Machine Background in Fast Simulation

Gabriele Simi
(UMD)

- Introduction
- Update since last meeting
- Sensitive window
- Performance

Introduction

- The background from the accelerator is important aspect to make the fast simulation results more realistic:
  - It can be significant in benchmark analyses: which ones?
  - It affects the optimization of some sub-detectors
- Fast Sim geometry not detailed enough => need Full GEANT4 Simulation

Plans at Warwick meeting

- Define which volume to use
  - Probably an iterative process
- Refine the exchange format:
  - Should store additional information in the root file on the type of background
    - Brehmstrahlung, beam gas, Touscek, hadronic or EM shower, material where shower was initiated...
  - Add this to Gvertex::Cause
- Ensure conventions are the same:
  - Axes, B Field, definition of the time

Sensitive window

- Need to consider tracks in a time interval corresponding to the window in which the detector is sensitive to background, usually the L1 accept window ~ 1μs
- Sub-detectors need to implement specific filters based on their reconstruction
  - For example SVT applies a cut on the time of the hots of ~50ns, the EMC applies a cut of ~100ns
- DCH max drift time is ~0.5 μs
  - =>Set window to 1 μs

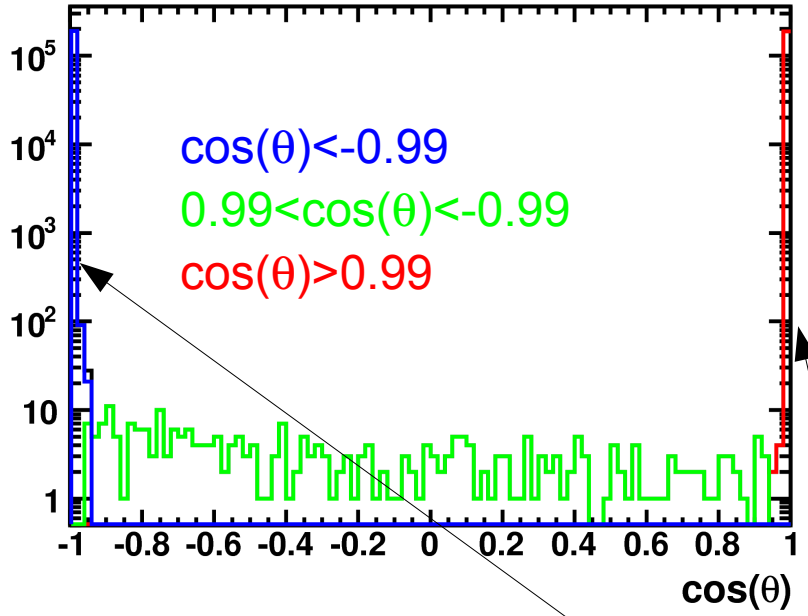
Background mixing

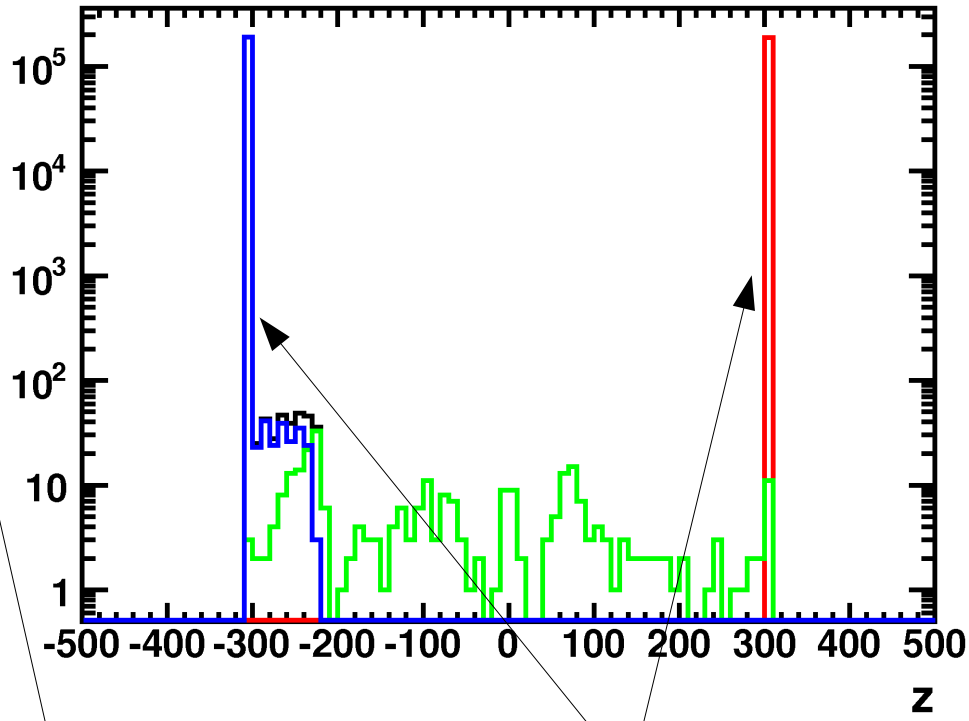
- Baseline is that Full Simulation provides events that correspond to one bunch crossing
- Fast Sim input module adds a number of events corresponding to the whole sensitive window
- Each particle has its time shifted by the time of the bunch relative to the center of the window
- Touscek needs special treatment, not contemplated yet in Fast Sim

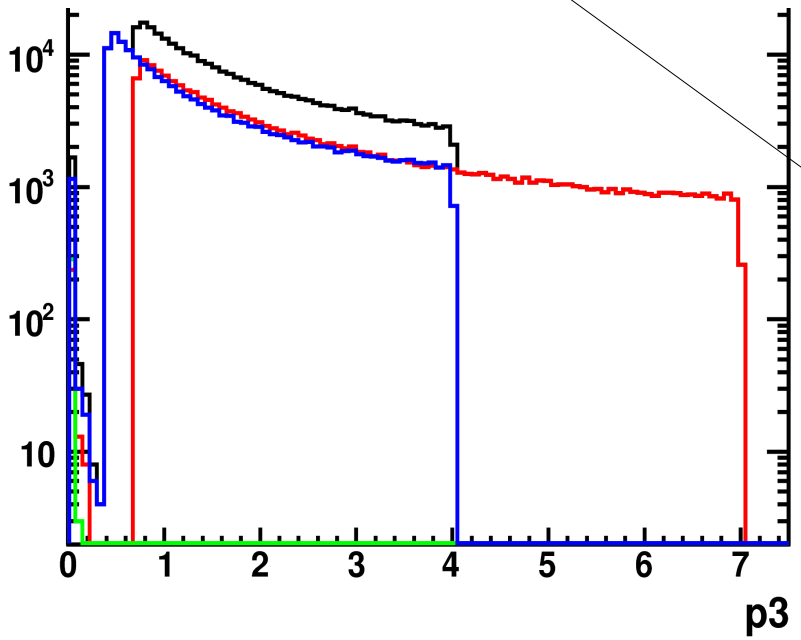
Performance

- With window=1us, bunch crossing=400MHz
  - 400 bunch crossings/event, 380 γ/b. crossing
  - PmcBkgInput: ~650 ms/event, ~150k γ/event
  - PmcSimulate: ~250 ms/event, ~150k γ/event
- =>Try to select improbable particles

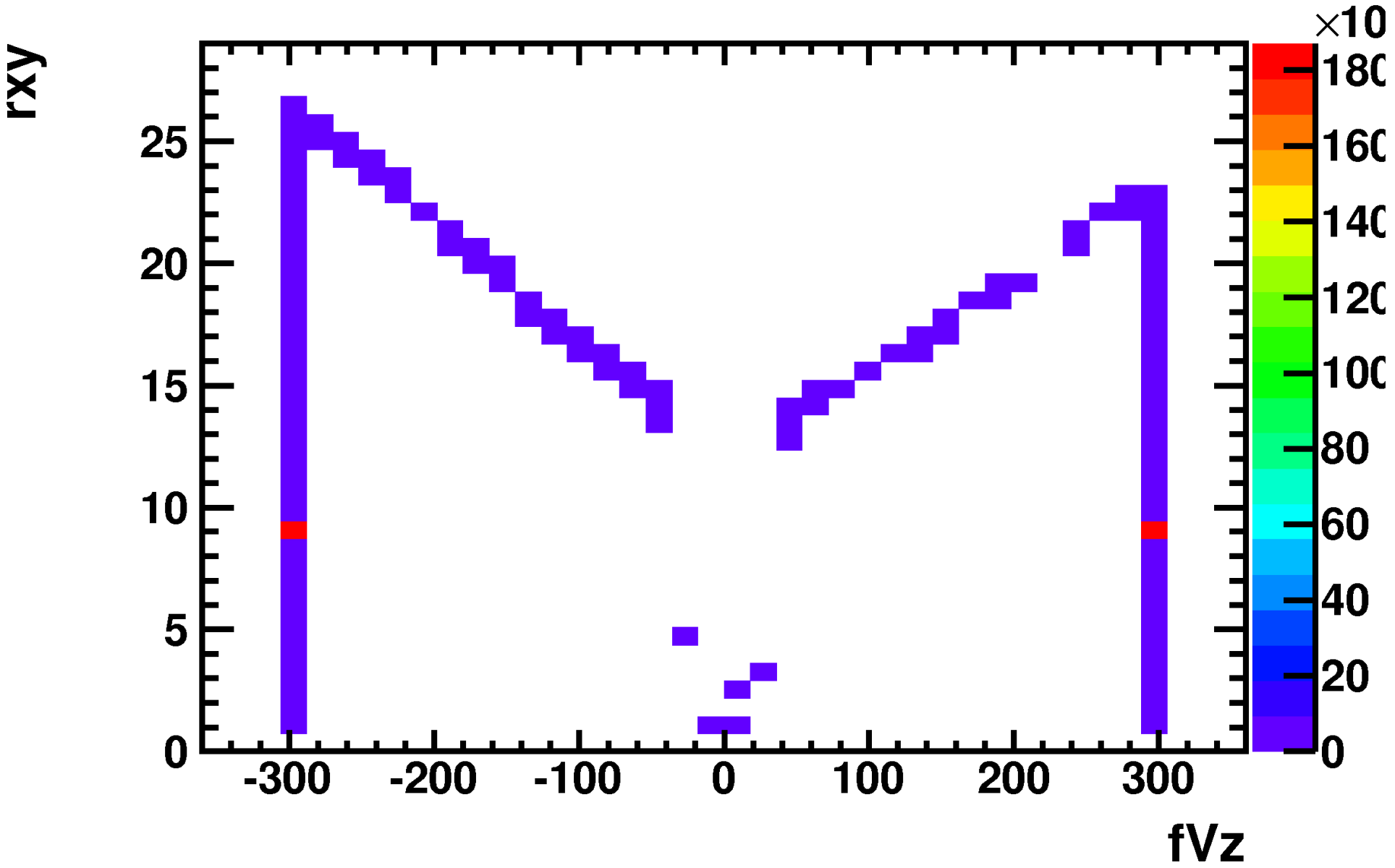
Bkg distributions







Most of the photons are parallel to the beams (+/-30mrad) and originate from z=+/-300 cm (the endcaps of the scoring volume)


rxy
25
20
15
10
5
0
-300
-200
-100
0
100
200
300
fVz
×10
180
160
140
120
100
80
60
40
20
0

Performance II

- zmin=-299cm, zmax=299cm
  - PmcBkgInput: ~400 ms, ~150k γ/event
  - PmcSimulate: 2.3ms, 300 γ/event
- overhead from reading each particle into memory~2.5 μs/particle
- =>need to read only the branch used to select
  - Can it be done with TClonesArray??

Conclusions/Plans

- Synchronized units with Bruno
- Using updated scoring volume
- Defined sensitive window
- Background events added in the sensitive window to build background frame.

BACKUP

Outlook

- Can we use this format to save a subset of the generated events and process them later ?
  - The code to build GTracks in PmcEventConverter base class can be re-used
  - We need to write code to read the decay tree from the root file
- Idea: save a selected list of generated events for later processing: useful?
  - Could be useful to optimize an analysis with a small background without re-running all the events
- Problem: when re-processing events the results will be different because of different random numbers

Conclusion

- Defined a preliminary exchange format
- Produced a test Full Sim background file
- Read into fast-sim and overlaid to generator event
- Need more information to define scoring volume
- Work can be expanded to store events selected in an analysis if useful

BACKUP

Test Full Sim Output (technical info)

- Output format is a TTree with three branches:
  - an array of TParticle
  - an event weight
  - an event number
- Particles have no mother but have an origin vertex
- The origin vertex is the point where the particle crossed the scoring volume
- Time is referred to the primary vertex

Fast Sim Input (technical info)

- Use the same code that converts StdHep events (those typically returned by generators)
- Produce a list of GTrack, Gvertex and add it to the existing one (or create a new one)
  - PmcEventConverter : base class implementing conversion from internal data to GTrack, GVertex objects
    - PmcStdHepConverter: uses as input a StdHep event
    - PmcTParticleConverter: uses as input an array of TParticle
  - PmcBuildGTracks: base module of framework to build GTrack and GVertex lists and save then into the event.

Questions

- How is the origin of the axes defined in full sim ? What abut axes direction ?
- What is the meaning of the time ? unit ?
  - Time is referred to origin of the particle, unit is 10ns
- How do we use the event weight ?
  - Filter on it using it as a prescale probability
- Should we loop on the events over and over again ?
  - Yes after reaching the end of the event-list

Questions(cont.)

- What Gvertex:cause to set ?
  - Using preassignedDecay for now
- Should we store additional information on the type of background?
  - Brehmstrahlung, beam gas, Touscek, hadronic or EM shower, material where shower was initiated...
  - Should we add items to Gvertex::Cause?
- Geant can produce unphysical particles (like geantino) that could be useful for testing.
  - Currently fast sim crashes using those particles. Should we try to fix this ?