



---

# **TPC Tracking Software Manual**

---

# TPC Tracking Software Manual

Ryuho Kudo  
Jennifer Klay, PhD

## **Contributors:**

Andrew Barton  
Uriel Gomez  
Ryan Harch  
Samson Truong

---

California Polytechnic State University, San Luis Obispo  
San Luis Obispo, CA 93407

2009

# Table of Contents

---

Disclaimer .....	v
1. INTRODUCTION .....	1
<b>1.1 What is NIFFTE’s TPC Tracking Software?</b> .....	1
<b>1.2 Current Flow of Operation</b> .....	1
<b>1.3 Future Flow of Operation</b> .....	2
2. INSTALLATION .....	4
<b>2.1 Regular Installation</b> .....	4
▪ 2.1.1 Hardware Recommendations .....	4
▪ 2.1.2 Dependencies .....	4
▪ 2.1.3 Compiling .....	4
<b>2.2 Installation Using Virtual Machine</b> .....	5
▪ 2.2.1 Installing Virtual Box .....	5
▪ 2.2.2 Importing Disk Image .....	5
<b>2.3 Connecting to a Server with Software</b> .....	6
<b>2.4 Maintenance</b> .....	7
▪ 2.4.1 SVN Update .....	7
▪ 2.4.2 Recompiling Code .....	7
3. OPERATION .....	8
<b>3.1 Running the Simulation &amp; Reconstruction</b> .....	8
▪ 3.1.1 Geant4 Physics Simulation .....	8
▪ 3.1.2 NIFFTE Detector Simulation (detSim.xml) .....	9
▪ 3.1.3 NIFFTE Reconstruction (configStd.xml) .....	9
<b>3.2 Options/Settings</b> .....	10
▪ 3.2.1 Physics Simulation .....	10
▪ 3.2.2 Detector Simulation .....	11
▪ 3.2.3 Standard Track Reconstruction .....	11
▪ 3.2.4 Hough Track Reconstruction .....	12
4. ANALYSIS .....	13
<b>4.1 Terminal View of Reconstruction</b> .....	13

▪ 4.1.1 What to Use This For .....	13
<b>4.2 Using the Event Display .....</b>	<b>14</b>
▪ 4.2.1 Running the Event Display.....	14
▪ 4.2.2 Using the Event Display .....	14
▪ 4.2.3 What to Use This For .....	15
<b>4.3 Writing a Macro in ROOT for Analysis.....</b>	<b>15</b>
<b>4.4 What Does All of This Mean? .....</b>	<b>15</b>
5. TROUBLESHOOTING/FAQ.....	16
<b>5.1 Common Mistakes .....</b>	<b>16</b>
▪ 5.1.1 Terminal Command Line Mistakes .....	16
▪ 5.1.2 Options Mistakes.....	17
<b>5.2 Known Bugs .....</b>	<b>17</b>
<b>5.3 Missing Functionalities .....</b>	<b>17</b>
6. RESOURCES .....	18
<b>6.1 Contact E-mails .....</b>	<b>18</b>
<b>6.2 Websites .....</b>	<b>18</b>
7. APPENDIX.....	19
<b>7.1 Subroutines .....</b>	<b>19</b>
▪ 7.1.1 Geant4 Physics Simulation .....	19
▪ 7.1.2 NIFFTE Detector Simulation Modules .....	19
▪ 7.1.3 NIFFTE Reconstruction Modules .....	19
<b>7.2 Options .....</b>	<b>19</b>
▪ 7.2.1 Geant4 Physics Simulation .....	19
▪ 7.2.2 NIFFTE Detector Simulation Modules .....	20
▪ 7.2.3 NIFFTE Standard Reconstruction.....	21
▪ 7.2.4 NIFFTE Hough Reconstruction .....	22
8. GLOSSARY .....	24
9. TABLE OF FIGURES .....	25
10. REVISION LIST .....	26

# Disclaimer

---

Please check the Revision List section to find out which SVN revision number this documentation was written for. There is a good possibility that the software has been significantly altered if the revision number has changed and will misinform you of how to operate this software. At the time this disclaimer was originally written, the NIFFTE software project was still being heavily developed. It is only natural the documentation for the software gets quickly outdated as well.

# 1. INTRODUCTION

---

## 1.1 What is NIFFTE's TPC Tracking Software?

---

The Neutron Induced Fission Fragment Tracking Experiment (NIFFTE) will employ a compact, pressurized Time Projection Chamber (TPC) to study neutron-induced fission of the major actinides. The software to reconstruct the fragment tracks in the TPC must be capable of sifting through the large volume of sampled data to identify them and determine their (A,Z). This reconstruction is done in several stages.

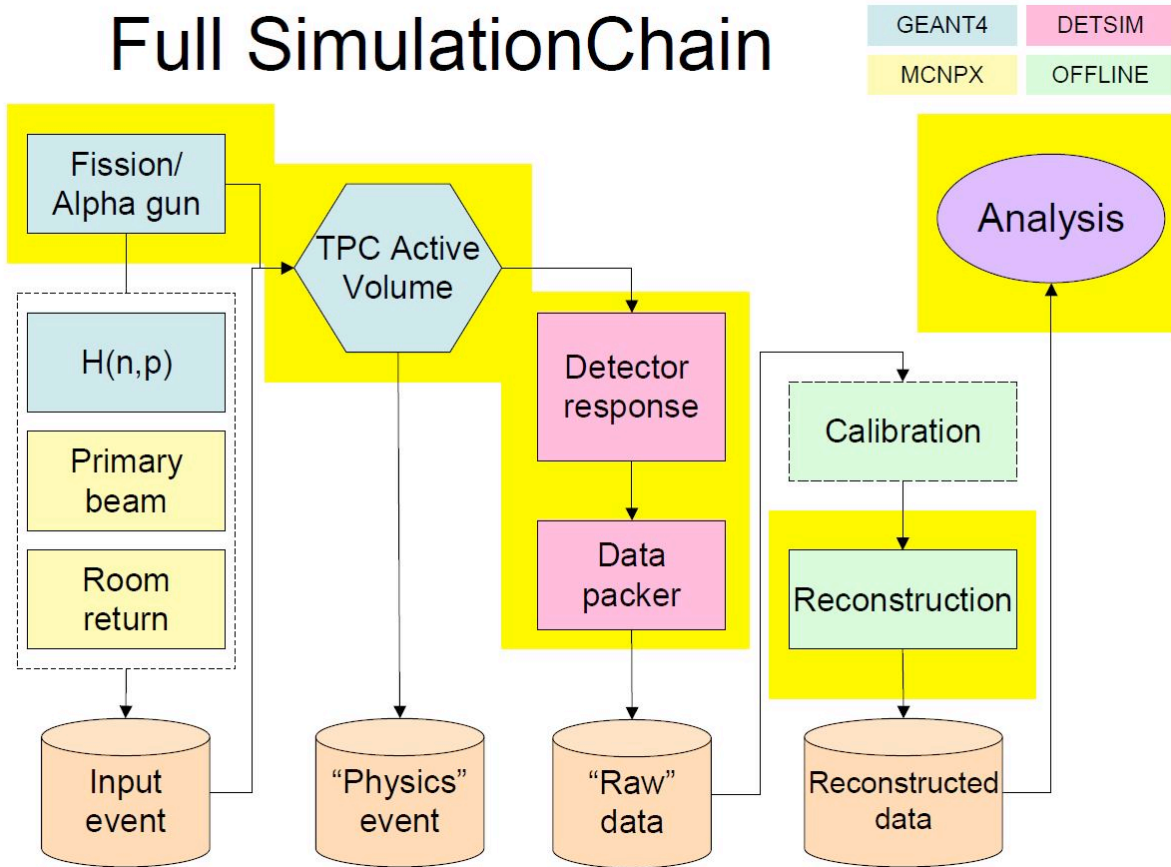
It first performs 2-dimensional clustering of raw TPC voxels, followed by mountain-finding and hit-fitting to form hits. It then searches for contiguous hits, starting from the outer edge of the TPC near the pad plane where the population is lower and working inward toward the target. At each step, the most likely location for the next hit is predicted by a straight-line extrapolation of the current list of hits associated to the track. Track-fitting and error correction are completed once no more hits are found to belong to a given track.

## 1.2 Current Flow of Operation

---

The TPC software is currently set up for 'mock data challenge' stage of the project. This means that our software is set to receive input from a simulated input rather than actual input from hardware. This is done for a few reasons. We are still developing the TPC software, which means options and algorithm for the software needs to be calibrated correctly so that it can reconstruct track efficiently. Knowing the input and being able to check what the output should be for the reconstruction is the most efficient way to calibrate the software. Please refer to the figure on the next page for a helpful visual.

# Full SimulationChain

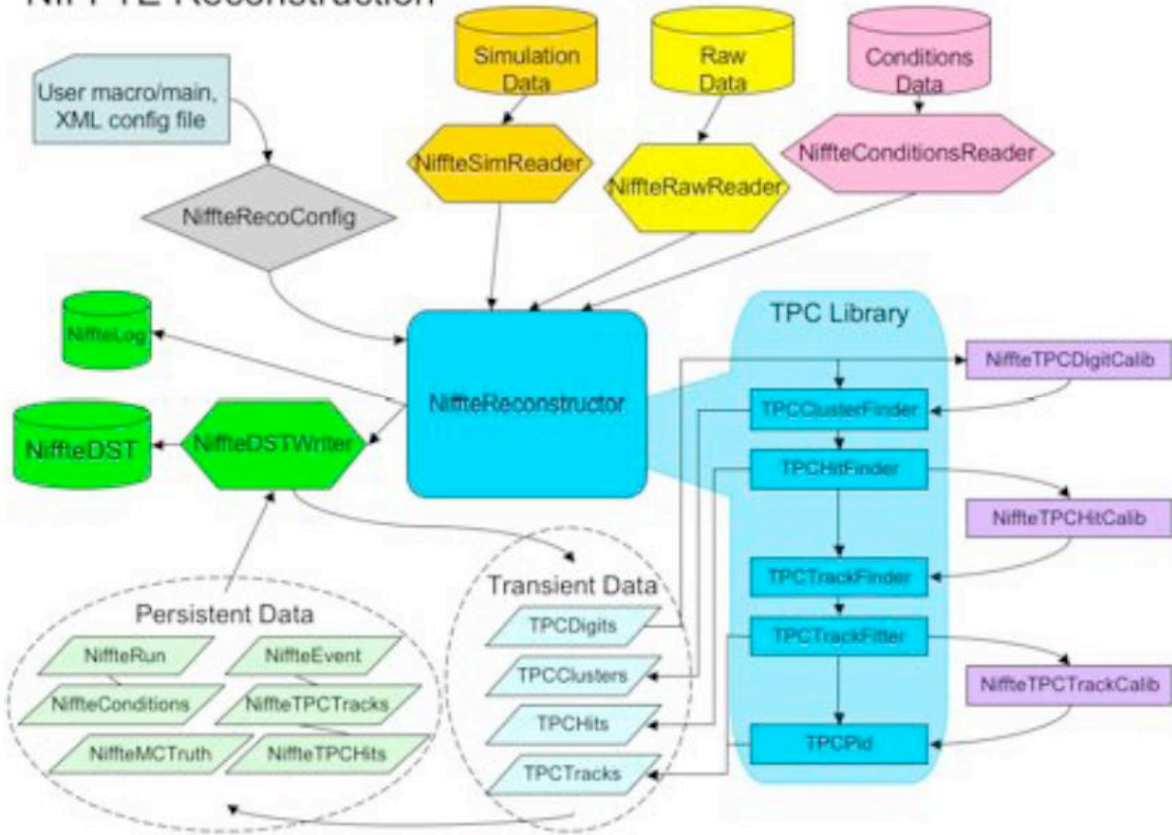


**Figure 1:** Diagram of Flow of Operation That Uses Simulation Input as Oppose to Using Hardware Input

## 1.3 Future Flow of Operation

When the hardware and software is ready, the software will be setup to receive input from the actual hardware and will analyze actual data. Please refer to the figure on the next page for a helpful visual.

## NIFFTE Reconstruction



**Figure 2:** Diagram of Flow of Operation That Uses Hardware Input as Oppose to Using Simulation Input



# 2. INSTALLATION

---

## 2.1 Regular Installation

---

### 2.1.1 Hardware Recommendations

---

There are no definite minimum hardware requirements for this software since it can work on almost any computers anyone uses. The problem is that it will run slow and at a certain point, it will just become time-consuming to be usable. So here are the recommended hardware specs:

- **2.4Ghz Single core CPU / 2.0Ghz Dual core CPU**
- **1GB of RAM**
- **6GB of HDD space** (needed if virtual machine is being used)
- **Nice Graphics card will help speed up the event display**

### 2.1.2 Dependencies

---

- **GEANT4 9.1.p02 and 9.2.p01** (both work)
- **CLHEP 2.0.4.2** (required by GEANT4)
- **ROOT v5-22-00 or v5-23-04**
- **GSL 1.11 or 1.12**
- **Xerces 2.8.0 or 3.0.1**

### 2.1.3 Compiling

---

The computer will need each dependency to be compiled independently, but the rule of thumb for compiling code is to:

- 1) Download the appropriate source file; it will most likely be a **.tar file** that is compressed with **.gz format**.
- 2) Uncompress it using programs like **gunzip** and **tar**
- 3) Go to the base directory
- 4) To configure the compiler, type in **./configure**
- 5) Then to compile and install, type in **make install**

There will most likely be a README file somewhere in the installation folder, so try to look for it if the directions are unclear.

## 2.2 Installation Using Virtual Machine

---

### 2.2.1 Installing Virtual Box

---



#### **Windows XP/Vista:**

- 1) Download the following:  
<http://download.virtualbox.org/virtualbox/3.0.2/VirtualBox-3.0.2-49928-Win.exe>
- 2) Run the program and follow the directions provided



#### **Linux:**

- 1) Go to this page:  
[http://www.virtualbox.org/wiki/Linux\\_Downloads](http://www.virtualbox.org/wiki/Linux_Downloads)
- 2) Find the flavor of Linux you are using and download the file
- 3) Install the program via however the website provides



#### **Mac OSX (Intel):**

- 1) Download the following:  
<http://download.virtualbox.org/virtualbox/3.0.2/VirtualBox-3.0.2-49928-OSX.dmg>
- 2) Install the program

### 2.2.2 Importing Disk Image

---

- 1) Download the disk image located here under the name NiffteVM.p7z:  
<http://nuclear.calpoly.edu/~niffte/NiffteVM/>
- 2) The file is compressed by a format called **7z**. Uncompress it by using the appropriate program for your OS.
  - **Windows:** <http://www.7-zip.org/>
  - **Linux/Mac OSX:** Install p7zip-full via package manager
- 3) Uncompress it and get the file **Fedora 10 GNOME Desktop.vmdk** and move it to a location where it has fast access (meaning not on USB)
- 4) Start the virtual box program. Go to **file -> virtual media manager**, and select **Add Hard Disks**; then choose the **.vmdk image** you downloaded and uncompressed.

- 5) Choose to start a new virtual machine, select the kind of OS to be **Linux – Fedora** and choose to use the already imported hard drive you added. Finish the wizard with default settings.
- 6) Start the Fedora OS.

## 2.3 Connecting to a Server with Software

---

This section will assume the server is connected via text based terminal via SSH.

### Windows XP/Vista:

- 1) Download and install SSH client
  - SSH client can be downloaded at:  
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>
- 2) Click the quick connect button
  - Fill out the following information (replace everything in <> with your own info):  
  
Host Name: <ip address or server name>  
User Name: <user name to login with>  
Port: <the port ssh uses, usually 22>  
Connection Type: **SSH**
- 3) Click connect
  - Enter password: <password to that user>

### Linux/Mac OSX:

- 1) The OS you are using will most likely come with ssh installed, but if it isn't there, install "ssh" with the package manager your OS is using.
- 2) Type **ssh -X <username>@<ip address / server name>** with everything in the <>s filled out with your information. Do not type the <>s itself.
- 3) Type in the password to that username and you should be logged on.

## 2.4 Maintenance

---

### ▪ 2.4.1 SVN Update

---

- 1) Go to the /trunk location by typing in  
**cd \$NIFFTE**
- 2) Update the source code by typing in  
**svn update**
- 3) After you type in the password to your svn server, everything should be updated to the latest code.

### ▪ 2.4.2 Recompiling Code

---

- 1) Go to the /trunk location by typing in  
**cd \$NIFFTE**
- 2) Then to compile the NIFFTE framework, run the compiling script by typing in  
**./autogen**
- 3) Next, to recompile the Geant4 physics simulation program, type in  
**cd \$NIFFTE/sim/tpc/TPCSimNiffteIO/**
- 4) To compile, just type in  
**make install**

# 3. OPERATION

---

## 3.1 Running the Simulation & Reconstruction

---

### ▪ 3.1.1 Geant4 Physics Simulation

---

- 1) Enter the following command:

```
$G4WORKDIR/bin/Linux-g++/TPC $NIFTE/sim/tpc/TPCSimNifteIO/run1.mac
```

Here is a breakdown of this command:

- The first part of the command,

```
$G4WORKDIR/bin/Linux-g++/TPC
```

specifies the executable file. It is a Geant4 program compiled to simulate the kind of particle physics used in the experiment

- The second part of the command,

```
$NIFTE/sim/tpc/TPCSimNifteIO/run1.mac
```

specifies the macro file that describes the particles used in the experiment.

- 2) If the program worked correctly, you should see the following terminal output at the end:

```
Graphics system detected.  
Visualization Manager detecting...
```

Data will then be outputted by the simulation. The physics simulation command will also create a file at the end, called **nifte\_mc\_output.root**. You can check this by typing the command: **ls**

### ■ 3.1.2 NIFFTE Detector Simulation (detSim.xml)

---

- 1) Enter the following command:

```
niffte -c $NIFFTE/build/testDetSimConfig.xml -i niffte-mc-output.root -o detSim.root
```

Here is a breakdown of this command:

- The first part of the command,

**niffte**

specifies the executable and runs the NIFFTE framework.

- The second part,

**-c \$NIFFTE/build/testDetSimConfig.xml**

is a configuration command. The **-c** means configuration, and the text following it is the text file being configured. Refer to Options/Settings for more information and options.

- 2) If the program worked correctly, you should see the following terminal output at the end:

```
I-NiffteRootIO.cxx-Close: Wrote X events. Closing file.
```

If **X** is a number less than 1, refer to the Troubleshooting section.

Data will then be outputted by the simulation. The physics simulation command will also create a file at the end, called **detSim.root**. You can check this by typing the command: **ls**

### ■ 3.1.3 NIFFTE Reconstruction (configStd.xml)

---

- 1) Enter the following command:

```
niffte -c $NIFFTE/build/configStd.xml -i detSim.root -o reco.root
```

Here is a breakdown of this command:

- The first part of the command,

**niffte**

specifies the executable and runs the NIFFTE framework.

- The second part,

**-c \$NIFFTE/build/configStd.xml**

is a configuration command. The **-c** means configuration, and the text following it is the text file being configured.

- The third part,

**-o reco.root**

specifies the output file. Refer to Options/Settings for more information and options.

- 2) If the program worked correctly you should see the following mess terminal output at the end:

*I-NiffteRootIO.cxx-Close: Wrote **X** events. Closing file.*

If **X** is a number less than 1, refer to the Troubleshooting section.

Data will then be outputted by the simulation. The physics simulation command will also create a file at the end, called **reco.root**. You can check this by typing the command: **ls**

## 3.2 Options/Settings

---

This section will introduce you to parameters that you can change by editing the **macro fire** or **.xml** file. Please refer to Appendix for example files of all modules.

### ▪ 3.2.1 Physics Simulation

---

This module has parameters and options that can be changed by editing the macro that will run with the program. They are usually located in **\$NIFFTE/sim/tpc/TPCSimNiffteIO/xxx.mac** and can be edited to your liking.

- **/gps/particle** - Defines what kind of particle it is
- **/gps/energy** - Defines how much energy in MeV this particle starts out with
- **/gps/pos/type** - Defines what type of particle it is

- **/gps/pos/shape** - Defines the shape this particle is starting out in
- **/gps/pos/centre** - Defines the x,y,z position in cm where the center is
- **/gps/pos/radius** - Defines the radius the particle it can start in
- **/gps/ang/type** - Defines the angular type of the particle

### ■ 3.2.2 Detector Simulation

---

This module has parameters defined in the **.xml** file that will be defined by the **-c** argument when the NIFFTE framework is run. The detector simulation's xml file is usually located in **\$NIFFTE/build/testDetSimConfig.xml** and has the following options:

- **TPCLatching**
  - NumberOfClocks - Defines the number of clocks that run the TPC
- **TPCDiffusion**
  - GeometryUsing - Know which geometry is being used
- **TPCPedestalNoise**
  - Sigma - Sigma for Gaussian noise added to the MCHit energy depositions in GeV
- **TPCPedestalSubtraction**
  - PedestalValue - Know where to put the old zero level in GeV
- **TPCChargeSharing**
  - SharedFractionPerNeighbor - Fraction of charge (actually Edep) to be capacitively shared with each neighboring pad
  - GeometryUsing – Know which geometry is being used (also used in TPCDiffusion)
- **TPCRandomDigits**
  - Sigma - Sigma for Gaussian noise added to a random digit in ADC
  - NumberOfDigits - The number of random hits to be added

### ■ 3.2.3 Standard Track Reconstruction

---

This module has parameters defined in the **.xml** file that will be defined by the **-c** argument when the NIFFTE framework is run. The detector simulation's xml file is usually located in **\$NIFFTE/build/configStd.xml** and has the following options:

- **TPCClusterFinder**
  - Threshold - Do not include digits below this threshold (units of ADC)
  - Geometry - Use the hexagonal geometry for cluster searching
  - MinNumOfDigits - Minimum number of digit needed to be considered a cluster



- **TPCHitFinder**
  - PeakADCThreshold - In determining peaks, only take those above this value (units of ADC)
  - PeakRadius - Keep digits within this radius around the peak
  - HitFindMode - Either Standard or Simple for mode of HitFinding
  
- **TPCTrackFinder**
  - LeeWay - Double measuring how much distance each hit can be apart in a track
  - SearchRadius - This is the starting search radius when the track finder starts (measurement in cm)
  - NarrowRatio - This is the % search radius will be narrowed at each bucket traversal (0.9 being 90% of former radius)
  
- **TPCTrackFitter**  
(No parameters here)

### ▪ 3.2.4 Hough Track Reconstruction

---

This module has parameters defined in the **.xml** file that will be defined by the **-c** argument when the NIFFTE framework is run. The detector simulation's xml file is usually located in **\$NIFFTE/build/configHough.xml** and has the following options:

- **TPCHoughD**
  - HoughRadius - First Hough radius used to pick tracks over Y-Z plane
  - HoughRadius2 - Second Hough radius used to pick tracks over X-Z plane
  
- **TPCKalman**
  - BlowupFactor – Re-scales the initial error matrices
  - ProcessNoise - Process noise assumed (from multiple scattering)

# 4. ANALYSIS

---

## 4.1 Terminal View of Reconstruction

---

Once you have the reconstruction result, it is now possible to see what kind of result the output file has by running a root macro. There is a simple list macro that resides in `$NIFSTE/ana/macro/list.C` that can show you what kind of data you have in the output `.root` file. Here is an example output display with this command line argument:

**Terminal Input** : `root -b -q '$NIFSTE/ana/macros/list.C("reco.root")'`

Notice that `reco.root` can be changed to point any other `.root` file you have access to.

**Terminal Output:**

*Loading Nifste Root Utilities*

*Processing \$NIFSTE/ana/macros/list.C("reco.root")...*

*#### Event 0 ####*

*There are 3 mc particles*

*There are 46011 mc hits*

*There are 1818 digits*

*There are 84 clusters*

*There are 84 hits*

*There are 6 tracks*

*Particle name is Ru101*

*Particle name is He4*

*Particle name is He4*

*...*

*..*

*.*

### ■ 4.1.1 What to Use This For

---

This macro is very useful for quickly checking what kind of content is inside of a `.root` file. It lists all of the `mcparticles`, `digits`, `clusters`, `hits` and `tracks` contained in the specified `.root` file.

## 4.2 Using the Event Display

---

The event display allows the user to visualize the results of the simulation. The display will show where the digits, clusters, hits, tracks, and MC particles are located in the reactor chamber for different event trials. This allows the user to compare his or her track findings against the simulation to check for accuracy or for any errors based on the alignment and the position of these points.

### ■ 4.2.1 Running the Event Display

---

To run the event display, enter the following code:

```
$ root '$NIFTE/ana/macros/testGeoEVD.C("$NIFTE/reco.root")'
```

Now the event display will show up. (Notice the input ("*\$NIFTE/reco.root*")' activates this command.)

### ■ 4.2.2 Using the Event Display

---

Once the event display is up and running, you will see a sidebar on the left and the event display figure on the right side of your screen.

#### **The Sidebar**

The sidebar allows you to manipulate what you want to display. To do this, press the **Eve** tab in the side bar. Pressing this tab will expand a list of what is displayed. The list includes digits, clusters, hits, the number of tracks, and MC particles. Next to all of these items is a check box . Check next to the items you want displayed. Unchecking them will hide what you don't want to see. You can reconstruct the event display anyway you like.

#### **Event Display**

Click and drag the display image if you wish to change the orientation of the view. Use the mouse scroll to zoom in or out to adjust the size appearance of the display. The red line is the particle trajectory and is mainly used to compare the tracks against the simulated version. Make sure it is aligned with the digits, clusters, etc., otherwise it is a problem.

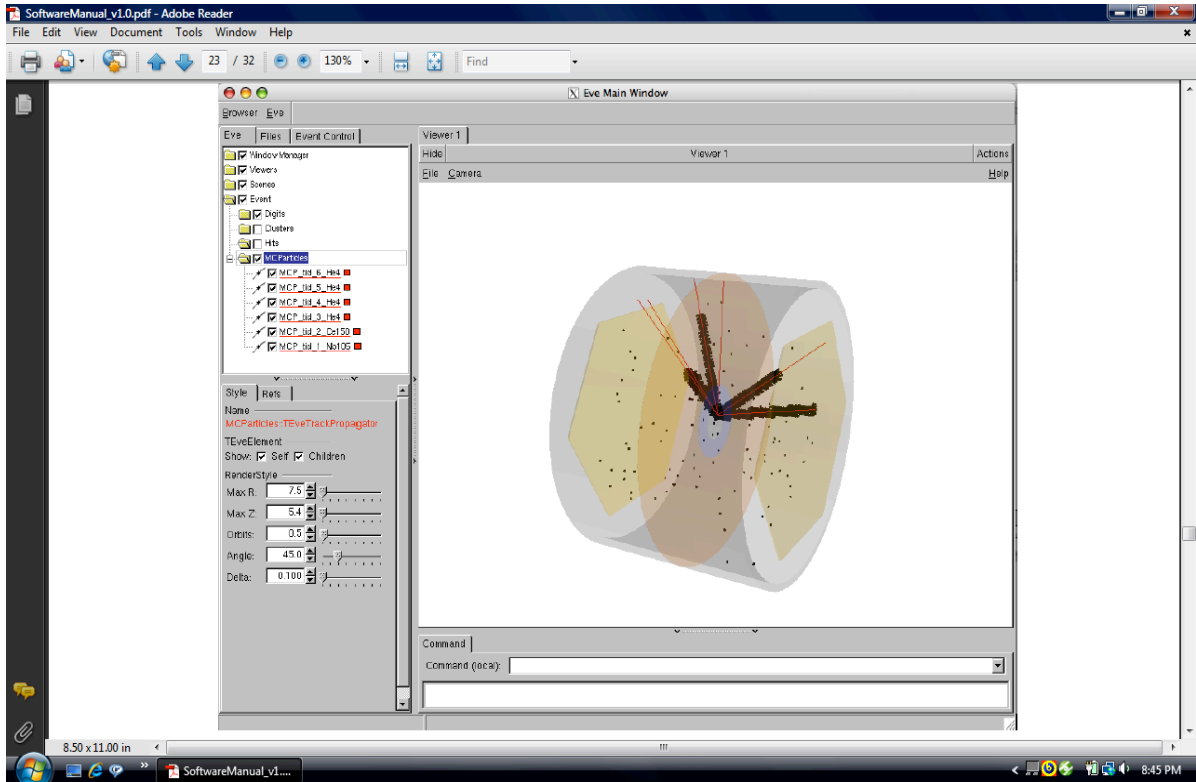


Figure 3: NIFFTE Event Display

### 4.2.3 What to Use This For

The event display is used for more in-depth analysis. The software developers will most likely use this to find out very hard-to-find bugs. Physicists will definitely use this to do the actual analysis of meaningful data once we start reconstructing actual data.

## 4.3 Writing a Macro in ROOT for Analysis

There are many examples of tools needed to create analysis macros in the **\$NIFFTE/ana/macros** folder. Physicists are assumed to have the skills to write one to fit each one's needs. If you need assistance in writing a ROOT macro file, please refer to the ROOT website listed on the resource section.

## 4.4 What Does All of This Mean?

There are a few primitive analysis tools in the **SVN repo** to get the physicists started. However, the collaboration expects each contributor to write their own ROOT macro and do their own analysis.

# 5. TROUBLESHOOTING/FAQ

---

## 5.1 Common Mistakes

---

Here are some of the most common mistakes while operating the NIFFTE software:

### ▪ 5.1.1 Terminal Command Line Mistakes

---

There are several error messages that NIFFTE can produce and most of them will be in the last few lines outputted by the program. This section deals with problems related directly to the command you give NIFFTE. The Options section in the Appendix deals with problems related to your XML configuration file.

For example:

```
E-TPCTaskConfig.cXX-TPCTaskConfig: Problem with configuration. Exiting.
```

is the last line of output from NIFFTE.

Additionally, if NIFFTE recorded **0 events**, then NIFFTE probably didn't complete properly.

➤ **Message:** *Error in <TFile::TFile>: file deSim.root does not exist  
E-NiffteRootIO.cxx-SetupInputFile: Failed to open file: deSim.root for read  
I-NiffteRootIO.cxx-Close: Wrote 0 events. Closing file.*

**Fix:** This message means that NIFFTE wasn't able to find and open your input file, double check the name and the file's location.

➤ **Message:** *E-NiffteTaskManager.cxx-SaveEvent: Unrecognized file extension. Choose a  
file of type \*.root (ROOT) or \*.dat (ASCII)  
I-NiffteRootIO.cxx-Close: Wrote 0 events. Closing file.*

**Fix:** This message means that your output file did not have the file type of .root. Check the command you gave NIFFTE for **-o someFile.root**.

## ▪ 5.1.2 Options Mistakes

---

Anytime NIFFTE outputs **Wrote 0 events. Closing file.**, there was probably some sort of problem. If NIFFTE wrote 0 events, then it didn't do anything, either because there were no events to record (rare) or there was some problem.

➤ **Fix:**

- 1) Check your input/output files; sometimes NIFFTE will complain if you give it a bad input file, sometimes it won't. If you have repeated problems with this consider adopting a naming convention or use separate folders for each type of root file. (detector sim output in one folder, reconstruction output in another).
- 2) Check your configuration (something.xml). NIFFTE doesn't check the values. It has defaults if you specify nothing, but it will work with anything you specify. For example: *PeakRadius* in TPCHitFinder is usually quite low, but it will accept a value of 3 million.

➤ **Message:** *E-TPCTaskConfig.cXX-TPCTaskConfig: Problem with configuration. Exiting.*

**Fix:** This message means that something was wrong with your configuration (something.xml) file. Unless you have worked with XML before, it is best to restore from an older configuration file and change the values again.

## 5.2 Known Bugs

---

- The NIFFTE framework segfaults if a **-d** argument is given without any number after.
- TrackFinder module's track finding algorithm does not consider slope calculations. It only does distance based neighbor search which can result in 'V' shaped tracks.
- The event display macro gives out a GL related warning message on terminal; however, no harm is done.

## 5.3 Missing Functionalities

---

- HitFinder module's standard mode option is missing, please use the simple mode until the module is fixed. This results in terrible Z space resolution which will cause a track to only have two or three hits.
- The module's HitFitter and TrackFitter are completely missing and are not even compiled at this time. There is currently no data fitting being done, which means the event display cannot display the projected track slope.

# 6. RESOURCES

---

## 6.1 Contact E-mails

---

- ❖ **Jennifer Klay – Software leader**  
jklay@calpoly.edu
- ❖ **Ryuhō Kudo – Documentation author and standard reconstruction module developer**  
rkudo@calpoly.edu
- ❖ **Shon Watson – Physics simulation and event display developer**  
tsw00c@acu.edu
- ❖ **Eric Burgett – Fission generator developer and cluster manager**  
eric.burgett@nre.gatech.edu
- ❖ **Saravagya Sharma - Hough/Kalman reconstruction module developer**  
sarvagya.s@gmail.com
- ❖ **Tyler Thornton – Detector simulator developer**  
rtt06a@acu.edu

## 6.2 Websites

---

- ❖ **Main wiki for NIFFTE Collaboration**
  - <http://nuclear.calpoly.edu/~niffte/index.html>
- ❖ **SVN Repository for NIFFTE**
  - <http://nuclear.calpoly.edu/~niffte/viewvc.cgi/?root=NIFFTE>
- ❖ **Basic website about running a bash terminal**
  - <https://help.ubuntu.com/9.04/basic-commands/C/>
- ❖ **Explanation of what TPC is**
  - <http://cerncourier.com/cws/article/cern/29014>
  - <http://www.answers.com/topic/time-projection-chamber>
  - [http://en.wikipedia.org/wiki/Time\\_projection\\_chamber](http://en.wikipedia.org/wiki/Time_projection_chamber)
- ❖ **ROOT Resources**
  - <http://root.cern.ch/drupal/>

# 7. APPENDIX

---

## 7.1 Subroutines

---

### ▪ 7.1.1 Geant4 Physics Simulation

---

Physics simulation is the module that simulates a physical fission event using Geant4 and output a file with all the results in it. Geant4 is a software package for simulating the passage of particles through matter. The module will simulate that a radioactive matter, most likely Californium, is naturally decaying in a pressurized TPC chamber. The fission fragments and alpha particles that appear from Californium will then react with the gas and make a trail of ionized gas. That charged gas will then drift, due to electromagnetic field, to sensors located at both ends of TPC chamber.

### ▪ 7.1.2 NIFFTE Detector Simulation Modules

---

Simulation Modules will then get the input from physics simulation and simulate the various distortions and limitations the hardware sensors will have. The result will then be written to a file. This simulation includes physical phenomenon, such as charge coupling, as well as electronic inaccuracies, such as random noise and TPC latching.

### ▪ 7.1.3 NIFFTE Reconstruction Modules

---

The Reconstruction modules will take the detector simulation and try to reconstruct what the trails of particles looked like. The track information that was reconstructed is written to a file.

## 7.2 Options

---

The following are the example macro and .xml files that define options for the modules:

### ▪ 7.2.1 Geant4 Physics Simulation

---

```
# The first integer seed should be within the range 0 to 2147483647 for
# the HepJamesRandom engine. The value of the second integer does not
# seem to matter for HepJamesRandom, but it must be present.
#/random/setSeeds 987654321 0
```



```

#####
# Define the alpha source
#
# A monoenergetic alpha randomly placed inside a volume which is the
# same size and location as the target in the simulation.
# Alphas are emitted isotropically from this volume.
#####
/gps/particle alpha
/gps/energy 7.244 MeV
/gps/pos/type Plane
/gps/pos/shape Circle
/gps/pos/centre 0. 0. 0. cm
/gps/pos/radius 0.1 cm
/gps/ang/type iso

# Add an additional alpha in the same manner
# This can also be used to add different particles to the same event
/gps/source/add 1
/gps/particle alpha
/gps/energy 8.344 MeV
/gps/pos/type Plane
/gps/pos/shape Circle
/gps/pos/centre 0. 0.3 0.1 cm
/gps/pos/radius 0.1 cm
/gps/ang/type iso

#####
# Define the isotope and place it randomly in the drift volume
# Isotopes will be emitted isotropically from their start location
#####
/gps/source/add 1
# new way of specifying ion: Z A Q E
# Q = charge in units of e, E = excitation energy
/gps/ion 44 101 1 0.
/gps/energy 58.33 MeV
/gps/pos/type Volume
/gps/pos/shape Sphere
/gps/pos/centre 0. 0. 0. cm
/gps/pos/radius 3. cm
/gps/ang/type iso

# This tells the generator to allow them to start from different locations
/gps/source/multiplevertex true

/tracking/verbose 1
/run/beamOn 1

```

## ■ 7.2.2 NIFFTE Detector Simulation Modules

---

```

<?xml version="1.0" encoding="UTF-8"?>
<TestDetSimConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="reco/NiffteRecoConfig.xsd">

```

```

  <Modules type="DetSim"> TPCLatching TPCDiffusion TPCChargeSharing
TPCDigitizer TPCRandomDigits NiffteCheckPad </Modules>

```

```

<Config name="TPCLatching" version="default">
  <Param name="NumberOfClocks"><int> 1 </int>
    Number of clocks that run the TPC
  </Param>
</Config>
<Config name="TPCDiffusion" version="default">
  <Param name="GeometryUsing"><string> hex </string>
    Knowing which geometry is being used
  </Param>
</Config>
<Config name="TPCPedestalNoise" version="default">
  <Param name="Sigma"><double> 2.0e-9 </double>
    Sigma for Gaussian noise added to MCHit energy depositions in GeV
  </Param>
</Config>
<Config name="TPCPedestalSubtraction" version="default">
  <Param name="PedestalValue"><double> 1.0e-9 </double>
    Where to put the old zero level in GeV.
  </Param>
</Config>
<Config name="TPCChargeSharing" version="default">
  <Param name="SharedFractionPerNeighbor"><double> 0.02 </double>
    Fraction of charge (actually Edep) to be capacitively shared with
    each neighboring pad
  </Param>
  <Param name="GeometryUsing"><string> hex </string>
    Knowing which geometry is being used
  </Param>
</Config>
<Config name="TPCRandomDigits" version="default">
  <Param name="Sigma"><double> 50 </double>
    Sigma for gaussian noise added to a random digit in ADC
  </Param>
  <Param name="NumberOfDigits"><int> 0 </int>
    Number of random hits to be added
  </Param>
</Config>
</TestDetSimConfig>

```

### ■ 7.2.3 NIFFTE Standard Reconstruction

---

```

<?xml version="1.0" encoding="UTF-8"?>
<ConfigStd xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="reco/NiffteRecoConfig.xsd">

  <Modules type="Reco"> TPCClusterFinder TPCHitFinder TPCTrackFinder
</Modules>

  <Config name="TPCClusterFinder" version="default">
    <Param name="Threshold"><int> 2 </int>
      Do not include digits below this threshold (units of ADC).
    </Param>
    <Param name="Geometry"><string> Hex </string>
      Use the hexagonal geometry for cluster searching.
    </Param>

```

```

    <Param name="MinNumOfDigits"><int>2</int>
      Minimum number of digit needed to be considered a cluster.
    </Param>
  </Config>

  <Config name="TPCHitFinder" version="default">
    <Param name="PeakADCThreshold"><int> 4 </int>
      In determining peaks, only take those above this value (units of
      ADC).
    </Param>
    <Param name="PeakRadius"><int> 1 </int>
      Keep digits within this radius around the peak.
    </Param>
    <Param name="HitFindMode"><string> Simple </string>
      Either Standard or Simple for mode of HitFinding
    </Param>
  </Config>

  <Config name="TPCTrackFinder" version="default">
    <Param name="LonerRadius"><double>1.0</double>
      double measureing how much distance each hit can be apart in a track
    </Param>
    <Param name="LonerCount"><int>0</int>
      int number of allowed hits with in the LonerRadius distance
    </Param>
    <Param name="InitialSphereRadius"><double>1</double>
      double deciding what is the initial radius when looking for the tail
      end of track
    </Param>kay
    <Param name="InitialMinimumSphereCount"><int>5</int>
      int number of minimum hits to be considered a track
    </Param>
    <Param name="MaxHitSlopeDistance"><double>0.25</double>
      double signifiying the maxmimum distance a hit can be away from the
      slope of track
    </Param>
    <Param name="MaximumChiSq"><double>1.5</double>
      double signifiying the maximum chi2 that's allowed in the final
      track
    </Param>
    <Param name="MaxHitDistance"><double>0.5</double>
      double signifiying the maximum chi2 that's allowed in the final
      track
    </Param>

  </Config>
</ConfigStd>

```

## ■ 7.2.4 NIFFTE Hough Reconstruction

```

<?xml version="1.0" encoding="UTF-8"?>
<ConfigStd xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="reco/NiffteRecoConfig.xsd">

  <Modules type="Reco">TPCHoughD TPCKalman</Modules>

```

```
<Config name="TPCHoughD" version="default">
  <Param name="HoughRadius"><double>0.6</double> <!-- 1 -->
    First Hough radius used to pick tracks over Y-Z plane
  </Param>
  <Param name="HoughRadius2"><double>0.4</double> <!-- .8 -->
    Second Hough radius used to pick tracks over X-Z plane
  </Param>
</Config>
<Config name="TPCKalman" version="default">
  <Param name="BlowupFactor"><double>10</double>
    Re-scales the initial error matrices
  </Param>
  <Param name="ProcessNoise"><double>1</double>
    Process noise assumed (from multiple scattering)
  </Param>
</Config>
</ConfigStd>
```

# 8. GLOSSARY

---

## **Digit:**

Digit are data containers used in the Reconstruction module. They represent a single sensor output with information such as padrow, padcoloum, bucket, volume, and ADC. The first 4 are position information and the last one is a measurement of the strength of the charge the sensor picked up.

## **Cluster:**

Cluster are data containers used in the Reconstruction module. Clusters are literally 'clusters' of digits. It has information such as list of digits used to form the cluster, cluster ID, cluster average location and total sum of ADC charge.

## **Hit:**

Hit are data containers used in the Reconstruction module. Hits are made from clusters. This data member is supposed to be what the actual ionized gas is supposed to be. It has information such as (X,Y,Z) positions, QTotal charge, QADC charge, the errors (delta) of all the information it has, hit ID, what cluster the hit was made from.

## **Track:**

Tracks are data containers used in the Reconstruction module. Tracks are made up of 'track' of hits. A single track signifies a single particle's trajectory. It has information such as list of hits included in the track, the chi squared value for the line, positional parameters for the event display and track ID.

# 9. TABLE OF FIGURES

---

Figure 1: Diagram of Flow of Operation That Uses Simulation Input as Oppose to Using Hardware Input .....2

Figure 2: Diagram of Flow of Operation That Uses Hardware Input as Oppose to Using Simulation Input .....3

Figure 3: NIFFTE Event Display .....15

# 10. REVISION LIST

---

SVN Tag #	Date	Modifications/Notes	Author(s)
v1-00	12-Aug-09	Initial Manual Documentation	Ryuhō Kudo Jennifer Klay, PhD Andrew Barton Uriel Gomez Ryan Harch Samson Truong