

# The SND Interactive 3D Event Viewer

I.V. Gaponenko\* *BINP, Novosibirsk, Russia*

## Abstract

This work describes the interactive 3D event viewer. The viewer aims to display the data acquired from SND (Spherical Neutral Detector) [1]. It reads and displays the experimental data objects specified in the COCHA [2] package format. The viewer is based on the Open Inventor object-oriented toolkit [3]. Local interactivity, provided by Open Inventor, allows a user to change the view of the displayed event rotating and zooming it. Additional features give the user a possibility to review information about various parts of the detector systems and the displayed event. The displayed event can be saved either in the VRML or Open Inventor format file. The viewer is part of the SND 3D event display.

Keywords: SND, Open Inventor; event display; VRML.

## 1 Introduction

The interactive 3D event viewer is based on a three-dimensional graphics model. It uses real three-dimensional geometric description of the detector and events acquired from it. The metric right-handed coordinate system is used to define the geometry and represent it as the scene graph, that is eventually drawn on the screen. All angles are specified in radians. To construct the detector scene graph, which is a complicated one, the Open Inventor file format is used to read some parts of the detector description from a file and some from a string in memory. The other parts are defined by the program. The Open Inventor examiner viewer is used to render the scene graph. The viewer allows one to interactively change the view of the scene graph by click-dragging the mouse buttons in the window or using the various controls along the window border (e.g., zoom). The three-dimensional scene is transformed automatically before being drawn onto a two-dimensional screen to show a particular view of the scene graph. In other words, the viewer provides all features, that are called local interactivity.

The viewer has been written to investigate the 3D Open Inventor [3] based approach for implementing the event display. This investigation is based on the real experiment and the real data is visualised. Since the most difficult part in this task is the description of the detector geometry, the SND geometry has been chosen as the relatively simple one.

Since the 3D visualisation demands large amount of the hardware resources (CPU, memory, etc.), one of the aspects to investigate has been the interactive response of the viewer. The idea has been not only to run the viewer on special SGI workstations, but also on the ordinary X-terminals without GLX extension. The investigations show that there are many useful features in the Open Inventor toolkit, which allow us to increase the performance of the viewer. One of such features is a possibility to specify various levels of detail for the same object. The **SoLevelOfDetail** node usage is discussed beneath. The viewer performance has been tested on various computers to define the requirements of the hardware needed for this purpose. The various aspects of the approach are discussed and the results of this investigation are presented in this article. If the results seem to be satisfactory, the approach might be used for detectors with more complex geometry.

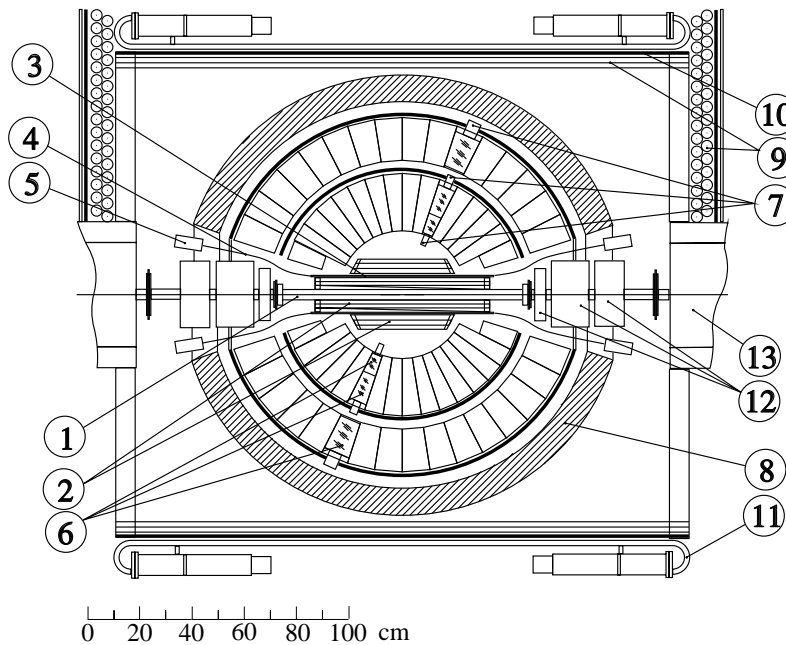
## 2 The Detector Overview and Events Signature

The experiments with SND have been carried out at the collider VEPP-2M in Novosibirsk since 1995. At present, before the beginning of the DAΦNE era, VEPP-2M is the only collider in the world operating in the energy range of  $\rho$ ,  $\omega$ , and  $\phi$  vector mesons. SND is a general purpose nonmagnetic

---

\* E-mail: I.V.Gaponenko@inp.nsk.su

detector. It consists of the tracking system, three spherical layers of NaI(Tl) crystals, and an outer system of streamer tubes and plastic scintillation counters (Fig. 1). The NaI(Tl) calorimeter consists of 1632 individual crystals of 3.6 tons total weight. The calorimeter thickness in the radial direction is 35 cm  $\sim 13.5 X_0$ . The angular size of each crystal is  $9^\circ$ . The solid angle coverage of the calorimeter is 90% of  $4\pi$ . For trigger needs the calorimeter crystals are logically organized into "towers". A tower consists of counters located within  $18^\circ$  interval in the same polar and azimuthal directions in all three layers. The number of counters in a tower is 12 at large angles and 6 in the regions, close to the beam. In addition to signals from individual counters each tower produces a total energy deposition analog signal and two trigger signals. The tracking system consists of two cylindrical drift chambers. Both chambers are divided into 20 jet-type cells in azimuthal plane. Each cell contains five sensitive wires. The longitudinal coordinate is measured by the charge division method with an accuracy of 3mm. In addition, cathode strip read-out for inner and outer layers provides the improvement of the latter value to 0.5mm. The muon/veto system consists of plastic scintillator counters and streamer tubes.



**Figure 1:** The SND Detector Overview: 1 - collider beam pipe; 2 - drift chambers; 3 - coincidence counter; 4 - fiber lightguides; 5 - photomultipliers; 6 - NaI(Tl) crystals; 7 - phototriods; 8 - iron absorber; 9 - muon tubes; 10 - 1cm iron plate; 11 - muon counters; 12 - magnetic lenses; 13 - bending magnets.

The physics program for SND experiment includes radiative decays, OZI and G-parity suppressed decays, electromagnetic decays,  $e^+e^-$  annihilation into hadrons, test of QED, search for rare  $K_S$  and  $\eta$  decays and C-even reactions. Typically there are no more than seven or ten particles in each event.

### 3 Requirements for a Visualisation System

A visualisation system, or event display, should meet the following requirements. It should be:

- interactive, so that the displayed event can be zoomed and rotated easily;
- flexible (modular design):
  - to integrate it in the data acquisition system;
  - to easily add new functionality in the future, when it's necessary;
  - to integrate it with the data processing system;

- suitable for debugging the reconstruction and selection algorithms;
- suitable for testing the detector systems;
- able to provide all the information about the displayed event and the detector systems, although some information might be hidden, and displayed only by the user request.

These are probably the most important, but incomplete, requirements for the detector visualisation system. From amongst these requirements, the most important one is the task of the debugging of reconstruction algorithms and event selection procedures. Obviously, integration of the visualisation system with the reconstruction and analysis parts provides faster debugging facilities. That is why the system should be flexible. The visualisation system should also easily allow the user to scan the detector data event by event, displaying vertices, tracks, and so on. The visualisation system should also be used to extract some interesting or rare events in a file with portable format to provide some facilities to publish the experiment results. That is why the visualisation system should have the output facilities.

One of the important parts of the system is its ability to visualise the detector systems, which are not operational, and provide additional information about them, in order to check the detector systems. This is particularly essential for detectors with more complex geometry.

#### 4 Why Open Inventor?

Firstly, both the detector and the event must be split into logical parts (objects), whose behavior is similar in particular cases. Physical entities of the detector (crystals, wires, etc.) are considered as objects also. This is so obvious, that there is no need to be discussed here. The approach of splitting the detector into the logical parts fully corresponds to the object-oriented model. That is one of the reasons why visualisation should be done with the object-oriented tools, such as Open Inventor. Being very intuitive, this approach gives us flexibility to change the properties of the objects and eases the addition of new objects. Object-oriented methodology is based on the physical and mechanical world we live in.

- **Conclusion 1: It is object-oriented.**

Secondly, one of the hardest task is transformation of the 3D object to the 2D display. Even if the definition of the objects is three-dimensional, which is very convenient in itself, the rendering of these objects should be done also by the toolkit, so that we do not need to implement complex transformation algorithms.

- **Conclusion 2: It provides 3D object description and automatic transformation to the 2D screen.**

Thirdly, the interactive manipulations, such as rotation, zoom, etc., allow one to change the view of the rendered objects. The interactive selections, picking, etc. provided by Open Inventor, enables us to include all information and to display it only when required.

- **Conclusion 3: It is interactive.**

Finally, based on the requirements of a visualisation system, it has been necessary to find the most convenient solution. The 3D graphics approach for event visualisation seems the most natural for this task. The Open Inventor toolkit provides all we need in this situation. Open Inventor is an object-oriented 3D toolkit offering a comprehensive solution to interactive graphics programming problems. It defines the standard file format for exchanging 3D data among applications. Moreover, it serves as the basis for the VRML (Virtual Reality Modelling Language). Inventor objects include database primitives, including shape, property, group, and engine objects; interactive manipulators, such as handle box and trackball; and components, such as the material editor, directional light editor, and examiner viewer. All information about the 3D objects - their shape, size, colouring, surface texture, location in 3D space - is stored in a scene database. This information can be used in a variety of ways. Changing the objects in the scene, adding to the scene, and interacting with the objects is a simple process, because such changes are the part of Inventor's interface, and because

they are anticipated by Inventor's basic design. The objects in the scene can be picked, highlighted, and manipulated as discrete entities. This programming model is intuitive, and hence, convenient.

## 5 The Visualisation System Structure

### 5.1 Nodes of the Detector Scene Graph

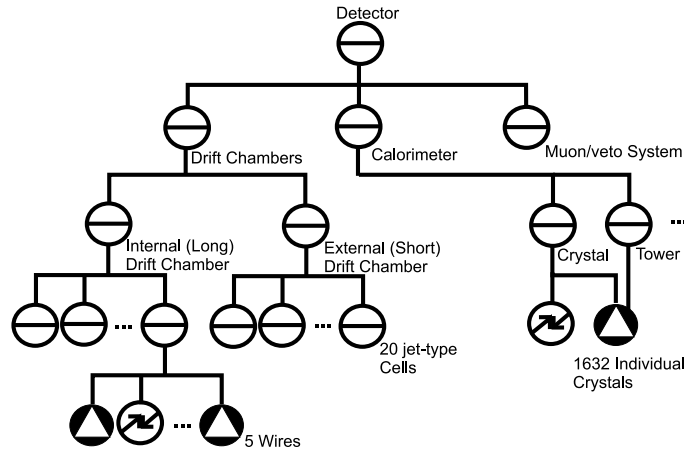


Figure 2: A Part of the Scene Graph for the SND Detector

Open Inventor is a fairly large and very general system. To implement the viewer only several features of it has been used. The most difficult part of the visualisation system is the detector geometry definition. Open Inventor facilitates the process, by supporting the specification of the detector geometry via an Open Inventor format file and a string in memory. This is an example of the definition via the Open Inventor ASCII file:

```

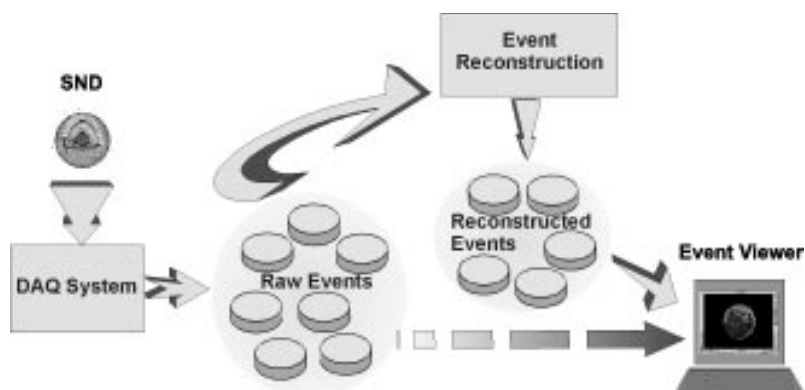
#Inventor V2.0 ascii
#Top level = single node gathered into Separator
Separator {
    #Coordinate3 node defines the eight points of a NaI(Tl) crystal.
    Coordinate3 {
        point [
            0 0 0, #First point, index 0
            1 0 0, #Second point, index 1
            ...
            0 1 1 #Eighth point, index 7
        ]
    }
    #Define the front of the crystal
    IndexedFaceSet {
        coordIndex[ 4, 5, 6, 7 ] #Four points in front
    }
    #Define the other sides of the crystal
    ...}
  
```

Open Inventor defines a set of objects useful for doing 3D graphics - nodes. The node is the basic building block used to create a 3D scene database in Open Inventor. The nodes defined in the detector are crystals, wires, and so on. Nodes are arranged in the hierarchical structure - scene graphs. Scene graphs are more than just a collection of nodes: the scene graph defines an ordering for the nodes. Nodes earlier in the scene graph can affect nodes that appear later in the scene. For

example, **Rotation** or **Material** affect nodes after in the scene. The group nodes are separator, level-of-details, and switch. The separator group nodes of the detector are the drift chambers, the calorimeter, the muon system. The separator nodes limit the effects of properties. It allows parts of the scene graph to be functionally isolated from other parts. A node can belong to more than one group. For example, a crystal belongs to a calorimeter and a tower (Fig. 2). The elements of the detector correspond to the nodes of the scene graph. Each element of the detector is defined as a separate node: each crystal, each wire of the drift chambers, etc. is presented in the scene graph as a shape node. The nodes are joined in groups. Some of these groups are defined physically, for example, wires of a drift chamber belong to one group. Some of the groups are defined logically, for example, crystals of the calorimeter that belong to a tower. Groups in the scene graph are: drift chambers that join the wires, towers that package some crystals (see above), the calorimeter that join all the crystals, etc. The group nodes are containers that collect nodes into the detector scene graph. The nodes are named to refer to them elsewhere. That is very convenient, since the crystals in the detector are named also. Each crystal has a unique number, which defines its position relative to the other crystals.

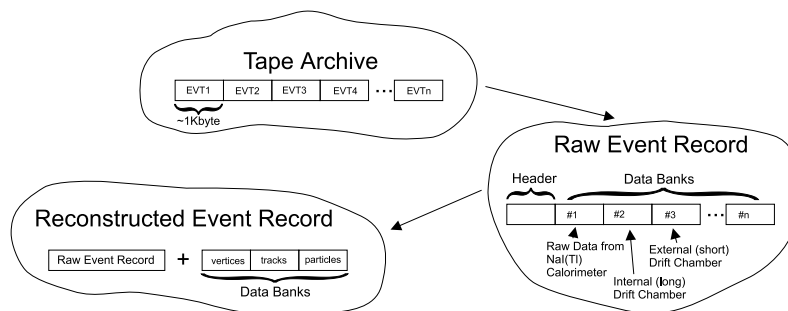
## 5.2 Data Flow

The data acquired from the SND detector by its DAQ system are stored in the tape archives (Fig. 3) as raw events. At the next stage of processing these data are read and processed by reconstruction



**Figure 3:** Simplified Data Flow Chart

and event selection programs using the calibration information and then are written back to the tape archives as reconstructed events. The formats of these files are quite similar with the difference that reconstructed events carry additional information coming from reconstruction procedure, such as particles, vertices, and tracks (Fig. 4).



**Figure 4:** Raw/Reconstructed Events

The typical length of the event record is about 1Kbyte. Each record has a number of information banks - one per detector subsystem, plus the reconstruction output mentioned above. The event converter is written to read experimental data from the archive file to create the Open Inventor database.

### 5.3 The Viewer: What can it do?

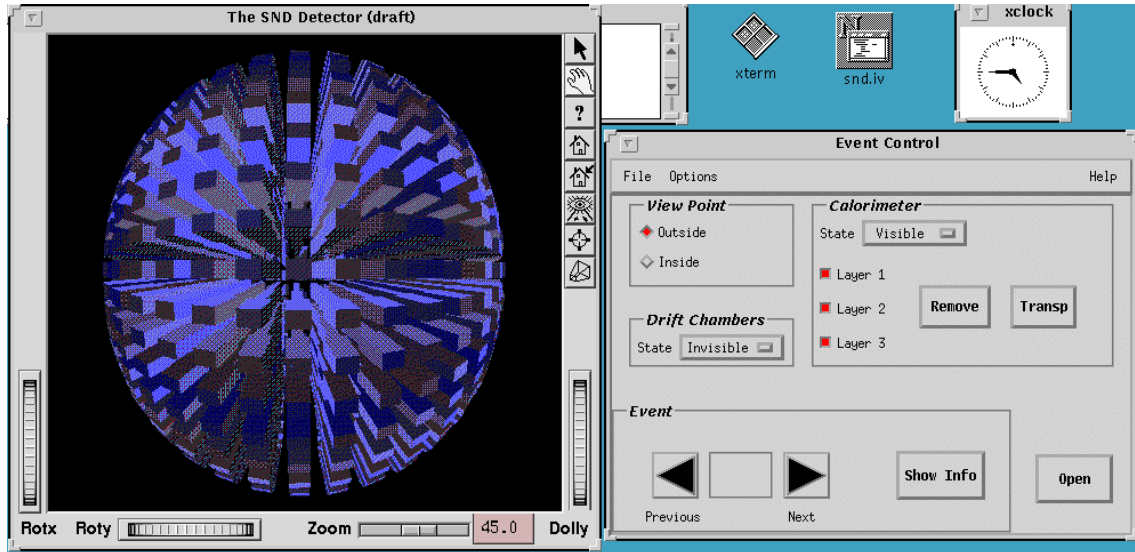


Figure 5: The 3D Interactive Event Viewer

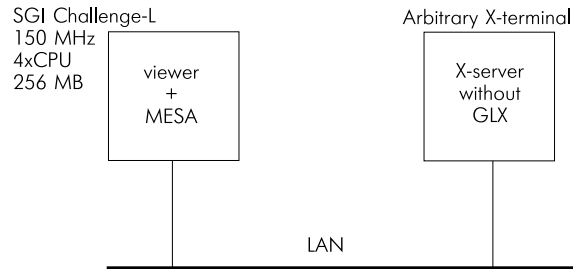
The visualisation system consists of two windows (Fig. 5). One of them is the examiner viewer. It allows a user to rotate and zoom the displayed event. Another is a control panel that provides event control facilities: the user can display additional information about the event and the detector systems, or save the scene graph in a file. There are two formats for output files supported in current version: VRML and Open Inventor (binary or ASCII). Various controls in the second window allow the user to remove or make transparent some displayed parts of the detector or to get additional information about the event.

## 6 Current Status

In the current implementation of the viewer the distributed client-server scheme has been chosen (Fig. 6). This choice has been motivated by the available hardware and software. This approach allows us to use any X-terminal with or without the GLX extension, because the MESA shared library, which emulates OpenGL, is installed on the host computer. However, there are drawbacks in this scheme. One of them is high network traffic, which can rise by up to tens of percent if the MESA library is used. The viewer has been tested on the SGI Server CHALLENGE-L. As a benchmark the rotation of the detector depending on the rendered area size and the number of rendered nodes has been chosen. The results are presented in Table 1. The time is measured with an accuracy of  $\pm 1s$ . The result in the third case is almost the same as in the second case, because the SoLevelOfDetail node is used.

## 7 Conclusion

The results of the investigation show that Open Inventor fully matches the task of event visualisation. However, new nodes aimed at a particular situation (for example, **SoNaIcrystal** shape node, that



**Figure 6:** The Distributed Client-Server Scheme

What is Displayed	Rotation Time (s)		
	Render Area Size		
	1024x800	600x400	300x200
<b>Drift Chambers and Event</b>	6	5	2
<b>Calorimeter and Event</b>	7	6	3
<b>Calorimeter, Drift Chambers, and Event</b>	7	6	3
<b>Event and Active Elements of the Detector</b>	4	3	1

**Table I:** Performance of the viewer.

would properly present the shape of calorimeter crystals) are necessary. Creating these new objects and methods requires the extension of Open Inventor by subclassing.

The issue seems to be obvious. The more complex the geometry to be rendered the more powerful a workstation is needed. However, the performance of the viewer is quite satisfactory, if the visible part of the scene graph is restricted and a possibility to add invisible parts of the detector and events is left to the user. It should be carefully analysed in each particular case well before implementing the Open Inventor based event display.

## Acknowledgements

I wish to thank M.S. Dubrovin and I.A. Gaponenko for valuable remarks and multiple discussions.

## References

- 1 M.N. Achasov et al., *Status of the experiments with SND detector at e+e- collider VEPP-2M in Novosibirsk*, Preprint BudkerINP 96-47 (1996).
- 2 V.N. Ivanchenko, *COCHA - the Data Structure Management Package*, Preprint BudkerINP 94-25 (1994).
- 3 *The Inventor Mentor*, Addison-Wesley Publishing Company.
- 4 *The Inventor Toolmaker*, Addison-Wesley Publishing Company.
- 5 *The Open Inventor C++ Reference Manual*, Addison-Wesley Publishing Company.
- 6 *The OpenGL Programming Guide*, Addison-Wesley Publishing Company.