Validation of improved iterative tracking in $\mu_{\text{iso}} + b$ jet trigger

Andrey Popov

17 Dec 2013

Outline

- Checked three approaches to decrease timing of iterative tracking proposed here:
  1. Consider at Iter0 only tracks from the first primary vertex
  2. In addition to option 1 increase p_T of seeds at Iter2 and use pixel triplets instead of pairs at Iter4
  3. In addition to option 1 increase p_T of a jet that defines a region to do Iter2 and use pixel triplets at Iter4 like in option 2
- Studied online performance of b-tagging in a CSV version of trigger `HLT_IsoMu17_eta2p1_CentralPFNoPUJet30_BTagIPIter_v5`
  - Some details on implementation are in the next slide
- Two 25 ns pile-up scenarii considered using datasets
  - `/TTbar_TuneZ2star_13TeV-pythia6-tauola/Summer13dr53X-PU25bx25_START53_V19D-v1/GEN-SIM-RAW`
  - `/TTbar_TuneZ2star_13TeV-pythia6-tauola/Summer13dr53X-PU45bx25_START53_V19D-v1/GEN-SIM-RAW`

Setup and implementation of new tracking

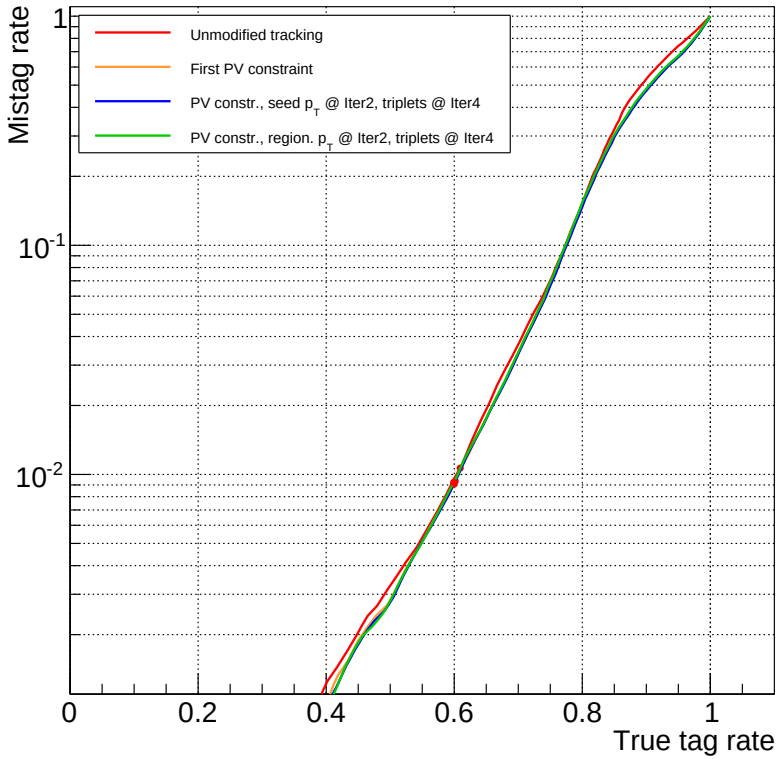
- Study is performed in `CMSSW_7_0_0_pre9`
  - Global tag is `START70_V2` as recommended by Stephanie
- Had some fun with the trigger implementation
  - ConfDB GUI fails to do anything to a menu from 70X, apparently also 62X
    - After an empty configuration is created, an attempt to save it or import a path into it hangs the GUI forever
    - At the same time it works for 53X
    - Have you faced this problem?
  - The trigger menu in 700 (`/dev/CMSSW_7_0_0/GRun/V30`) is buggy
    - Nothing serious is visible: wrong types of parameters in configuration, illegal values for some optional parameters
    - Although it can be corrected manually, it looks suspicious
    - Do you know if we are supposed to start with this menu when developing for 70X? A menu from 52X cannot be run in 70X directly
- Finally, reimplemented the CSV version of the trigger from scratch in 70X starting from the IP-based trigger
  - Use `hltGoodOnlinePVs` in *b*-tagging modules
  - Had to switch on filtering in `hltGoodOnlinePVs` (details below)

Setup and implementation of new tracking

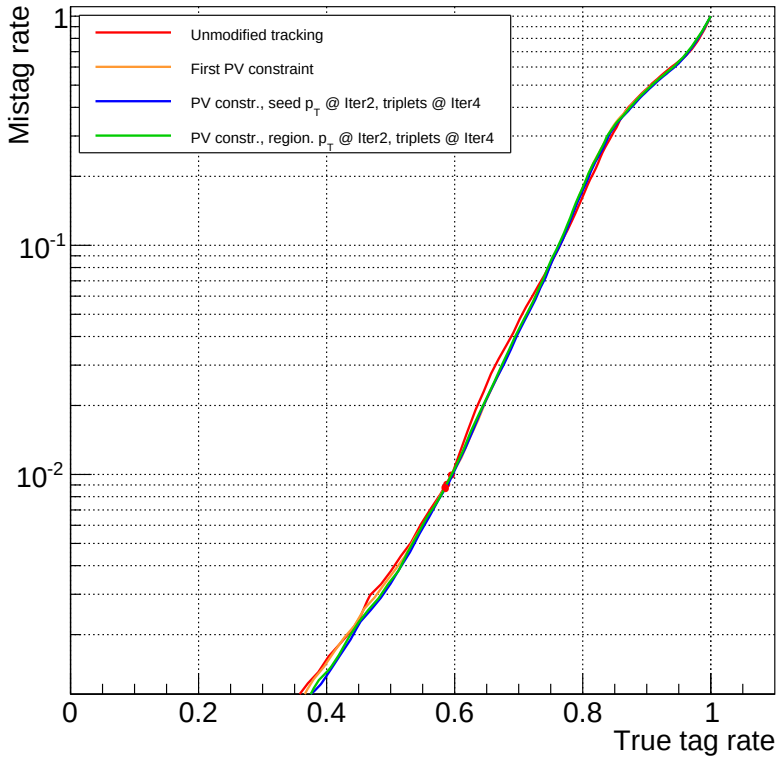
- To create paths with modified tracking, replaced all modules referenced in the `HLTIterativeTracking` sequence by ones in the three menus cited in this page
- In addition, needed to add/replace several modules for event setup
  - Can be deduced from `diff` output
- All configurations are stored as python files, nothing put into ConfDB

b vs *uds*

$\lambda_{PU} = 25$ b vs uds

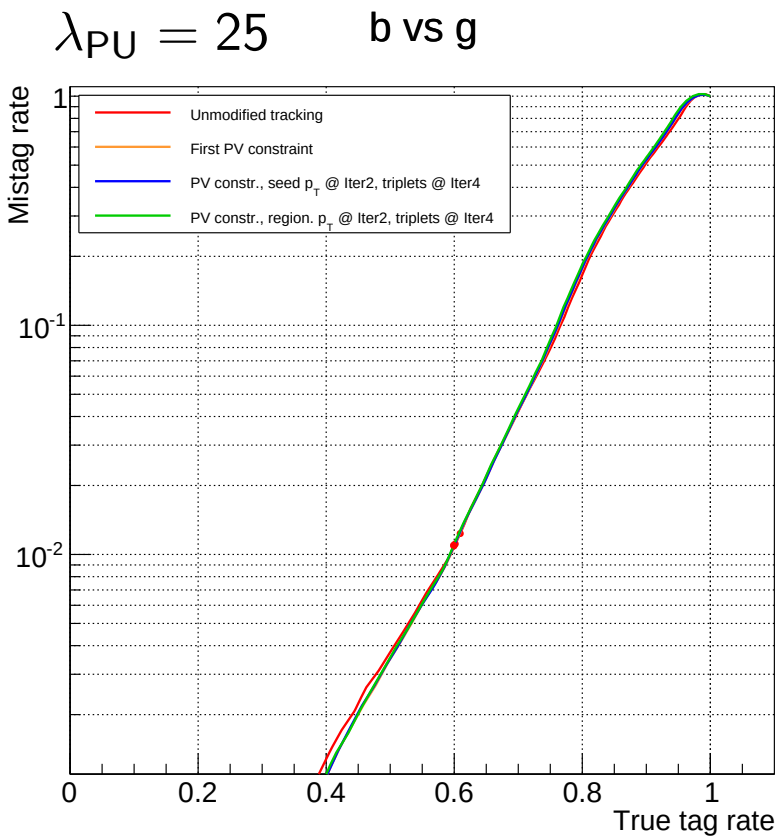


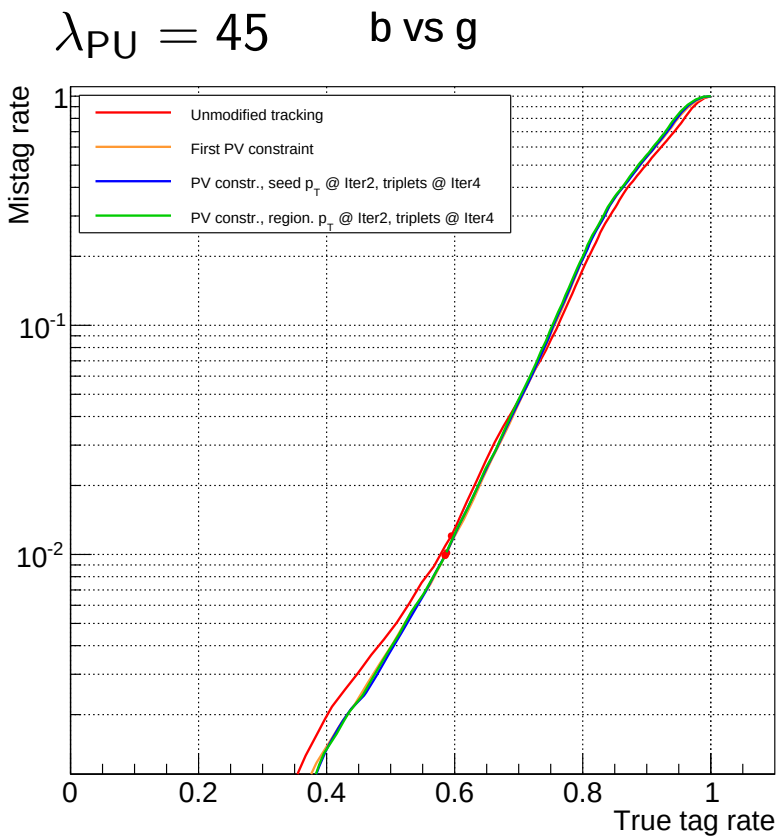
$\lambda_{PU} = 45$ b vs uds



Marked working point corresponds to discriminator value of 0.7

b vs g

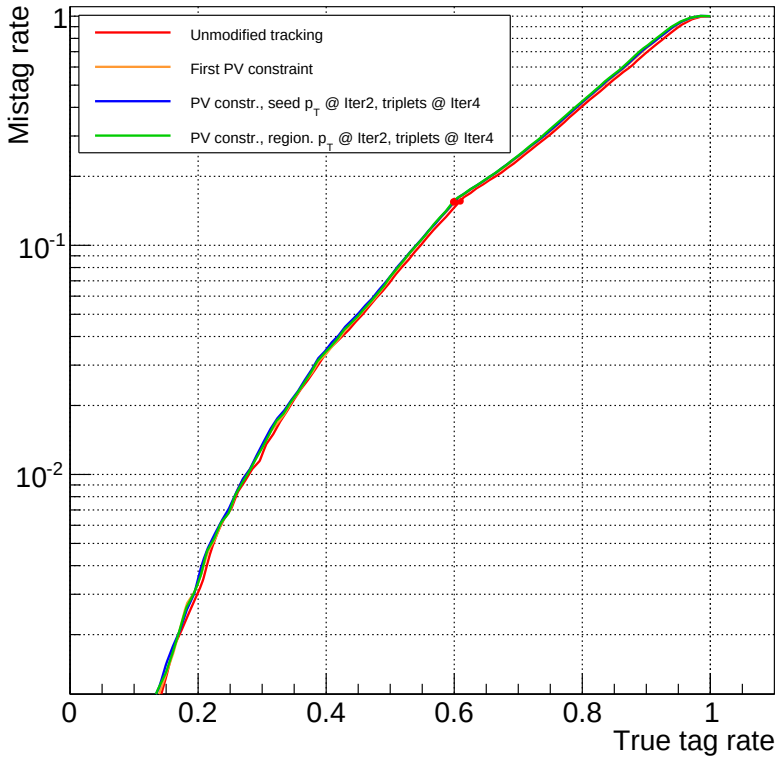




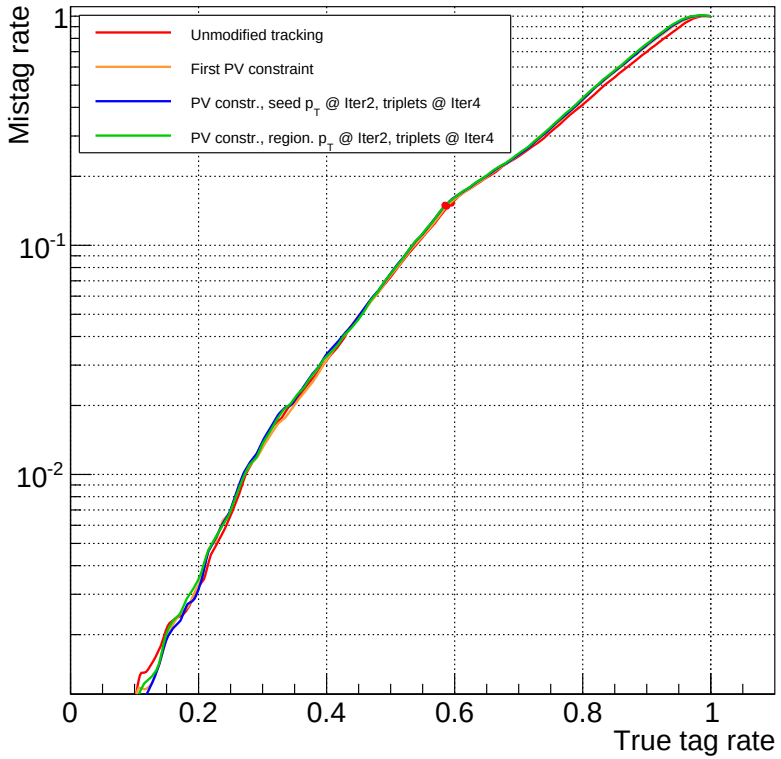
Marked working point corresponds to discriminator value of 0.7

b vs c

$\lambda_{PU} = 25$ b vs c



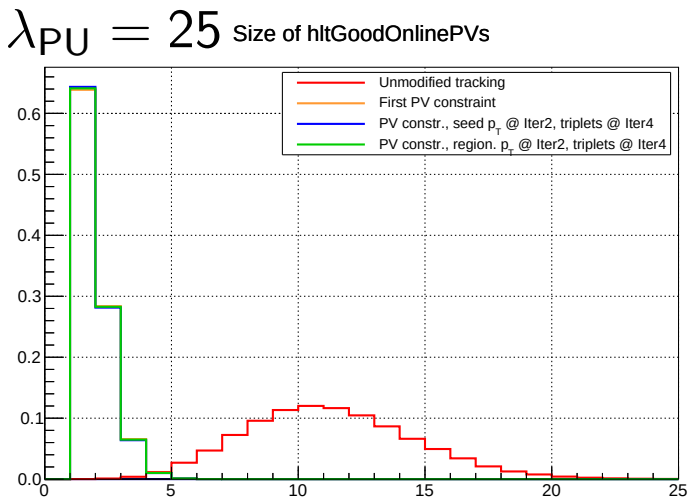
$\lambda_{PU} = 45$ b vs c

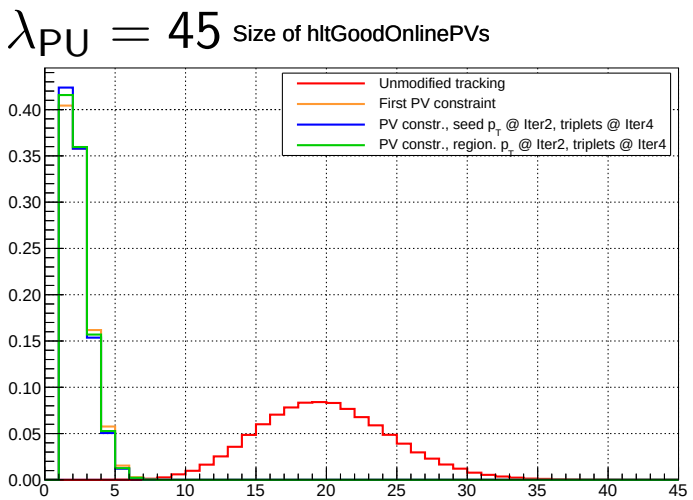


Marked working point corresponds to discriminator value of 0.7

Primary vertices

- All paths with modified tracking have a very different distribution over number of good PVs compared to the original path
  - It is reasonable as good PVs are clustered from only those tracks that are assigned to the first PV at Iter0
- But `SecondaryVertexProducer` expects `TrackIPTagInfos` to reference a valid PV
  - Otherwise an exception is thrown from here
- With modified tracking there are some events with no good PVs
  - Had to switch on filtering in `hltGoodOnlinePVs`





Summary

- No significant changes in online performance are observed
- A drastical change in distribution over number of good primary vertices is spotted, but the effect is understood