СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

E10-98-278

O.Yu.Barannikova, G.A.Ososkov, Yu.A.Panebratsev

INVESTIGATION OF FAST ALGORITHMS FOR THE TRACK RECONSTRUCTION AND VERTEX FINDING USING NEW COMPUTATION MODEL WITH REAL **STAR-SVT** GEOMETRY

1998

Introduction

In the previous note the method of SDD alignment (position reconstruction) within SVT system and fast algorithms for track and vertex reconstruction were described [1]. The precision of the proposed algorithms was investigated using a simplified Monte Carlo model. Here we analyse the possibility of using these algorithms for processing real experimental data a computational model with the geometry maximally close to that of the real STAR-SVT was developed. SVT coordinate system and labeling scheme is chosen as it was proposed by R.Bellwied [2]. We studied the accuracy and the resistance to the data contamination of the proposed methods.

1 Computational model

Following [2] we consider the SVT structure as consisting of three barrels. The innermost barrel is barrel 1, the middle one is barrel 2, the outermost barrel is barrel 3.

The barrels consist of 8, 12, 16 ladders respectively. They are labeled clockwise, in the same way as in [2]. The ladders contain 4, 6, 7 wafers respectively. The detectors are labeled from West to East.

In the model sizes of wafers, distances between their centers in the ladders and radial distances between the detector center and ladders of each of the 3 barrels are taken to be equal to those of the real detector.

The global coordinate system (X_g,Y_g,Z_g) is chosen in the following way. The coordinates origin is chosen to be right in the detector center. X_g axis was taken to go along the beam line from West to East (so that if viewed in this direction ladders were labeled clockwise and wafer label numbers increased together with X_g. Y_g axis was taken to be vertical directed upwards and Z_g axis - horizontal from left to right (if viewed from West). Besides, for each wafer a 2D-coordinate system is chosen. The local coordinate center was in the wafer center, the direction of X axis is coincided with that of the X_g axis of the global coordinate system and Y axis is perpendicular to it in the wafer plane.

When simulating an event, at first, the vertex coordinates (in global coordinate system) are randomly chosen inside the beam pipe. After that a wafer on the 3-rd barrel is calculated by the random choice of the label number for a ladder and a wafer. The local coordinates of the point are determined by two uniform random numbers. After that ladders and wafers of the 2-nd and 1-st, barrel which are crossed by the track drawn through the simulated vertex and the point chosen in the 3-rd barrel, are determined. For this purpose we recalculate the coordinates of the chosen point from the local coordinate system to the global one.

In the Y_gZ_g plane

$$Y_{g1} = R\cos\Theta + Y_1\sin\Theta \qquad Z_{g1} = R\sin\Theta - Y_1\cos\Theta$$

where R is the radial distance between wafer and detector center, Θ is the angle between R and Y_g axis (see Fig.1a)

$$\Theta = \begin{cases} k\pi/4, & for\ 1-st\ barrel \\ k\pi/6, & for\ 2-nd\ barrel \\ k\pi/8, & for\ 3-rd\ barrel, \end{cases}$$

where k is the ladder number.

In the X_gZ_g plane (see Fig.1b):

$$X_{1g} = X_1 + (n-1)STEP - c$$

where STEP is the distance between neighouring wafer centers in the ladder, n is the wafer label number, c is the constant, calculated on the following way:

$$c = \begin{cases} 3STEP/2, & for\ 1-st\ barrel \\ 5STEP/2, & for\ 2-nd\ barrel \\ 3STEP, & for\ 3-rd\ barrel \end{cases}$$

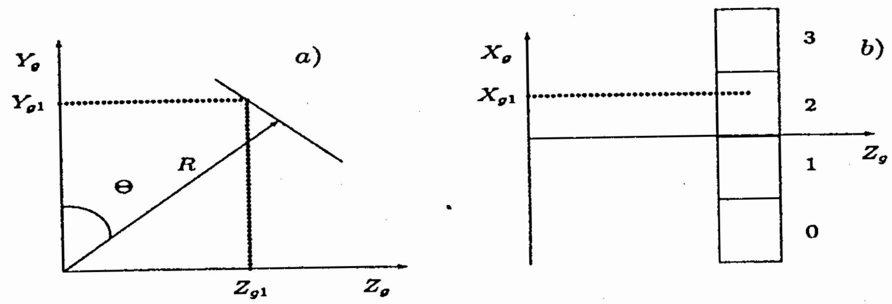

Figure 1: Coordinate recalculation from the local coordinate system to the global one

Then after defining the ladder label number in the 2-nd barrel (using the angle value ϕ), we recalculate the coordinates of the chosen point from the global coordinate system to the one connected with this ladder (see Fig.2).

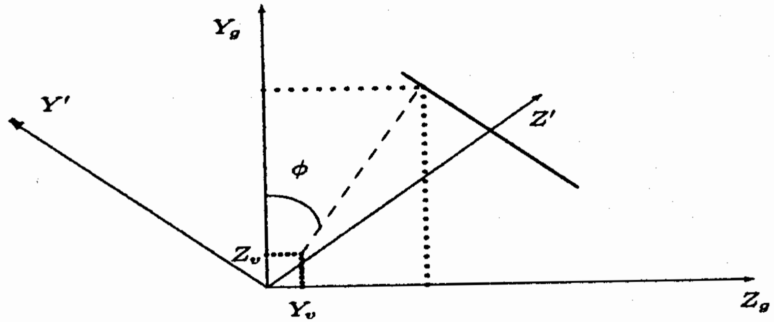

Figure 2: The coordinate system connected with some ladder [1]

In this coordinate system we have

$$Y_1' = Y_1\sin\Theta - Z_1\cos\Theta \qquad Z_1' = Y_1\cos\Theta - Z_1\sin\Theta$$

and for the vertex coordinates one obtains

$$Y_v' = Y_v\sin\Theta - Z_v\cos\Theta \qquad Z_v' = Y_v\cos\Theta - Z_v\sin\Theta$$

In this system we can define one of the local coordinates of the intersection point on the 2-nd barrel:

$$Y_2 = Y_v' + \frac{(Y_1' - Y_v')(R - Z_v')}{Z_1' - Z_v'}$$

and using the 2-nd coordinate in the global system

$$X_{g2} = X_{gv} + \frac{(X_{g1} - X_{gv})(R - Z_v')}{Z_1' - Z_v'}$$

it is easy to obtain its value in the local coordinate system. For example for the 5-th wafer it is:

$$X_2 = X_{g2} - 3STEP/2$$

etc. Then in the same way we obtain the coordinates of the track intersection point with the 3-rd barrel.

To simulate an uncertainty of the experimental data a normally distributed random distortion is added to the local coordinates of points, in which the simulated tracks crossed the detectors. The mean square deviation (distribution width) σ of this displacements is taken to be 20 μm. Besides, the Coulomb scattering is simulated for the 2-nd and 3-rd barrels. For this purpose the slope of the line simulating the track is changed by some small angle θ after it crosses the wafers of the 1-st barrel. The value of this angle is determined by a normally distributed random number with some σ_θ [3]. The analysis of the vertex reconstruction precision is done with $\sigma_\theta = 1$ mrad. The azimuth direction of scattering (in xy plane) is determined by some angle ϕ uniformly distributed in the interval $(0, 2\pi)$. The procedure is repeated in oder to get sufficiently representative statistics (1000 tracks for our model). Then a number of uniformly distributed points is added to each detector in oder to simulate a background measurements. We study the precision of vertex reconstruction with 30% background level. The output file of each event contains the local coordinates of all simulated points for each of 216 wafers. The structure of data presentation in the output file closely resembles that obtained from the real experiments.

2 Vertex reconstruction

The methods of track and vertex reconstruction described in [1] was used to test the proposed model. To accomplish this the track parameterization must be changed, since it is impossible to choose a coordinate system, which would allow to parameterize all the possible tracks with the help of two projection slopes without loosing some tracks being parallel to some of the axes. The new parameterization is chosen in the following way. Point $P_i(X_{ig}, Y_{ig}, Z_{ig})$ belonging to the track is defined as follows

$$\vec{P_i} = \vec{a} + \lambda_i \vec{b} \quad (1)$$

where $\vec{P_i}$ is the i-th point on the track, $\vec{b}$ is the vector, determining track direction in space, $\vec{a}$ is the vector defining the minimal distance between the beam axis and the track in a plane orthogonal to the beam axis (by definition, one has $(\vec{b}\cdot\vec{a})|_{Y_g Z_g} = 0$), and λ is a parameter that determines the position of any point along the track (see Fig.3). The above parameterization allows a quick, non-iterative, determination of the track parameters.

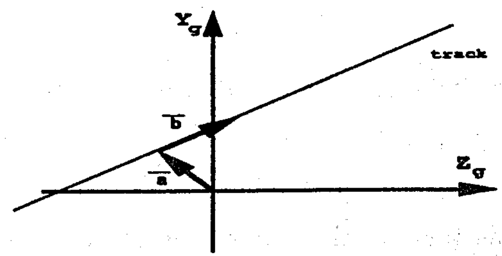

Figure 3: Track parameterization

The track reconstruction is done in the way described in the previous note [1]. For every point of the 3-rd barrel (P_3) a line is drawn passing through P_3 and P_1, where P_1 is such a point that the line drawn through P_3 and P_1 didn't pass farther then the determined distance ($Vertex_W$) from the beam axis in $Y_g Z_g$

plane and its projection in X_gZ_g plane intersects X_g axis within the interval ($-Vertex_L$, $+Vertex_L$) (where $Vertex_L$=3cm and $Vertex_W$=1mm are constants defining the region of possible vertex location). For each found line the ladder and wafer numbers and local coordinates of the point, where this line crossed the 2-nd barrel, are defined in the above mentioned way. If there is a point P_2 on the defined wafer in the vicinity of the found point, then these three points (P_1, P_2, P_3) are assumed to belong to the same track. Then the track parameters are determined with the help of the least square method.

While optimizing the track using 3 defined points the main problem is to choose the functional, which is quadratic in respect all track parameters. That means, it is possible to find them at once, without iterations. For this purpose a preliminary vector $\vec{b}_o = \vec{P}_1 - \vec{P}_2$, where $\vec{P}_1$ and $\vec{P}_2$ are radius-vectors of points P_1 and P_2 respectively, is formed in the Y_gZ_g plane (transverse to the beam axis).

We calculate for each point P parameter λ_i as:

$$\lambda_i = \frac{(\vec{p}_i \cdot \vec{b}_o)_{Y_gZ_g}}{(\vec{b}_o \cdot \vec{b}_o)_{Y_gZ_g}} \tag{2}$$

then the following functional is to be minimized:

$$L = \sum_{i=1}^{3} \left(\vec{a} + \lambda_i \vec{b} - \vec{P}_i \right)^2 \tag{3}$$

We choose this functional for our minimization because its minimizing would lead to a system of the simple linear equations of the track parameters, which allows one to determine these parameters by a simple one step procedure without any time consuming iterations.

Given an estimate $\vec{b}_o$, one finds a corresponding estimate for $\vec{a}$ with:

$$\vec{a}_o = \vec{P}_1 - \lambda_1 \vec{b}_o. \tag{4}$$

Better estimates are sought by rewriting the functional L as a function of adjusted vectors $\vec{a}$ and $\vec{b}$ defined as follows:

$$\vec{a} = \alpha \vec{a}_o \tag{5}$$
$$\vec{b} = \vec{b}_o + \beta \vec{a}. \tag{6}$$

Derivatives of L with respect to α and β must be zero.

$$\frac{\partial L}{\partial \alpha} = 2 \sum_{i=1}^{3} \left(\alpha \vec{a}_o + \lambda_i (\vec{b}_o + \beta \vec{a}_o) - \vec{P}_i \right) \cdot \vec{a}_o \;=\; 0 \tag{7}$$
$$\frac{\partial L}{\partial \beta} = 2 \sum_{i=1}^{3} \left(\alpha \vec{a}_o + \lambda_i (\vec{b}_o + \beta \vec{a}_o) - \vec{P}_i \right) \cdot \lambda_i \vec{a}_o \;=\; 0. \tag{8}$$

That yields the solution :

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{a_o^2 (3 \sum \lambda_i^2 - \sum \lambda_i)} \begin{pmatrix} \sum \lambda_i^2 & -\sum \lambda_i \\ -\sum \lambda_i & 3 \end{pmatrix} \begin{pmatrix} \sum \vec{P}_i \cdot \vec{a}_o \\ \sum \lambda_i \vec{P}_i \cdot \vec{a}_o \end{pmatrix} \tag{9}$$

Having minimized L in the Y_gZ_g plane, one then consider the x-projection component of the L functional to be so far neglected.

$$L_x = \sum (a_x + \lambda_i b_x - \vec{i} \cdot \vec{P}_i)^2 \tag{10}$$

This function can be minimized by zeroing derivatives with respect to a_x and b_x. One gets

$$\begin{pmatrix} a_x \\ b_x \end{pmatrix} = \frac{1}{(3 \sum \lambda_i^2 - (\sum \lambda_i)^2)} \begin{pmatrix} \sum \lambda_i^2 & -\sum \lambda_i \\ -\sum \lambda_i & 3 \end{pmatrix} \begin{pmatrix} \sum \vec{P}_i \cdot \vec{i} \\ \sum \lambda_i \vec{P}_i \cdot \vec{i} \end{pmatrix} \tag{11}$$

The solution $(\vec{a}, \vec{b})$ can then be expressed as

$$\vec{a}_n \;=\; \alpha \vec{a}_o + a_x \vec{i} \tag{12}$$
$$\vec{b} \;=\; \vec{b}_o + \beta \vec{a}_o + b_x \vec{i} \tag{13}$$

In general, $\vec{a}_n$ is not perpendicular to $\vec{b}$. So for our convenience, we redefine $\vec{a}$ to remove any component along $\vec{b}$.

$$\vec{a} = \vec{a}_n - \frac{\vec{b} \cdot \vec{a}_n}{\vec{b} \cdot \vec{b}} \tag{14}$$

After all the possible tracks are found we can start with the preliminary vertex fit. For this purpose the previously described region of the possible vertex locations is breaking into a set of cylinders with $R = Vertex_W$ with the step $= 2mm$ and the common symmetry axis coinciding with a beam axis. Then a particular cylinder that is crossed by the biggest number of tracks must be found. After that, for track, that crosses the found cylinder, the vertex position is determined precisely.

One considers the optimal estimate of the vertex position as the position which minimizes the sum of the distance of a point nearest to all tracks simultaneously.

Given the (unknown) vertex position $\vec{V}$, and a track, j, whose direction is given by $\vec{b}_j$, the point of the closest approach to the vertex, $\vec{P}_c$ can be obtained by solving the following equastion:

$$(\vec{P}^c - \vec{V}) \cdot \vec{b} = 0. \tag{15}$$

Using the same track model (1) one finds

$$(\vec{a} + \lambda^c \vec{b}_j - \vec{V}) \cdot \vec{b}_j = 0. \tag{16}$$

Clearly, this yields

$$\lambda_j^c = \frac{(\vec{V} - \vec{a}_j) \cdot \vec{b}_j}{b_j^2}. \tag{17}$$

The optimal vertex position is obtained by minimizing of the functional K defined as the sum of the distance squares of the closest approaches to all tracks.

$$K = \sum_{j=1}^{N_t} \omega_j \left(\vec{a}_j + \lambda_j^c \vec{b}_j - \vec{V}\right)^2. \tag{18}$$

We use here the special robust approach (see, for example [4]) in oder to reduce a contaminating effect of ghost tracks and other tracks properly reconstructed but not pointing to the primary vertex. At this approach a conventioanl least square functional is replaced by the sum of weighted least squares with optimally chosen weight functio, which values also depend on the track parameters and must be reculculated on each iteration. We use suboptimal weight function [4] which are, in fact, the Tukey's be-weights calculated as foolows:

$$\omega_j = \begin{cases} \left(1 - \frac{g_j}{(3\sigma^{n-1})^2}\right)^2 & \text{for } g < [3\sigma^{n-1}]2, \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where "n" is the iteration number, and

$$g_j = (\vec{V} - \vec{a}_j) \cdot (\vec{V} - \vec{a}_j) - \frac{((\vec{V} - \vec{a}_j) \cdot \vec{b}_j)^2}{(\vec{b}_j \cdot \vec{b}_j)}, \tag{20}$$

$$\sigma^n = \frac{\sum_{j=1}^{N_t} \omega_j^{n-1} g_j^{n-1}}{\sum_{j=1}^{N_t} \omega_j^{n-1}}, \tag{21}$$

N_t is the number of tracks in the event. The notation X^{n-1} refers to quantities calculated on the previous iteration.

The vertex position, $\vec{V}$, is determined by minimizing the functional K relative to the vertex position. Substituting λ_j^c in (18) by (17) we obtaine the normal equation system:

$$\frac{\partial K}{\partial \vec{V}} = \sum_{j=1}^{N_t} \omega_j \left(\vec{a}_j + \frac{(\vec{V} - \vec{a}_j) \cdot \vec{b}_j}{b_j^2} \vec{b}_j - \vec{V}\right) \cdot \left(\frac{\vec{b}_j \vec{b}_j}{b_j^2} - 1\right) = 0, \tag{22}$$

which yields the solution

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = M^{-1} \begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix}, \qquad (23)$$

where the matrix "M" is defined as

$$M = \begin{pmatrix} \Sigma\omega\frac{1-(\vec{b}\cdot\vec{i})(\vec{b}\cdot\vec{i})}{b^2}) & \Sigma-\omega\frac{(\vec{b}\cdot\vec{i})(\vec{b}\cdot\vec{j})}{b^2} & \Sigma-\omega\frac{(\vec{b}\cdot\vec{i})(\vec{b}\cdot\vec{k})}{b^2} \\ \Sigma-\omega\frac{(\vec{b}\cdot\vec{j})(\vec{b}\cdot\vec{i})}{b^2} & \Sigma\omega\frac{1-(\vec{b}\cdot\vec{j})(\vec{b}\cdot\vec{j})}{b^2}) & \Sigma-\omega\frac{(\vec{b}\cdot\vec{j})(\vec{b}\cdot\vec{k})}{b^2} \\ \Sigma-\omega\frac{(\vec{b}\cdot\vec{k})(\vec{b}\cdot\vec{i})}{b^2} & \Sigma-\omega\frac{(\vec{b}\cdot\vec{k})(\vec{b}\cdot\vec{j})}{b^2} & \Sigma\omega\frac{1-(\vec{b}\cdot\vec{k})(\vec{b}\cdot\vec{k})}{b^2}) \end{pmatrix} \qquad (24)$$

and the matrix "I" is defined as

$$I = \begin{pmatrix} \Sigma\omega((\vec{a}\cdot\vec{i}) - (\vec{a}\cdot\vec{b})\frac{(\vec{b}\cdot\vec{i})}{b^2}) \\ \Sigma\omega((\vec{a}\cdot\vec{j}) - (\vec{a}\cdot\vec{b})\frac{(\vec{b}\cdot\vec{j})}{b^2}) \\ \Sigma\omega((\vec{a}\cdot\vec{k}) - (\vec{a}\cdot\vec{b})\frac{(\vec{b}\cdot\vec{k})}{b^2}) \end{pmatrix} \qquad (25)$$

Here the sums are taken over all "N_t" tracks reconstructed in an event.

3 Simulation results

In this section we present the results for track fitting routine, compare the results for the vertex reconstruction obtained be means of simple non-iterative minimization of non-weighted functional K ($w_j = 1$) and the robust approach. The accuracy of proposed vertex finder is studied.

The underlying sample consists of 1000 Monte Carlo events simulated for the above mentioned values of σ and σ_θ.

Fig.4 shows histograms of the track χ^2 distributions for reconstructed tracks. It is to be noted that, the obtained χ^2 distribution is quite close to the theoretical expectation for a χ^2 distribution with 1 degree of freedom (theoretical distribution is

shown with dashed line). Besides, the proposed algorithms allows us recognize about 98% of simulated tracks.

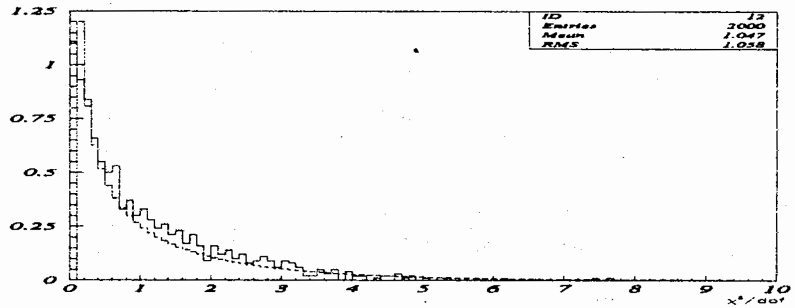

Figure 4: Track Chi-squares

Fig.5 show the vertex x, y and z coordinate distribution for the cases of the simple non-weighted vertex fit (plots a, b and c, respectively) and of the robust method (d, e and f). One can see from the resulting histograms the advantages of using the robust weight functions: it allows significant increase of vertex position determination accuracy. All peaks have Gaussian form, which is illustrated by the fitted Gaussians, and have practically unbiased mean values.

The different resolutions obtained by fitting each of the distributions individually for both cases are shown in Table 1. The robust approach gives a significantly better resolution for each coordinate.

Table 1: Precision of the vertex reconstruction

Method	σ_x	σ_y	σ_z
Unweighted fit	23.43	10.35	10.91
Robust fit	7.15	4.92	5.36
	μm		

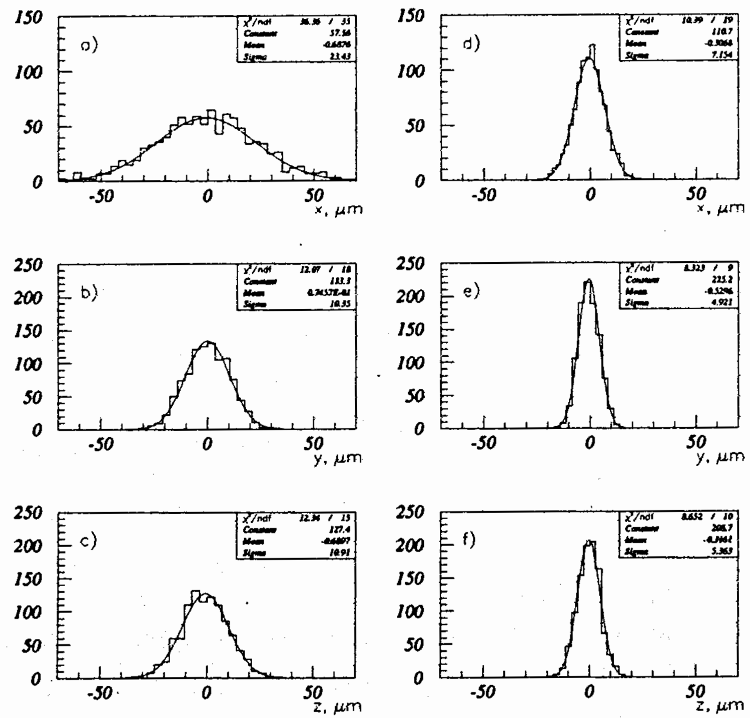

Figure 5: Vertex coordinate definition errors a,b,c - for unweighted fit; d,e,f - for robust fit

To determine the precision of the vertex reconstruction the set of thousand events is treated for each value of σ and σ_ϕ parameters. Fig.6, a demonstrates the distribution of the vertex reconstruction errors for $\sigma = 20\mu m$ and $sigma_{phi} = 1mrad$. Fig.3 b shows the dependence of vertex reconstruction precision $\sigma_{vertex} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$ on values of σ and σ_ϕ. The results obtained show the good precision of the method proposed, even when hit dispersion is significant. That allows us to use this method for wafer position reconstruction (alignment).

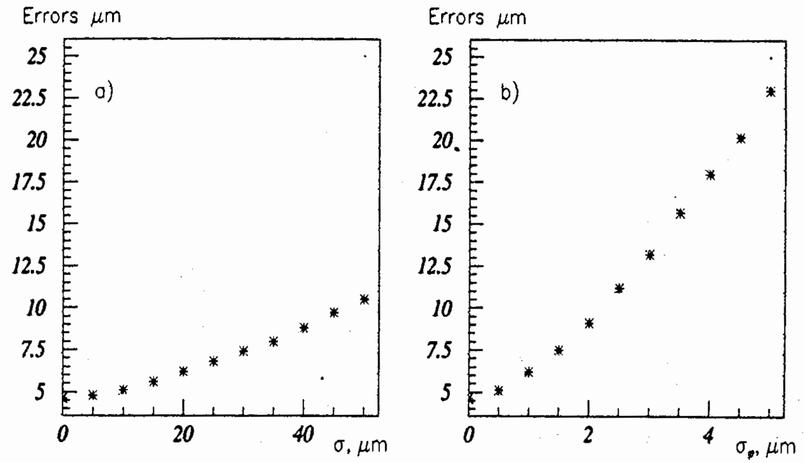

Figure 6: σ_{vertex} dependencies on σ **a)** and σ_ϕ **b)**

4 Computational matters

For implementing of the worked out algorithms and their inclusion in the STAR software C++ programming language with all benefits of object oriented approach is used. The problem itself demands the application of abstract data types, that are connected on the one hand to the complex geometrical configuration of the setup and on the other hand to the physics nature of the problem.

Several types of abstract classes were elaborated: *wafer, ladder, barrel, event, work* etc. The first three classes are closely connected to the setup geometry and allows us, for instance, to make quite complex point coordinate recalculations (e. g. for different recalculations switches between 216 local, 36 intermediate and one global coordinate systems are used). All details are hidden from the user by the help of the C++ encapsulation feature. Thus, each new task connected to similar geometry structure of the given setup can be solved with the help of these classes (with minimal alterations) or some new classes can be derived.

It is quite natural to create for our task such the abstract data types as *point, track, event* that not only considerably simplify access to data, but also make the program more logical and readable. Besides, the use of the proposed classes is for sure to enhance a further development of this software for different programmers.

The need to work simultaneously with the whole data ensemble as well as with each event separately (i. e. preserving the possibility of event identification in the ensemble) leads to create an operating class *work* that is introduced in order to simplify considerably the monitoring of large data amounts.

The program code organized in such the way simplifies the debugging process, increases program stability and also makes easier to perform its proper modification in cases of inevitable future changes of the setup geometry. The latter issue is especially important for working out software for any research-and-development experiment, since in the process of its designing and developing its geometry configuration has usually to be changed several times.

5 Conclusion

We propose the fast algorithms for the tracks and vertex reconstruction and study the their precisions and the possibility of using these algorithms for processing of experimental data. For this purpose the computational object model of vertex detector with the geometry maximally close to that of the real SVT is created. For track and vertex reconstruction for minimization the functionals were chosen in such way, that it led to the simple linear equations on the track and vertex parameters and allowed to determine these parameters by a simple one step procedure without any time consuming iterations. The results obtained show the good precision of the method proposed, that allows us to use

this method for wafer position reconstruction (alignment) for processing of real experimental data. And besides, in our program packet everything that is connected with the detector is placed in a single block, this simplified the program structure, increased its reliability, decreased the possibility of errors. One and the same code (describing classes) can be easily used without any changes by different programs (different users).

References

[1] O. Barannikova, G. Ososkov, Yu. Panebratsev "The Alignment of SDD within SVT system", E10-96-344.

[2] R. Bellwied (WSU) "SVT coordinate system and labeling scheme"

[3] A. Klatchko, B. C. Choudhary, T. Huehn, "Estimation of the Multiply Coloumb Scattering Error for Various Numbers of Radiation Lengths", FERMILAB-Pub-92/289, 1992.

[4] F. Mosteller, W. Tukey, "Data analysis and regression: a second course in statistics", Addison - Wesley (1987).

Received by Publishing Department
on October 1, 1998.

Баранникова О.Ю., Ососков Г.А., Панебратцев Ю.А. E10-98-278
Исследование быстрых алгоритмов для восстановления треков и определения положения вершин с использованием новой вычислительной модели с полной STAR-SVT геометрией

Предложены быстрые алгоритмы для восстановления треков и вершин событий для детектора SVT в эксперименте STAR. Исследованы точность предлагаемых алгоритмов и возможности их использования для обработки реальных экспериментальных данных, для чего была разработана объектная модель вершинного детектора с геометрией, максимально приближенной к реальной STAR-SVT установке. В созданных на C++ программных кодах все объекты, связанные с геометрией установки, выделены в отдельный блок, что существенно упрощает структуру программы, и увеличивает надежность ее работы. Кроме того, код, описывающий классы, может быть использован пользователями без всяких изменений для различных программ.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 1998

Barannikova O.Yu., Ososkov G.A., Panebratsev Yu.A. E10-98-278
Investigation of Fast Algorithms for the Track Reconstruction and Vertex Finding Using New Computation Model with Real STAR-SVT Geometry

We propose the fast algorithms for the tracks and vertex reconstruction for the STAR-SVT setup and investigate their precision and the possibility of using these algorithms for processing real experimental data. For this purpose the computational object model of the vertex detector with the geometry maximally close to that of the real STAR-SVT is created. In our C++ program all objects related to the detector are placed in a single block. That simplified the program structure, increased its reliability, minimized the possibility of errors. Besides, the same code (describing classes) can be easily used without any changes by different programs or users.

The investigation has been performed at the Laboratory of High Energies, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 1998