

# Multiple variables data sets visualization in ROOT.

**O. Couet**

CERN EP/SFT, Geneva

E-mail: Olivier.Couet@cern.ch

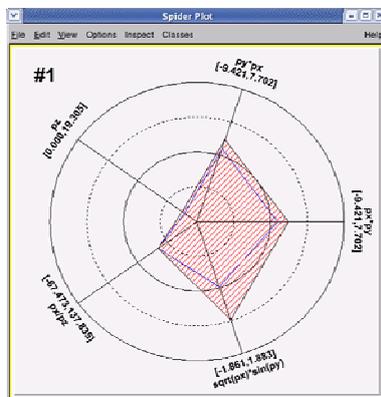
**Abstract.** The ROOT [1] graphical framework provides support for many different functions including basic graphics, high-level visualization techniques, output on files, 3D viewing etc. They use well-known world standards to render graphics on screen, to produce high-quality output files, and to generate images for Web publishing. Many techniques allow visualization of all the basic ROOT data types, but the graphical framework was still a bit weak in the visualization of multiple variables data sets. This paper presents latest developments done in the ROOT framework to visualize multiple variables (>4) data sets.

## Introduction

The ROOT's trees (**TTree** and **TNtuple** classes [5]) provide many functionalities to handle multiple variables data sets: statistical analysis, cuts, variable combinations etc ... but it was quite poor on the visualization side: only four variables maximum could be represented simultaneously on the same plot (a 3D scatter plot with the 4<sup>th</sup> variable mapped on a color palette). Several techniques exist to visualize many variables. Three of them have been recently implemented in the ROOT context: Spider plot, Parallel Coordinates plots and Box plots.

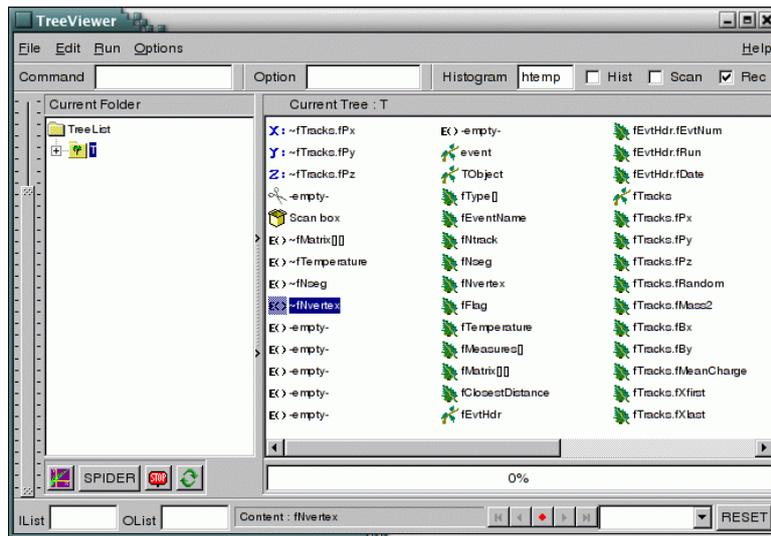
## Spider (Radar) Plots

Spider plots (sometimes called “web-plots” or “radar plots”, Fig. 1) are used to compare series of data points (events). They use the human ability to spot un-symmetry.



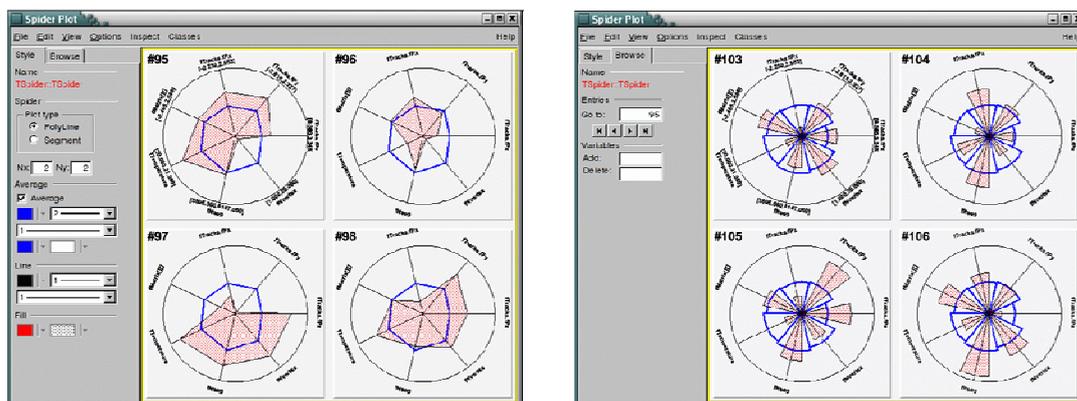
**Fig. 1** Example of spider plot.

Variables are represented on individual axes displayed along a circle. For each variable the minimum value sits on the circle's center, and the maximum on the circle's radius. Spider plots are not suitable for an accurate graph reading since, by their nature, it can be difficult to read out very detailed values, but they give quickly a global view of an event in order to compare it with the others. In ROOT the spider plot facility is accessed from the tree viewer GUI (Fig. 2). The variables to be visualized are selected in the tree viewer and can be scanned using the spider plot button.



**Fig. 2** The tree viewer Graphical User Interface.

The spider plot graphics editor [2] provides two tabs to interact with the spider plots' output: the tab "Style" defining the spider layout and the tab "Browse" to navigate in the tree.



**Fig. 3** The spider plots graphical editor.

### Parallel Coordinates Plots

The Parallel Coordinates Plots are a common way of studying and visualizing multiple variables data sets. They were proposed by in A.Inselberg in 1981 [3] as a new way to represent multi-dimensional information. In traditional Cartesian coordinates, axes are mutually perpendicular. In Parallel coordinates, all axes are parallel which allows representing data in much more than three dimensions. To show a set of points in Parallel Coordinates, a set of parallel lines is drawn, typically vertical and equally spaced. A point in n-dimensional space is represented as a polyline with vertices on the

parallel axes. The position of the vertex on the  $i$ -th axis corresponds to the  $i$ -th coordinate of the point. The three following figures show some very simple examples:

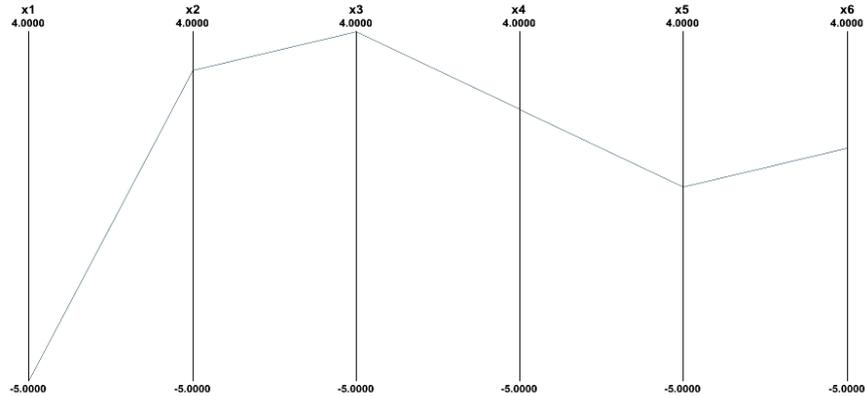


Fig. 4 The Parallel Coordinates representation of the six dimensional point  $(-5, 3, 4, 2, 0, 1)$ .

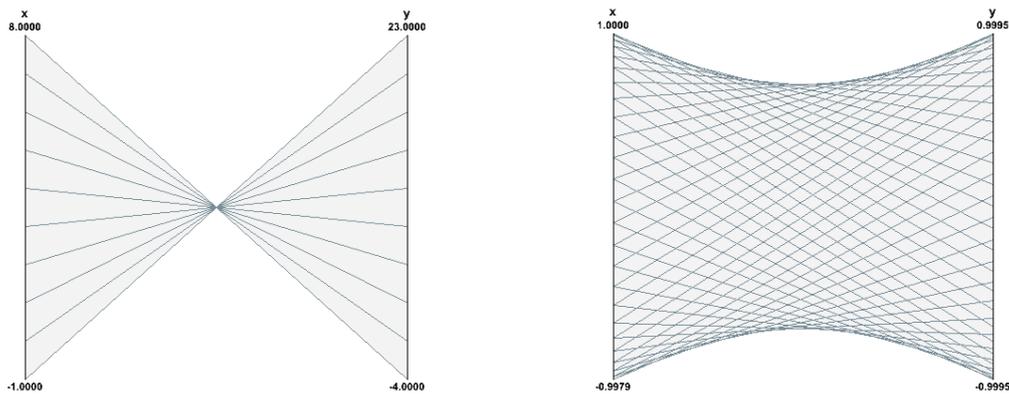


Fig. 5 The line  $y = -3x + 20$  and a circle in Parallel Coordinates.

The Parallel Coordinates technique is good at: spotting irregular events, seeing the data trend, finding correlations and clusters. Its main weakness is the cluttering of the output. Because each “point” in the multidimensional space is represented as a line, the output is very quickly opaque and therefore it is difficult to see the data clusters. Most of the work done about Parallel Coordinates is to find techniques to reduce the output’s cluttering. The Parallel Coordinates plots in ROOT have been implemented as a new plotting option “**PARA**” in the **TTree::Draw()** method. To demonstrate how the Parallel Coordinates works in ROOT we will use the tree produced by the following “pseudo C++” code:

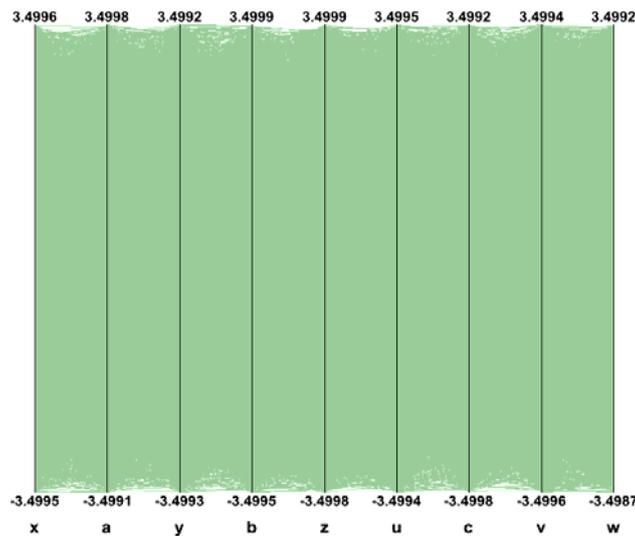
```
void parallel_example() {
    TTuple *nt = new TTuple("nt","Demo tuple","x:y:z:u:v:w:a:b:c");
    for (Int_t i=0; i<3000; i++) {
        nt->Fill(   rnd,   rnd,   rnd,   rnd,   rnd,   rnd, rnd, rnd, rnd );
        nt->Fill(  s1x,  s1y,  s1z,  s2x,  s2y,  s2z, rnd, rnd, rnd );
        nt->Fill(   rnd,   rnd,   rnd,   rnd,   rnd,   rnd, rnd, rnd, s3y, rnd );
        nt->Fill( s2x-1, s2y-1, s2z, s1x+.5, s1y+.5, s1z+.5, rnd, rnd, rnd );
        nt->Fill(   rnd,   rnd,   rnd,   rnd,   rnd,   rnd, rnd, rnd, rnd, rnd );
        nt->Fill( s1x+1, s1y+1, s1z+1, s3x-2, s3y-2, s3z-2, rnd, rnd, rnd );
        nt->Fill(   rnd,   rnd,   rnd,   rnd,   rnd,   rnd, rnd, s3x, rnd, s3z );
        nt->Fill(   rnd,   rnd,   rnd,   rnd,   rnd,   rnd, rnd, rnd, rnd, rnd );
    }
}
```

The data set generated has:

- 9 variables: **x, y, z, u, v, w, a, b, c**.
- $3000 \times 8 = 24000$  events.
- 3 sets of random points distributed on spheres: **s1, s2, s3**
- Random values (noise): **rnd**
- The variables **a, b, c** are almost completely random. The variables **a** and **c** are correlated via the 1st and 3rd coordinates of the 3rd “sphere” **s3**.

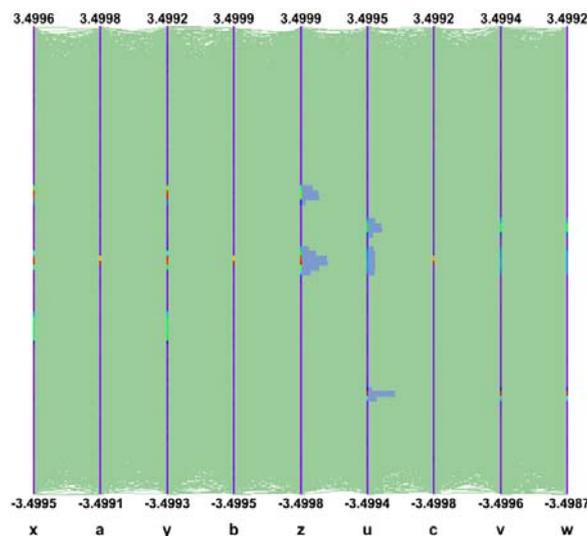
The command used to produce the Parallel Coordinates plot is:

```
nt->Draw("x:a:y:b:z:u:c:v:w", "", "PARA");
```



**Fig. 6** Cluttered output produced when all the tree events are plotted.

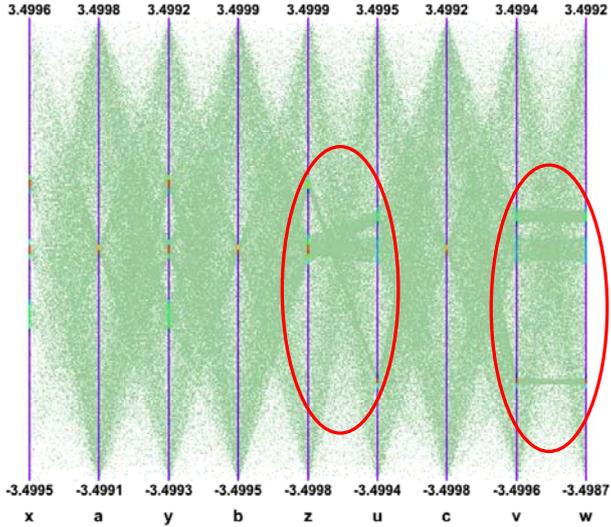
If the 24000 events are plotted as solid lines and no special techniques are used to clarify the picture, the result is the picture on Fig. 6 which is very cluttered and useless. To improve the readability of the Parallel Coordinates output and to explore interactively the data set, many techniques are available. We have already implemented a few in ROOT. First of all, in order to show better where the clusters on the various axes are, a 1D histogram is associated to each axis. These histograms (one per axis) are filled according to the number of lines passing through the bins.



**Fig. 7** The histogram's axis can be represented with colors or as bar charts.

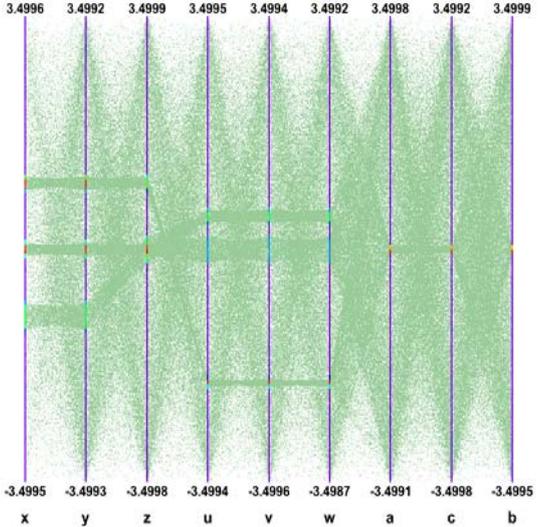
These histograms can be represented with colors (get from a palette according to the bin contents) or as bar charts. Both representations can be cumulated on the same plot (Fig. 7). This technique allows seeing clearly where the clusters are on an individual axis but it does not give any hints about the correlations between the axes.

We have implemented a very simple technique to make the clusters appearing. Instead of painting solid lines we paint dotted lines. The cluttering of each individual line is reduced and the clusters show clearly as we can see on the Fig. 8. The spacing between the dots is a parameter which can be adjusted in order to get the best results.



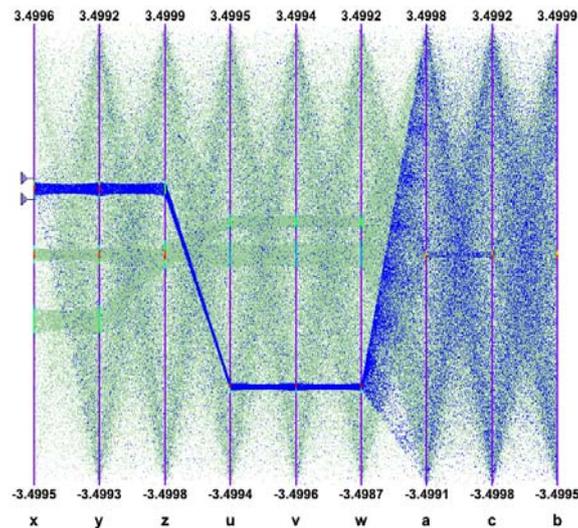
**Fig. 8** Using dotted lines is a very simple method to reduce the cluttering.

Interactivity is a very important aspect of the Parallel Coordinates plots. To really explore the data set it is essential to act directly with the events and the axes. For instance, changing the axes order may show clusters which were not visible in a different order. On Fig. 9 the axes order has been changed interactively. We can see that many more clusters appear and all the “random spheres” we put in the data set are now clearly visible. Having moved the variables  $u, v, w$  after the variables  $x, y, z$  the correlation between these two sets of variables is clear also.



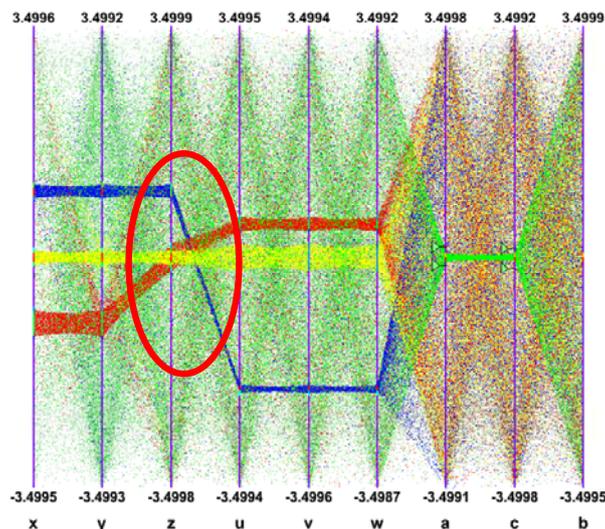
**Fig. 9** Axis order is very important to show clusters.

To pursue further data sets exploration we have implemented the possibility to define selections interactively. A selection is a set of ranges combined together. Within a selection, ranges along the same axis are combined with logical **OR**, and ranges on different axes with logical **AND**. A selection is displayed on top of the complete data set using its own color. Only the events fulfilling the selection criteria (ranges) are displayed. Ranges are defined interactively using cursors, like on the first axis of Fig. 10. Several selections can be defined at the same time, each selection having its own color.



**Fig. 10** Selections are set of ranges which can be defined interactively.

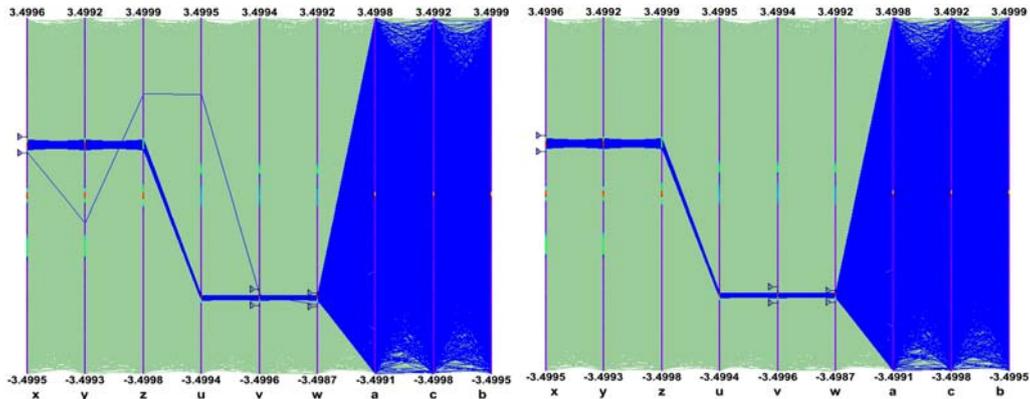
On Fig. 11 several selections have been defined. Each cluster is now clearly visible and the zone with crossing clusters is now understandable whereas, without any selection or with only a single one, it was not easy to understand.



**Fig. 11** Several selections can be defined each of them having its own color.

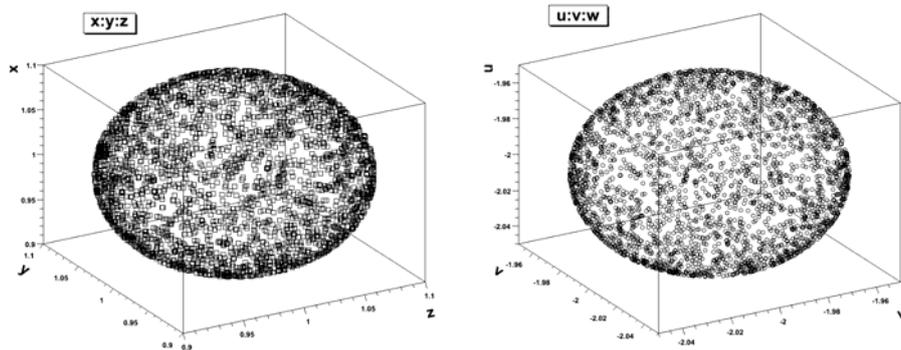
Interactive selections on Parallel Coordinates are a powerful tool because they can be defined graphically on many variables (graphical cuts in ROOT can be defined on two variables only) which allow a very accurate events filtering. As shown on Fig. 12, selections allow making precise events choices: a single outlying event is clearly visible when the lines are displayed as “solid” therefore it is

easy to make cuts in order to eliminate one single event from a selection. Such selection (to filter one single event) on a scatter plot would be much more difficult.



**Fig. 12** Selections allow to easily filter one single event.

Once a selection has been defined, it is possible to use it to generate a **TEntryList** which is applied on the tree and used at drawing time. In our example the selection we defined allows to select exactly the two correlated “random spheres” as shown on the Fig. 13.



**Fig. 13** Output of `nt->Draw("x:y:z")` and `nt->Draw("u:v:w")` after applying the selection.

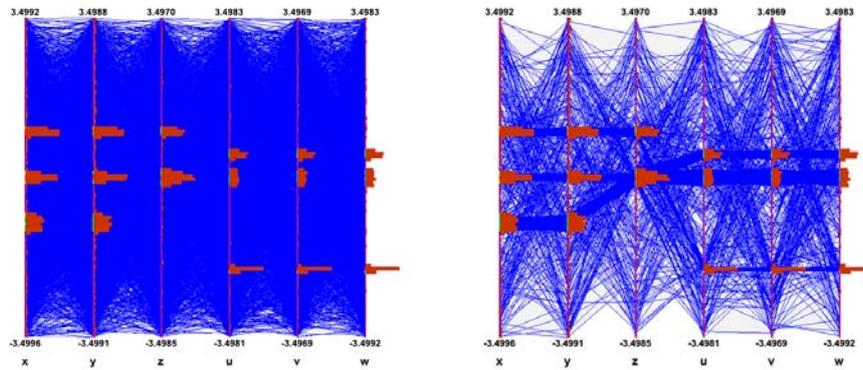
Another technique has been implemented in order to show clusters when the picture is cluttered. A weight is assigned to each event. The weight value is computed as:

$$weight = \sum_{i=1}^n b_i$$

Where:

- $b_i$  is the content of bin crossed by the event on the  $i$ -th axis.
- $n$  is the number of axis.

The events having the bigger weights are those belonging to clusters. It is possible to paint only the events having a weight above a given value and the clusters appear. On Fig. 14 the “weight cut” applied on the right plot is 50. Only the events with a weight greater than 50 are displayed.



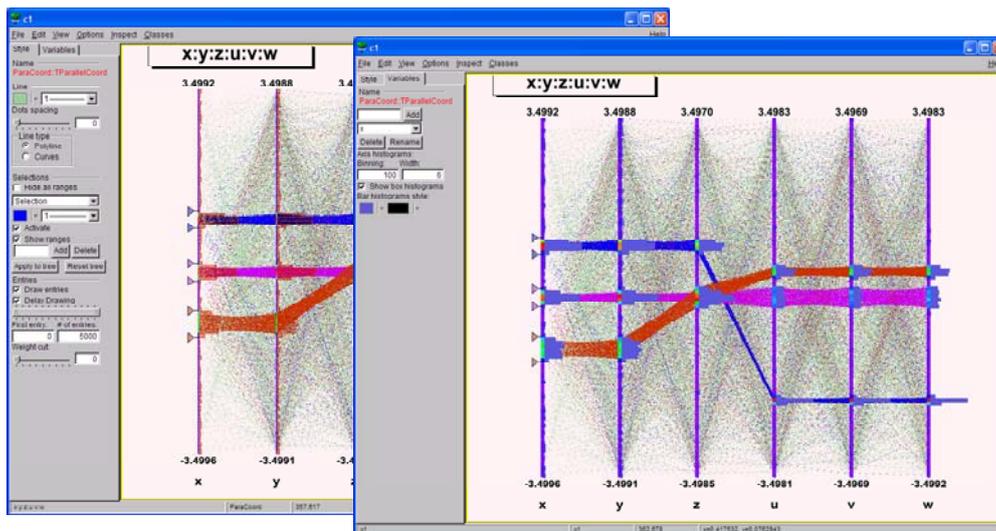
**Fig. 14** Applying a “weight cut” makes the clusters visible.

In case only a few events are displayed, drawing them as smooth curves instead of straight lines helps to differentiate them as shown on Fig. 15.



**Fig. 15** Zoom on a Parallel Coordinates plot detail: curves differentiate better events.

Interactivity and therefore the Graphical User Interface are very important to manipulate the Parallel Coordinates plots. The ROOT framework allows to easily implement the direct interactions on the graphical area and the graphical editor facility [2] provides dedicated GUI.



**Fig. 16** Parallel Coordinates graphical editors.

### Box (Candle) Plots

A Box Plot (also known as a “box-and whisker” plot or “candle stick” plot) is a convenient way to describe graphically a data distribution ( $D$ ) with only the five numbers (Fig. 17). It was invented in 1977 by John Tukey [4]. The five numbers are:

1. The minimum value of the distribution  $D$  (Min).
2. The lower quartile (Q1): 25% of the data points in  $D$  are less than Q1.
3. The median (M): 50% of the data points in  $D$  are less than M.
4. The upper quartile (Q3): 75% of the data points in  $D$  are less than Q3.
5. The maximum value of the distribution  $D$  (Max).

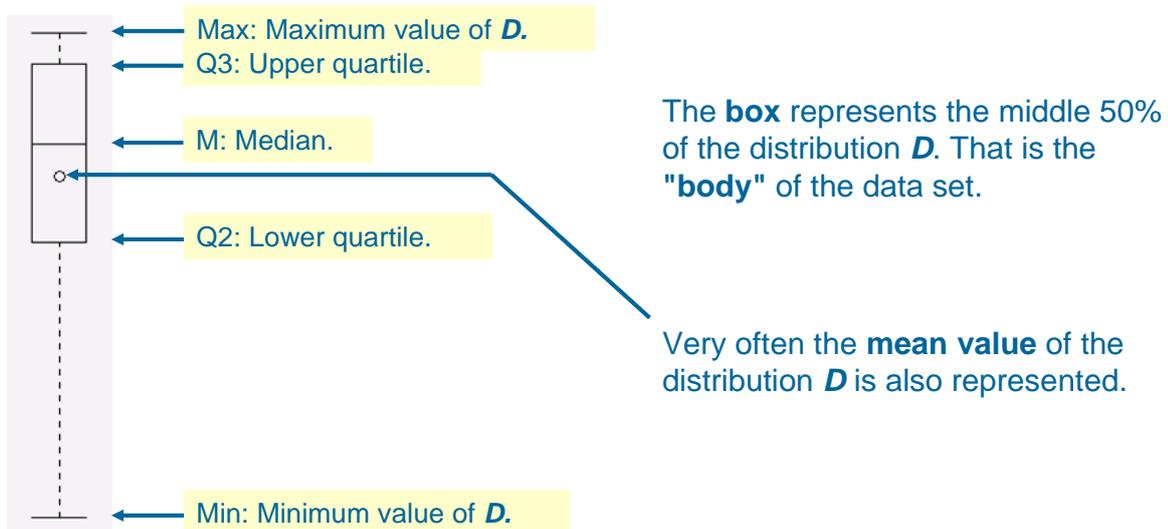


Fig. 17 A box plot describes a distribution with only five numbers.

In ROOT Box Plots (Candle Plots) can be produced from a **TTree** using the “**candle**” option in **TTree::Draw()**. See Fig. 18.

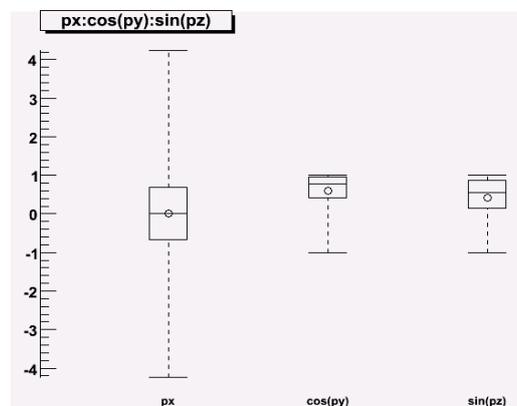


Fig. 18 `tree->Draw("px:cos(py):sin(pz)","","candle");`

It is possible to combine Parallel Coordinates and Candle-Plots as shown on Fig. 19.

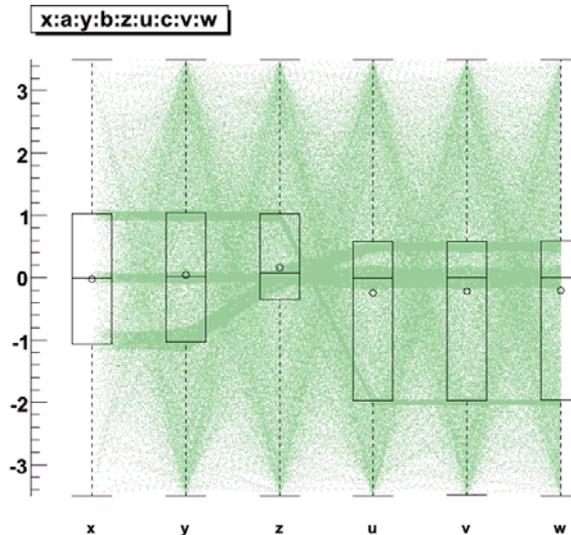


Fig. 19 Candles plots and Parallel Coordinates plots combined.

## Conclusion

The multivariable's visualization techniques introduced in ROOT are very promising. In particular the Parallel Coordinates show the clusters over many variables which allows to make precise selections. Nevertheless, the cluttering of the output is still a problem to be fight. Several techniques are available, like the transparency and shading. We should explore these tracks in the future. Right now we have implemented simple techniques which give good results already. Also it would be very interesting to have tools to find the clusters automatically or to sort the axes automatically. Finally other techniques than Parallel Coordinates to represent multiple variables data sets should be investigated.

## References

- [1] ROOT Web site: <http://root.cern.ch>
- [2] The graphics editor in ROOT, I. Antcheva, R. Brun, C. Hof, F.Rademakers, Nuclear Instruments and Methods in Physics Research, Volume 559, Issue 1, 1 April 2006, Pages 17-21, Proceedings of ACAT05
- [3] A. Inselberg. The plane with parallel coordinates. Special Issue on Computational Geometry, The Visual Computer, 1:69–97, 1985.
- [4] John W. Tukey. "Exploratory Data Analysis". Addison-Wesley, Reading, MA. 1977.
- [5] ROOT Users Guide 5.14 – Trees chapter page 183.