vertex.

Purity and efficiency are plotted against each other in a curve inspired by the “ROC curves” used in signal detection theory. ROC curves show the performance of binary classifiers as the signal threshold is varied. The purity-efficiency curves in this article however show the performance of a vertex finding algorithm. A binary classifier either accepts or rejects an object presented for classification whereas a vertex finder searches for vertices. Because of this difference, purity needs to be calculated in a different way. For a binary classifier, purity can be calculated using the total number of background objects presented for classification and
the number of objects that the classifier failed to reject:

\[ P = 1 - \frac{N_{\text{false positive}}}{N_{\text{total}}} \]  \hspace{1cm} (5)

In the case of a vertex finder, purity can only be calculated using the total number of *found* vertices like in equation (4). Due to this difference the shape of the curves changes: ROC curves are always monotone, the purity-efficiency curves used here are not because \( N_{\text{total reco}} \) in equation (4) changes while \( N_{\text{total}} \) in (5) is fixed when the signal threshold is varied.

![Figure 3: Purity and Efficiency for different vertex qualities and LHC collisions at high luminosity. A definition of the vertex quality is given in the text. Each event contains one hard vertex. The number of minimum bias vertices is Poisson-distributed. Labels indicate the mean of this distribution. Figure 3 shows such a curve for both algorithms for LHC collisions at the design luminosity \( 10^{34} \text{ cm}^{-2} \text{ s}^{-1} \) (mean of 25 interactions per bunch crossing, \( \sigma_{\text{tot}} = 80 \text{ mb} \)) [12] and for luminosities beyond this value. The figure shows that the operating point for the MIV Vertex Finder can be chosen such that it reaches both higher efficiency and higher purity than the adaptive vertex fitter. Figure 4 shows the performance for LHC at low luminosity (mean of 5 interactions per bunch crossing). Comparing with RAVE, it is apparent that the medical imaging approach works best with high numbers of tracks and vertices. This is a property of the filtered backprojection: with more data clearer images can be produced. For this reason it is easier to separate the vertex density into signal-like and background-like regions when many vertices (i.e. many tracks) are present. With very few vertices, background-like regions like random crossings of tracks may be misclassified as vertices because the signal-like regions are less abundant in the filtered track density. To reduce this background, the vertex density must be filtered with a rather soft filter.
Figure 4: Purity and Efficiency for different vertex qualities and LHC collisions at low luminosity. A definition of the vertex quality is given in the text. Each event contains one hard vertex and a Poisson-distributed number of minimum bias vertices (mean = 5).

This also decreases the efficiency of vertex finding as seen in figure 4. RAVE is the better choice if 5 or fewer vertices are to be found.

3.4. Speed

Figure 5 shows how the CPU processing time is affected by the number of vertices. For high numbers of vertices, the medical imaging approach is much faster than AVR in finding the vertices. The reason is that the CPU time depends approximately linearly on the number of vertices whereas the execution time for AVR scales with the number of two-track-crossings and

Figure 5: Average CPU time on a 2.33 GHz CPU for reconstruction of one event. \(<\mu>\) is the mean number of interactions per bunch crossing.
therefore scales like $O(<\mu>^2)$. The following list shows how the different steps of vertex finding depend on the number of vertices (or tracks). Measured CPU times are shown in Table 1:

**Backprojection** Linear: The backprojection is very fast and linear in the number of tracks.

**Prepare Fourier transformation** Constant: The whole histogram must be copied to an array compatible with FFTW. Therefore the time is independent of the track density. If instead of a ROOT histogram a data structure compatible with FFTW was used, this step could be skipped. However, this was not pursued because this requires a reimplementation of many functions provided by ROOT histograms.

**Fourier transformation** Constant: Independent of the data, the transformation takes a constant amount of time.

**Filtering** Constant: Each entry in the transformed array is multiplied with a number obtained from the filter equation (1) and from the window functions.

**Backtransformation** Constant: Inverse of Fourier transformation.

**Extraction of the vertex density** Maximally constant: All non-zero entries of the array are written to the histogram. For low numbers of vertices many zero entries can be skipped to save time. This step could also be skipped if a data structure compatible to FFTW was used.

**Finding vertices** Approximately linear in $N_{\text{Vtx}}$: The execution time for this step depends on the number of vertices: The clustering of the vertex density takes longer if more clusters are present in the filtered density.

It should be noted that speed was not the primary goal for the implementation of the algorithm. Still, a big improvement in the time consumed for vertex finding with many tracks has been achieved, because the MIV algorithm is independent of the number of two-track-combinations that scales quadratically with the number of tracks. It only depends linearly on the number of tracks and on the number of vertices (see Table 1): Processing a single track is very fast. Therefore, the only step significantly depending on the size of the input data is the clustering of the vertex density. It is likely that a different clustering algorithm designed for maximum speed might improve this situation.

As mentioned above, a speed-up is also possible for the steps that do not depend on the input data if different data structures were used. Furthermore, the current implementation is only single-threaded. A parallel implementation of selective steps may also speed up the algorithm.

### Table 1: Average CPU time for the steps in the reconstruction of one event (Intel Xeon CPU, 2.33 GHz)

| $N_{\text{Vtx}}$ | 5       | 25      | 50      | 100     |
|------------------|---------|---------|---------|---------|
| Backprojection   | 0.010 ± 0.006 | 0.036 ± 0.009 | 0.07 ± 0.01 | 0.11 ± 0.01 |
| Prepare Transformation | 1.38 ± 0.06 | 1.36 ± 0.03 | 1.40 ± 0.04 | 1.38 ± 0.19 |
| Transformation   | 0.55 ± 0.01 | 0.55 ± 0.01 | 0.57 ± 0.02 | 0.57 ± 0.03 |
| Apply filters    | 0.53 ± 0.01 | 0.53 ± 0.01 | 0.53 ± 0.01 | 0.53 ± 0.02 |
| Backtransformation | 0.56 ± 0.01 | 0.56 ± 0.01 | 0.59 ± 0.02 | 0.55 ± 0.03 |
| Extract density  | 0.30 ± 0.09 | 0.65 ± 0.13 | 0.88 ± 0.13 | 1.18 ± 0.18 |
| Find clusters    | 0.06 ± 0.04 | 0.36 ± 0.12 | 0.70 ± 0.14 | 1.19 ± 0.19 |
| MI Vertex Finder total | 3.39 ± 0.12 | 4.05 ± 0.18 | 4.74 ± 0.20 | 5.52 ± 0.33 |
| avr rave-0.6.6   | 0.18 ± 0.01 | 2.79 ± 0.10 | 14.77 ± 0.44 | 70.79 ± 2.52 |
4. Summary and Outlook

A new algorithm based on medical imaging methods was developed for vertex finding in events with large numbers of tracks or vertices. In comparison with the open-source vertexing package RAVE (AVR algorithm) our algorithm is able to find primary vertices with higher efficiency and purity at high pile-up whereas RAVE performs better at low luminosities. When we developed the algorithm emphasis was put on good performance in events with large track numbers. It should be noted however that the algorithm still features the unavoidable degradation of performance if one considers events with more and more pile-up.

Execution time for the vertex finding step scales linearly with the number of tracks and vertices. Because of this dependence, a significant speed advantage compared to standard algorithms exist at high pile-up although the tested implementation has not yet been fully optimized for speed.

4.1. Outlook

The focus of this study has been on finding primary vertices. An extension of this method to flavor tagging via secondary vertices is currently under investigation and will be presented in a future publication. Another extension of the algorithm may be the combination of our algorithm with a standard fit-based vertex finder. MIV would be used as a vertex seed finder and the exact vertex position would be found by a standard method, for example the multi-vertex fitter of the RAVE package. A presentation of our algorithm as an open-source package is currently being considered.

Acknowledgments

The authors would like to thank Harris Kagan (The Ohio State University) who suggested to apply medical imaging methods to the vertexing problem. E.V.T. acknowledges support by the German ministry for education and research (BMBF) and S.H. from the Bonn-Cologne Graduate School of Physics and Astronomy (BCGS) and the Cusanuswerk.

References

[1] Gutbrod H H et al. 2006 Baseline technical report (btr) for fair, Tech. rep. GSI Darmstadt URL http://www.fair-center.de/de/fair-nutzer/publikationen/fair-publikationen.html
[2] High luminosity large hadron collider, Tech. rep. URL http://hilumilhc.web.cern.ch/HilumiLHC/
[3] Dougherty G 2009 Digital Image Processing for Medical Applications (Cambridge University Press) ISBN 9780521860857
[4] Jackson D J 1997 Nucl. Inst. & Meth. A 388 247–53
[5] Strandlie A and Frühwirth R 2010 Rev. Mod. Phys. 82(2) 1419–58
[6] Smith S W 1997 The scientist and engineer’s guide to digital signal processing (San Diego, CA, USA: California Technical Publishing) ISBN 0-9660176-3-3
[7] Brun R and Rademakers F 1997 Nucl. Inst. & Meth. in Phys. Res. A 389 81–6
[8] Frigo M and Johnson S G 2005 Proceedings of the IEE pp 216–31
[9] Colsher J G 1998 Physics in Medicine and Biology 25 103–15
[10] Defrise M, Townsend D W and Clack R 1989 Physics in Medicine and Biology 34 573–87
[11] Aad G et al. 2009 Expected performance of the ATLAS experiment: detector, trigger and physics (Geneva: CERN)
[12] Brüning O S, Collier P, Myers S, Ostojic R, Poole J and Proudlock P 2004 LHC Design Report (Geneva: CERN)
[13] Waltenberger W 2011 Nuclear Science, IEEE Transactions on 58 434–44 ISSN 0018-9499
[14] Sjostrand T, Mrenna S and Skands P Z 2006 JHEP 0605 026 (Preprint hep-ph/0603175)