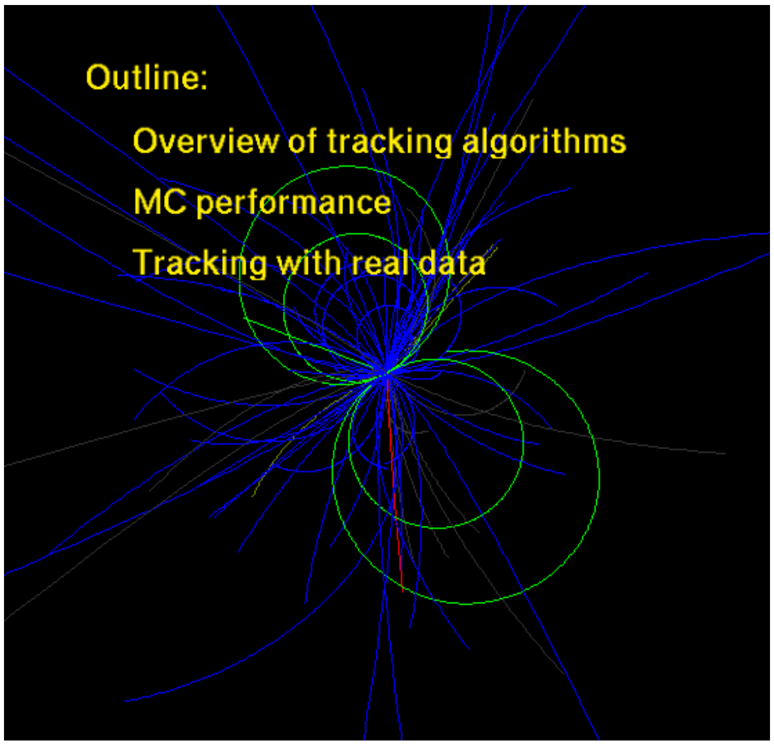
Tracking in DØ

Global Tracking Group



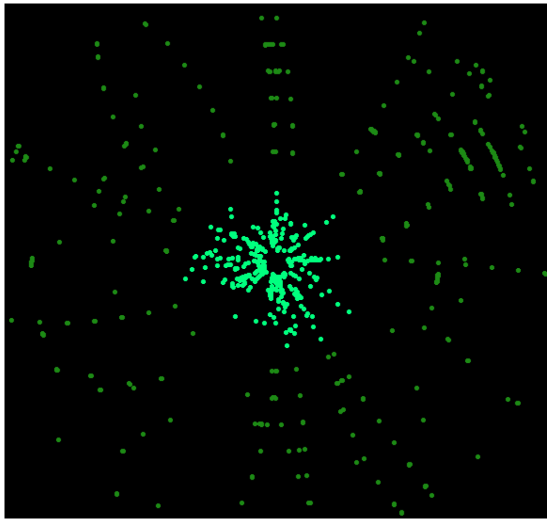
Herb Greenlee
Valentin Kuznetsov
Slava Kulik
Dave Adams
Avto Kharchilava
Mike Hildreth
Gavin Hesketh
Marian Zdrazil
Sasha Khanov
Andrew Haas
Gennadii Borisov

Prepared by V. E. Kuznetsov

Saclay, Dec. 19-21, 2001

Tracking (introduction)

Tracking system is a major part of any HEP experiment

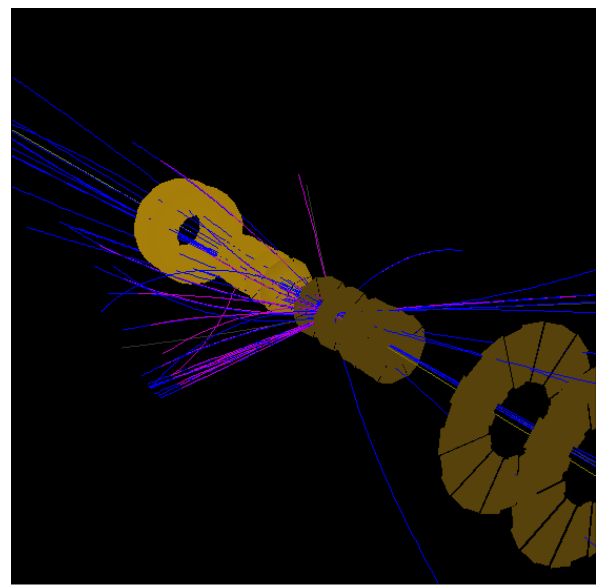


Problem: find association between hits (detector measurements) and construct out of them tracks (particle trajectories)

Solution: human scanning, track algorithms

Requirements: fast, efficient algorithm

Complications: low momentum tracks, scattering, noise, high track density in jets, etc.



You need to balance between CPU, reconstruction time and efficiency vs. scattering, noise, p_T threshold

Consequences: many techniques have been developed and adapted in HEP community

Experiments try to find the best algorithm for their detector

Tracking Algorithms (random search and road approach)

Random Search: try all hit combinations to construct track candidates

Track selection based on track model (χ^2 cut and # of shared hits).

Used in experiments with a small number of measurements

Advantages: easy to implement

Disadvantages: time grows significantly with number of measurements, may be inefficient

Road Approach: tracks are searched for within a road

Track parameters updated using Kalman filter

Hit association with the track is based on χ^2

Used in both fixed target and collider experiments

Advantages: very flexible, proven, quite efficient

Disadvantages: requires many roads to be efficient, number of operations grows quickly with number of hits

Tracking Algorithms (histogramming method)

Histogramming Method: based on Hough transform which translates a single hit into a line in (φ, ρ) space

Lines from all hits on a track intersect at one point corresponding to the vector of track parameters

Those lines make bands in a histogram whose peacks are track templates

Each template is a track candidate consisting of several hits, with approximately known track parameters. The templates are futher processed applying the Kalman filtering. During this stage, fake templates are discarded, wrong hits removed, and track parameters accurately calculated

Used mostly in collider experiments

Advantages: number of operations proportional to number of hits, small number of candidates within a histogram bin

Disadvantages: sensitive to vertex position: assumes small impact parameter and therefore is not sensitive to K_s

Tracking Algorithms (elastic arms)

Elastic-Template Approach: based on estimation of track parameters concurrently with assignment of hits

Defines a track-hit probability matrix that is a probability for any hit to be on any track at a given "temperature"

Assignment probability is chosen on the competition basis:

- competition between all hits for each track, but no competition between the tracks
- competition between all tracks for each hit, but no competition between the hits
- global competition between all entries that are incompatible, i.e. belong to the same hit or the same track

Tracking begins with "seeds" filtered from a histogramming technique

Seeds are fit "elastically" they continuously change their probabilities of being connected with each hit as their track-parameters evolve

Hits not originally associated with a track seed can become associated through the elastic fitting process

Developed for LHC/SSC collider experiments, considered for ATLAS

Advantages: equal performance with road-finding algorithm at low track densities, significantly higher performance at high track densities.

Disadvantages: new technique, untested, sensitive to vertex position

Tracking Algorithms (Kalman filter)

Kalman filter is a common algorithm for prediction and evaluation of track parameters based on known information

A track in space can be described by 5D state vector

Evolution of the state vector is described by a discrete system of linear equations:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \omega_{k-1}$$

which define the change in status of this vector based on the previous measurement point $\mathbf{x}_{k-1}$

Matrix $\mathbf{F}_{k-1}$ is the track propagator from measurement $k-1$ to k and ω_{k-1} describes the random noise of the system

The Kalman filter proceeds by performing three distinct operations:

- ⋆ *Prediction*, **where the status of the state vector is estimated at a future measurement point;**
- ⋆ *Filtering*, **where the current estimation of the state vector is carried out based on the previous measurements; and**
- ⋆ *Smoothing*, **where the estimation of the state vector at a previous measurement is re-evaluated with the new information from the present measurement.**

DØ implementation of Global TRacking software (GTR)

Object oriented approach (C++ language) and modular design of DØ software allows development of many tracking algorithms

May be run simultaneously/separately to achieve best performance for different physics tasks

Currently, four algorithms are under investigation:

- Road approach (default algorithm): based on TRF++ uses specific paths (roads) during track finding
- HTF (histogramming track finder): divides DØ detector into slices in (φ, ρ) and uses Hough transform to reduce initial number of combinations. Existing algorithm can use either CFT, SMT hits or combinations of them to construct tracks
- Elastic reco (elastic-template algorithm): can use existing tracks (from other track finders) as initial seeds, existing vertex (run after of GTR and vertex code) and/or run in stand-alone mode (construct own seeds and vertices)
- SBF (silicon barrel finder): based on random search. Track candidates are formed in SMT detector and than extended to CFT

Having a variety of track reconstruction algorithms allows us to cross-check the tracking performance in DØ detector on MC and real data

Algorithms Status

Road-following	Histogramming	Elastic-template	SBF
✓ released	✓ released	✓ released	under development
✓ supported	✓ supported	✓ supported	untested
✓ tested	✓ tested	✓ tested	✓ promising
✓ complete	needs tuning	needs tuning	

All tracking algorithms are implemented as framework packages

Each can specify a unique chunk ID to allow comparison

All of them use one final track refitting step based on D0Propagator

One common user interface: GTrack

Default algorithm is road following, used for data processing

Physics groups are encouraged to use any tracking algorithm in post-processing analysis

GTR Management

GTR packages are released and tested on the farm every week

Farm usage allows rapid response to various problems with design and implementation of the tracking algorithm(s)

GTR by numbers (release t01.71.00)

	gtr	trf	smt	cft	total
packages	22	23	24	21	90
classes	268	399	252	148	1401
RCP's	120	0	134	80	334
OBS's	5	2	33	39	79

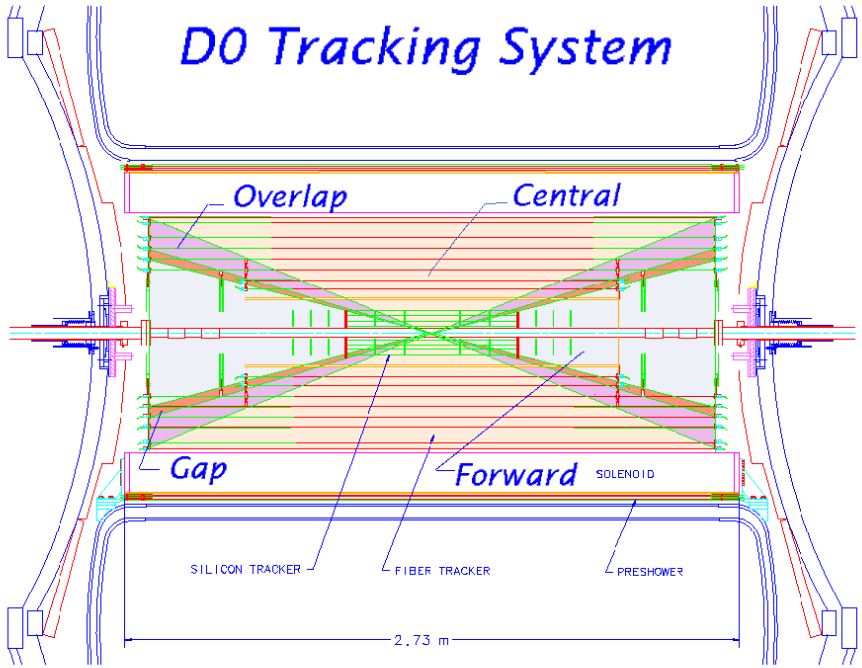
Global cuts and objects (e.g. propagators) are constructed once during initialization in ObjTable

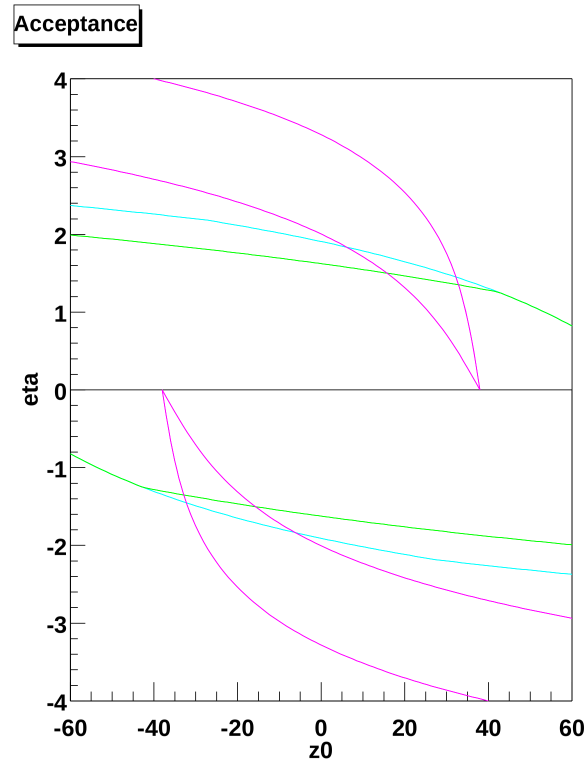
Parameters and objects are controlled by RCP (run control parameters) and OBS (object streams) files

GTR extensively uses RCP files for external parameters

GTR uses OBS files to define a class objects on fly. The OBS files contain instructions for track finding, i.e. the list of roads to follow

DØ Acceptance





Divide acceptance into four regions:
central - full CFT extended into SMT
forward - forward SMT with three F-disks
overlap - partial CFT extended into SMT
gap - between overlap and forward

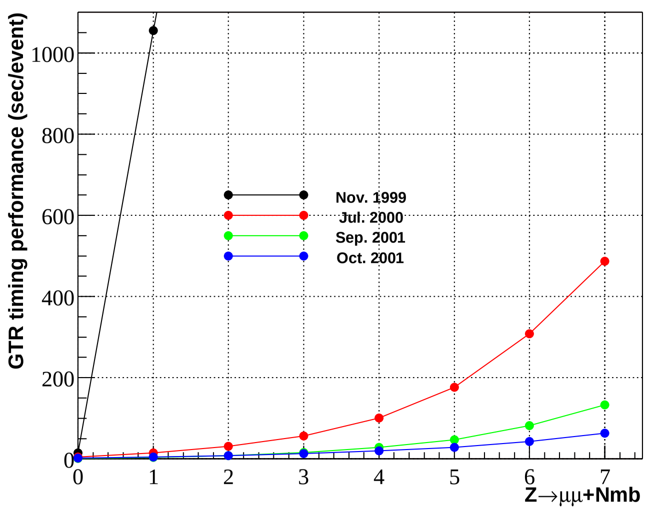
A Path for road-following algorithm is constructed for each range by requiring it to cross a particular number of sub detector layers

GTR Performance

Test global tracking on standard samples of $Z \to \mu\mu + Nmb$, $N = 0, \ldots, 7$

GTR performance on DØ farm is measured every week

We monitor time usage during each step of track reconstruction program(s)



Major improvements

- ☞ compiler optimization (up to 30%)
- ☞ interface change (LTrack⇔VTrack)
- ☞ new algorithms (e.g. overlap region CFT → SMT or SMT → CFT)
- ☞ better programming style (e.g. STL containers vs. size-limited arrays)
- ☞ debugging and profiling the code

Total time spent by track reconstruction program for a sample of $Z \to \mu\mu + Nmb$

version/Nmb	0mb	1mb	2mb	3mb	4mb	5mb	6mb	7mb
non-optimzed	1.5	5.6	11.6	21.6	39.3	65.4	114.2	178.1
optimized	1.2	3.7	8.2	15.8	28.3	46.9	81.9	132.9
new overlap	1.6	4.3	8.0	12.8	19.8	28.8	43.2	62.9

Final goal to achive 10 sec/event

GTR Performance (cont'd)

Definitions:

Track-finding Performance **is assessed by comparing reconstructed and Monte Carlo tracks using kinematic matching**

Match χ^2: **average value** $\simeq 5$

Nearness: **reconstructed tracks are matched to the nearest MC track using the match χ^2. The maximum allowed match $\chi^2 = 500$**

If multiple reco tracks are matched to one MC track, then all but the closest are left unmatched.

Matched MC tracks are *found tracks*

Found tracks with match $\chi^2 > 25$ are called misreconstructed. Those below are *well reconstructed*

Unmatched reco tracks are called *fakes*

Performance Metrics:

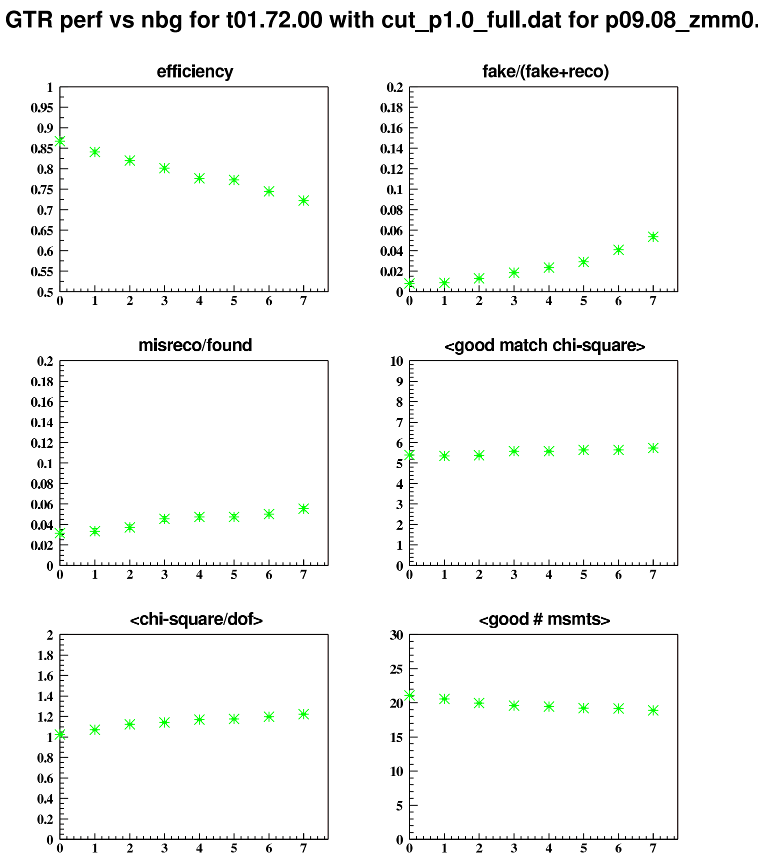
Efficiency **is the fraction of Monte Carlo tracks which are matched**

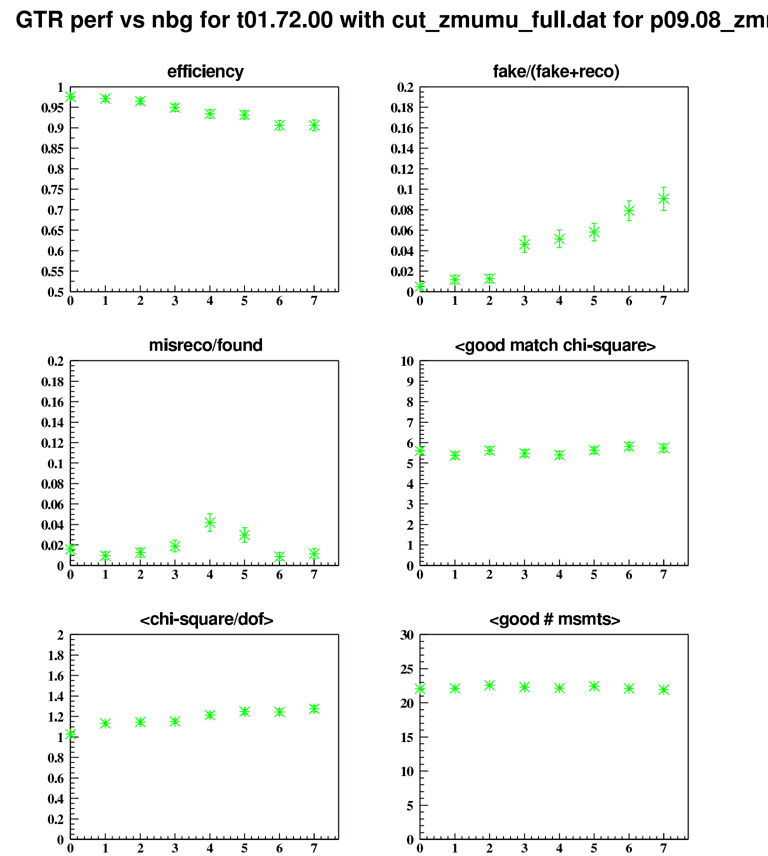
Misreco fraction **is the fraction of found tracks that are misreconstructed**

Fake ratio **is the number of fake tracks divided by the number of (found+fake) tracks**

GTR Performance (cont'd)

Performance of track reconstruction program:



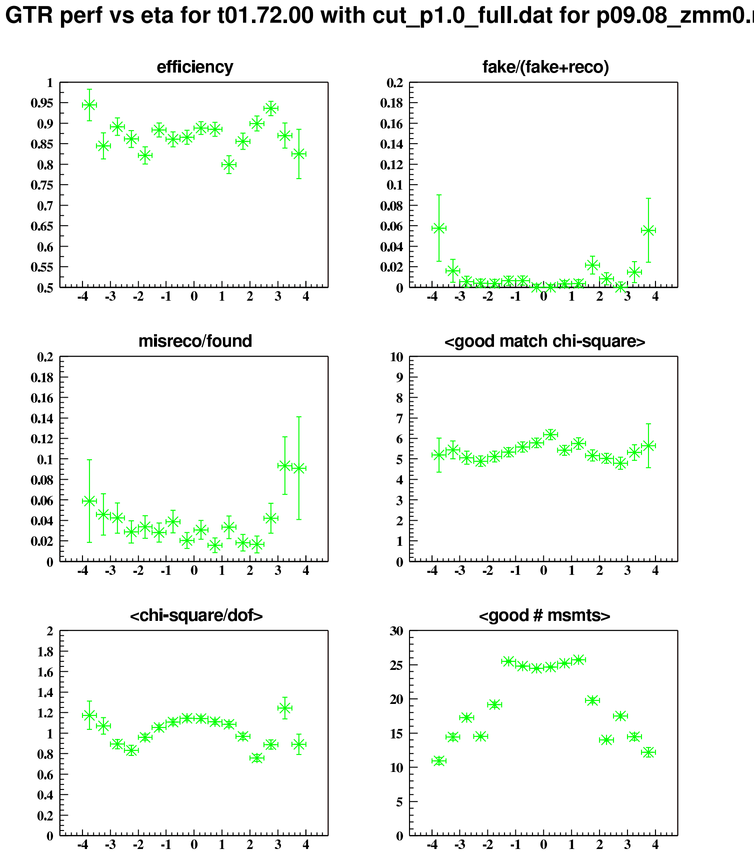


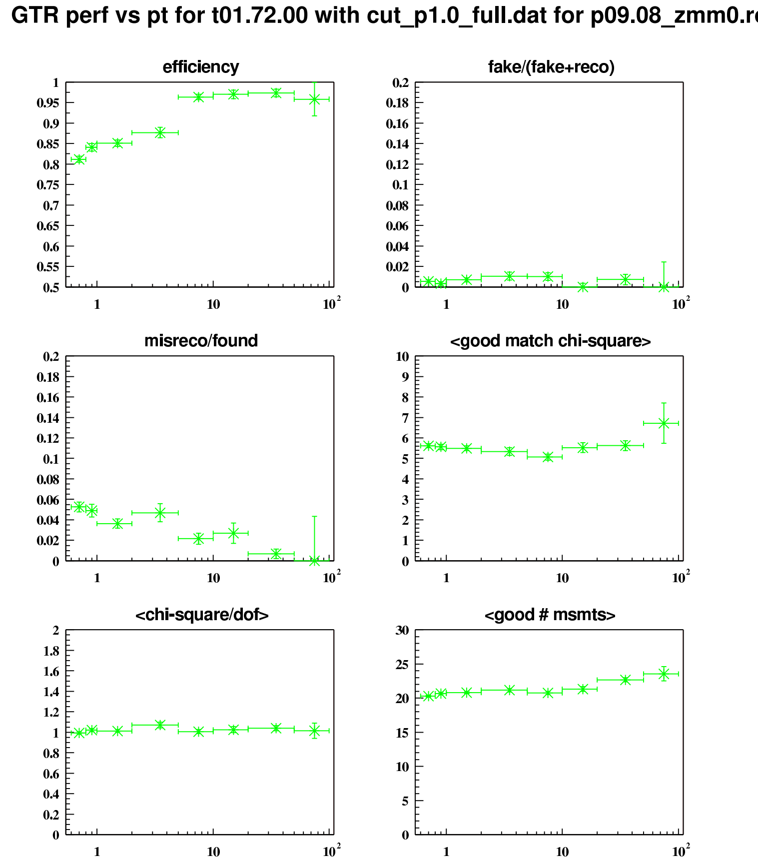
perf. vs. # min. bias events, all tracks above 1GeV

perf. vs. # min. bias events, μ's from Z above 10GeV

GTR Performance (cont'd)

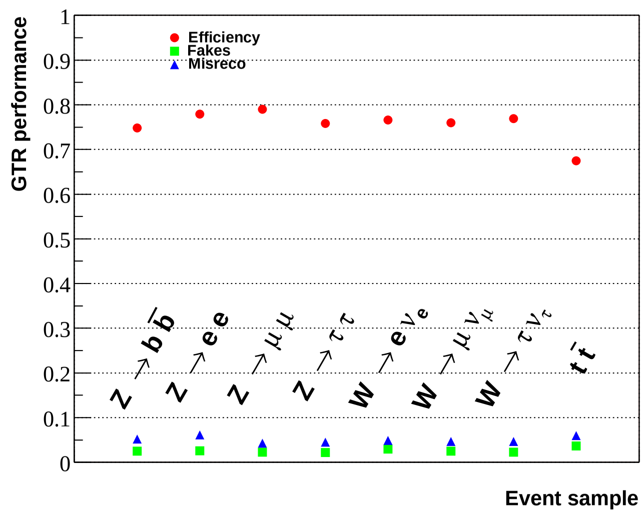
Performance of track reconstruction program:





perf. vs. η, tracks above 1GeV | **perf. vs. p_T, μ's from Z above 10GeV**

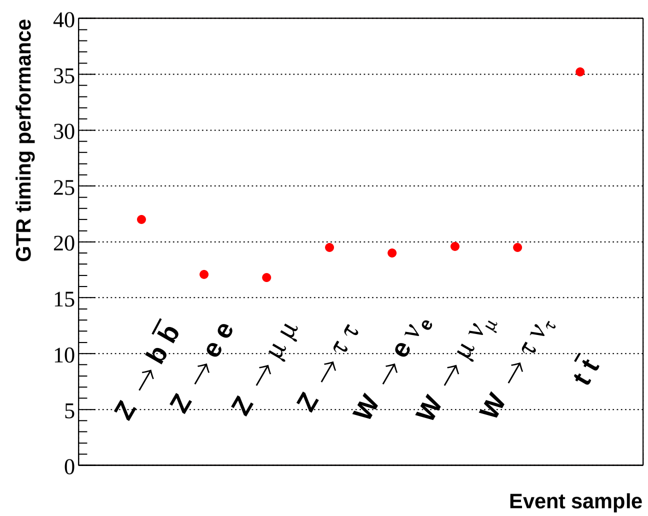
GTR Performance (cont'd)



- **GTR performance is measured for many interesting physics samples ($Z \to \ell\ell, b\bar{b}$, $W \to \ell\nu_\ell, t\bar{t}$ plus 2.5mb average)**
- **We achieved quite stable performance, additional tunning may be possible using different algorithms and/or cuts**
- **Our plan to measure performance for these samples on bi-weekly basis**

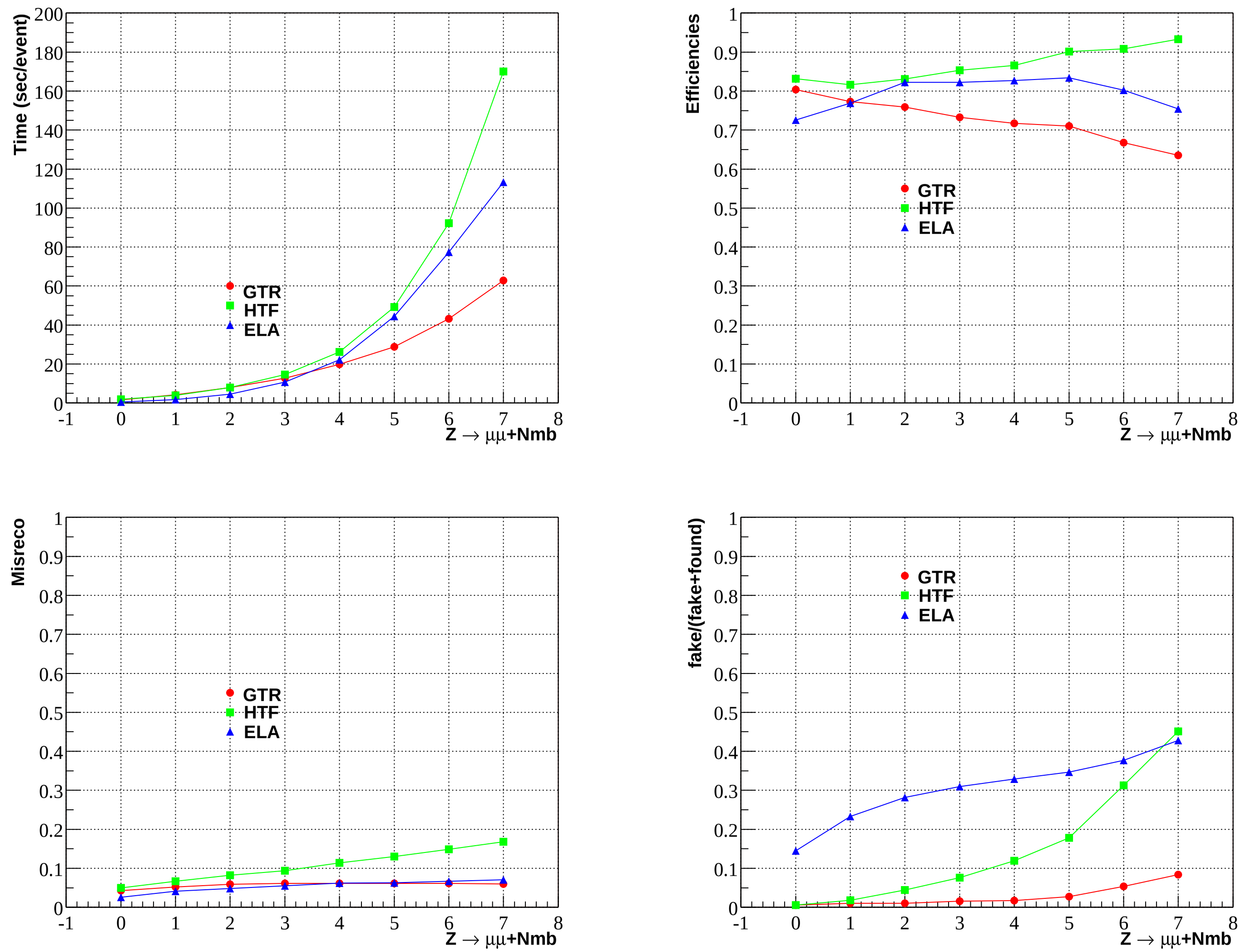
Future plans:

- **GTR performance in jets**
- **double track resolution studies**
- **feedback from physics groups (search for optimal algorithm)**



Algorithms Comparison

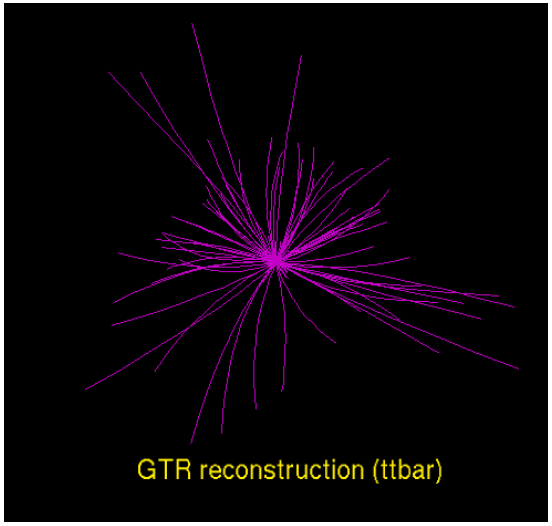
We have started to compare the different track finders

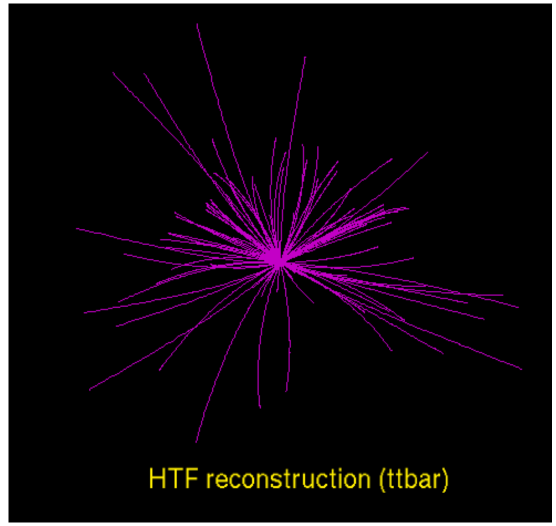


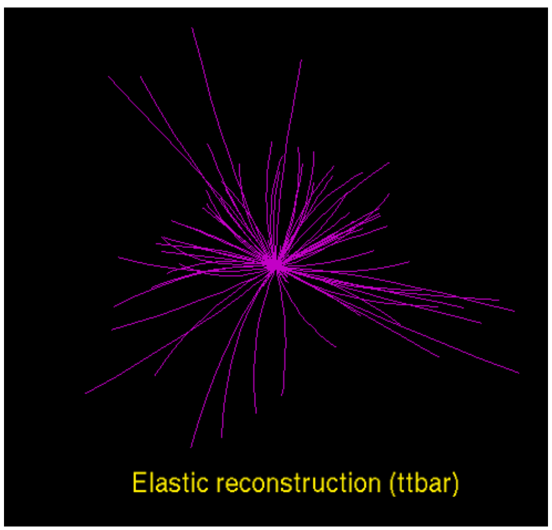
Time, efficiency, fake rate, # of misreco as a function of Nmb.

Algorithms Comparison (cont'd)

MC $t\bar{t}$ event reconstructed in DØ







- **algorithms are working**
- **we can perform cross-check**
- **compare them visually**
- **identify problems**
- **to solve problems we need new debugging and analysis tools**

Extrapolating to the Real Detector

All studies used Monte Carlo events. Expect differences with real data. Areas of concern:

Event Generation

Event generators may not match well with real events or even with each other. In our benchmark $Z \to \mu\mu$ events ISAJET produces much more underlying event than PYTHIA. PYTHIA with one background event is roughly comparable to ISAJET with none.

Detector Response

The response of the real detector may differ from our simulation (D0GSTAR plus digitization)

Detector Inefficiencies

Reduced detector response, dead channels, lead to inefficiencies in detector elements. Present track-finding paths make optimistic assumptions about these efficiencies especially in the CFT. We can expect an increase in CPU time if we allow for these inefficiencies or some loss in track-finding efficiency if we do not.

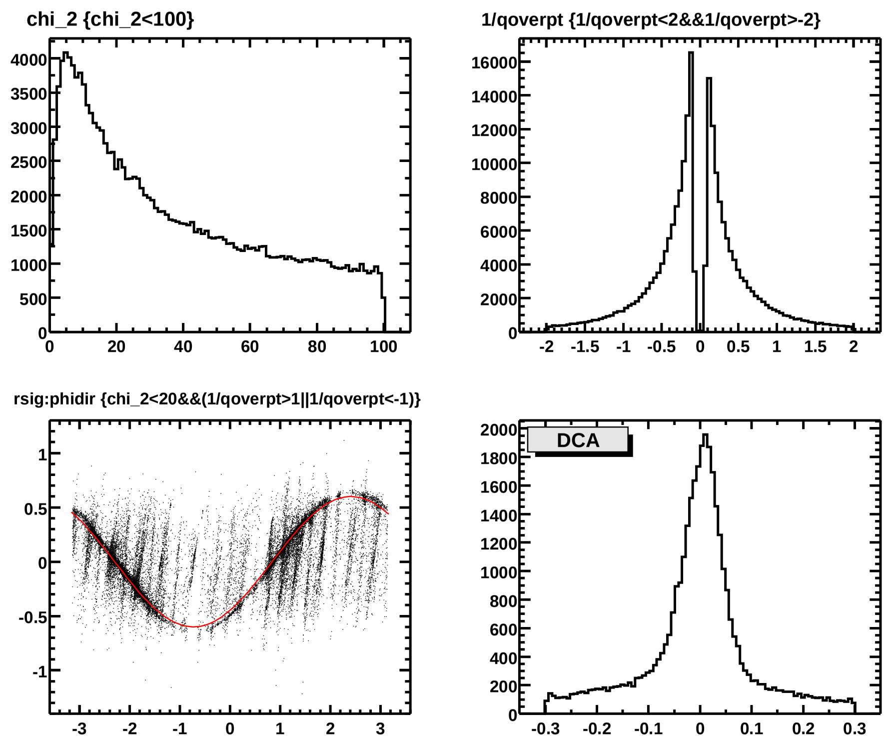
Detector Noise

If detectors have more noise that expected, then expect an increase in CPU time. If the noise is severe, resolution may be degraded.

Global Tracking with Real Data

Data-taking starting in March 2001. DØ detector was partially instrumented. We get data from SMT barrels and one sector of CFT layers.

A new set of paths was developed to adapt tracking algorithm for existing detector layout, but data surprised us:



Global Tracking with Real Data (cont'd)

A few observations

- ✍ beam size as measured with SMT is much large ($\sim 300\mu$m) than expected ($\sim 30\mu$m)
- ✍ too few CFT+SMT global tracks in magnet ON data
- ✍ DCA vs φ problem
- ✍ # of tracks too low
- ✍ φ structure

<u>Detector status</u>

SMT fully read out

- only barrels are used in current tracking
- subtract pedestals strip-by-strip

CFT partially instrumented in a sector of $\phi = 0.6 - 1.3$ rad (pre-shutdown)

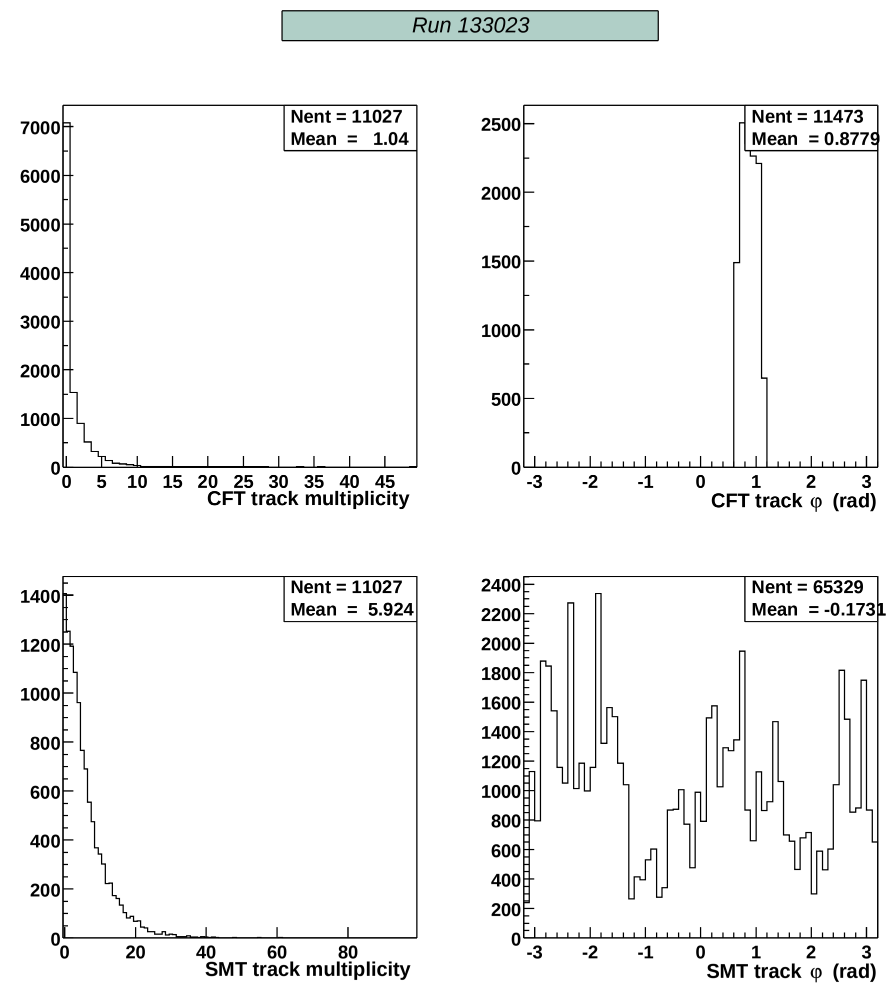
- 5 axial doublets out of 8 axials + 8 stereos
- supress channels with < 35 ADC counts
- clusters size do not quite well agree with MC

Tracking task force formed to solve problems:

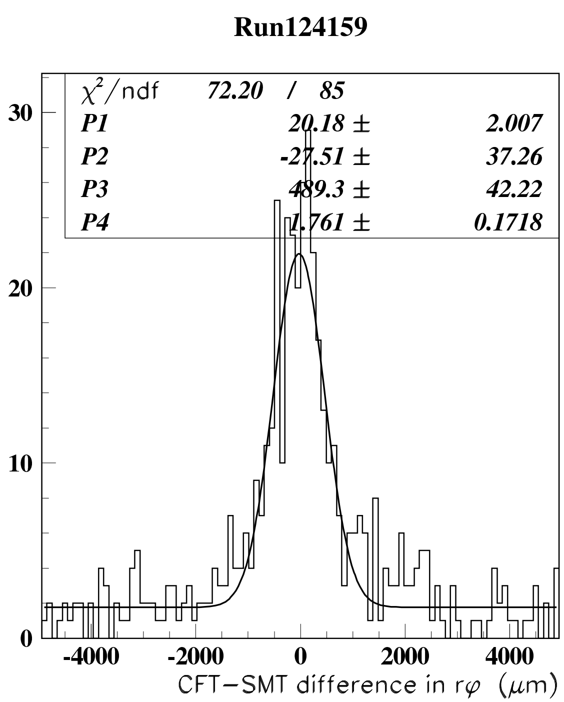
- ✓ 5mm beam offset found ➠ track reconstruction modified for this
- ✓ tracking with real data cross-checked using three algorithms
- ✓ new pedestals
- ✓ tracking $\leftrightarrow$ alignment

Global Tracking with Real Data (cont'd)

CFT and SMT track multiplicity and azimuthal angle distributions in pre-shutdown data, B=0



Global Tracking with Real Data (cont'd)

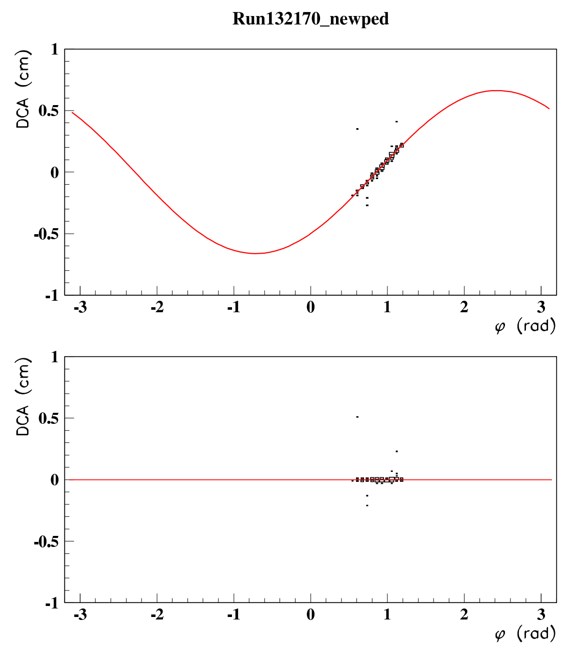


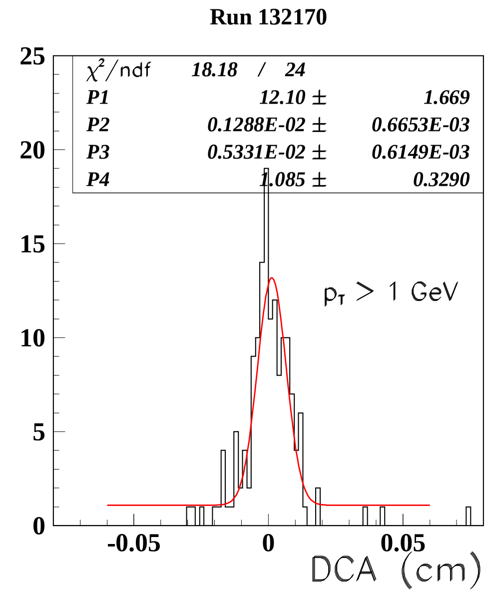
- find SMT and CFT tracks separately
- propagate SMT tracks to cylinder # 2 (first measurement)
- plot differentces in, e.g. $r \times \varphi$
  - typical cuts:
    number of missed ladders = 0
    $q/p_T < 0.25$ (B=0, soft tracks)
    $\chi^2 < 50$
- in magnet OFF data the SMT vs CFT relative alignment is good to $\Delta r\varphi = -27 \pm 37 \mu$m.

Global Tracking with Real Data (cont'd)

IP resolution with global tracks

- have limited φ acceptance in CFT
- expect much better performance
  - less fakes as compared to SMT stand-alone tracking
  - more precise momentum measurement

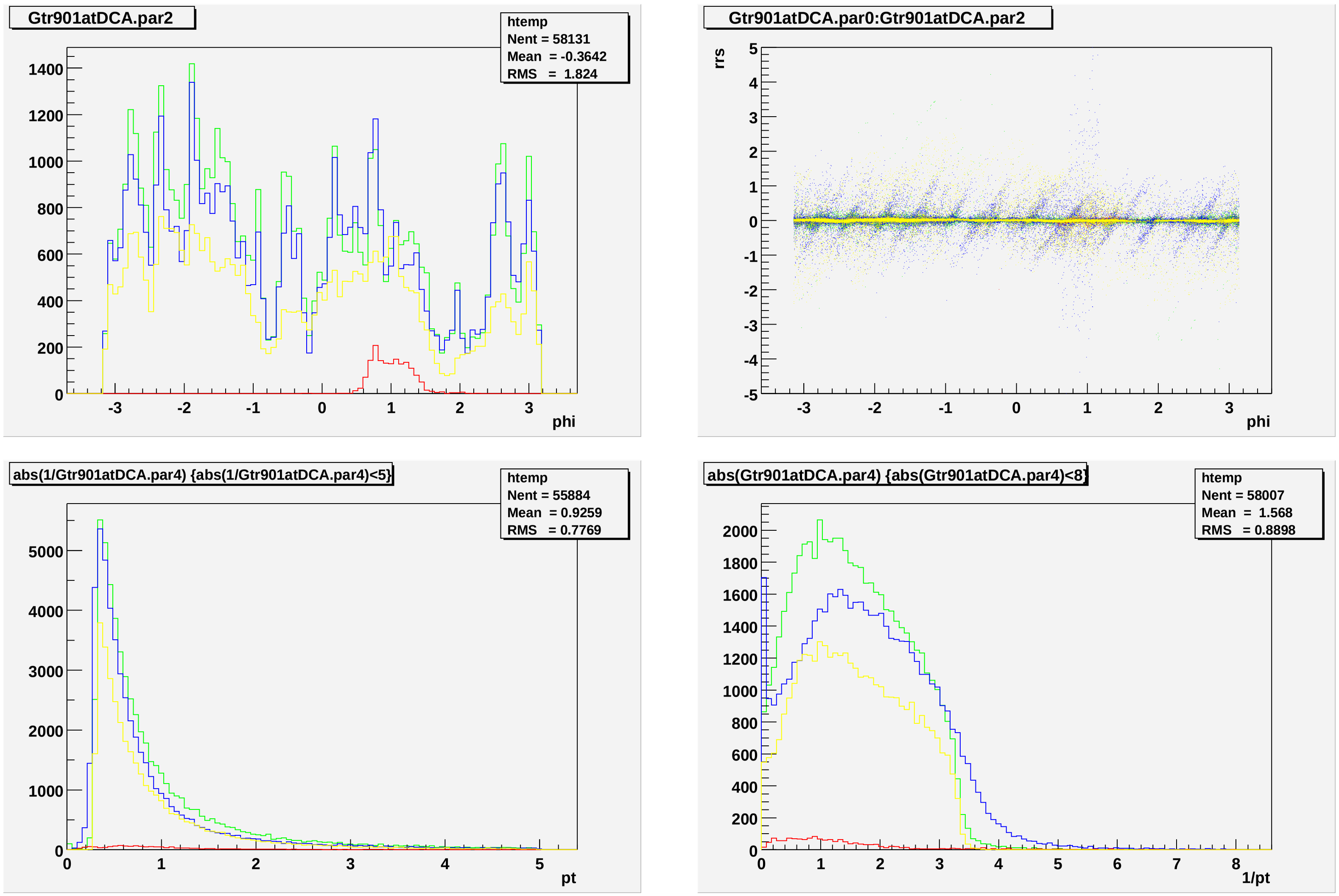




- with global tracks measure much narrower (and closer to the expected) beam width
- a (moderate) $p_T > 1$ GeV cut gives the best result up to now on: beam position (x,y)=(0.435, 0.505) cm and width 53 μm.

Global Tracking with Real Data (cont'd)

Run 132947, GTR, HTF tracking, new pedestals, DCA to vtx=(0.435 0.505).



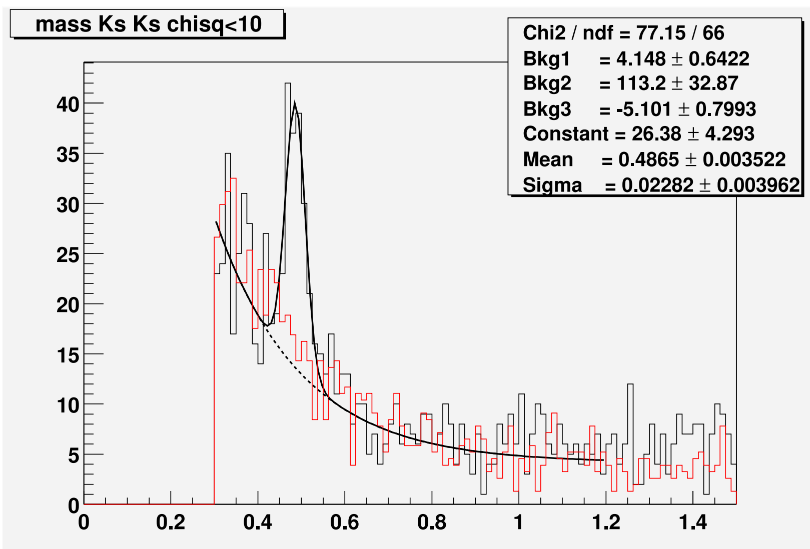
φ, DCA vs φ, $p_T < 4$ GeV, $1/p_T$, GTR - blue, HTF(smt) - green, SBF - yellow, HTF(full) - red.

Global Tracking with Real Data (cont'd)

Plenty of problems remain

- ➢ correction for dead ladders/detector regions
- ➢ alignment of SMT and CFT
- ➢ possible biases in tracking (e.g. charge-dependent φ bias)
- ➢ number of tracks per event still seems too low

But we have made significant progress:



- ✓ modified tracking to handle arbitrary beam position
- ✓ identified and diagnosed noise in SMT detectors
- ✓ fixed SMT sequencer controller timing problem
- ✓ we're able to see physics, $m(K_s) = 0.49 \pm 0.02$ GeV

DØ was moved so it is now centered within ~ 0.5 mm of zero w.r.t the nominal beam position, DØ news Nov. 30, 2001

SUMMARY

Track reconstruction software is working

We develop/support four different tracking algorithms

Recent innovations in DØ global tracking include:

- **significant improvements made in GTR timing performance**
- **design and implementation of global DØ multiple scattering propagator**
- **development of track reconstruction in non uniform magnetic field, including track extrapolation to H disks**
- **track extrapolation between DØ subdetectors from distance of closest approach out to muon system**
- **track reconstruction in gap and overlap regions, most vulnerable parts of the DØ detector**
- **integration of four tracking algorithms into unique working structure**
- **hit mask for thumbnail**
- **global objects, ObjTable $\leftrightarrow$ (cuts, propagators, mag. field update, etc.)**
- **design and implementation of analysis tools**