STAR-Note # 89
S. Margetis and D. Cebra
December 1992

Main Vertex reconstruction in STAR

1 Introduction

A Least Squares Method (L.S.M) combined with a simple technique of outlier removal, has been used for main vertex fitting in STAR. The first section of this document briefly describes the method and the following section contains information on where to find and how to use the various routines. In the last section we present and discuss results from simulations on the expected vertex resolution in STAR with or without the SVT.

2 The method

After the track reconstruction programs finish their task there is a pool of tracks described by a set of helix parameters, usually evaluated at the first reconstructed point of the track. There will be at least two sets of tracks, one containing the tracks reconstructed in the TPC and another with tracks reconstructed in the SVT. A matcher might then create a third set with the combined information of both sets.

The next step is to reconstruct the main vertex for the event. The accurate determination of the main vertex is essential for two reasons. First, it determines the interaction point and we can re-calculate the momentum vector for each primary track at the vertex with the vertex point included in the fit. Second the ability to discriminate between primary and secondary tracks depends on it. We describe below a LSM method which was used to extract first estimates on main vertex resolution in STAR.

The first thing one does in this reconstruction process is to project the helices to a reference point which is close to an estimated vertex. We can only reference a helix to either a point in the transverse plane or along the beam axis (the beam is the z-axis). At RHIC this reference point is in the transverse plane because the knowledge of the interaction point in the direction transverse to the beam plane is planed to be much better than along the beam axis ($\sigma_{xy} \approx 0.5$mm and $\sigma_z \approx 22$cm, see [1]). The natural choice of the reference point would be the center of the beam profile which in 3-D is the central axis of the beam diamond. In the transverse plane, the real vertex can only be a few millimeters away from the central axis. For distances of this order of magnitude we can approximate the helices as straight lines:

$$H_i \Longrightarrow T_i + n\hat{e}_i$$

where $T_i = (x_i, y_i, z_i)$ is the point of closest approach to the center of the beam and $\hat{e}_i = \dot{x}_i\hat{x}+\dot{y}_i\hat{y}+\dot{z}_i\hat{z}$ is the unit vector in the momentum direction ($\dot{x} = dx/du = p_x/|\vec{p}|$). This linearization of the track model in the vertex neighbourhood has the advantage of leading to an analytical, and therefore fast, solution. If $V = (x_0, y_0, z_0)$ is the requested vertex point, the distance of the vertex from track i is the cross product of the vector from T_i to V and the unit vector $\hat{e}_i$ (see also fig. 1).

$$
\begin{array}{rclcl}
d_i & = & |\,\overrightarrow{T_iV} \times \hat{e}_i| & = \{ & [(y_i - y_0)\dot{z}_i - (z_i - z_0)\dot{y}_i]^2 \\
 & & & + & [(z_i - z_0)\dot{x}_i - (x_i - x_0)\dot{z}_i]^2 \\
 & & & + & [(x_i - x_0)\dot{y}_i - (y_i - y_0)\dot{x}_i]^2 \quad \}^{1/2}
\end{array}
$$

Let us define a merit function:

$$\chi^2 = \sum_{i=1}^{n} \frac{d_i^2}{\sigma_i^2}$$

For the current derivation we put $\sigma_i = 1$. In our LSM approach we will minimize this sum. The partial derivatives then are:

$$\frac{\partial \chi^2}{\partial x_0} = 2\sum_i \{(z_i - z_0)\dot{x}_i\dot{z}_i + (y_i - y_0)\dot{x}_i\dot{y}_i - (x_i - x_0)(\dot{y}_i^2 + \dot{z}_i^2)\}$$

$$\vdots$$

We set them to zero and we now have 3 equations and 3 unknowns:

$$x_0 \sum_i (\dot{y}_i^2 + \dot{z}_i^2) - y_0 \sum_i \dot{y}_i\dot{x}_i - z_0 \sum_i \dot{z}_i\dot{x}_i = \sum_i \{x_i(\dot{y}_i^2 + \dot{z}_i^2) - y_i\dot{y}_i\dot{x}_i - z_i\dot{z}_i\dot{x}_i\}$$

$$\vdots$$

We can write these equations in matrix notation by defining:

$$M_i = \begin{pmatrix} \dot{y}_i^2 + \dot{z}_i^2 & -\dot{x}_i\dot{y}_i & -\dot{x}_i\dot{z}_i \\ -\dot{x}_i\dot{y}_i & \dot{x}_i^2 + \dot{z}_i^2 & -\dot{y}_i\dot{z}_i \\ -\dot{x}_i\dot{z}_i & -\dot{y}_i\dot{z}_i & \dot{x}_i^2 + \dot{y}_i^2 \end{pmatrix}$$

and our equations take the form:

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \begin{pmatrix} \sum M_{11} & \sum M_{12} & \sum M_{13} \\ \sum M_{21} & \sum M_{22} & \sum M_{23} \\ \sum M_{31} & \sum M_{32} & \sum M_{33} \end{pmatrix} = \sum_i \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} M_i$$

By solving this system of equations using matrix manipulation we obtain the vertex position. In our choice of Helix parameters [2] we get:

$$
\begin{array}{rclcl}
\dot{x} & = & p_x/|\vec{p}| & = & cos\lambda\, cos\phi \\
\dot{y} & = & p_y/|\vec{p}| & = & cos\lambda\, sin\phi \\
\dot{z} & = & p_z/|\vec{p}| & = & sin\lambda
\end{array}
$$

and after substitution:

$$M_i = \begin{pmatrix} sin^2\lambda_i + cos^2\lambda_i sin^2\phi_i & -cos^2\lambda_i cos\phi_i sin\phi_i & -cos\phi_i cos\lambda_i sin\lambda_i \\ -cos^2\lambda_i cos\phi_i sin\phi_i & sin^2\lambda_i + cos^2\lambda_i cos^2\phi_i & -sin\phi_i cos\lambda_i sin\lambda_i \\ -cos\phi_i cos\lambda_i sin\lambda_i & -sin\phi_i cos\lambda_i sin\lambda_i & cos^2\lambda_i \end{pmatrix}$$

In the above discussion we put $\sigma_i = 1$ (the errors on d_i). This is because σ_i is a rather complicated function of track momentum (multiple scattering) and fitted track parameters (dp/p etc.) [3]. On the other hand, it is known that including tracks in the sample which are at large distances from the central cluster (outliers) has a strong influence on the results. This might be tracks emerging from weak decays or low momentum tracks which spiral in the detector [4].

We use a simple 'truncation' method in an iterative way to remove the outliers and therefore improve the 'robustness' of the fit[1]. In this method we first use all reconstructed tracks to get an estimate of the vertex parameters. Then we use that as a seed to the next iteration and we remove from the track pool all tracks with distances greater than a certain cut value. The new vertex is then used as a seed to the next iteration. The whole process converges after 3–4 iterations. The cut values of the last iteration were estimated by looking at the (simulated) distributions of track impact parameters [4].

We found that in some rare cases the presence of outliers at large distances from the vertex (e.g. > 20 cm) could pull the estimated vertex during the first (global) pass away from the true point. In this case, if the cut values for the first iteration are not large enough, the program might fail finding the real vertex. In order to avoid this, one should either use relatively large cut values in the first iteration or try to get a rough estimate of the vertex position. In this latter case using a simple (i.e. fast) sorting of the tracks along the beam (z) direction and then find the maximum of the distribution will return an estimated ("seed") vertex position with an accuracy of about the size of the bin used (0.5 cm in our application).

This method of vertex fitting should be used in relatively high multiplicity events (such as Au+Au events) where, even after the removal of a large fraction of tracks, there are still enough tracks left for the fit. For low multiplicity events (less than 20 tracks) another method might be used instead (see pages 253 and 254 in ref. [3]).

3 The programs

The fitting program is a modified version of a routine used in the EOS experiment [5]. The package resides on rsgi00 at BNL in the directory */u2/margetis/svt/vertex*. The name of the executable is *main_vertex_fit* [2] and it is produced by the make file *Makefile.main_vertex* . There is one parameter file which is supposed to be edited

[1] An estimator is called robust if it is insensitive to measurements which deviate from the expected behaviour (outliers).

[2] In the same directory the files with the word *new* in their name are basically the same function, the only difference is that a "seed" vertex is calculated for the first iteration using the track sorting method described above.

by the user before running the program. This is the file *main_vertex_fit.inc* which is listed below:

```
*
*              CUT VALUES FOR OUTLIERS IN MAIN VERTEX FIT
*
* cutxy is smaller than cutz because of the better beam definition in
* the transverse dimension. Real values in cm.
*----------------------------VALUE IN USE        NOMINAL VALUES
*                                               SVT+TPC      TPC

        DATA    cutxy           /1.0/           !1.0         2.0
        DATA    cutz            /10.0/          !10.0       15.0
        DATA    cut2            /2.0/           !2.0         4.0
        DATA    cut3            /0.5/           !0.5         1.0
*
*--- smear the vertex locally (true) or it was done in GEANT (false)--
*
        smear=.false.
*
*--- svt_hits >0  use only tracks which have SVT hits and fit was good
*--- svt_hits =0  same as above plus TPC tracks without SVT hits.
*--- svt_hits <0  same as =0 plus TPC tracks with SVT hits and bad fit.
*--- It should be negative when the SVT is not in the simulation.
*--- integer values
*
        svt_hits=1
*----------------------------------------------------------------------
```

where:

cutxy,cutz	are the cuts for the first iteration
cut2,cut3	are the cuts for the 2^{nd} and 3^{rd} iteration
smear	is .true. for internal smearing of the vertex point
svt_hits	is a cut on the track quality.

The script file *main_vertex_fit.com* runs the program which generates the binary output file *fort.9* and the log file *main_vertex.out*. In the log file the user can look at the results as they come out after each iteration. The binary file is put in the directory */u2/margetis/svt/data* after being renamed (e.g. 5auau_str_vertex.out). In the same directory the user can find sample input data files.

The user can find in the directory */u2/margetis/svt/ana* the sample routine *vertex.f* which reads the binary output file into an PAW–ntuple. There is a script file *run_vertex.com* to run the executable as well as a *Make.file* (self documented) which builds it.

4 Results and discussion

We ran the program on a set of 15 central Au+Au events. We studied two cases, one with the SVT points not used in the calculation, and one with the complete SVT+TPC tracking information. The matching (and refit) of the SVT points with the tracks reconstructed in the TPC was done using GEANT information, which corresponds to 100% matching efficiency. A track quality cut (in the TPC+SVT case) was rejecting tracks with no SVT hits (e.g. secondaries) and tracks with SVT hits but bad re-fit (e.g. tracks changing direction because of multiple scattering). The assumed position resolution of the SVT was 50 microns.

In fig. 2 and 3 the distribution of the residuals for 15 central Au+Au events is plotted separately for each coordinate[3]. Table 1 summarizes the results of the fit.

RMS (μm)	x	y	z
TPC	330	220	250
TPC+SVT	40	40	35

We see that the resulting distributions have an RMS in the range 250–350 microns for the TPC and 30–40 microns for the TPC+SVT which reflects the accuracy of the fit. This accuracy of defining the primary vertex is a result of two factors, the position resolution of the corresponding detectors and the number of tracks (degrees of freedom) used in the fit, which in this case is around one thousand. We observe that in this case the vertex was reconstructed with better accuracy than the assumed position resolution of the detectors ($\approx 700\mu$m for the TPC and 50μm for the SVT). This, combined with the increase of the overall track length, makes the reconstructed vertex a valuable point in the track fitting process, improving the overall momentum resolution[4]. Especially without the SVT in the experiment it makes a rather dramatic improvement ($dp/p = 9.6$ % without, and 2.2 % with the vertex included in the fit) [6].

We also observe that the error on the estimated position is much smaller than the impact parameter of an average track, which is a few millimeters for the TPC and few hundred microns for the TPC+SVT [4]. This means that track impact parameter information can be used in order to discriminate vertex from non–vertex tracks, an invaluable tool in the search for secondary vertices.

As a conclusion we can make the statement that main vertex fitting in STAR's heavy ion program is a relatively easy and straightforward task.

5 Acknowledgement

The authors would like to thank M. Bloomer, W. Christie, P. Jacobs, P.G. Jones, G. Odyniec, I. Sakrejda and L.P. Teitelbaum for their comments and suggestions.

[3]There is a factor of ten difference in the horizontal scale of the two sets of histograms.

[4]The momentum resolution scales as $dp/p \propto (p \cdot \sigma_i)/\sqrt{l^5}$, where σ_i is the error on the space points used in the fit and l is the track length [7].

References

[1] Conceptual Design of RHIC, BNL 52195, May 1989

[2] Margetis S.;"Vertex programs documentation"; SVT-note, Nov. 1992

[3] R. Bock, H. Grote, D. Notz and M. Regler, (ed. by M. Regler). *Data Analysis Techniques in High Energy Physics Experiments*. Cambridge Univ. Press, Cambridge (UK, 1990)240.

[4] STAR – Conceptual Design Report, PUB–5347(1992)4C-44 and 4H-13

[5] D. Cebra and the EOS Collab., LBL-30798(1991)191

[6] H. Wieman et al.; STAR Collab. meeting, BNL September 1992

[7] CERN Yellow report #26(1967)25

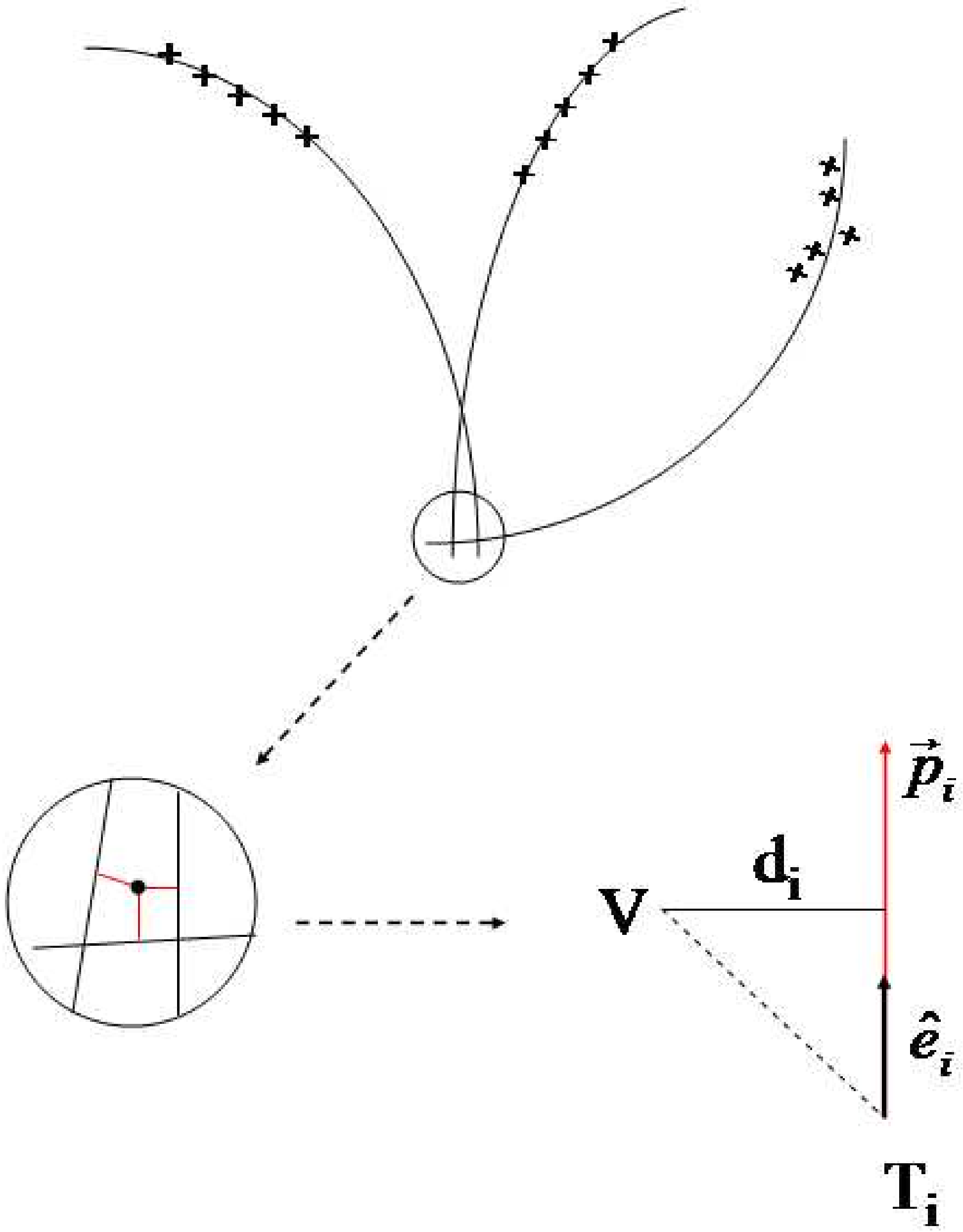

Figure 1: Concept of minimization of distances of closest approach.

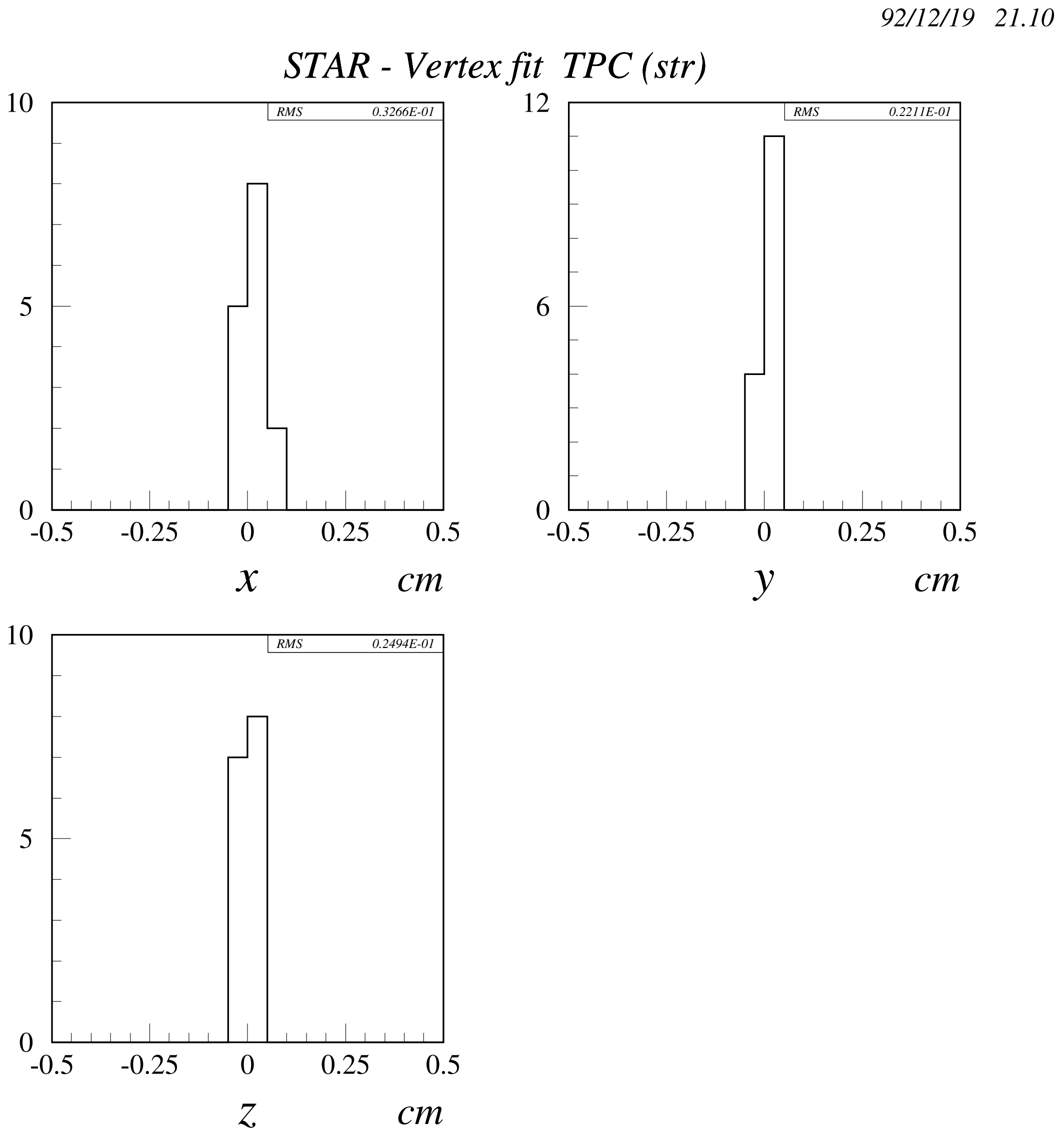

Figure 2: Vertex resolution with the STAR-TPC. The difference between the true and reconstructed value is plotted for each coordinate for 15 Au+Au events. The TPC design used in the simulation has straight pad–rows with both high and low resolution pads.

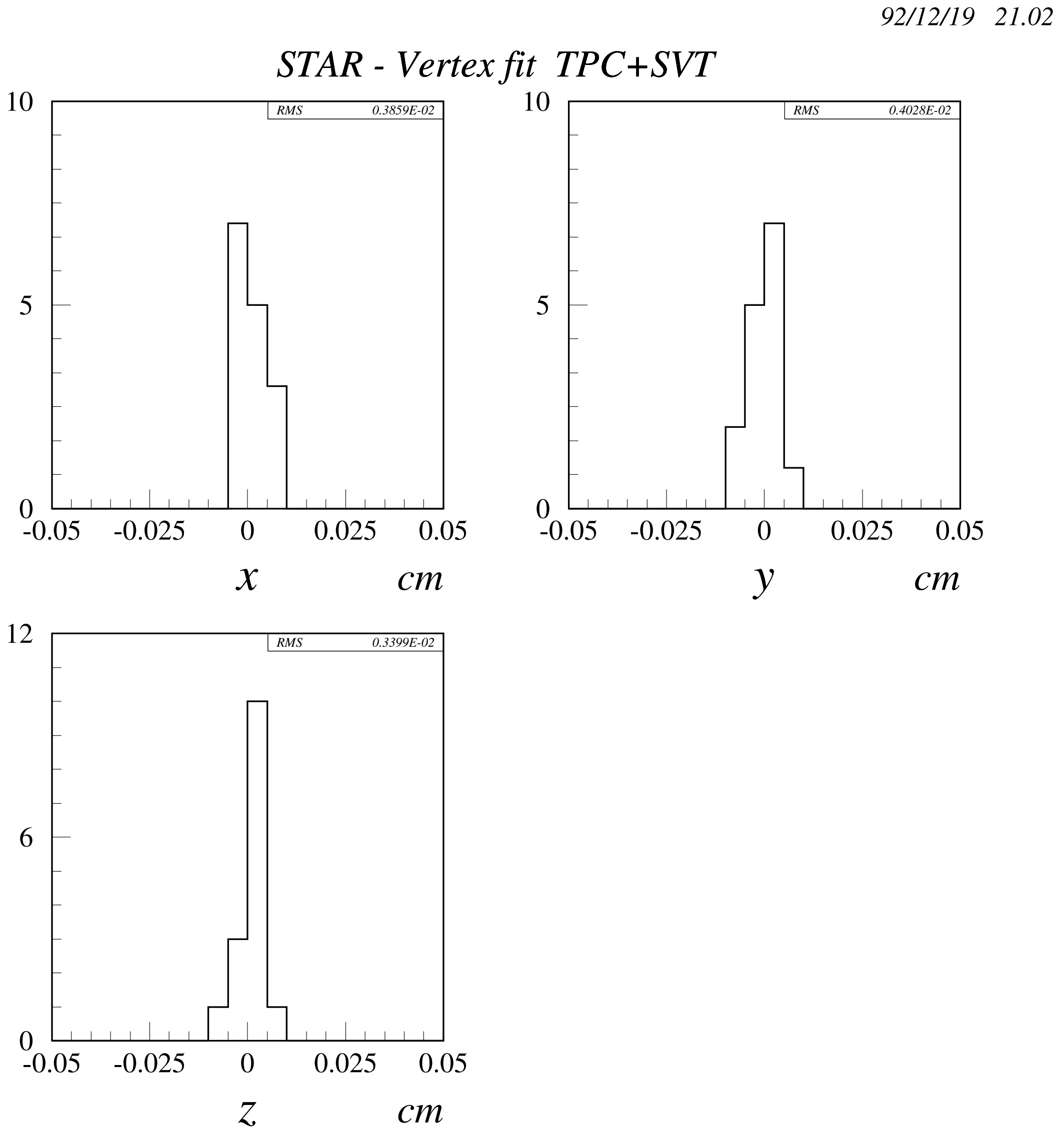

Figure 3: Vertex resolution with the STAR-TPC+SVT. Notice the factor of ten difference in the scale from the corresponding TPC plots.