Three dimensional cleaning of data from HEP events

H.Drevermann[1], D.Kuhn[2],
P.Luthaus[3] and B.S.Nilsson[4]

Abstract

In large HEP detectors typically a very high number of 3D hits is recorded simultaneously. We present a fast method to clean these data from unwanted hits, which significantly improves event visualisation. The accepted hits can be grouped, each group containing practically all hits of one or a few single tracks. The execution time is proportional to the number of hits.

[1]CERN, CH-1211 Geneva 23
[2]Inst. für Experimentalphysik der Universität Innsbruck, A-6020 Innsbruck,
supported by grant of Austrian Federal Ministry of Science and Transport
[3]Experimentelle Physik IV, Universität Dortmund, D-44221 Dortmund
[4]Niels Bohr Institute, DK-2100 Copenhagen

1 Introduction

Many HEP detectors record a very large number of hits simultaneously. This large amount of data creates problems for further data processing, e.g. event visualisation. For the case of 3-dimensional space coordinates, called hits, we propose a fast cleaning method, which we exemplify with data from the LHC detector ATLAS.

When ATLAS will run with full luminosity a typical data set will contain the hits of the event, which triggered the recording, the hits of a few tens of background events and random noise. The data set contains also hits from delta electrons etc.. Here we are concerned with the 3-dimensional hits sampled in the two innermost parts of the ATLAS detector, namely the pixel detector and the silicon strip detector, called silicon detectors in the following. In the case of the silicon strip detector there will be furthermore ghost hits, if several charged particles pass through the same module.

For visual analysis of these data it is of interest to separate the "good" hits belonging to sufficiently high p_t tracks of the triggering event from the "bad" hits, which consist of hits from low p_t tracks of the triggering event, of ghost hits, of all hits coming from the background events, of delta rays and noise etc.. To solve this task, we will propose in the following a method, which is fast despite the very high number of hits in the silicon detectors.

The examples, on which we will exemplify the proposed method, stem from a data set containing 10400 3-dimensional hits.

2 Cleaning Method

In standard track following methods, hits are flagged as good if they are found to be associated to tracks identified beforehand. With a large number of hits most of these methods become very time consuming.

In contrast, we base the method of separation on the difference between the features of the good and bad hits, thus avoiding to go through the track following process.

In the case considered here the good hits lie on tracks of rather high p_t pointing to one primary vertex. Most of the bad hits lie on tracks of rather low p_t pointing to several primary vertices at other positions along the beam line. Noise and ghost hits etc. do not lie on tracks. We try to exploit these features, which can be expressed more formally in the following way:

A hit on layer l_1 is accepted as good if it is sufficiently close in ϕ and θ - i.e. in direction as seen from the primary vertex - to a set of other hits on different layers $l_2, l_3, ...l_k$ with $(l_1 \neq l_2 \neq .. \neq l_k)$, where $k \geq L$. L is the minimum number of different layers to accept the hit.

ϕ and θ are calculated in a coordinate system with the primary vertex of the triggering event at its origin. All hits not fulfilling this criterion are assumed to be bad.

The potentially very large computing time of combinatorial methods can be avoided by a histogram method, where a 2-dimensional ϕ/θ histogram of all hits is generated. Normally, a hit is then accepted as good, if it falls into a histogram bin with a sufficient number of entries. However several hits on only few or even one layer might be accepted as good. Therfore, we request in addition that the hits lie on at least L different layers as mentioned above, so that we extend the conventional 2-dimensional cleaning method, which neglects the third dimension, to a 3-dimensional cleaning method.

The procedure is as follows: each histogram bin is represented by a number N, which has at least as many bits as there are different layers in the detector. The histogram is filled by setting that bit to 1, which corresponds to the layer of a given hit, so that hits of the same layer falling into the same θ, ϕ interval are only counted once. This histogram we call a "layer histogram". A hit is then accepted as good if at least L bits of N are equal to 1.

This method makes use of a fast bit handling routine. The number of bits set to 1 in the number N can be evaluated through a look up table or, if there are too many layers, by a combination of a few look up tables.

As an example assume just three hits inside one bin in the histogram, where two of them are located on layer 4 of the detector and one on layer 5. Initially the bin number N has to be set to zero. When the first hit is filled, bit 4 of N is set to 1, thus $N = 8$. Filling the second hit on layer 4 does not change anything, as the bit 4 is already set to 1. The third hit which comes from layer 5 sets bit 5 to 1 which means that:

$N = 24 = 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 1 \cdot 16$

From this number it can be derived that 2 layers have been hit inside the bin. This is done via a look up table:

number of layers = LUT(24) = 2.

In order to avoid problems due to rounding errors, each hit is entered in all eight neighbouring bins as well. Bins containing at least one genuine activated layer will be called "direct bins", neighbouring bins without a genuine hit will be called "indirect bins" in the following.

The ATLAS silicon detectors are placed in a homogeneous magnetic field parallel to the Z-axis. It can be shown [1], that for tracks with sufficiently high p_t the polar angle θ is aproximately constant, whereas the variation of ϕ inside the silicon detectors is proportional to ${p_t}^{-1}$. The minimal interval size $\Delta\phi * \Delta\theta$ depends therefore in the case of $\Delta\theta$ on the measuring accuracy, in the case of $\Delta\phi$ on the minimum p_t value to be accepted. The minimum number L of layers required in such a θ, ϕ cone depends on the detection efficiency of the silicon detectors.

For the cleaning the following values were used:

$\Delta\phi = 2$ deg

$\Delta\theta = 0.18$ deg

minimum number of layers required: $L = 4$

For the data set containing the triggering event and background events a small section of a normal histogram is shown in figure 1a, where the number filled is the

total number of hits inside each bin. In the upper part of the figure three histogram bins have 5 or 6 entries, so that the hits falling into these bins would be accepted as good. To exemplify the power of 3-dimensional cleaning the same region of the layer histogram is shown in figure 1b, where one sees that the hits stem from only one layer and hence are discarded

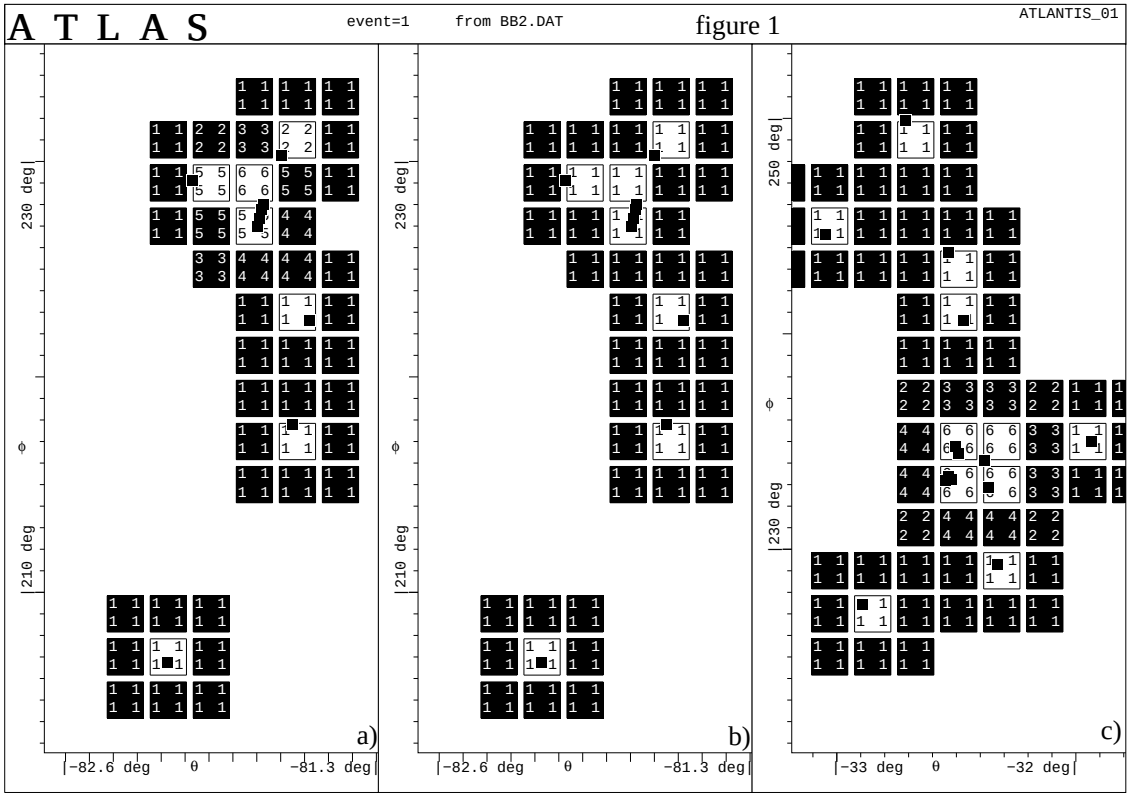

Figure 1: **Cleaning:** A histogram section is shown as normal histogram (a) and as layer histogram (b). Another section of the layer histogram is shown in (c). Direct bins are coloured white, indirect bins black. The hits, from which the histogram section was filled, are superimposed (small black squares). The number of entries (1a) and the number of activated layers, respectively, (1b,c) is shown four times close to the corners of the respective histogram bin, in order not to be masked out by the hits.

Figure 1c gives another section of the layer histogram which contains a group of seven close hits on 6 different layers. Six of these hits belong to a simulated track. All seven hits will be accepted as good, whereas the other hits on the picture will be flagged as bad. In effect these hits can be identified from the Monte Carlo as bad.

This method, which was applied to high p_t tracks, can easily be modified to work also in other momentum intervals.

3 Grouping of Hits

Figure 2a shows a third section of the layer histogram. Requiring $L = 4$ to accept hits as good many bad hits and four clusters of good hits can be discerned. These hits lie in direct bins with at least L activated layers. These bins we will call good bins. In order to facilitate visualisation further, the method described so far to clean hits may be extended by "grouping" those hits which fall into the same cluster of good bins.

For this purpose, we associate a unique cluster number to all good bins which touch each other. This is shown in figure 2b, where the four clusters got the numbers 13, 35, 9 and 7. In this way, groups of hits may be identified and lists may be formed, so that for visual analysis one can selectively work with one group only.

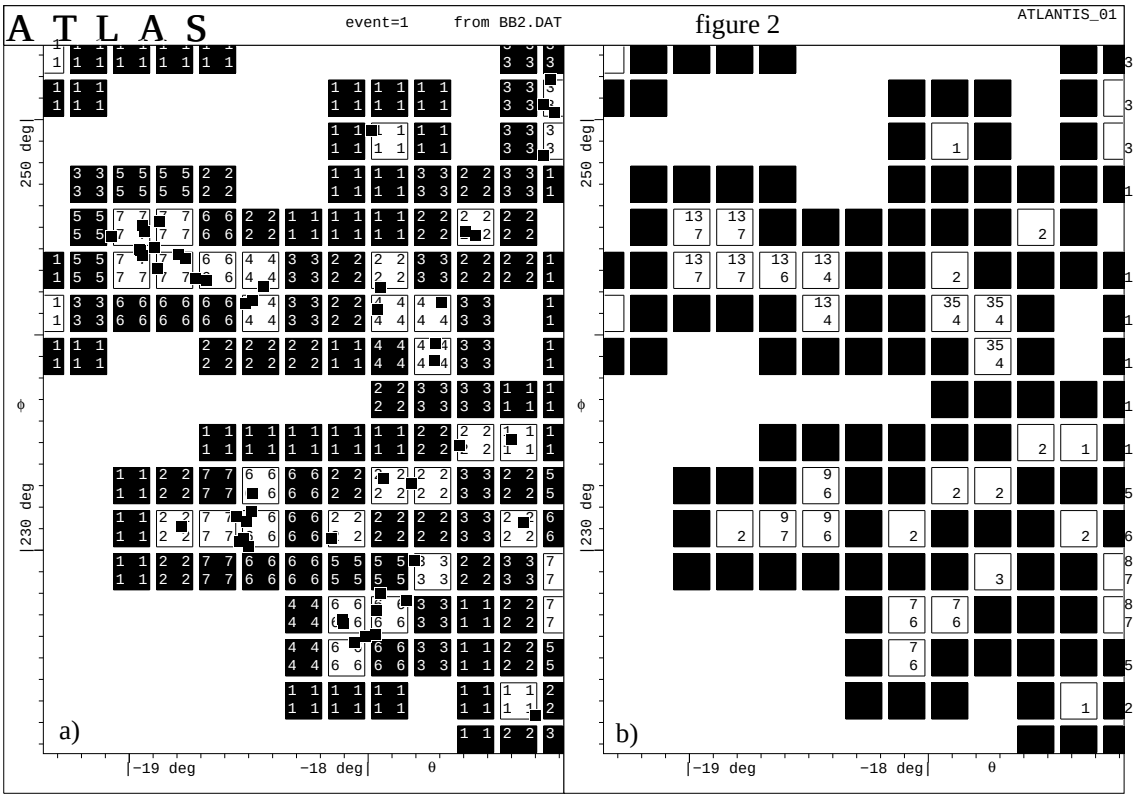

Figure 2: **Grouping** Grouping:(a) A small section of a layer histogram with background hits and with hits associated to five tracks, (b) the same section, where all good bins are clustered. In the clustered bins the cluster number is printed on top of the number of activated layers.

In order to find and number all histogram clusters of good bins, one normally loops over all histogram bins. This is too time consuming in our case, as the total number of histogram bins is much larger than the number of hits. Instead, it is faster to loop over all hits and to execute the following algorithm:

- If a hit does not fall into a good bin, it is flagged as bad.

- Else:
  - If a hit falls into a good bin not yet treated, it gets a new cluster number and the histogram clustering starts from the corresponding bin in the layer histogram until the new cluster number is allocated to all connected good bins.
  - otherwise the hit gets the cluster number already allocated to this bin.

4 Event Visualization

In the following we will demonstrate, how cleaning and grouping acts on hits in a section of the detector containing a b jet and background events. Figure 3 shows these hits in a ρ/Z projection (3a). No tracks are recognized. Even when showing the cleaned hits only, which are seen in a ρ/Z and a Y/X projection in figures 3b and 3c, no tracks are recognized. Only the layers are recognized, as in these pictures the pattern recognition of the human brain groups all points into layers instead of tracks, because the mean distance of points in the single layers is much smaller than the mean distance of points of single tracks.

These projections are therefore not suitable to judge the efficiency of cleaning. Instead we use the V-plot representation [1] (developed originally for the ALEPH experiment), which improves track recognition considerably.

In the V-plot a 3D hit is represented by two points drawn left and rigth to its ϕ/θ position, with a distance proportional to $\rho_{max} - \rho$, where ρ is the distance of the hit from the Z-axis and ρ_{max} the outer radius of the silicon detectors. Helix segments at the primary vertex are represented by a V pattern. The gradient of the arms of the V pattern is inversely proportional to p_t. No track assignment of the hits is used to generate the V-plot. A V-plot of the cleaned hits is shown in figure 3d, where in most cases the V-pattern is clearly recognized.

In the simulated data it is assumed that the origin of the hits is known, i.e. that their track assignment is correctly given. Figure 4 shows V-plots of the four combinations:

- cleaned jet hits (4a)
- cleaned background hits (4b)
- rejected jet hits (4c)
- rejected background hits (4d).

Each hit belongs to only one of these combinations. If the cleaning would deliver exactly the jet hits, the two combinations 4a and 4d should contain all hits while the other two, 4b and 4c, should be empty. If not, the combination 4c should contain

- hits of low p_t tracks of the triggering event,

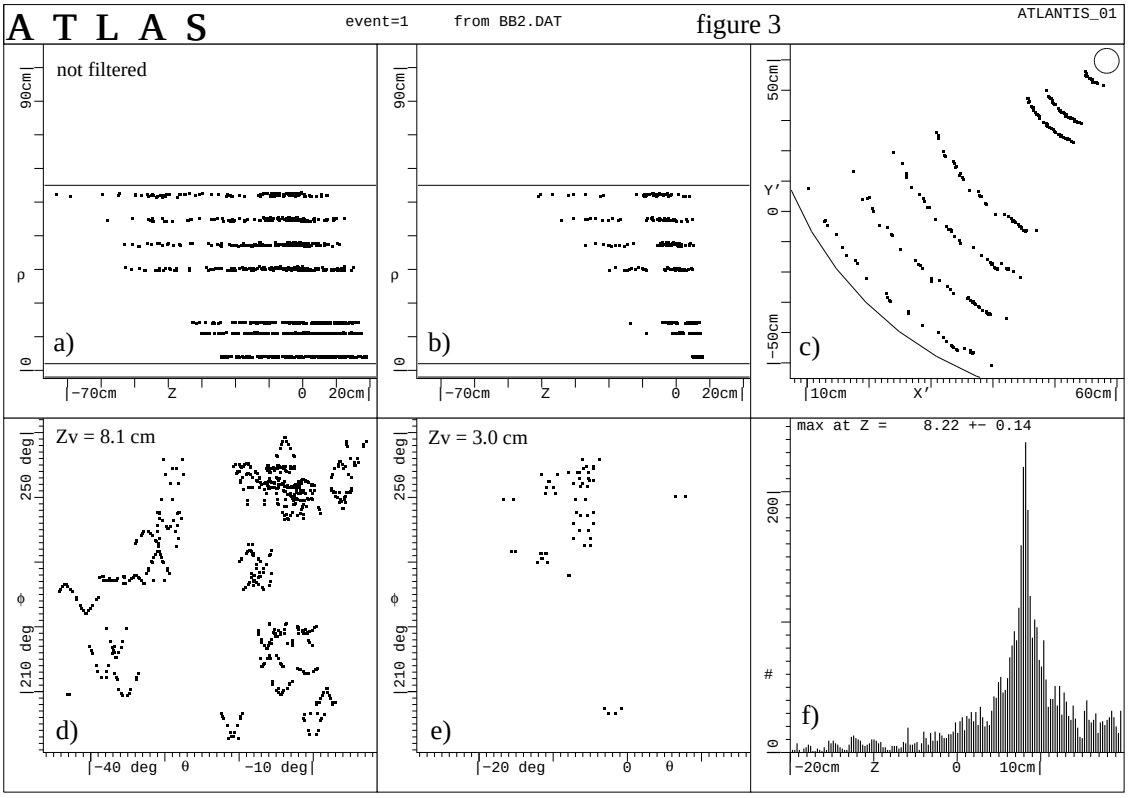

Figure 3: **Check of Tracks with the V-plot:** Event visualization and Z_{vertex} finding: Region of a jet of the triggering event: (a) ρ/Z projection, (b) ρ/Z projection of cleaned hits, (c) Y/X projection of cleaned hits, (d) V-plot of cleaned hits with vertex position at 8.1 cm, (e) V-plot of cleaned hits with vertex position wrongly assigned to 3.0 cm, (f) number of hits passing the cleaning as a function of assumed vertex position Z.

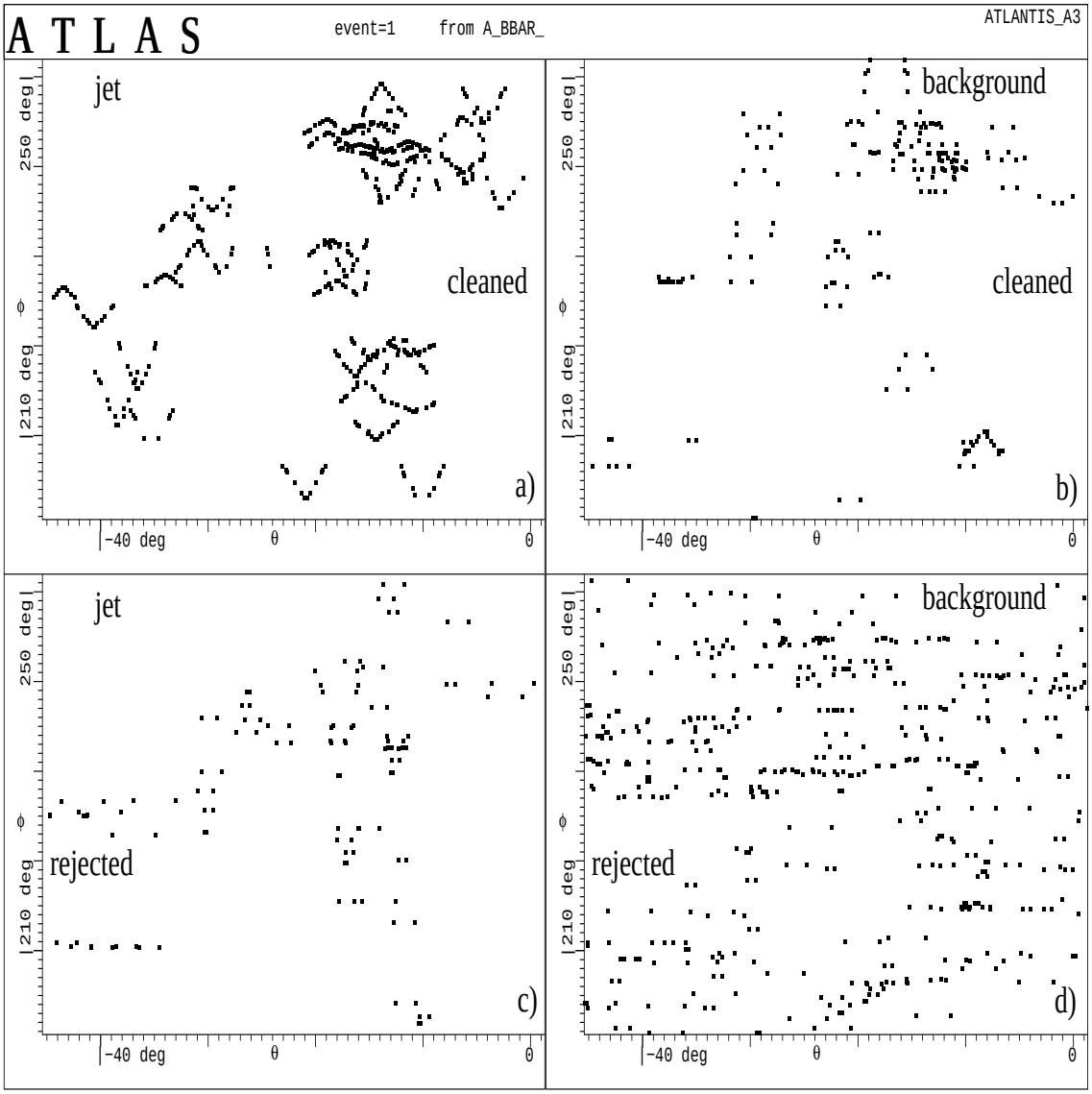

Figure 4: **Visual check of cleaning**
Visual check of cleaning A region of a V-plot for four exclusive combinations of hits. (a) Cleaned jet hits, (b) cleaned background, (c) rejected jet hits, (d) rejected background hits.

- wrongly rejected hits of high p_t tracks,
- hits, which do not lie on a primary track (e.g.hits from delta electrons, which are also flagged as belonging to the triggering event, see below).

Combination 4b contains

- background hits, which the cleaning failed to reject, especially hits of background tracks, which point to the primary vertex of the triggering event,
- jet hits, which were not flagged correctly.

On the lower right corner of figure 4b, a V-pattern is clearly seen revealing the existance of a track, the hits of which were not flagged by the simulation as belonging to the triggering event.

Two examples of hit grouping are demonstrated in figure 5. All hits which were flagged to belong to a selected track are shown in figure 5 as V-plot (a), in a ϕ/ρ projection (b) and a ρ/Z projection (c). Only the hits drawn as solid squares passed the cleaning process and were indeed assigned to one group containing no other track in this case. The straight lines represent the corresponding tracks from the MC truth. As seen from the V pattern, the cleaned hits are the good hits of the track; the rejected ones (open squares) are probably hits of secondary delta electrons, which are also flagged as hits belonging to the track.

As a second example, a group was selected which contains more than one simulated track. The hits belonging to the group are shown in the same projections as for the first example (see figures 5d, e, g). In this case some bad hits passed the cleaning process and the grouping could not separate the three tracks in this group. Figure 5f presents figure 5e in compressed form, where the tracks are easily recognized even without the suggestive lines.

5 Finding the Vertex Position along the Beam Line

The cleaning method (as well as the V-plot, which is a special form of a ϕ/θ projection) relies on a precise determination of θ, which in turn implies the knowledge of the Z-position of the vertex of the triggering event. If the Z-position is not known - e.g. from earlier trigger levels - or if a given Z-position should be checked, a method is needed to find the Z-position of the vertex. Due to the small beam diameter (e.g. $15\mu \times 15\mu$ at LHC) the X-Y-position of the vertex is known well enough to determine ϕ.

We propose a method, which allows to determine the vertex position without track following. The method can be applied either if the triggering event has many high p_t tracks and the background events have practically only hits of low p_t tracks, or if a region, where the above condition is fulfilled, is known from other sources.

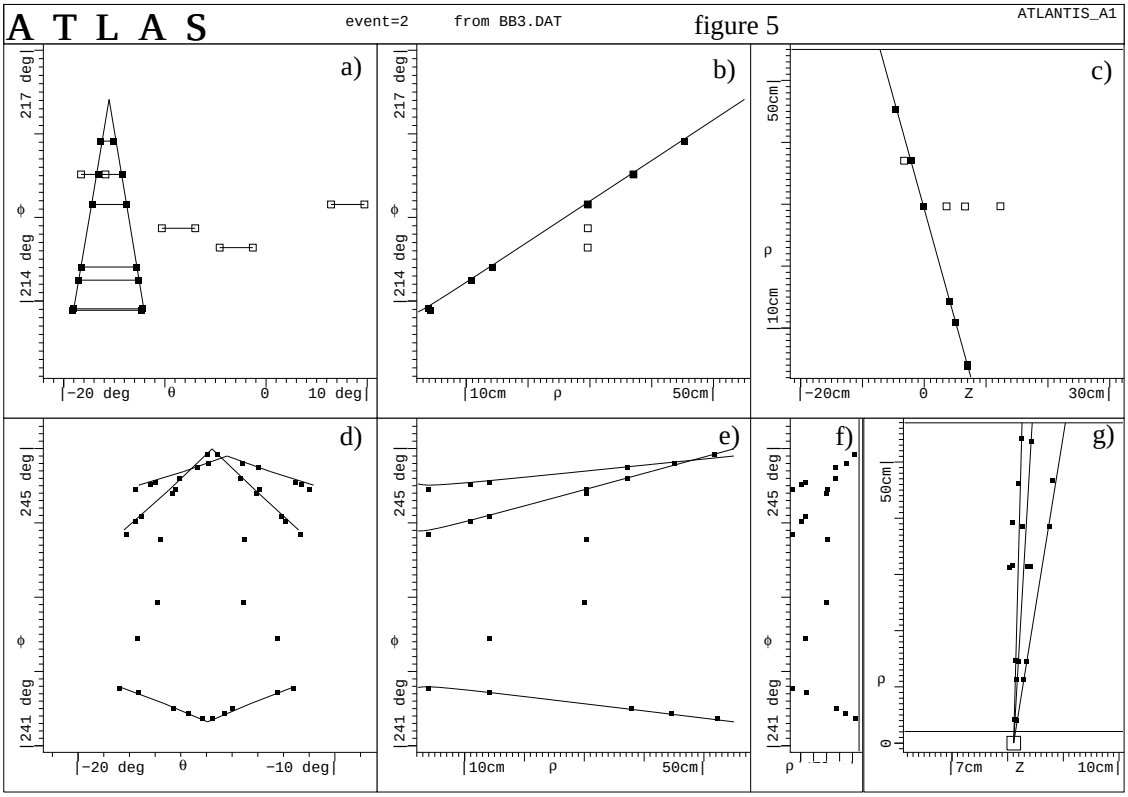

Figure 5: **Hits of a selected track and hits of a selected group:**
Hits of a selected track and hits of a selected group:
The lines represent the MC tracks, which created the hits.
(a), (b), (c) Example of hits created by a track of the jet. Solid squares: cleaned hits, open squares: rejected hits, (a) V-plot, (b) ϕ/ρ projection, (c) ρ/Z projection, (d), (e), (f), (g): all hits belonging to a selected group, (d) V-plot, (e) ϕ/ρ projection, (f) ϕ/ρ projection compressed without tracks, (g) ρ/Z projection,

The method consists of varying Z_{vertex} and counting the number of hits N_h passing the cleaning. The maximum number of hits indicates the correct vertex position.

This may be seen by comparing figure 3d and 3e which shows the region of the b-jet. In figure 3d the nominal vertex position as known from the simulation was used, whereas in figure 3e a different, arbitrarily selected Z-position was applied. We recognize that only few hits pass the cleaning process in the latter case. By applying the cleaning process to the subset of hits from this b jet region at subsequent Z positions we get the 1-dimensional histogram seen in figure 3f, where N_h is shown as a function of Z. This histogram shows a clear peak in the bin between 8.08 and 8.36 cm, which contains the nominal value of 8.11 cm, known from the simulation. The precision of the determination of the vertex position may be improved in a second pass by sampling with a smaller interval size around the peak found in the first pass.

6 Time Consumption

The cleaning method consist of three parts:

- The filling of the 2-dimensional layer histogram, which is done by a single loop over all hits.
- A second loop over all hits, where
  - either all good hits are counted, if the z-position of the vertex is searched for,
  - or bad and good hits are flagged,
  - or bad hits are flagged and good hits are grouped.
- A third loop over all hits to reset the histogram to 0 as preparation for the next filling.

It follows that execution time is proportional to the number of hits to which the method is applied. For the case of cleaning and grouping it was measured to be 50 msec / 10000 hits on an AlphaServer 1000 model 4/200.

The third loop may be replaced - if faster - by a loop over all histogram bins in order to reset the histogram to 0.

When calculating $\theta = \arctan((Z - Z_{vertex})/\rho)$ the function arctan is calculated for each hit. In order to reduce time furthermore, the polar angle θ may be replaced by the "cylinder angle" ω, defined as:

- $\omega = -\rho/Z' - 2$, if $Z' < -\rho$,
- $\omega = -\rho/Z' + 2$, if $Z' > \rho$,
- $\omega = Z'/\rho$ else, with $Z' = Z - Z_{vertex}$.

By use of a ϕ/ω histogram, the cleaning process works as well.

7 Conclusions

For visual analysis the cleaning of hits based on a layer histogram, studied on a typical MC event, results in an efficient and very fast separation of good and bad hits, so that pictures of events cleaned in this way can be more easily interpreted. The hit grouping delivers furthermore groups of hits which are strongly correlated to the good hits of the simulated tracks. This further facilitates visual analysis of the results of simulation and pattern recognition etc.. It might be of interest to investigate the application of these methods to other fields of online and offline analysis, especially track following. The method can also be used to determine the Z-position of the primary vertex.

8 Acknowledgements

We aknowledge the help of A. Poppleton, from whom we got the data in an easily readable form.

References

[1] ' Event display: can we see what we want to see? '
H. Drevermann, D. Kuhn and B.S. Nilsson,
Proc. of 1995 CERN School of Computing, Arles, France, CERN Yellow Report 95-05.