I Workshop CMS-Italia del Software & Computing

Roma, 19 Novembre 2001

La ricostruzione dei muoni in CMS

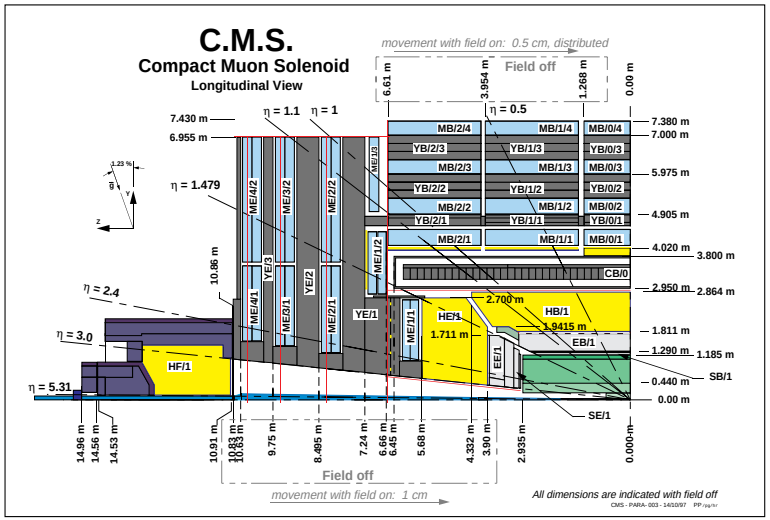
Stefano Lacaprara, *INFN, Padova.*

Sommario:

- Descrizione geometria;
- Simulazione;
- Ricostruzione locale;
- Ricostruzione regionale: L2 e L3;
- Performances;
- Problemi aperti e conclusioni.

Descrizione geometria:

- Geometria da rz (GEANT3)
- Anche per GEANT4!
- Implementata “á la” CommonDet;
- Ogni camera (DT,CSC,RPC) e’ una DetUnit;
- Anche layer CSC sono DetUnit;



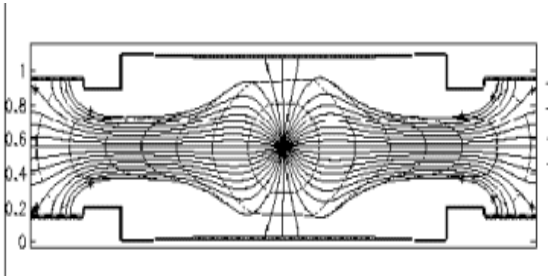
- Le DetUnit di ogni stazione sono raggruppate in DetLayer (cilindri nel barrel, dischi nell’endcap);
- Funzionalita’ DetUnit: accesso a Geometria (frame locale-globale), SimHits, Digi, RecHits, allineamento, ...

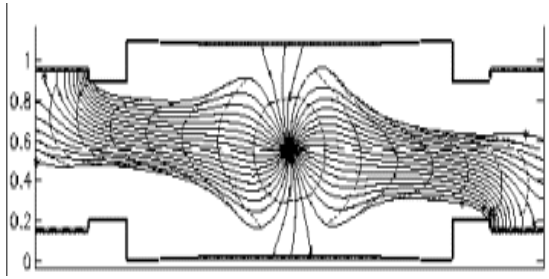
Simulazione:

SimHit (fz GEANT3 cmsim) $\Longrightarrow$ Objy $\Longrightarrow$ Digi (ORCA) $\Longrightarrow$ Objy

Drift Tube:

- Drift cell simulata con GARFIELD in funzione di $\vec{B}$.





- Drift time dipende da $\vec{B}$ (sia $B^{\parallel}$ che $B^{\perp}$) e angolo incidenza.
- TOF, propagazione segnale, soft delta-ray,segnali vicino al filo, ...
- Cella simulata e' quella vecchia (40 mm) poi riscalata a 42 mm.

Volontari per nuova parametrizzazione??

Cathode Strip Chamber:

- Simulazione ionizzazione, interazioni con gas, δ-ray.
- Trasporto e^- secondo $\vec{E}, \vec{B}$ locale, moltiplicazione vicino anodo.
- Simulazione elettronica (jitter, noise, ...)

Resistive Plate Chamber:

- Risposta 20 ns dopo passaggio particella con jitter 3 ns per elettronica e propagazione.
- Simulazione cluster size (piu' strips con segnale).
- Noise (e neutron background) impementati ma non utilizzati.

Ricostruzione locale (RecHit):

- Responsabilita' delle DetUnit (camere);
- Completamente *on demand*, risultati salvati in cache per successive richieste (RecDet);
- Utilizza interfaccia RecHit (CommonDet);

Drift Tube:

- Ricostruiti separatamente segmenti in $r - \phi$ ($4 + 4$ layers separati da honeycomb spacer) e $r - z$ (4 layers, 0 in MB4);
- Fit lineare e risoluzione ambiguita' left/right;
- Segmenti $3D$ associando le due proiezioni.

Cathode Strip Chamber:

- 6 layers per camera, RecHit $3D$ in ogni layer (strip $\sigma \sim 100$ μm e wire $\sigma \sim 0.5$ cm);
- Fit lineare in $3D$ per associare RecHit in un segmento $3D$ (che e' sempre un RecHit).

Resistive Plate Chamber:

- cluster hits: misura $1D$, solo perpendicolare alle strips. Altra coordinata centro della camera, $\sigma = size/\sqrt{12}$.

Ricostruzione regionale:

"Naturalmente" regionale per l'implementazione *on demand*: solo le camere interrogate ricostruiscono localmente;

4 fasi distinte:

SeedGeneration: Da dove parto a ricostruire e in che direzione?

TrajectoryBuilding: e' il vero steering. 3 sub-fasi:

- Loop su camere compatibili (Navigation)
- Scelta dei RecHit da utilizzare (MeasurementEstimator).
- Inclusione dei RecHit nella trajectory (Updator)

TrajectorySelection: ghost suppression;

TrajectorySmoothing: per aggiornare parametri trajectory.

Seed Generation:

- Interna: dai segmenti DT e CSC
  - Posizione e direzione dei segmenti;
  - p_T da parametrizzazione $p_t = p_t(\phi)$ con o senza vertex constrain (BARREL) o ipotesi di traiettoria a elica dal I.P.(ENDCAP);
- Esterna: dal L1 Global Muon Trigger
  - p_T da L1 (90% p_T scale) $\sigma \sim 15\%$ barrel, $\sigma \sim 25\%$ endcap;
  - $\sigma(x, y, dx/dz, dy/dz)$: dimensioni camera;

Trajectory Building:

Navigation - MeasurementEstimator - Updator

- Quali DetLayer compatibili con attuale trajectory (all'inizio con seed)?

  navigazione implementata "one shot": subito tutti i DetLayer compatibili. Alternativa (CommonDet) e' "step-by-step", solo i primi DetLayers compatibili, ripetuta ad ogni step.

- Quali DetUnit sono compatibli con mia trajectory estrapolata a DetLayer?
- Ricostruzione locale nelle DetUnit compatibili (e solo in quelle);
- Test compatibilita' dei RecHit(s) con trajectory estrapolata;
- Includo RecHit compatibile nella trajectory: per ogni DetLayer usato solo il migliore RecHit (χ^2). Kalman filtering: uso segmenti DT, hits RPC, e hits dentro segmenti per CSC (alto $\vec{B}$).

2 Fasi Trajectory Building:

- **Forward filtering o Seed Refiner**: dall'interno (dove e' definito il seed, 2^a stazione) a fuori: taglio χ^2 per RecHit molto lasco;

  $\Longrightarrow$ costruisco state vector nel DetLayer piu' esterno: e' il seed migliorato da cui partire per la fase successiva;

- **Backward filtering**: vero filtering. Da seed migliorato verso l'I.P..

- Taglio χ^2 piu' duro. Costruizione trajectory: lo stato piu' updated (contiene info di tutti i RecHit usati) e' quello piu' interno:

  $\Longrightarrow$ no smoothing.

Trajectory selection:

- Per costruzione: 1 seed $\Rightarrow$ 1 trajectory;
- ma ghosts L1 $\Rightarrow$ seed simili!
- se trajectory diverse condividono RecHits: $\Rightarrow$ salvata solo la trajectory con miglior χ^2;
- spesso da seed vicini viene creata la *stessa* trajectory;
- possible problema per μ fisici vicini (e.g. b/c cascade $\longrightarrow \mu\mu$).

Fit al vertice I.P.:

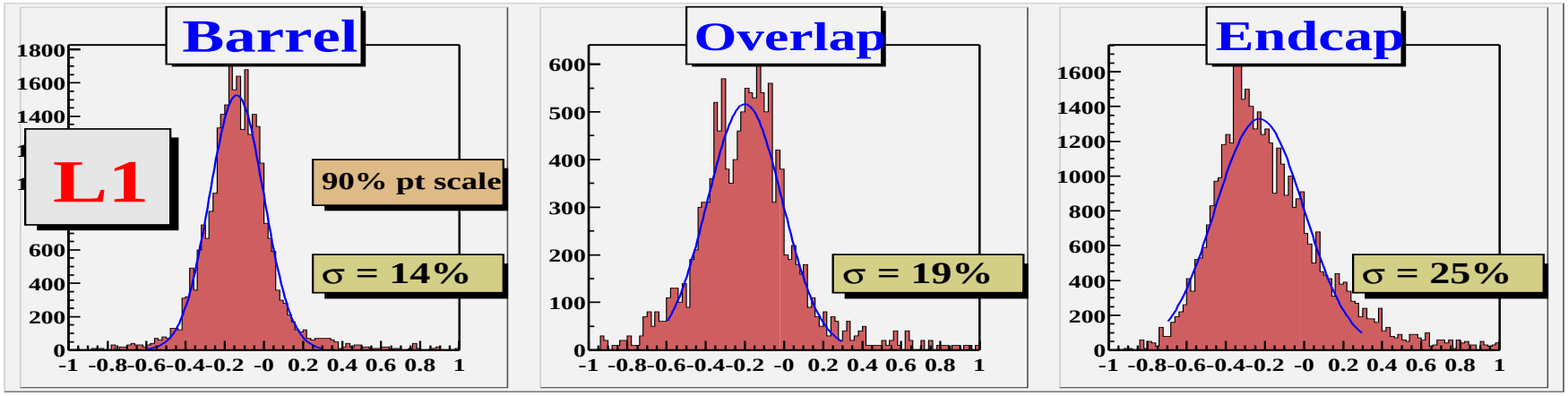
- Estapolazione trajectory al punto di massimo approccio all'I.P. nominale $(0, 0, 0)$;
- RecHit in $(0, 0, 0) \pm (0.015, 0.015, 5.3)$ cm;
- Update della trajectory (se recHit compatibile).
- Fit fallito significa problemi!
  - μ non prompt (da π, K decay in flight);
  - problemi di ricostruzione (locale o regionale).

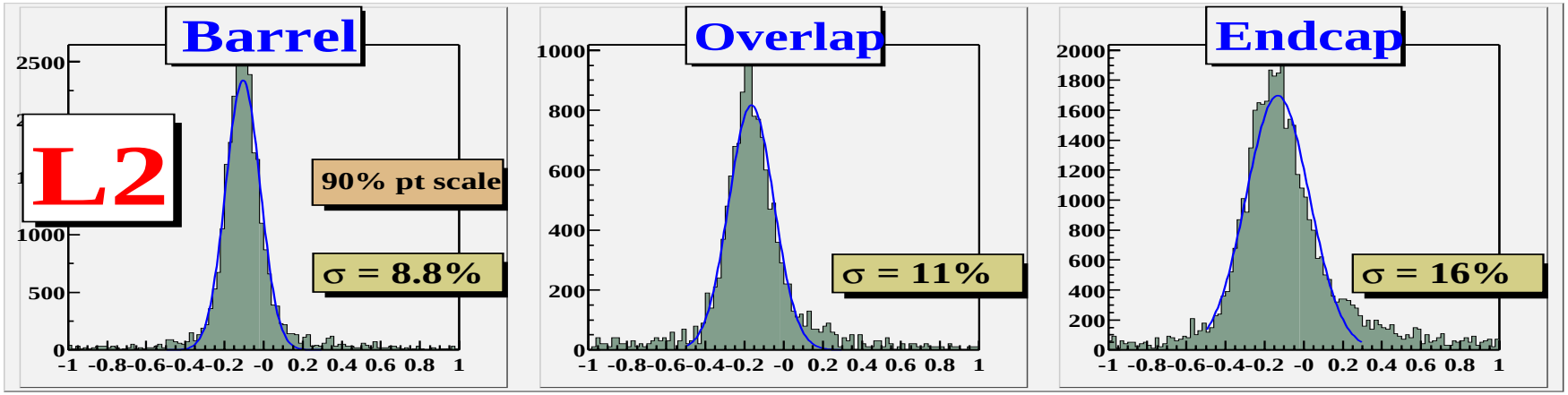
Ricostruzione L3:

- μ ricostruito a L2 come seed per ricostruzione Tracker;
- RecHit compatibili in Tracker DetLayer per costruire seed;
- inside$\longrightarrow$out oppure ouside$\longrightarrow$in;
- ricostruisco trajectory nel Tracker a partire da seed;
- alla fine smoother della trajectory usando anche i RecHit trovati dal L2.
- alternativa e' ricostruzione trajectory nel Tk e Muon usando L2 seed e navigazione combinata.

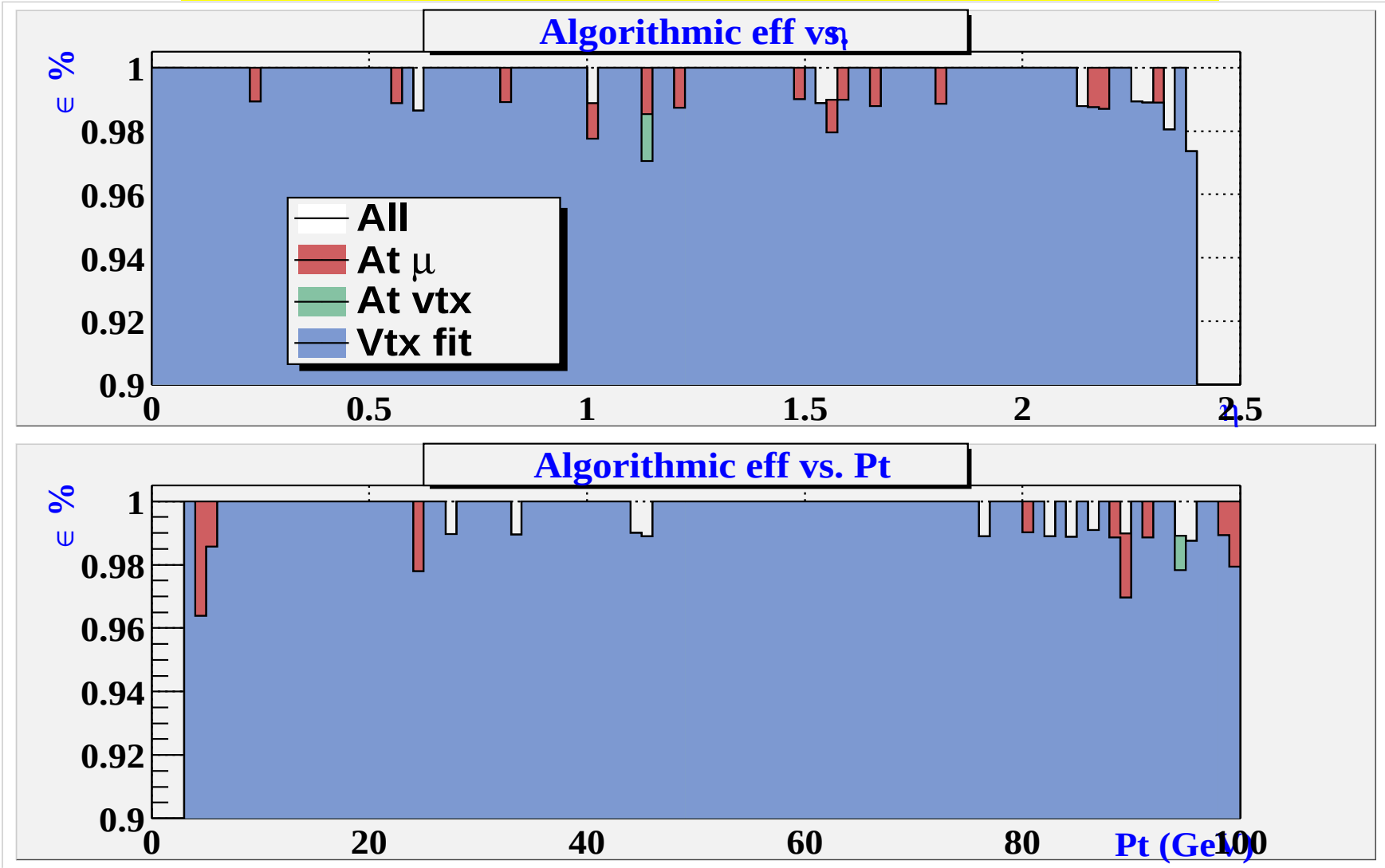
Performance $\Delta(1/p_t)$:

Sample: 10k mu singoli, $p_t = 3 \div 100$ GeV, $|\eta| < 2.5$

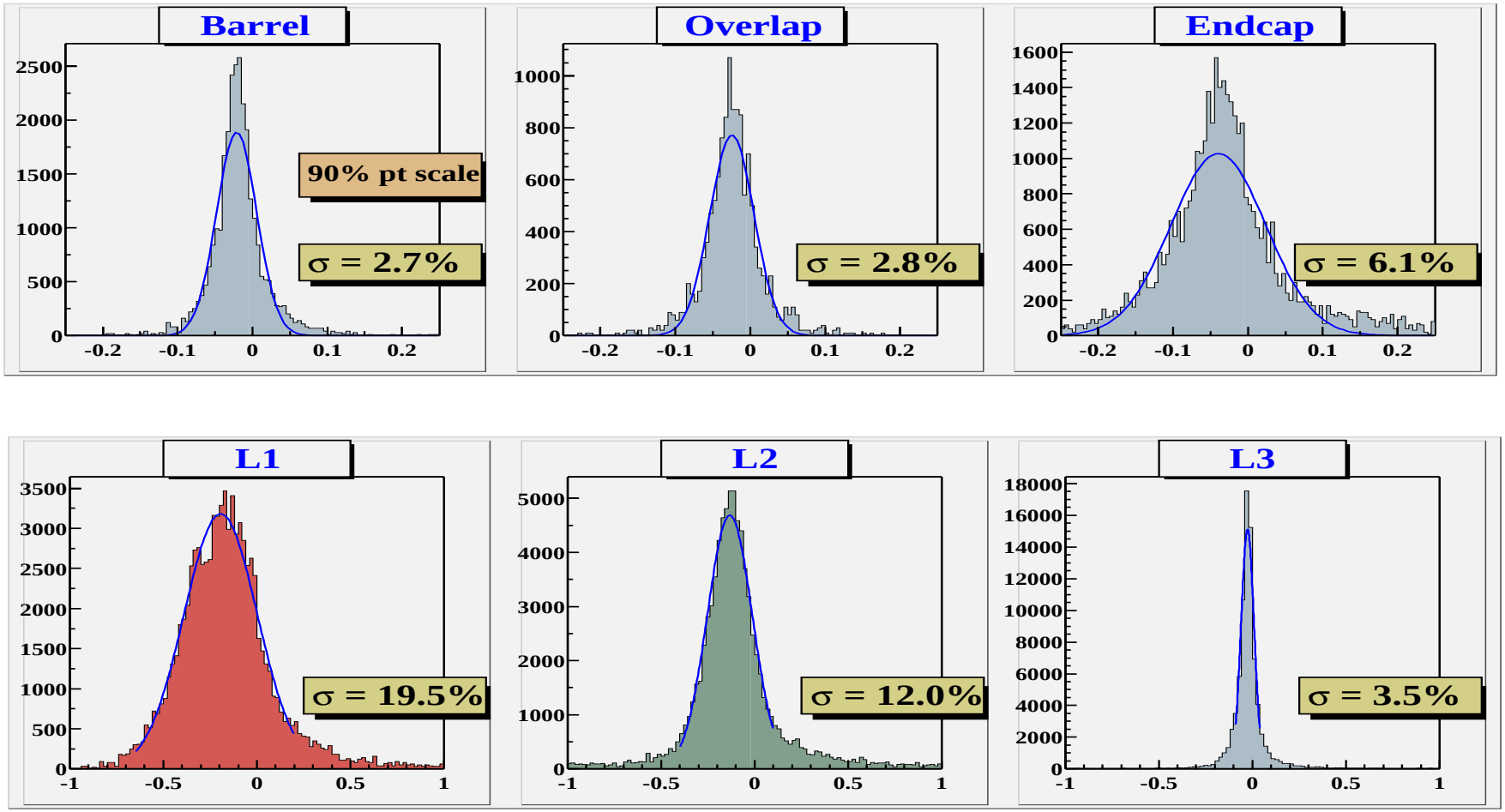




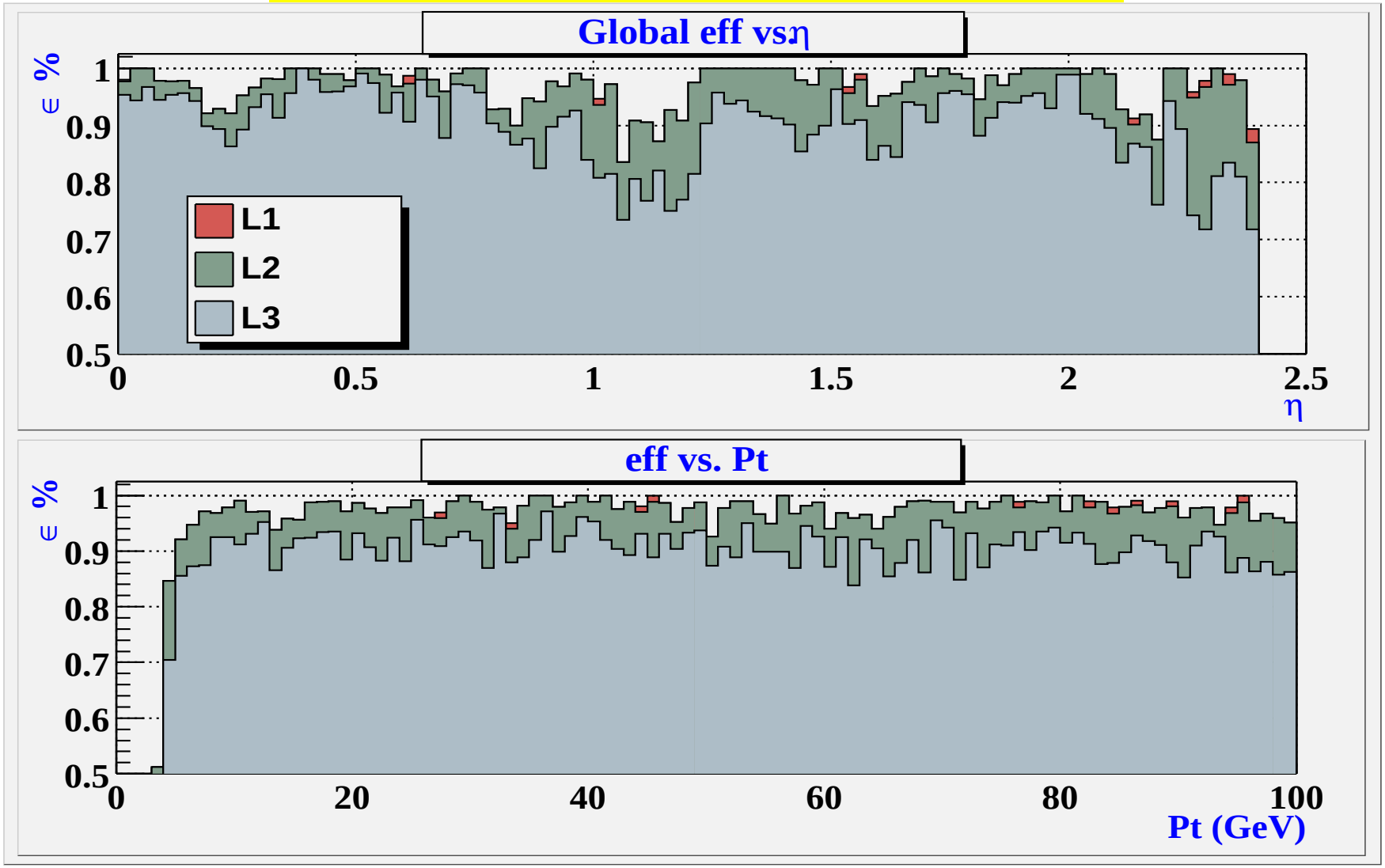
Efficenza algoritmica L2 (wrt L1):



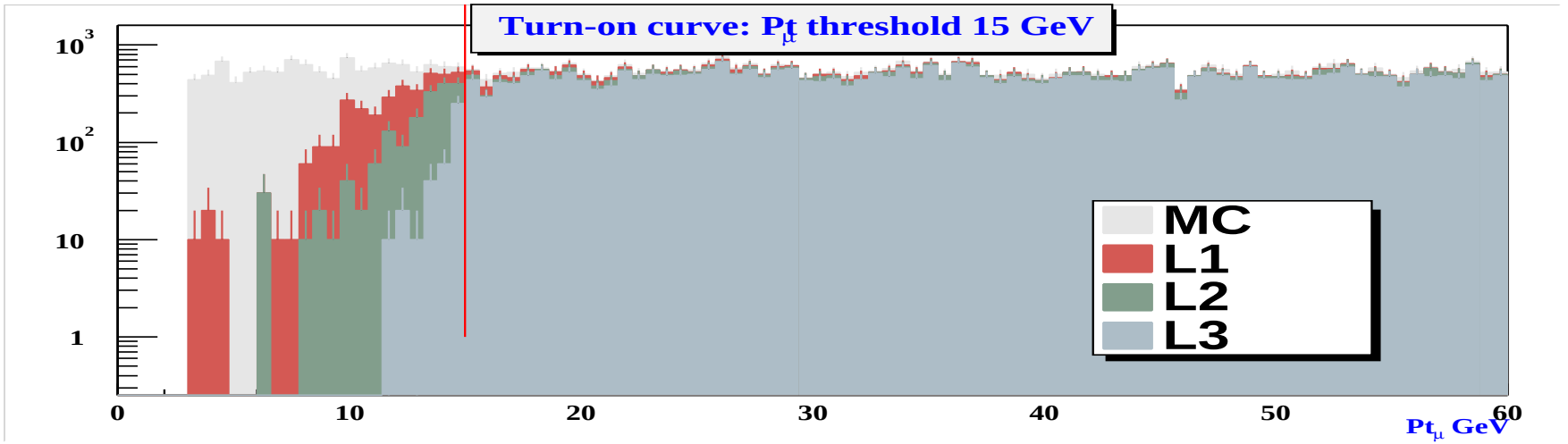
Performace L3 $\Delta(1/p_t)$:

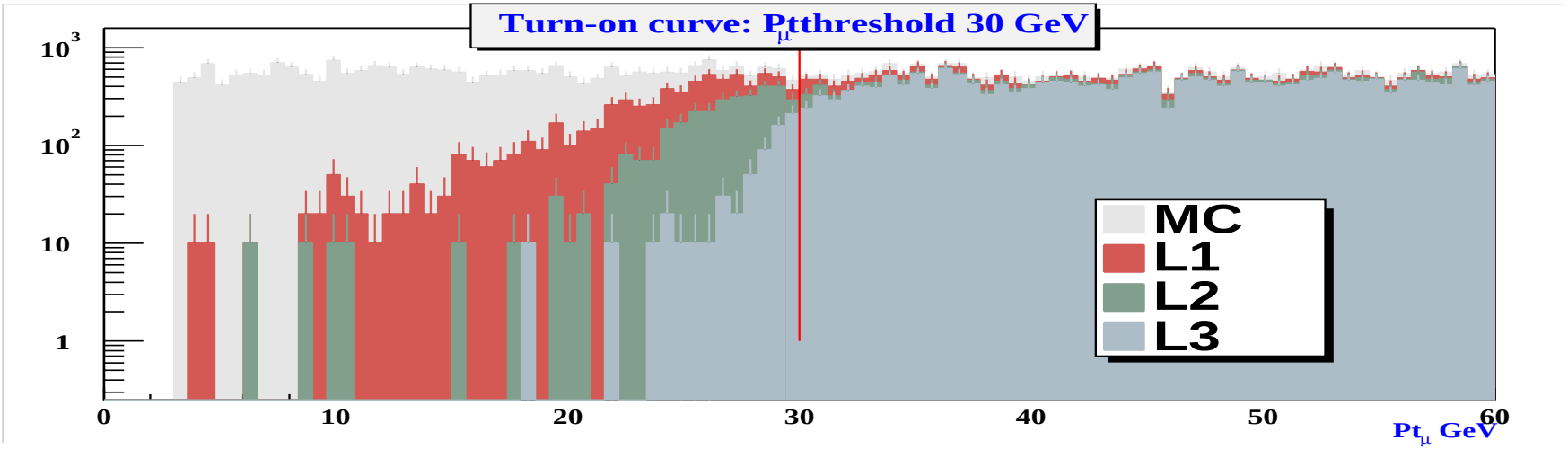


Efficenza globale L1/L2/L3:



L1/2/3 turn-on curves:





Problemi aperti:

- Efficienza del L3 e' troppo bassa! Uno dei problemi e' un cattivo fit a L2.

  $\Longrightarrow$ debugging per migliorare ricostruzione locale e definizioni errori, riconoscimento segmenti "problematici", studio dei pool locali...

- Timing: L2 e' lento! $\sim 1\ s/ev$ su CPU standard. Maggior parte del tempo e' perso nell'estrapolazione (geane) tra le stazioni:

  $\Longrightarrow$ ottimizzazione ricerca DetUnit entro DetLayer. Risultati interessanti con algoritmo migliorato: da testare per verificare l'efficienza.

- Comprensibilita' codice: alcune parti molto complesse e difficili da capire (e/o debuggare/migliorare) per non esperti. Documentazione e re-architecturing.

Conclusioni:

- Ricostruzione Muoni in CMS funzionante per tutto il rivelatore;
- Efficenza $\gtrsim 99\%$;
- Velocita' algoritmo mogliorabile;
- Debugging e fine tuning in corso;
- L3 ancora problematico per l'efficienza;
- Codice migliorabile per facilitare l'uso per utente non esperto.