CMS Conference Report

12 Dec 2003

Vertex reconstruction in CMS

E. Chabanat, N. Estre

Institut de Physique Nucléaire de Lyon, Villeurbanne, France

J. D'Hondt^{a)}, P. Vanlaer^{b)}

Interuniversity Institute for High Energies, ULB-VUB, Belgium

K. Prokofiev, T. Speer

Physik-Institut der Universität Zürich, Switzerland

R. Frühwirth, W. Waltenberger^{c)}

Institut für Hochenergiephysik der ÖAW, Vienna, Austria

Abstract

Because of the high track multiplicity in the final states expected in proton collisions at the LHC experiments novel vertex reconstruction algorithms are required. The vertex reconstruction problem can be decomposed into a pattern recognition problem (“vertex finding”) and an estimation problem (“vertex fitting”). Starting from least-squares methods, robustifications of the classical algorithms are discussed and the statistical properties of the novel methods are shown. A whole set of different approaches for the vertex finding problem is presented and compared in relevant physics channels.

^{a)} Supported by the FWO-Vlaanderen.

^{b)} Supported by the Belgian Federal Office for Scientific, Technical and Cultural affairs through the Interuniversity Attraction Pole P5/27 and by the FNRS.

^{c)} Supported by the Fonds zur Förderung der wissenschaftlichen Forschung, Project 15177. Corresponding author.

1 INTRODUCTION

LHC experiments present a new challenge in the field of event reconstruction. Algorithms have to cope with an immense background in high-luminosity scenarios. Vertex finding algorithms have to disentangle the tracks of different vertices in very challenging topologies, such as decays of B^0 or D^0 mesons, where the separation of the vertices is typically of the order of a few hundred microns up to a few millimeters. Vertex fitters need to cope with mismeasured and non-Gaussian track parameters and mis-associated tracks. Studies of robust vertex fitting techniques are discussed in Section 2; novel approaches to vertex finding are presented in Section 3.

2 VERTEX FITTING

The classical method is the least-squares method (LS), usually formulated as a Kalman filter [1]. This method can be shown to be optimal in the case that the model is linear, the measurement uncertainties are Gaussian, and the data is not contaminated by mis-associated tracks. None of these conditions will hold for real data in LHC experiments. Hence robustified methods [2, 3] – methods that are less sensitive to outliers – are desirable. The three following methods have been implemented and studied:

- Adaptive method: An iterative reweighting least-squares method. Instead of the sum of squared residuals a weighted sum of squared residuals is minimized: $\hat{\vec{x}}_{\text{Adaptive}} = \text{argmin}_{\vec{x}} \sum_{i=1}^n (w_i \cdot r_i^2(\vec{x}))$. Here n denotes the number of tracks, while $r_i^2(\vec{x})$ is the squared residual of track i as a function of the vertex position $\vec{x}$. The weight function w_i is $w_i = \phi(r_i) (\phi(r_{\text{cutoff}}) + \phi(r_i))^{-1}$, where ϕ is the probability density function of a model of the track residuals. Note that $w_i = 0.5$ for $r_i = r_{\text{cutoff}}$. The most common choice for ϕ is a Gaussian distribution and results in $w_i = (1 + \exp((r_i^2 - r_{\text{cut}}^2) \beta))^{-1}$, where r_{cut}^2 denotes a *cutoff* parameter, and $\beta \equiv 1/(2T)$ introduces a temperature. This temperature can (and shall) be lowered in a well-defined annealing schedule.
- Trimming method: Only a (user-defined) fraction of the sum of the squared residuals is minimized: $\hat{\vec{x}}_{\text{Trimming}} = \text{argmin}_{\vec{x}} \sum_{i=1}^{h < n} r_i^2(\vec{x})$. For finding this minimum a variation of the “Fast-LTS” algorithm [4] is used.
- LMS: The median of squared residuals is minimized: $\hat{\vec{x}}_{\text{LMS}} = \text{argmin}_{\vec{x}} \text{med}(r_i^2(\vec{x}))$. An LMS estimator that works on the individual components has been implemented. This method ignores the spatial structure of the input data. A full 3d method that works within our time constraints would be desirable.

Table 1 shows a comparison of the different fitting techniques in 1000 fully simulated $c\bar{c}$ signal events, with a transverse jet energy of 100 GeV, in the “barrel” region of CMS ($|\eta| < 1.4$). The trimmer was used with a trimming ratio of 20%, the adaptive method had a cutoff at $r_{\text{cut}}^2 = 16$. The time was measured on a 2.5 GHz Intel Xeon processor. A failure was defined as a result that meets at least one of the following criteria:

- The reconstructed vertex has less than 50% in common with the set of reconstructed tracks associated with the simulated primary vertex.
- The reconstructed vertex is more than 3mm displaced from the simulated vertex in either the x or the z coordinate.
- The absolute value of the standardized residual is greater than 50, in either the x or the z coordinate.

The following conclusions can be drawn [5]:

- The adaptive fitter is able to downweight the most dangerous outliers, i.e. the ones which would introduce the largest bias, while making optimal use of the inliers. The error estimate is also close to optimal.
- The trimming vertex fitter has only a very specific use case. It only makes sense when the fraction of outliers is exactly known *a priori*.
- The least-squares method is still the fastest fit, and optimal in the case of “perfect” data. Obviously it is very sensitive to outliers.
- The coordinate-wise LMS is very insensitive to outliers, but has a worse resolution than the other robust algorithms. No use case has been found.

Name	res, x [μm]		st.res, x		res, z [μm]		st.res, z		time [ms]	fail
	RMS	σ_{Fit}	RMS	σ_{Fit}	RMS	σ_{Fit}	RMS	σ_{Fit}		
LS	660	287	18.6	17.7	640	261	17.7	10.1	40	349
Trimmer	61	22	4.7	1.4	81	26	4.3	1.5	74	8
Adaptive	21	15	1.3	1.15	24	18	1.3	1.14	67	2
LMS	48	31	2.8	2.0	42	31	3.2	2.6	40	3

Table 1: Comparison of different fitting methods in $c\bar{c}$ events. “res” denotes the residuals, “st.res” the standardized residuals, and “fail” the number of failures.

3 VERTEX FINDING

Vertex finding algorithms can be divided into two groups: hierarchic and non-hierarchic methods. Hierarchic methods [3] are algorithms whose working can be visualised by a tree graph; all other methods are called non-hierarchic. The following set of non-hierarchic (“learning”) methods has been implemented:

- Vector quantisation [6]: It starts with a set of vertex candidates (“prototypes”). The data points then sequentially “pull” the closest prototype towards themselves, with a given strength (“learning rate”), that is decreased after every iteration. The prototypes that “win” (i.e. move) very often, receive a penalty. This makes sure that less “fortunate” prototypes get to move as well. The specific algorithm employed is called “frequency sensitive competitive learning”.
- K-means [7]: It is similar to the vector quantisation. The only difference is that it works in a “batch mode”: all data points are associated to the prototypes at once. An overall update of all prototypes is then performed.
- Deterministic annealing [8, 9]: This algorithm searches the minima of a function which has the same form as the Gibbs free energy. The temperature parameter is used to smooth out the function in order to avoid local minima. The temperature is decreased (annealing) and when the global minimum for a given T becomes unstable (negative second derivative), the prototype is split in 2 prototypes, each converging towards a different minimum.

Currently all non-hierarchical algorithms are implemented on top of the *apex point* concept [3] which maps every track to one or more representative points. Clustering is then performed on these representative points. The algorithms can additionally exploit a 3-by-3 error matrix that comes with an apex point. This error matrix implicitly has the information of the track direction, since this direction usually forms the main axis of the error ellipsoid.

3.1 Simulation experiment

The non-hierarchic methods have been compared with more established hierarchic vertex finding procedures – one “divisive” method (the principal vertex reconstructor – “PVR”), and one “agglomerative” method (“Agglo”). The results were obtained using fully simulated $b\bar{b}$, $c\bar{c}$, and $q\bar{q}$ decay events with jet energies at 100 GeV, in the barrel region ($|\eta| < 1.4$). In order to simulate more realistic use cases, all tracks and vertices that were found outside the jet cones supplied by the calorimetry subsystems were filtered out. The events were generated with the CMSIM simulation program. All algorithms were fine-tuned to maximize the following “score function”:

$$S \equiv 100 \cdot E_{\text{Pr}}^{0.5} \cdot E_{\text{Sc}}^1 \cdot P_{\text{Pr}}^{0.25} \cdot P_{\text{Sc}}^{0.25} \cdot A_{\text{Pr}}^{0.25} \cdot A_{\text{Sc}}^{0.25} \cdot (1 - F)^1,$$

where E denotes “vertex finding efficiency”, P “vertex purity”, A “vertex assignment efficiency”, and F “fake rate”. “Pr” denotes primary vertices, “Sc” stands for secondary vertices. E is defined as the ratio of the number of simulated vertices which are reconstructed with a track purity strictly higher than 50%, to the number of simulated vertices which have at least two tracks reconstructed. F is defined as the ratio of the number of reconstructed vertices with a track purity equal to or lower than 50%, to the total number of reconstructed vertices. For a precise definition of the other “performance estimators” see [10]. The exponents reflect the importance of the various ingredients.

3.2 Results

Table 2 shows the result of the simulation experiment. It can be seen that the agglomerative method has higher vertex finding efficiencies than the divisive method, albeit at the price of a higher fake rate. All novel non-hierarchic methods score better than the classical methods, except for the $q\bar{q}$ case, where PVR’s tendency towards low fake rates is crucial in this specific setup. K-means scores most consistently well. The results are especially encouraging, since further improvements can be expected from all three novel methods.

Name	$b\bar{b}$				$c\bar{c}$				$q\bar{q}$			
	E_{Pr}	E_{Sc}	F	S	E_{Pr}	E_{Sc}	F	S	E_{Pr}	E_{Sc}	F	S
PVR	.84	.46	.48	17	.95	.34	.46	15	.98	.50	.40	24
Agglo	.92	.69	.67	19	.98	.59	.63	19	.99	.62	.66	19
VQuant	.92	.64	.56	23	.97	.46	.46	22	.99	.55	.54	23
DetAnn	.93	.56	.48	24	.96	.34	.33	20	.99	.43	.40	22
K-means	.97	.71	.60	23	.99	.62	.57	22	1.0	.58	.54	23

Table 2: Comparison of performance of three vertex finding algorithms, tested with Pythia di-jet event samples.

4 CONCLUSIONS

A good understanding of the statistical properties of the novel robustified fitting techniques has been reached. The adaptive fitting method performs better than the classical least-squares method in a wide range of use cases. It has also been shown that the non-hierarchic methods based on apex points outperform more conventional methods in relevant physics channels. A few relative advantages and disadvantages of different finding techniques could be identified. Future research will mainly go into the following areas:

- Apex point finding algorithms: New ideas are a deterministic annealing in one dimension, a gravitational clustering algorithm [11], and a wavelet analysis.
- More non-hierarchic cluster methods: Algorithms that are considered are super-paramagnetic clustering [12], or Potts neurons [13].
- Multi-vertex fitting: An algorithm that performs simultaneous estimation of several vertices which compete for the tracks.

References

- [1] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nucl. Instrum. Meth. A, 262 (1987) 444–450.
- [2] R. Frühwirth et al., *New developments in vertex reconstruction for CMS*, Nucl. Instrum. Meth. A, 502 (2003) 699–701.
- [3] R. Frühwirth et al., *New vertex reconstruction algorithms for CMS*, Proc. of the CHEP 2003, ePrint physics/0306012.
- [4] K. Van Driessen and P.J. Rousseuw, *Computing LTS regression for large data sets*, technical report, University of Antwerp, 1999.
- [5] J. D’Hondt, R. Frühwirth, P. Vanlaer and W. Waltenberger, CMS Note, in preparation.
- [6] A. Gersho and R. M. Gray, *Vector Quantization And Signal Compression*, Kluwer, Boston, 1992.
- [7] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.
- [8] K. Rose, *Deterministic annealing for clustering, compression, classification, regression and related optimization problems*, Proc. IEEE, vol. 86, num. 11 (1998) 2210–2239.
- [9] N. Estre, E. Chabanat and P. Vanlaer, *Deterministic Annealing for Vertex Finding at CMS*, CMS Note, in preparation.
- [10] T. Boccali et al., *Vertex reconstruction framework and its implementation for CMS*, Proc. of the CHEP 2003, ePrint physics/0306014.
- [11] S. Kundu, *Gravitational clustering: a new approach based on the spatial distribution of the points*, Pattern Recogn. 32 (1999) 1149–1160.
- [12] M. Blatt, S. Wiseman and E. Domany, *Super-paramagnetic clustering of data*, Phys. Rev. Lett. 76 (1996) 3251–3254.
- [13] M. Bengtsson and P. Roivainen, *Using the Potts glass for solving the clustering problem*, Int. J. Neural Syst., vol. 6, num. 2 (1995) 119–132.