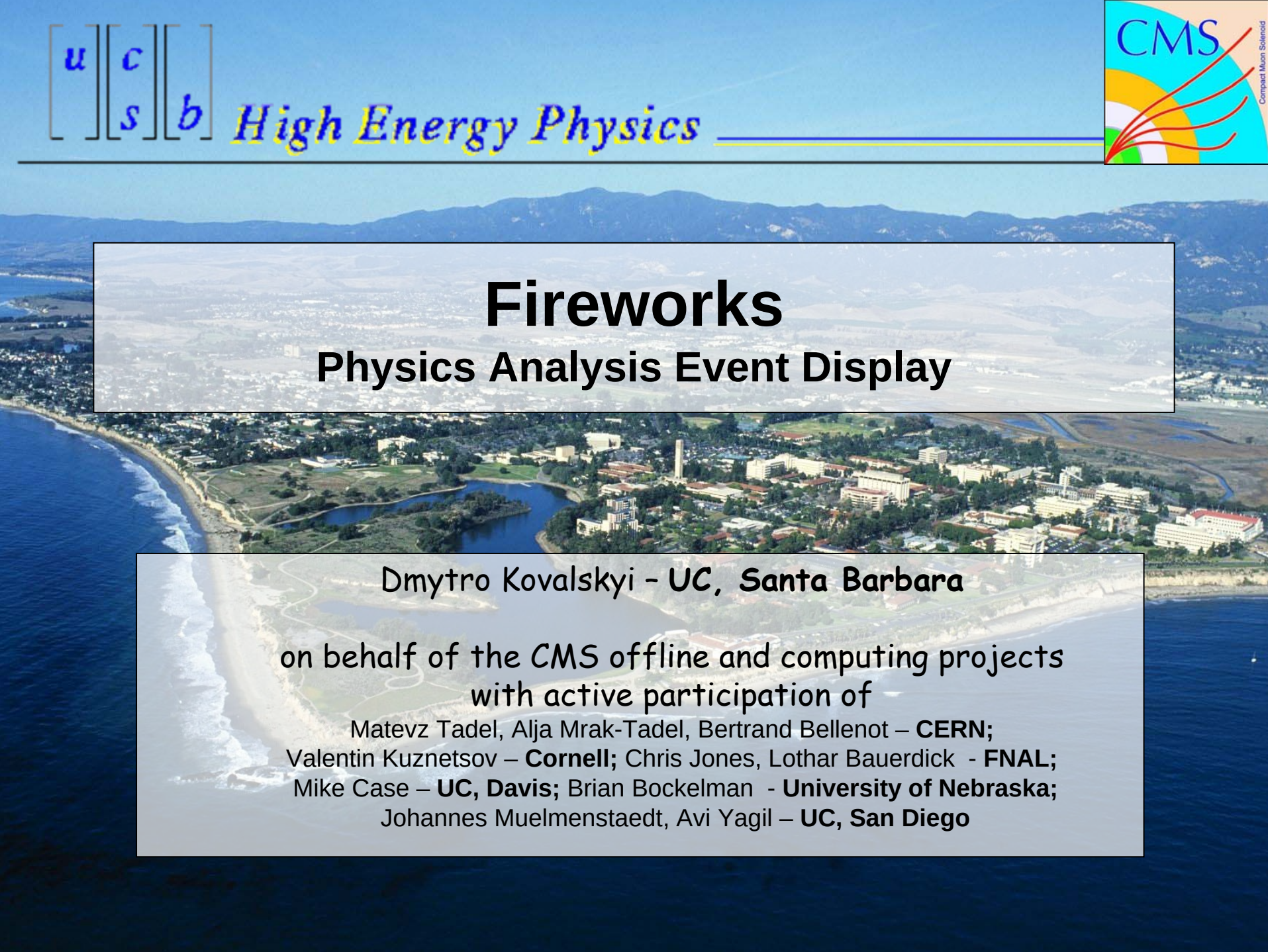


High Energy Physics

Fireworks

Physics Analysis Event Display

Dmytro Kovalskyi – **UC, Santa Barbara**

on behalf of the CMS offline and computing projects
with active participation of

Matevz Tadel, Alja Mrak-Tadel, Bertrand Bellenot – **CERN;**
Valentin Kuznetsov – **Cornell;** Chris Jones, Lothar Bauerdick - **FNAL;**
Mike Case – **UC, Davis;** Brian Bockelman - **University of Nebraska;**
Johannes Muelmenstaedt, Avi Yagil – **UC, San Diego**

Introduction

FIREWORKS

List View

Views | Physics objects | Triggers | Tracking

3D

- CMS has two event displays developed and supported by the collaboration, which complement each other:
  - Iguana – a general purpose full framework based application with ability to run reconstruction code on the fly. (See poster by Ianna Osborne)
  - Fireworks – a light-weight physics analysis oriented event display, showing only what is available in a standard CMS ROOT file.
- Fireworks development started in December 2007 and by Summer 2008 we had a first stable release available to users. By the end of 2008, the active development phase was replaced with maintenance and operation.
- The display is distributed as a stand alone application pre-build for Scientific Linux and Mac OS X.

Staying Focused

FIREWORKS

Views | Physics objects | Triggers | Tracking

3D

- In order to make the project manageable, we concentrated on a specific use case: physics analysis
- Staying focused on the primary objectives allowed us to:
  - Limit complexity – ideal geometry, no reconstruction on the fly, simple track propagation, trivial magnetic field model etc
  - Use schematics and abstraction where it's adequate and don't try to be absolutely precise unless it's necessary
- These choices allowed us to:
  - Present a prototype in 4 months and release first stable version in 6 months since project started
  - Make display light enough that it can be easily distributed as a single tar-ball with minimal number of external dependencies
  - Port display to Mac OS X (and Windows – prototype only)

Direct Data Access

FIREWORKS

Views | Physics objects | Triggers | Tracking

3D

- CMS data model is based on ROOT file format
- FWLite is ROOT plus a bunch of libraries with Reflex dictionaries, which allow direct data access from ROOT based applications.
- Our choice to show only what is stored in data allows us to use direct data access with limited resources.
- In practical terms it means a user downloads 100-200MB size event display ("player") and copies files of interest to her computer, which is all that she needs to get started.

ROOT - Eve

FIREWORKS

List View

Views | Physics objects | Triggers | Tracking

3D

- Last CHEP07 conference Matevz Tadel presented new ROOT package: Eve – a full scale framework for event visualization using OpenGL in ROOT
- Active participation of ROOT developers in Fireworks design and development allowed to extend Eve and add elements that can be shared by many experiments (Alice and CMS currently).
- Using Eve as a visualization framework we separated CMS specific tasks from object rendering, which is done by ROOT.
- This greatly speeds up the development and allowed physicists to contribute directly without special knowledge of such advanced technologies as OpenGL

ROOT Reflex

FIREWORKS

Views | Physics objects | Triggers | Tracking

3D

- ROOT Reflex is used extensively in the event display
- It allows for rich object and event filtering capabilities, where methods are not hard-coded, but user defined
- Given large number of methods available for most physics object classes, we use Reflex to help users to explore available methods and use "Tab"-completion to improve usability
- Reflex also allows us to explore inheritance structure of the data object types. As a result we can use the base class rendering if we do not find a render for the specific class.
- In practice almost all CMS physics object classes inherit from a single base class (Candidate), which provide most basic information like 4-momentum of a particle. So any new data can be shown in some simple way without writing new code.

Usability

FIREWORKS

List View

Views | Physics objects | Triggers | Tracking

3D

- Typical use case – looking at events for a physics analysis on a laptop traveling to/from CERN.
- Make it simple and usable
  - Given that event displays are not everyday tools, program interface is optimized for intuitive and simple use.
  - 3D accelerator is recommended, but not required. The event display is routinely used and tested on non-accelerated computers to make sure that performance is reasonable.
  - The display is distributed as a stand alone player with all necessary components including FWLite in a compact form.
  - No need for remote use over X11, which should be avoided on slow networks anyway.

Design Choices Summary

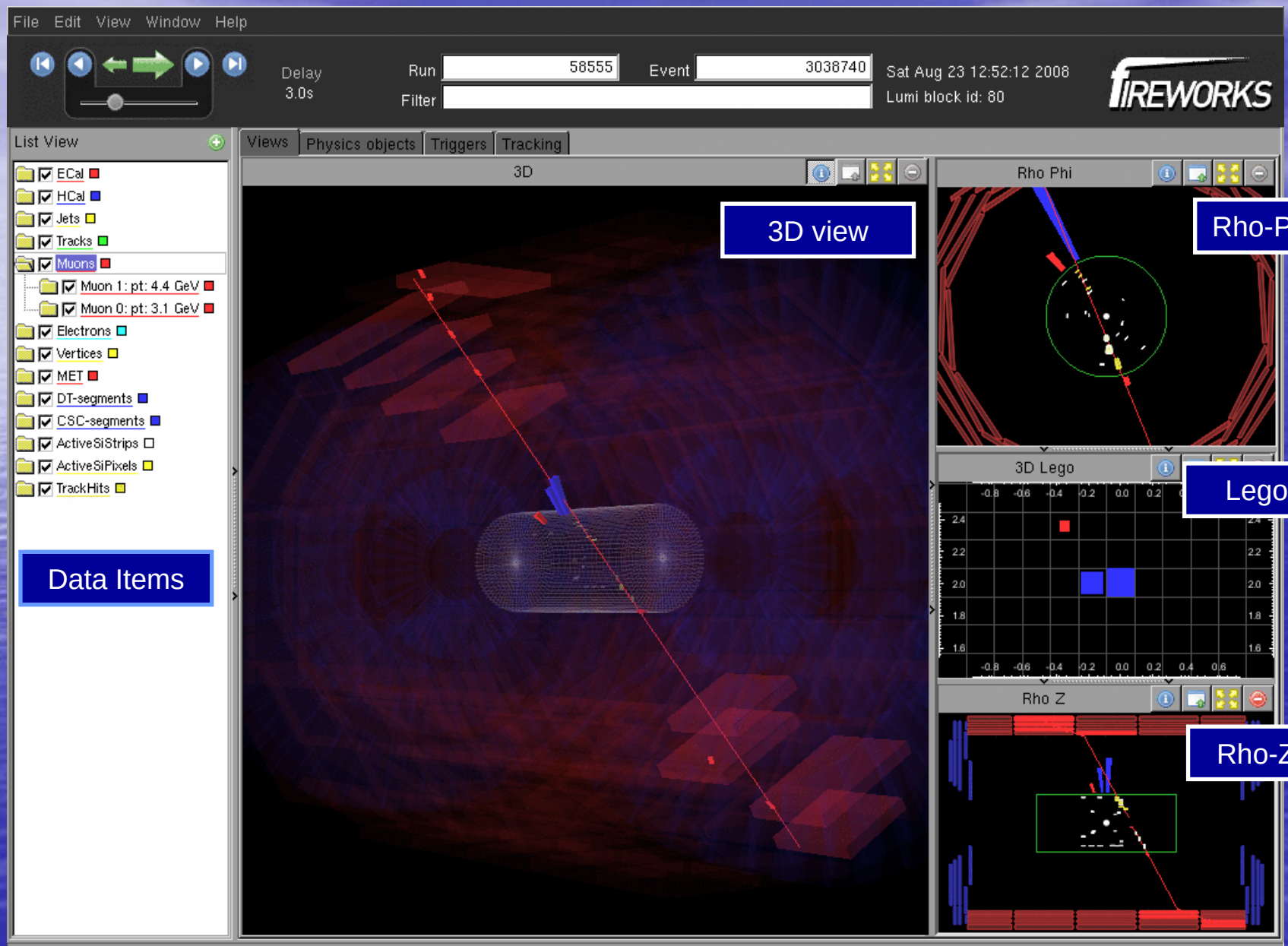
FIREWORKS

Views | Physics objects | Triggers | Tracking

3D

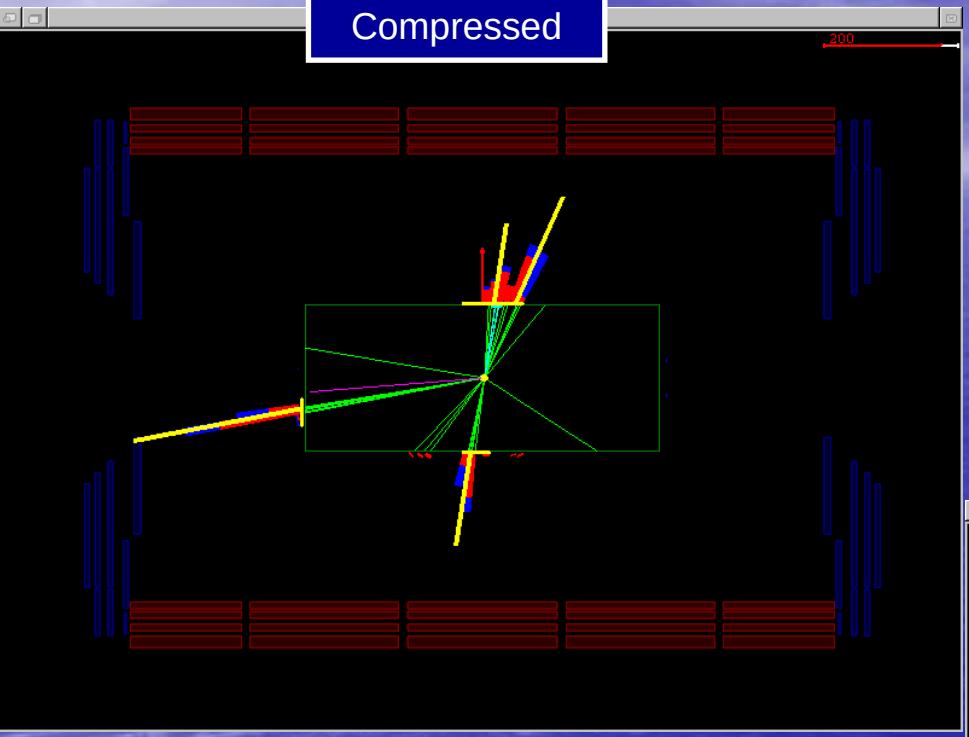
- Fireworks relies on:
  - CMS side
    - EDM - Event Data Model
    - Code Factorization – Data Formats are kept separately
    - FWLite – ROOT with CMS libraries to access data
  - ROOT side:
    - Eve – visualization framework
    - Reflex – ability to explore object information in C++
- Key Design Choices:
  - Show only available information (no reconstruction on demand)
  - Ability to display events off-line without access to databases and event stores (Airplane test)
  - No intermediate data formats – you see what you have in data, the way it is stored and processed. Less room for "operator error".

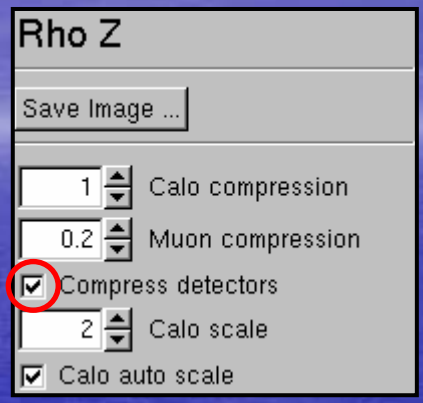
Display overview



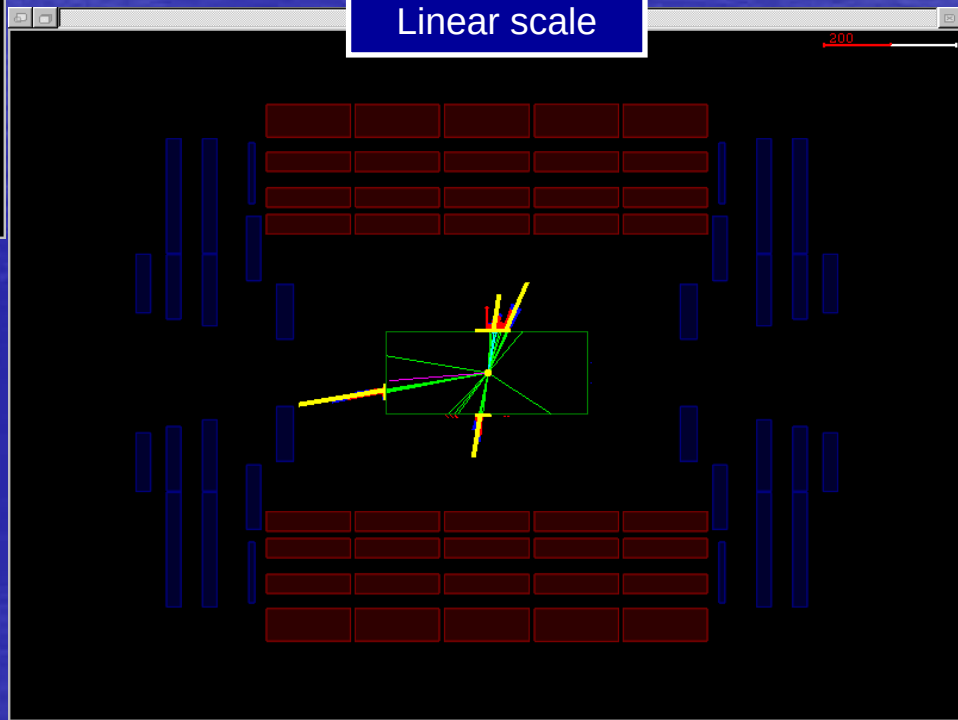
Rho-Z View Options

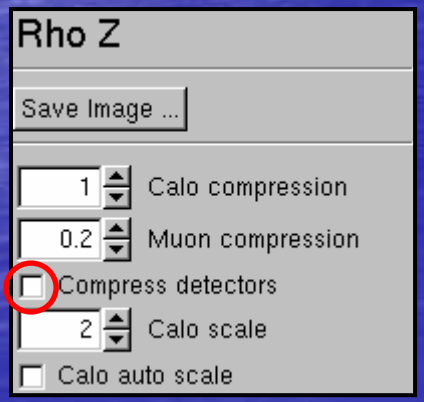
Compressed





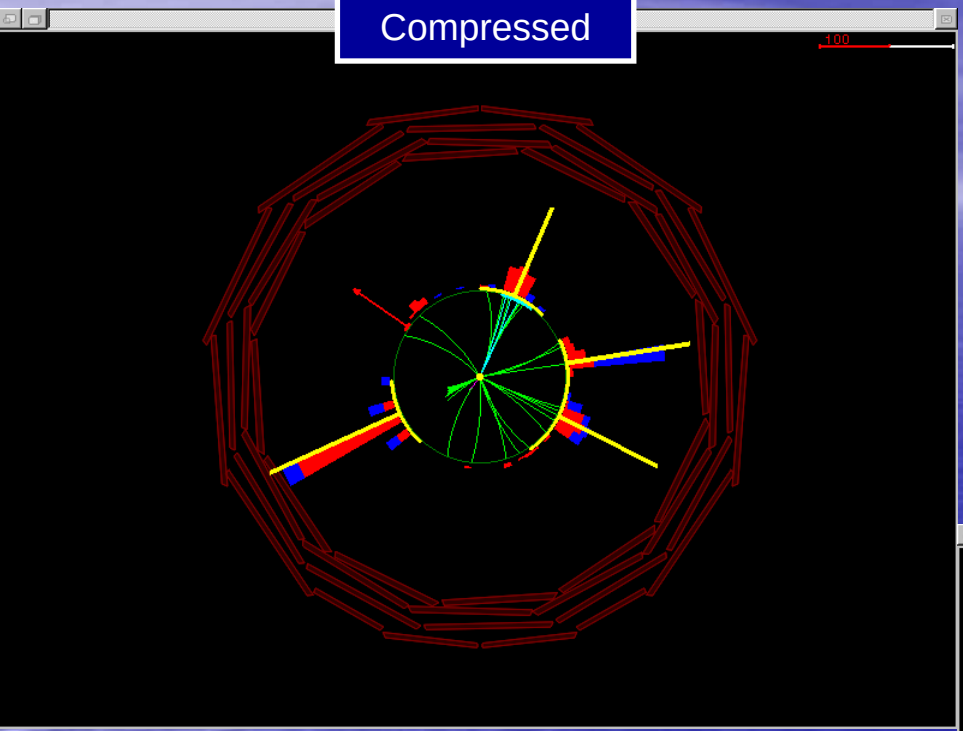
Linear scale





Rho-Phi View Options

Compressed



Rho Phi

Save Image ...

1 Calo compression

0.2 Muon compression

☑ Compress detectors

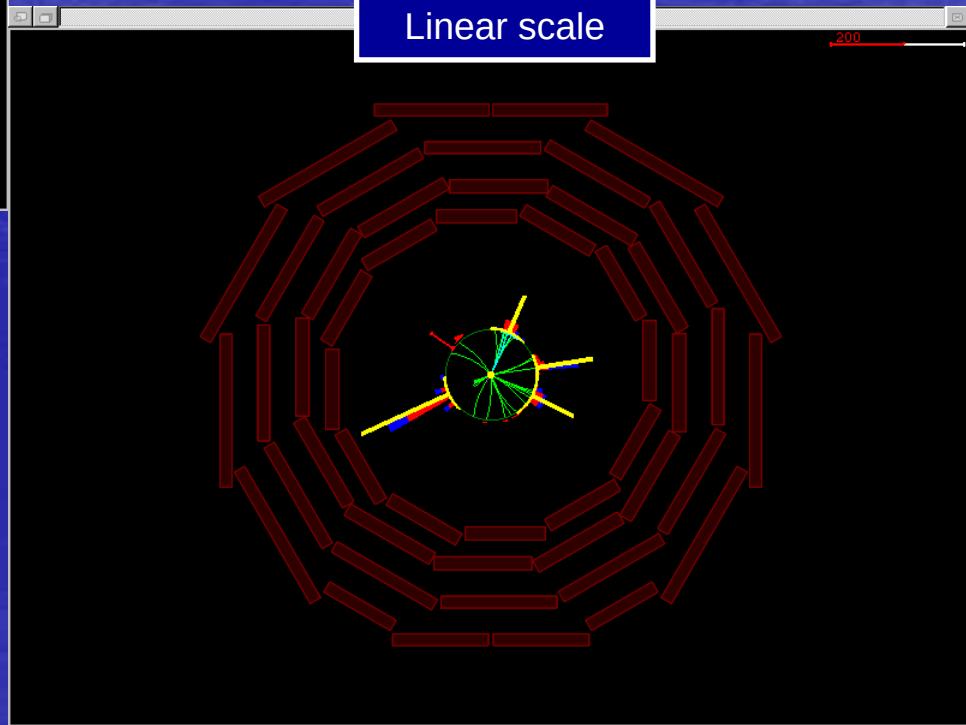
2 Calo scale

☑ Calo auto scale

☑ Show calo endcaps

☐ Show HF

Linear scale



Rho Phi

Save Image ...

1 Calo compression

0.2 Muon compression

☐ Compress detectors

2 Calo scale

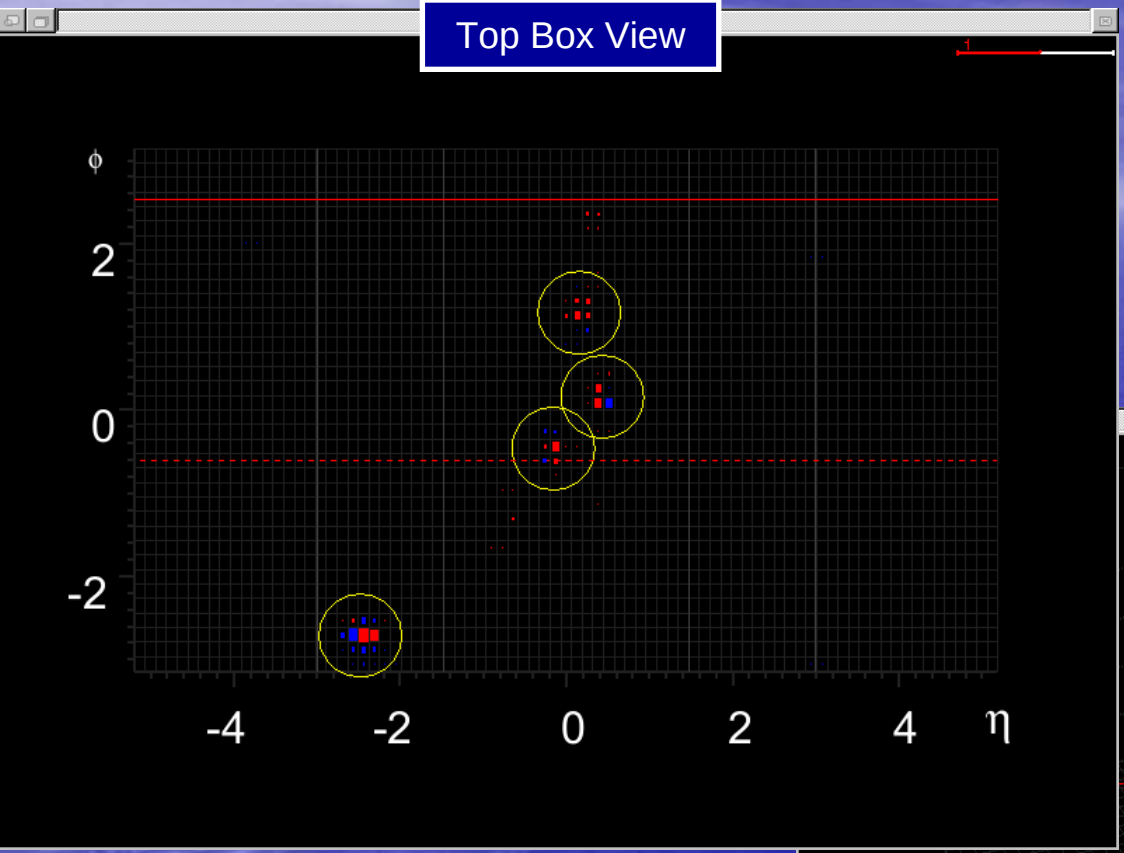
☐ Calo auto scale

☑ Show calo endcaps

☐ Show HF

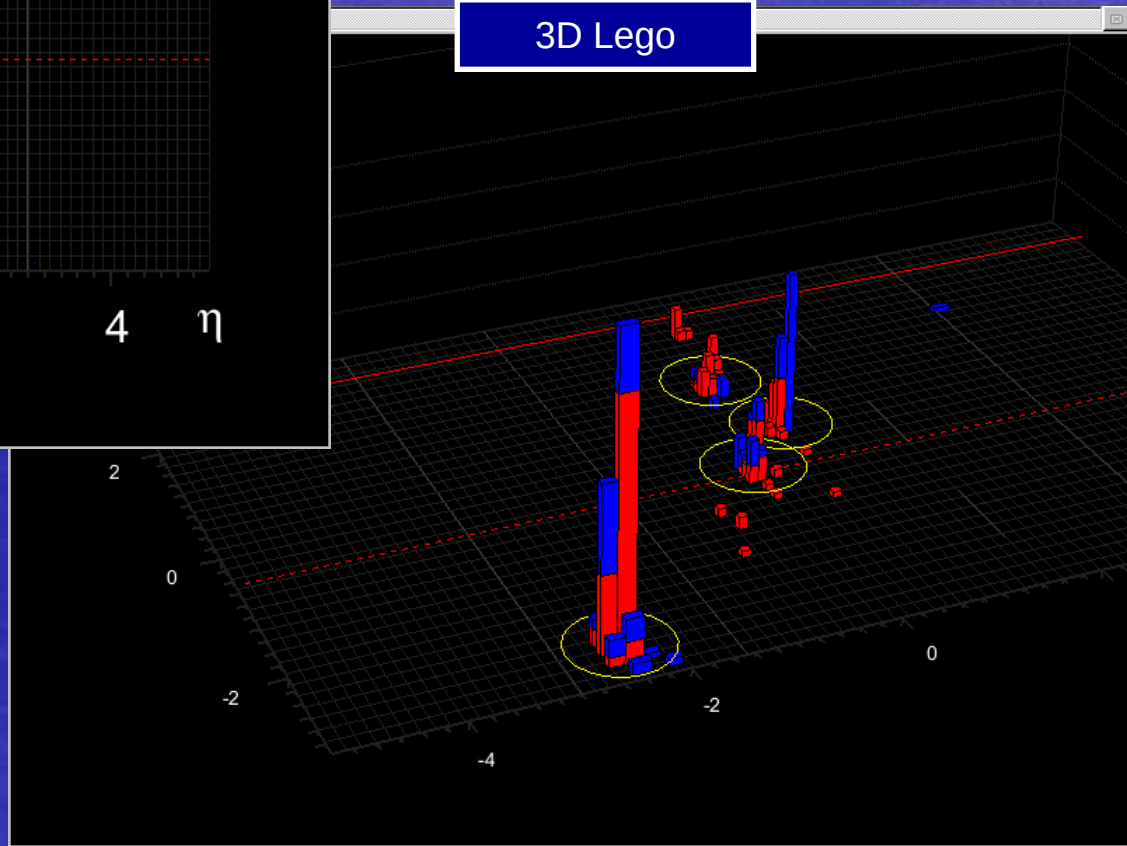
Lego View Options

Top Box View



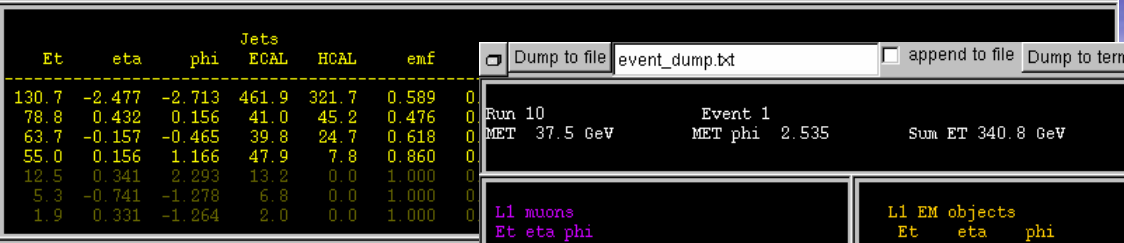
- Top view shows color of the dominant energy contribution in each bin using log scale to map total energy to box size
- Bin size is automatically adjusted depending on the size of the view area

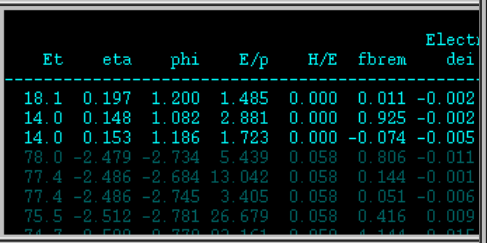
3D Lego

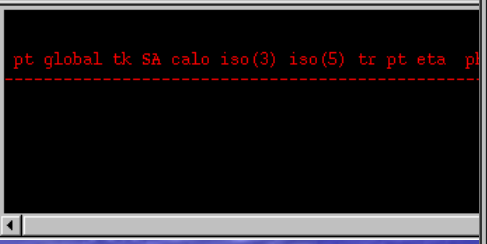


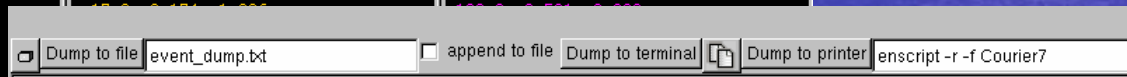
- At high resolution (large view area) binning is done according to the trigger tower size
- Currently the smallest bin is 2x2 tower, but this will be changed as soon as we test new code to support smaller bins

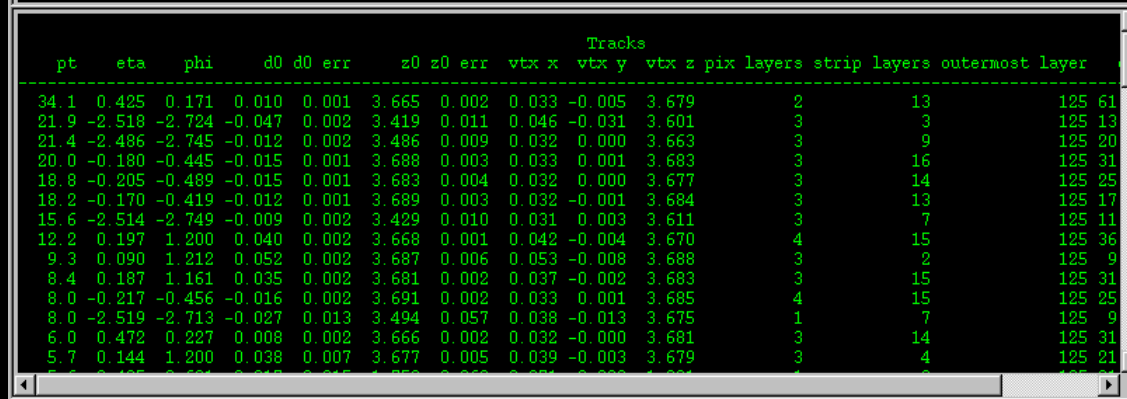
Text View Options

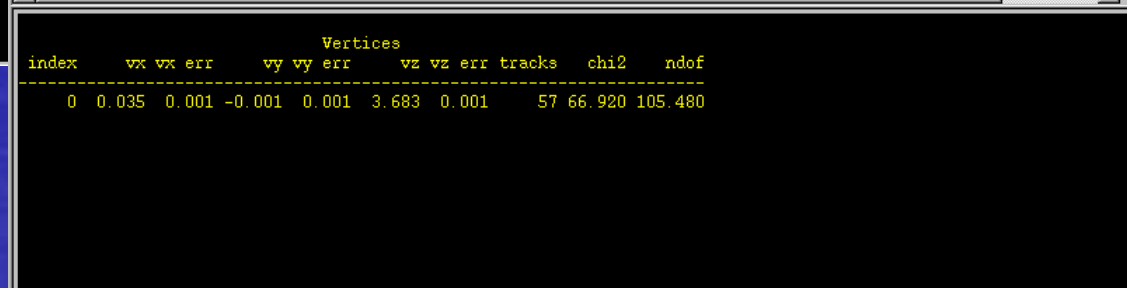












- All columns are sort-able
- Objects can be selected and selection propagates to all other views

Plug-in System

FIREWORKS

Views | Physics objects | Triggers | Tracking

3D

- Plug-in's allow users to write their own code to visualize what they need.
- Input is a vector of objects, called a collection, which represents a typical data unit in CMS.
- Output is a ROOT Eve container of graphical primitives that ROOT knows how to render.
- User contribution is the code that translates the CMS data object to a set of graphical primitives. Ideal geometry is available for most detectors.
- The plug-in's can be compiled without touching the main code with no exposure to internals of the display.

Remote Data Access

FIREWORKS

List View

Views | Physics objects | Triggers | Tracking

3D

- Data access in not a problem specific to event displays, but nevertheless ignoring it may render even the best display useless.
- CMS uses distributed storage model with data distributed across the world.
- We are trying to make sure that users will get access to events that they need.
- "Pick Event" – service to find and extract a few events with known identifiers like run and event number.
  - Typical use case: found an interesting event in ntuples and want to look at it in details.
- "Random Access" – access to any event within reasonable amount of time anywhere in the world. The main reason why it can be done is that in principle we don't need that much information and requests are rare.

Conclusion

Views | Physics objects | Triggers | Tracking

3D

- CMS developed a new light-weight event display - Fireworks
- Direct participation of ROOT developers in the project allowed for quick implementation and integration of new ideas.
- Fireworks became a working tool for many users at CMS. We get lots of positive feedback and it becomes normal to see Fireworks pictures in presentations.
- Many things still can be done and many features implemented, but the main goal of having a useful tool for a physicist to analyze data is achieved.