Kalman filter algorithms for neutrino experiments

ICATPP 2011
Villa Olmo, 5/10/2011

Anselmo Cervera Villanueva
IFIC-Valencia

Why General Reconstruction Packages ?

- In general, every HEP experiment implements its own reconstruction software
- This was the case in the past for:
  - Simulation software
  - Analysis tools
  - Data bases
  - …
- But we have learned that this is not optimal
- Instead code reuse and general packages are preferred solutions

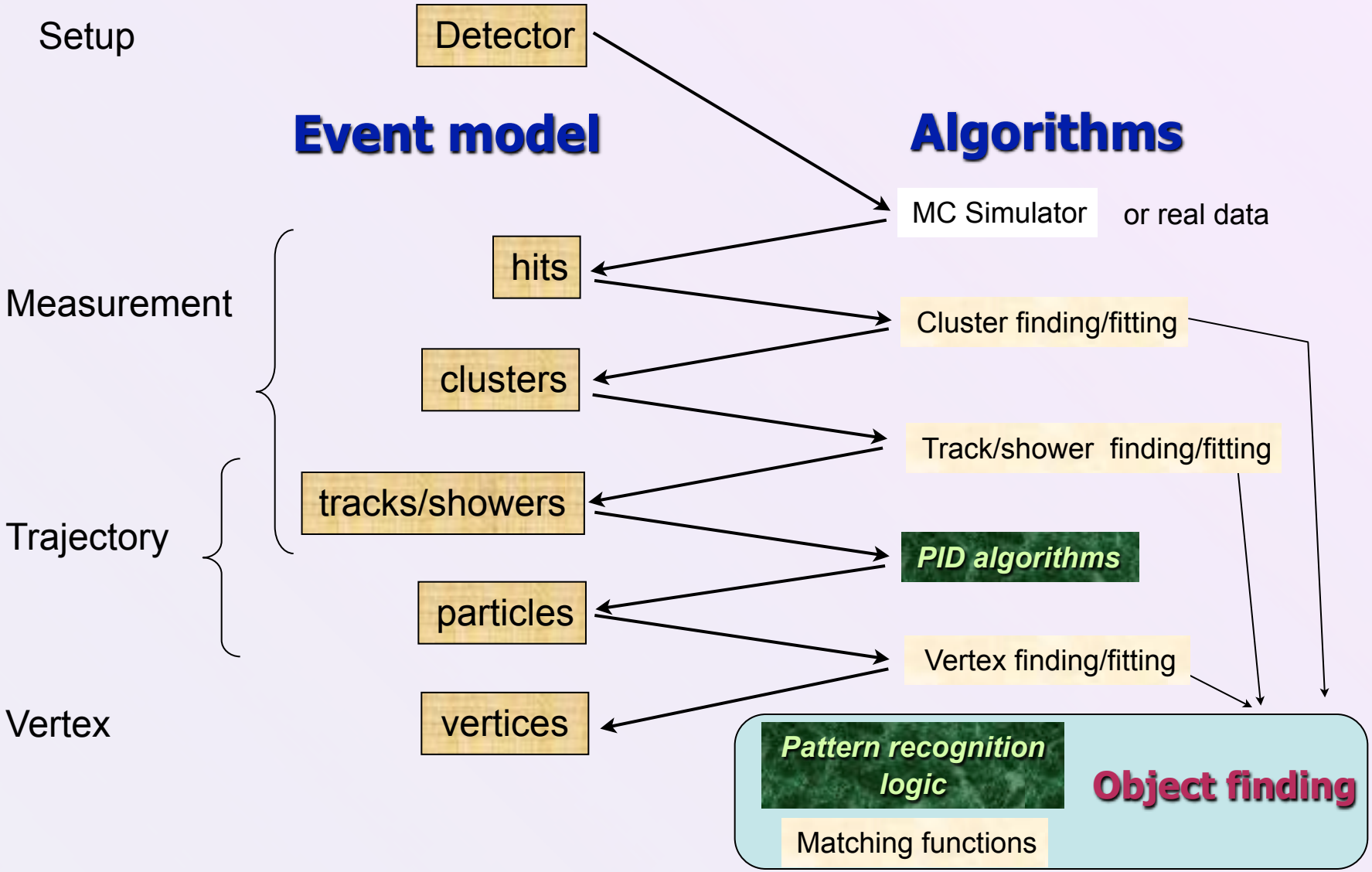
RecPack

- RecPack is a general reconstruction tool-kit
  - Track and vertex fitting
  - Propagation through complicate setups with multiple scattering, energy loss and inhomogeneous fields
  - Matching functionality
- Written in C++
- It separates data classes and algorithms
  - It's modular structure allows extensions
- Code can be downloaded from SVN repository
  - svn://neutrinos1.ific.uv.es:/usr/local/svn/recpack/alpha/trunk/recpack
- A web site exists (but it's a bit obsolete)
  - http://ific.uv.es/recpack (new web site under construction)

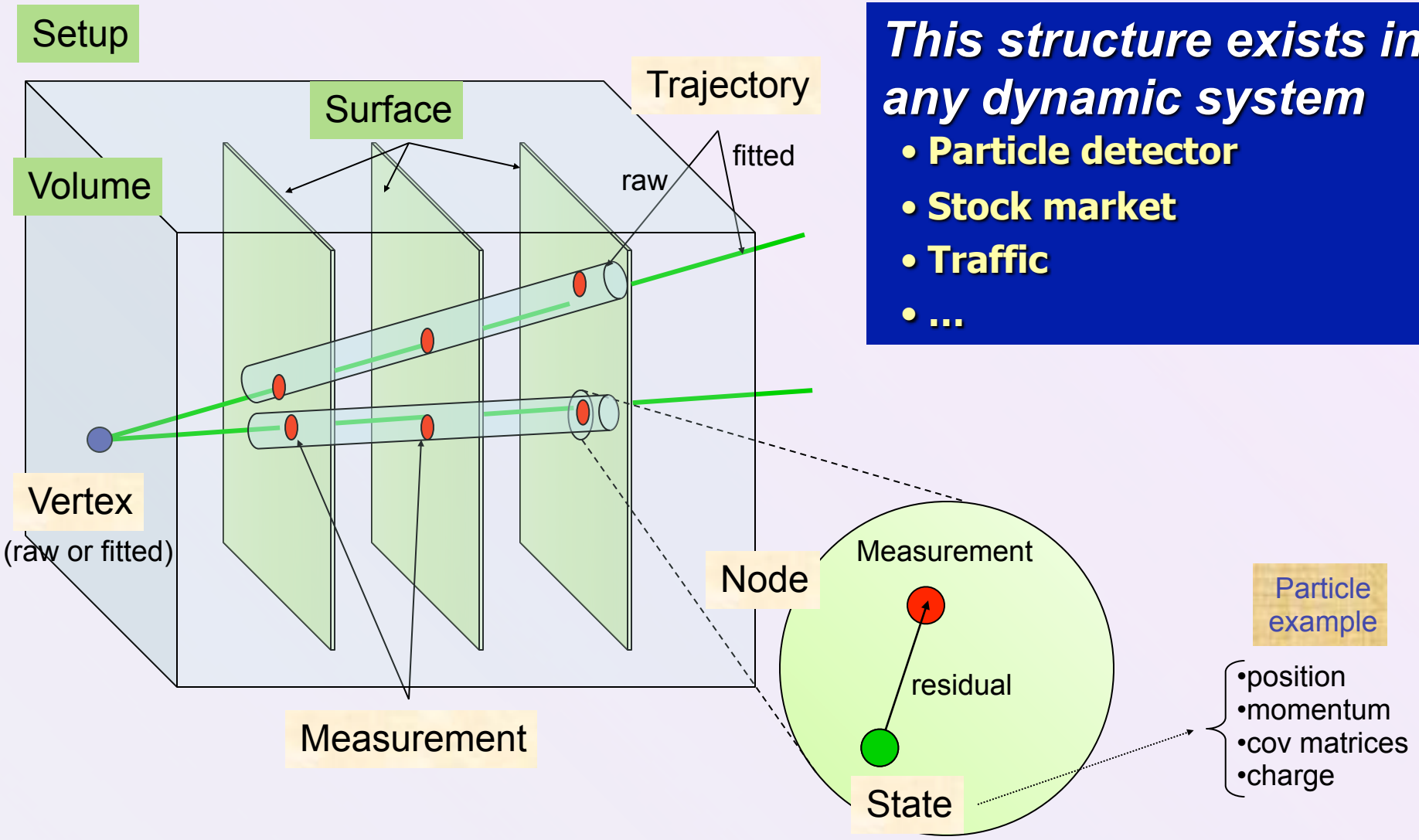
Reconstruction in HEP

RecPack classes

A simplified view



Generic Event Model (GEM)



This structure exists in any dynamic system

- **Particle detector**
- **Stock market**
- **Traffic**
- **…**

Measurement and Trajectory

- A Measurement represents a basic (i.e. hit) or a complex (i.e cluster) experimental measurement
  - The evolution of the system (i.e. particle) through the Setup produces Measurements
- Trajectory is the collection of those Measurements
- The relevant information for the spatial evolution of the particle is stored in the main measurement vector

$$\vec{\mathbf{m}} = (\mathrm{x}, \mathrm{y})$$ + cov matrix

- Additional info (i.e. energy deposited) is stored in secondary measurement vectors

$$\vec{\mathbf{m}}_{\mathbf{E}} = (\mathrm{E}_{\mathrm{dep}})$$ + cov matrix

State

- Information about the local state of a Trajectory
  - Main state vector + its covariance matrix
  - Additional state vectors (+ covariances)
- The same State can be expressed in different representations

default representation

$$\vec{\mathbf{v}} = (\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{u_x}, \mathrm{u_y}, \mathrm{u_z}, \mathrm{q/p})$$

used internally for propagation

slopes:z representation

$$\vec{\mathbf{v}} = \left(\mathrm{x}, \mathrm{y}, \mathrm{z}, \frac{\mathrm{dx}}{\mathrm{dz}}, \frac{\mathrm{dy}}{\mathrm{dz}}, \mathrm{q/p}\right)$$

main state vector

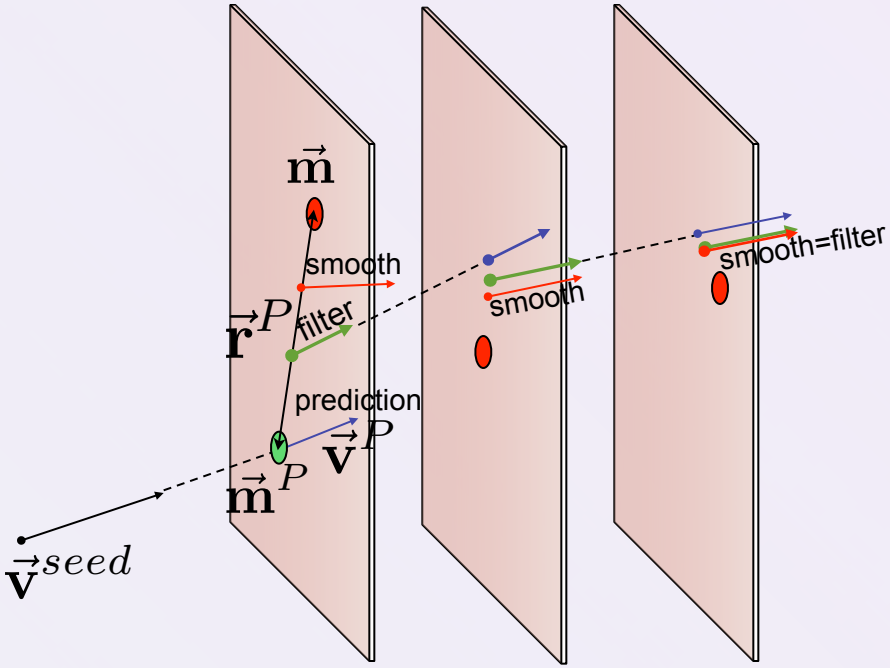
$$\vec{\mathbf{v}}_{\mathrm{sense}} = (\pm 1)$$

additional (static) state vector

more suitable for track fitting

Fitting: Kalman Filter

- Used for track and vertex fitting by most of HEP experiments
- Easy to include random noise processes (multiple scatt.) and systematic effects (eloss)
- It is a local and incremental fit



Propagation

model equation

$$\vec{\mathbf{v}}^P = f(\vec{\mathbf{v}}) \xrightarrow{\text{linear}} \vec{\mathbf{v}}^P = \mathbf{F} \cdot \vec{\mathbf{v}}$$

Transport matrix

$$\mathbf{C^P} = \mathbf{F C F^T} + \mathbf{Q}$$

Noise matrix (ms)

Projection

projection equation

$$\vec{\mathbf{m}}^P = h(\vec{\mathbf{v}}^P) \xrightarrow{\text{linear}} \vec{\mathbf{m}}^P = \mathbf{H} \cdot \vec{\mathbf{v}}^P$$

Projection matrix

$$\mathbf{M^P} = \mathbf{H C^P H^T}$$

Residual = prediction-measurement

$$\vec{\mathbf{r}}^P = \vec{\mathbf{m}}^P - \vec{\mathbf{m}}$$

Propagation

	RecPack sub-tools	
1. Computes distance to surface	SurfaceIntersector	s
2. Propagates state vector to that distance	ModelEquation	$f(\vec{\mathbf{v}}, s)$
3. Applies model corrections (energy loss, etc)	ModelCorrection	$\Delta\vec{\mathbf{v}}$
4. Propagates the covariance matrix	Propagator	$\mathbf{F}$
5. Adds random noise (multiple scattering, eloss fluctuations, etc)	Noiser	$\mathbf{Q}$

model equation — path length — model correction

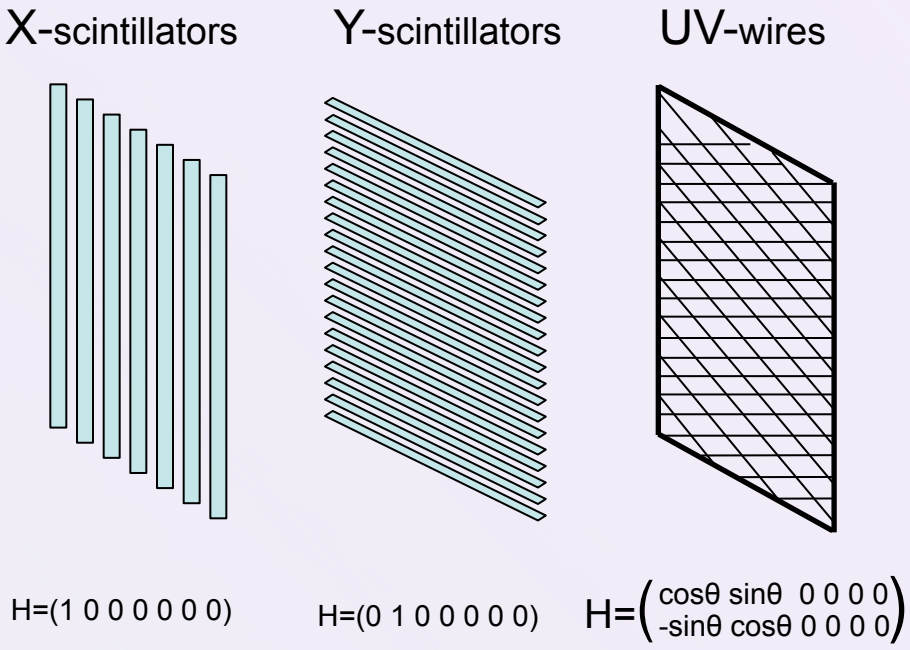
$$\vec{\mathbf{v}}^{P} = f(\vec{\mathbf{v}}, s) + \Delta\vec{\mathbf{v}}$$

Transport matrix — Noise matrix

$$\mathbf{C}^{\mathbf{P}} = \mathbf{F}\mathbf{C}\mathbf{F}^{\mathbf{T}} + \mathbf{Q}$$

Projection into measurement

- Consider a detector with several subdetectors, each providing a different measurement type



projection equation

Projection matrix

$$\vec{\mathbf{m}}^P = h(\vec{\mathbf{v}}^P) \xrightarrow{\text{linear}} \vec{\mathbf{m}}^P = \mathbf{H} \cdot \vec{\mathbf{v}}^P$$

$$\mathbf{M^P} = \mathbf{H C^P H^T}$$

- A **Projector** projects the state vector into the measurement space
- There is a Projector for each measurement type

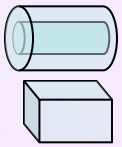
Geometry (I)

- Reconstruction processes can use a simplified geometry
- because measurement errors hide geometrical details:
  - screws, …
- The important objects are:
  - Simplified volumes delimiting regions of different propagation properties:
    - Radiation length, interaction length, dE/dx
    - Magnetic or electric field
  - Measurement surfaces
- This allows fast reconstruction using analytical extrapolations whenever is possible (homogeneous properties) and dynamic stepping otherwise

Geometry (II)

add_volume("parent_name", "vol_name", "box3D", pos, axis1, axis2, size)

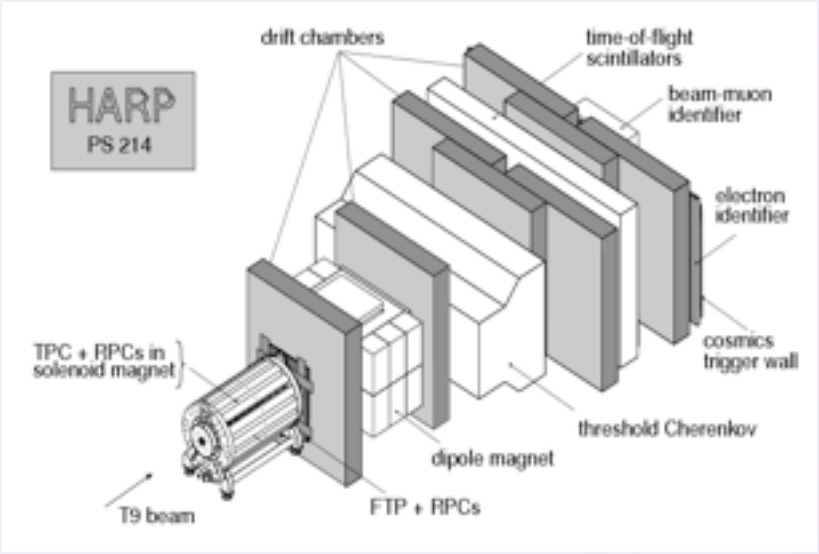
- tube
- box
- ….



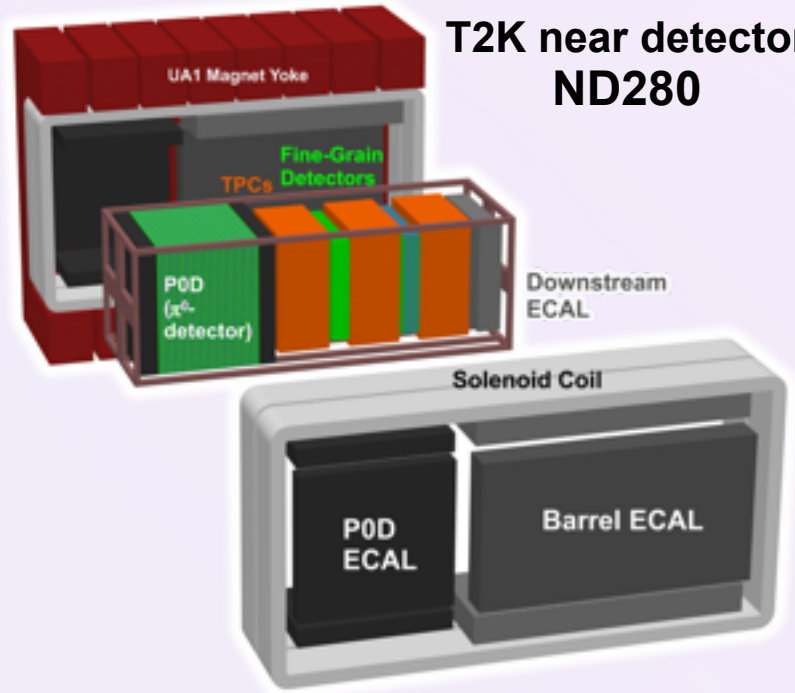
add_surface("parent_name", "surf_name", "rectangle", pos, axis1, axis2, size)

- plane
- ring
- ring_sector
- cylinder_sector
- …

HARP



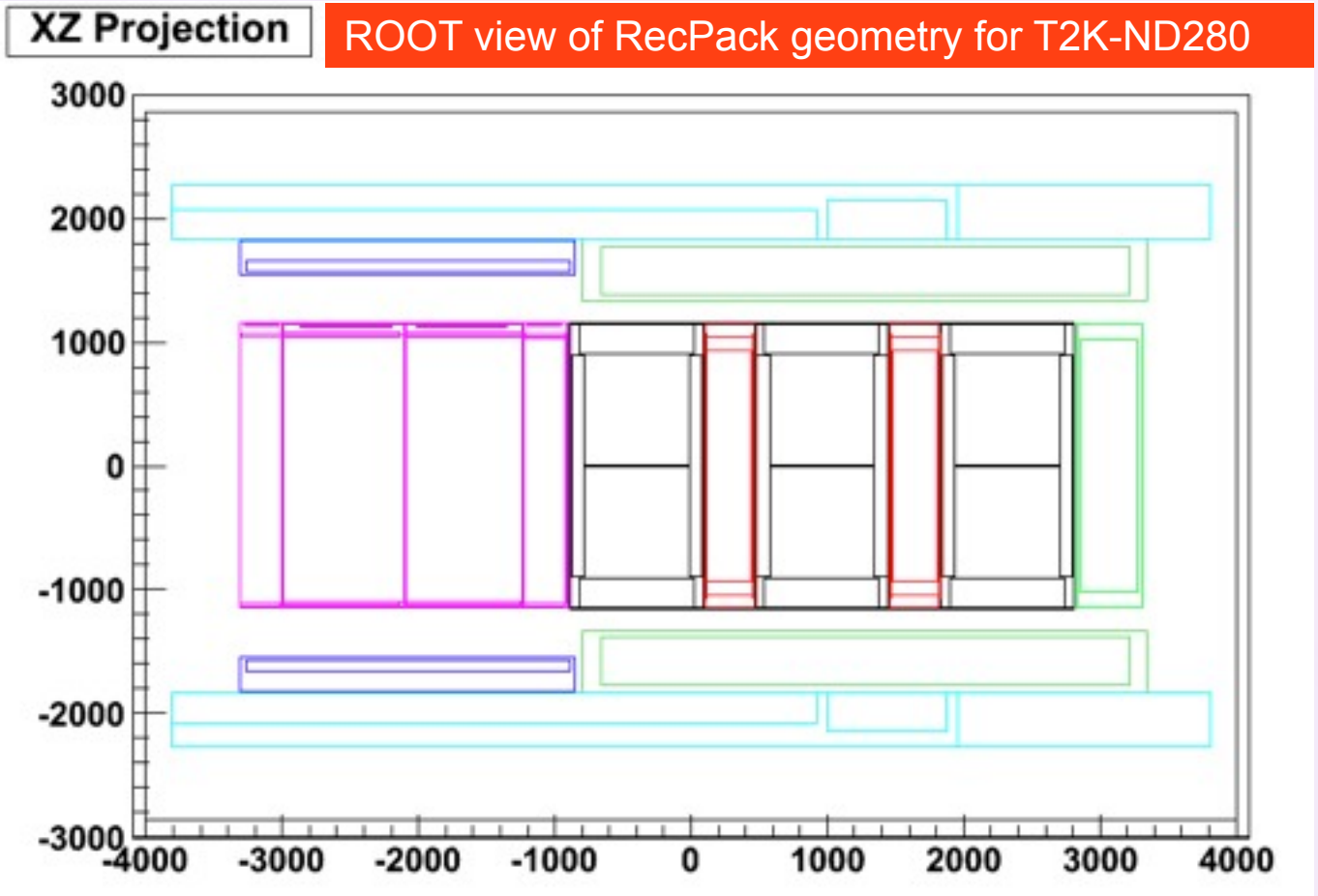
T2K near detector ND280



ROOT geometry interface

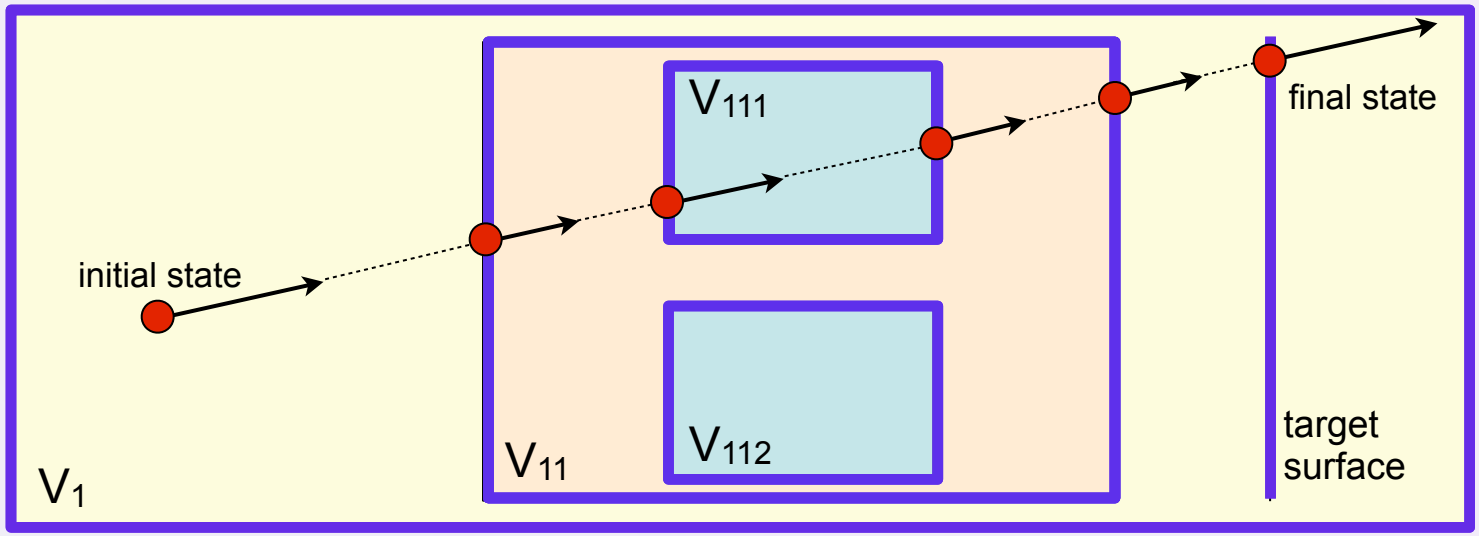
- The user selects by name which volumes from the complete ROOT geometry should be used for reconstruction
- Selected volumes and their properties are added automatically

Virtual volumes:
a RecPack volume
can also be the addition
of several ROOT volumes



Navigation

1. Intersect all active surfaces:
    1. Border surfaces of the current volume and its subvolumes
2. Propagate vector and covariance matrix to the closest surface
3. Repeat step 1 until target surface is reached



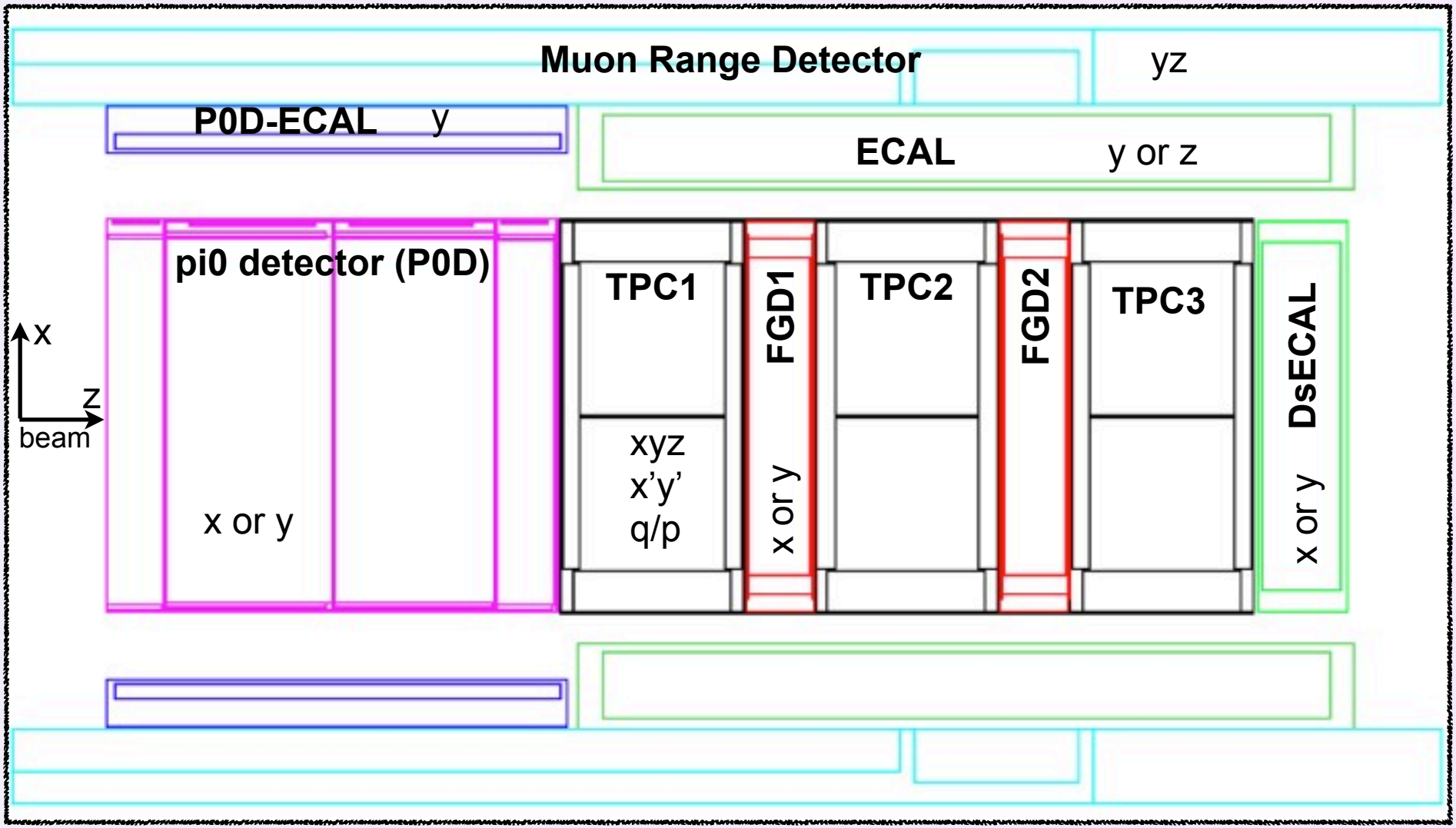
Matching

- Check whether two objects are originated by the same particle
- Propagate both objects to the same surface
- Compute the matching X^2 in that surface:

$$\chi^2_{match} = (\vec{\mathbf{v}}_1 - \vec{\mathbf{v}}_2)(\mathbf{C}_1 + \mathbf{C}_2)^{-1}(\vec{\mathbf{v}}_1 - \vec{\mathbf{v}}_2)^T$$

- Available functions
  - Trajectory-Trajectory
  - Trajectory-Measurement
  - State-Trajectory
  - State-Measurement

T2K-ND280 example



- TPCs are special case: meas vector = state vector
  - Projection matrix H is identity matrix

RecPack services

RecPack MANAGER

service name	Description
Model	mathematical equations for: propagation, projection, intersection with surfaces, random noise (ms), model corrections (eloss), model conversions, etc
Fitting	• track fitting (Kalman Filter and least squares) • vertex fitting • cluster fitting
Geometry	• volumes and surfaces to build complicated setups • volume and surface properties
Navigation	• propagate to any surface or length taking into account the volume hierarchy • navigation logics
Matching	• matching functions between trajectories, measurements, states, vertices etc. • pattern recognition methods
Simulation	Simulate trajectories and measurements for debugging purposes

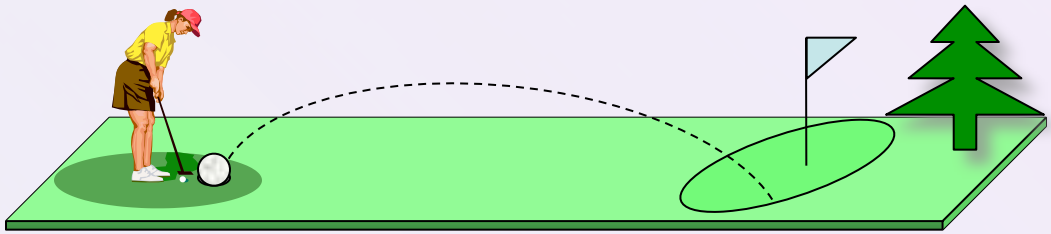
Extensions

Example for Model service

Models	surface intersectors	Model corrections	Noise estimators	projectors
• straight line in any dimension • helix in variable B field	• plane • cylinder • numerical	• energy loss	• multiple scattering	• **1D**: X,Y,Z,U • **2D**: XY,YZ,XZ,UV, **r**ϕ • **3D**: XYZ
parabola	earth surface	wind	wind fluctuations	earth surface

All these pieces live in independent containers & implement the corresponding interface

→ ***Extensions are straight forward***



Clients

- Hadron production experiments
  - **HARP** (CERN), where **RecPack** was born
  - **MIPP** (FNAL)
- Neutrino oscillation experiments
  - **T2K** (Japan) and **NOvA** (US, not yet official)
  - Design of future neutrino experiments:
    - **MIND** and **TASD** for Neutrino Factory
- Other:
  - **MuScatt** (FNAL)
  - **MICE** (RAL, UK)
  - **NEMO** and **Super-NEMO** (France)

Future extensions for HEP

- Extension of vertex concept → singularity
  - Vertex, kink, decay
- And trajectory → dynamic system
  - Collection of trajectories connected through singularities
- Common pattern recognition logics:
  - 3D tracking: TPC, …
  - 2D tracking: planar, cylindrical, spherical, …
  - Clustering: 2D uncorrelated, 2D correlated, 3D, …
- Common PID algorithms
- GUI: (HEP and non-HEP)
  - Detector design and geometry maker
  - Event display
  - Visual debugger: interactive reconstruction (probably via Python)

Conclusions

- **RecPack** is a c++ reconstruction toolkit
- It provides the common tools of any reconstruction program ⟶ avoids reinventing the wheel !!!
- Its modular structure allows extensions in any direction

  data types ⟶ volumes, surfaces, measurements, …
  tools ⟶ models, noisers, fitting methods, …

- It is setup independent
- It is being successfully used by several HEP experiments
- Interested people (users or developers) may contact:

Anselmo.Cervera@cern.ch
Juan.Jose.Gomez.Cadenas@cern.ch
Jose.Angel.Hernando@cern.ch

Coming soon:
- **Detailed documentation**
- **updated WWW site**