Report on the `VXDTF` performance under current background conditions containing QED-Background

`Very preliminary draft!`
`Do not circulate!`

Jakob Lettenbichler, Rudolf Frühwirth

July 3, 2014

Contents

1 Introduction

1.1 Introductory remarks

The aim of this document is to have a better and more complete picture of the current state of the low momentum track finder for the inner tracking detectors of Belle II (`VXDTF`). Three main aspects are relevant for this snapshot of the current performance:

- All important background types are finally provided for the PXD and for the SVD-sensors (including the QED background (also called 2-photon) which was not available for other detectors except the PXD).
- The `VXDTF` will soon be replaced by a redesigned version optimized for higher reliability, higher speed and better modularized design.
- I will write my PhD thesis about the `VXDTF` and its successor, therefore having some more detailed notes about the state of the `VXDTF` before its redesign will help me to do a better job in the end.

Another purpose of this report is to have a thorough documentation of the current situation. Related aspects which would be too detailed for a thesis, can be discussed here, since it is concentrating on one single topic: performance of the `VXDTF` in realistic cases.

1.2 Disclaimer

The current version of the `VXDTF` is neither optimized for speed nor for high efficiencies. A thorough optimization can only be done when having a clear picture of the conditions, which now begin to form after having a realistic estimation for the background to be faced by the `VXDTF`. Therefore the focus of the development was set on implementing functionality and testing concepts so far. Therefore now is a good time to use the knowledge collected so far for a complete redesign, which focuses more on bottlenecks in the reconstruction code than the current version. Nonetheless the current version is now to be tested and therefore some points have to be mentioned to put the following picture into the correct frame.

The settings for the filters and cutoffs are educated guesses based on preliminary estimates. The three passes with the pT-cuts at 30-125 MeV, 125-500 MeV, 500-x MeV are not chosen because of detailed studies, but simply by taking reasonably big pT-ranges. The same approach has been used for the sector size and the choice of filters. Since there are about 20 filters (each with individual tuning parameters), an arbitrary number of passes, pT-cuts and sectors per sensor, the number of parameters influencing the outcome

of the **`VXDTF`** is pretty high. This asks for a highly automated test and tuning suite, which is already under construction (started by Thomas Fabian).
But this also means that the efficiency and time consumption of the **`VXDTF`** should not be interpreted as the real limits of the **`VXDTF`**, but their relations between each other should be considered as hints where bottlenecks can be found.

1.3 Basic information

what is U and V, distinction between hits and clusters, SVD & PXD, ...

2 Background

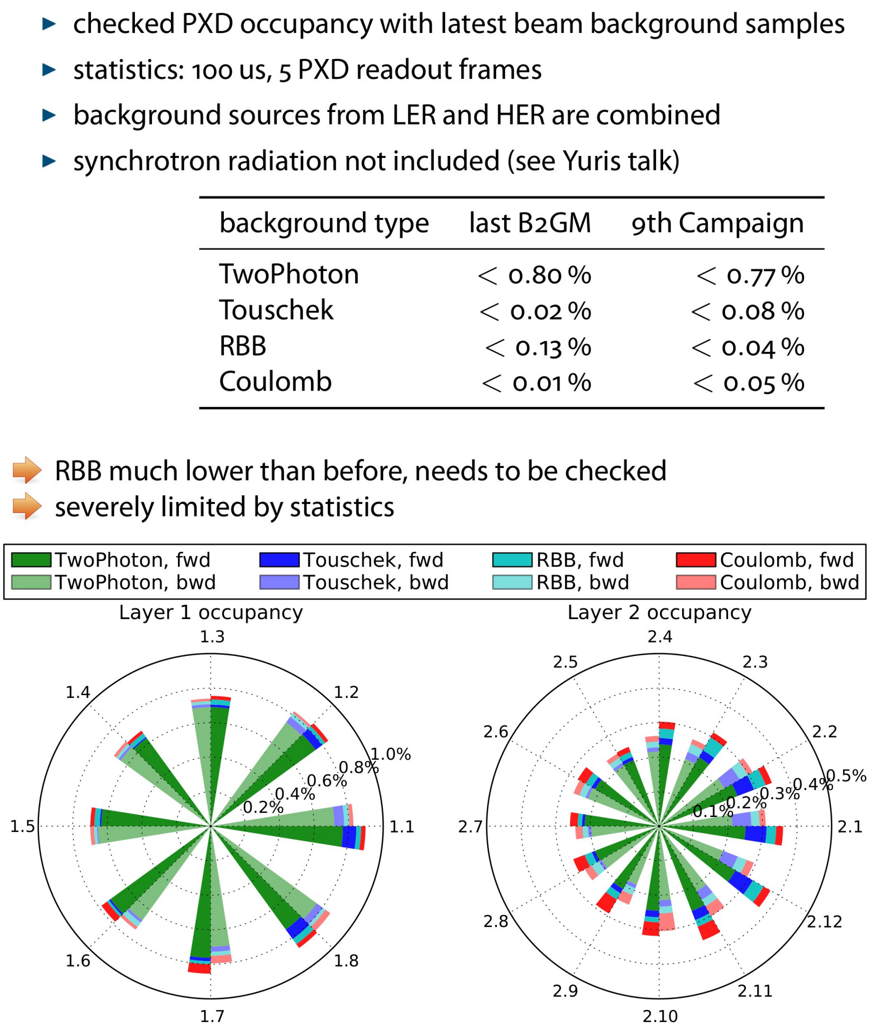
The read-out frames (ROFs) are taken from the 9th background campaign in June 2014 (Missing: download link). Most information can be found in Fig. 2.1 for the PXD and in Fig. 2.2 and Fig. 2.3 for the SVD. These figures are taken from the talks at the background session at the last B2GM in June 2014, where Martin Ritter (PXD) and Peter Kvasnicka (SVD) presented their results of the campaign.

Some of the main aspects of this campaign are:

- QED background, which is currently the most dominant and therefore the most important background type for tracking, was generated for the PXD and, actually the for the first time, for the SVD.
- There was a new implementation of the TrueHits, which increases speed and reduces size in the memory.
- PXD: 10 ROFs (total amount: 200 µs) for the QED-data and 5 ROFs (total amount: 100 µs) for all other background types.
- SVD: the same amount of time as for the PXD for the different background types which results in a much higher number of ROFs due to the smaller time window used for the SVD.

The occupancy for PXD layer 1 was ∼0.8 % and ∼0.3 % for layer 2. The occupancy is not evenly distributed, but is largest in ladders 1, 7 and 8 in layer 1, and in ladders 1, 2, 10, 11 and 12 in layer 2. (Fig. 2.1). The dominating effect is clearly the QED background. it is the source of more than 90 % of the background in the first layer and more than 75 % in the second layer. This effect can be seen in the inner SVD layers too (see Fig. 2.3). Although the QED background is the dominating effect up to layer 6, the total background level drops by a factor of 3–4 between layers 3 and 4, starting with an u/v-occupancy of about 1.5%/1% in layer 3 and dropping down to about 0.2%/0.2% in layer 6. An interesting detail can be seen when comparing the background ratios between u- and v-strips (Fig. 2.3), where e.g. the QED background is more severe for u-strips than for v-strips.

In conclusion it can be said that although the QED background dominates the other background types in all 6 layers, its effect is only problematic for the PXD. The comparatively small time window of a SVD ROF effectively suppresses any severe effects and therefore reduces the amount of ghost hits. The detailed settings are described in the next chapter (3.1.1).

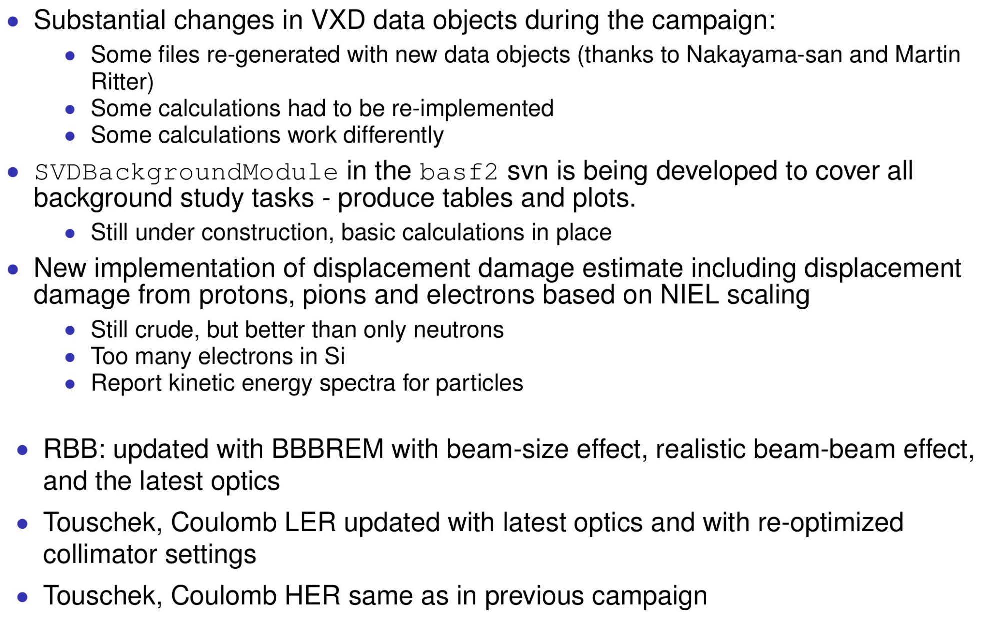


- checked PXD occupancy with latest beam background samples
- statistics: 100 us, 5 PXD readout frames
- background sources from LER and HER are combined
- synchrotron radiation not included (see Yuris talk)

background type	last B2GM	9th Campaign
TwoPhoton	$< 0.80\,\%$	$< 0.77\,\%$
Touschek	$< 0.02\,\%$	$< 0.08\,\%$
RBB	$< 0.13\,\%$	$< 0.04\,\%$
Coulomb	$< 0.01\,\%$	$< 0.05\,\%$

➡ RBB much lower than before, needs to be checked
➡ severely limited by statistics

Figure 2.1: Overview of the conditions for the 9th background campaign. The plots show the current occupancies in percent for the sensors of the PXD.



- Substantial changes in VXD data objects during the campaign:
    - Some files re-generated with new data objects (thanks to Nakayama-san and Martin Ritter)
    - Some calculations had to be re-implemented
    - Some calculations work differently
- `SVDBackgroundModule` in the `basf2` svn is being developed to cover all background study tasks - produce tables and plots.
    - Still under construction, basic calculations in place
- New implementation of displacement damage estimate including displacement damage from protons, pions and electrons based on NIEL scaling
    - Still crude, but better than only neutrons
    - Too many electrons in Si
    - Report kinetic energy spectra for particles

- RBB: updated with BBBREM with beam-size effect, realistic beam-beam effect, and the latest optics
- Touschek, Coulomb LER updated with latest optics and with re-optimized collimator settings
- Touschek, Coulomb HER same as in previous campaign

Figure 2.2

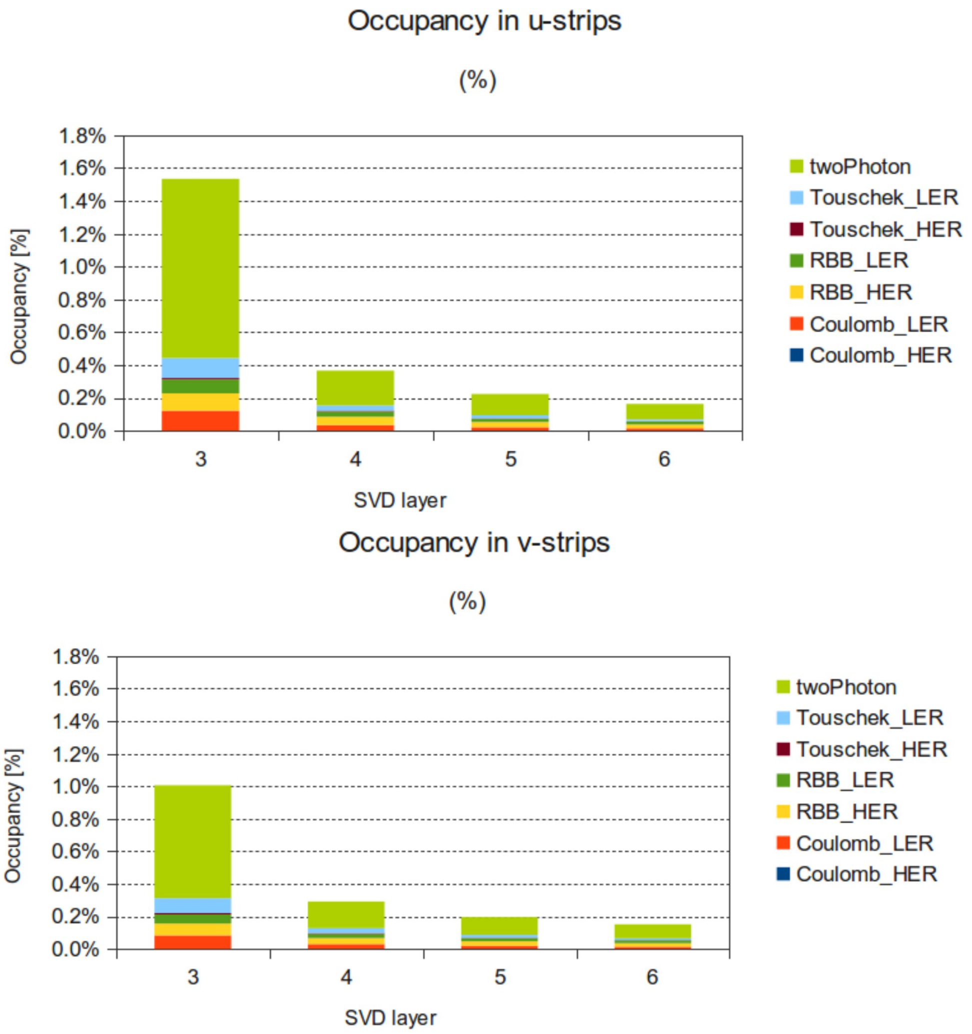


Figure 2.3: The occupancies of the 9th background-campaign for u- and v-strips separately.

3 Study

TODO: comparison of ghost rate for different cases, detailed look at time consumption of the individual parts of the **`VXDTF`** to identify bottlenecks

3.1 Settings

The framework was run using the following revision:
`Last Changed Author: piilonen`
`Last Changed Rev: 11193`
`Last Changed Date: 2014-06-20 22:05:24 +0200 (Fri, 20 Jun 2014)`
The following list gives a short overview over the most important modules for the present study and their settings, inparticular those that differ from the standard settings. Some modules are discussed in a separate subsection since they have to be described in more detail in order to present all the settings relevant for this study.

- 2000 Events were created by the `EvtGen`-module, using its standard-settings (afaik Υ4S). The only exception is the case of full background for both SVD and PXD, where due to the small number of available ROFs (see Chapter 2 for more details) for the PXD, the number of events was limited to 100.

- The events were simulated using the `FullSim`-module(geant4.10), where `'StoreAllSecondaries'` was set to `true`. This is necessary for the correct assignment of clusters to their mc particles, which is only relevant for the analysis of the outcome of the **`VXDTF`**.

- The `Geometry` module was reduced to the following detector components:

  `'BeamPipe', 'MagneticFieldConstant4LimitedRSVD', 'PXD', 'SVD'`

  This implies that there the only curlers in the test data are low momentum particles not leaving the VXD.

- The ROF-Files are loaded using the new `BeamBkgMixer` module. Details about the settings will be given in Section 3.1.1.

- The `Digitizer` and `Clusterizer` modules for the PXD and the SVD were used with their standard settings.

- The **`VXDTF`** will be discussed in Section 3.1.2.

- To be able to compare the outcome of the **`VXDTF`**, the `TrackFinderMCTruth` was run with the following settings:

```
param_mctrackfinder = {
  'UseCDCHits': 0,
  'UseSVDHits': 1,
  'UsePXDHits': mcPXDHits,
  'Smearing': 0,
  'UseClusters': True,
  'MinimalNDF': 6,
  'WhichParticles': ['primary'],
  'GFTrackCandidatesColName': 'mcTracks' }
```

where `'mcPXDHits'` was `true` if PXD was used for tracking and `false` if not.

- Track candidates (TCs) produced by both track finders were compared using the `TFAnalyzer`-module. Its settings were as follows:

```
param_analyzer = {
'printExtentialAnalysisData': False,
'caTCname': 'caTracks',
'collectorDisplayId': activateCollectorAnalyzer,
'rootFileName': [analyzerRootOut, 'RECREATE'] }
```

3.1.1 `BeamBkgMixer`

Five different runs werde performed with the **VXDTF**, two runs with SVD only (with and without BG) and three runs with VXD (noBG, smallBG and fullBG):

- `runSVDnoBG`: `BeamBkgMixer` deactivated.
- `runSVDwithBG`: `BeamBkgMixer` activated using the following settings:

```
bgmixer = register_module('BeamBkgMixer')
bgmixer.param('backgroundFiles', ['../../Coulomb_LER_100us.root'])
bgmixer.param('backgroundFiles', ['../../Coulomb_HER_100us.root'])
bgmixer.param('backgroundFiles', ['../../RBB_HER_100us.root'])
bgmixer.param('backgroundFiles', ['../../RBB_LER_100us.root'])
bgmixer.param('backgroundFiles', ['../../Touschek_HER_100us.root'])
bgmixer.param('backgroundFiles', ['../../Touschek_LER_100us.root'])
bgmixer.param('backgroundFiles', ['../../twoPhoton_200us.root'])
bgmixer.param('components', ['SVD'])
```

- `runVXDnoBG`: `BeamBkgMixer` deactivated.
- `runVXDsmallBG`: `BeamBkgMixer` activated using the following settings:

```
bgmixer = register_module(’BeamBkgMixer’)
bgmixer.param(’backgroundFiles’, [’../../Coulomb_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Coulomb_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../RBB_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../RBB_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Touschek_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Touschek_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../twoPhoton_200us.root’])
bgmixer.param(’components’, [’SVD’, ’PXD’])
```

- `runVXDfullBG`: `BeamBkgMixer` activated using the following settings:

```
bgmixer = register_module(’BeamBkgMixer’)
bgmixer.param(’backgroundFiles’, [’../../Coulomb_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Coulomb_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../RBB_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../RBB_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Touschek_HER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../Touschek_LER_100us.root’])
bgmixer.param(’backgroundFiles’, [’../../twoPhoton_200us.root’])
bgmixer.param(’components’, [’PXD’])
bgmixer.param(’maxTime’, 10000) # 800.0  Time window upper edge in nano seconds
bgmixer.param(’minTime’, -10000) # -1000.0 Time window lower edge in nano seconds

bgmixerSVD = register_module(’BeamBkgMixer’)
bgmixerSVD.param(’backgroundFiles’, [’../../Coulomb_LER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../Coulomb_HER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../RBB_HER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../RBB_LER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../Touschek_HER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../Touschek_LER_100us.root’])
bgmixerSVD.param(’backgroundFiles’, [’../../twoPhoton_200us.root’])
bgmixerSVD.param(’components’, [’SVD’])
```

The `BeamBkgMixer` does not create realistic ROFs for the PXD in its standard settings, where it reduces the ROFs of the PXD to 1.8 ms instead of the full 20 ms. This can be changed when modifying the parameters `minTime` and `maxTime` to cover a full 20 ms-window. To see the behavior for the reduced amount of background hits, the standard setting of the `BeamBkgMixer` is used also for `runVXDsmallBG`. It therefore contains only 1/12th of the full number background hits per event.

For the worst realistic scenario, two different instances of the `BeamBkgMixer` were used because this allows the SVD to cut out more ROFs of the same BG-Block, since it uses a smaller window for cuts (smaller slices, more different pieces). This behavior is part of the approach used in the `BeamBkgMixer`, which selects a random slice of fixed size within the range of the background data for each event. If all parts are covered, a new round starts with other random selections of the same size.

However, using two instances has a downside too: it destroys the correlation of the BG between the two detectors. This is of course not the best approach since this also means that fake tracks created by the BG will not be reconstructed and therefore do not add to the fake rate of the **VXDTF**. On the other hand, the BG rates in the SVD are so low compared to the PXD and the number of hits in the PXD so high, that in my opinion this should not have a relevant effect on the result.

3.1.2 VXDTF

For the runs of the **VXDTF** the following settings were made for the case of SVD-only tracking:

```
secSetup = ['secMapEvtGenOnR10933June2014SVDStd-moreThan500MeV_SVD', \
    'secMapEvtGenOnR10933June2014SVDStd-125to500MeV_SVD', \
    'secMapEvtGenOnR10933June2014SVDStd-30to125MeV_SVD']
TFDebugLevel = 2
killThreshold = 2000
hiocThreshold = 500
cutoffTune = 0.06
minState = 2
minLayer = 4
qiType = 'circleFit'
filterOverlaps = 'hopfield'
activateCollector = 0

param_vxdtf = {
    'activateBaselineTF': 0,
    'tccMinState': [minState],
    'tccMinLayer': [minLayer],
    'standardPdgCode': 211,
    'sectorSetup': secSetup,
    'calcQIType': qiType,
    'debugMode': 0,
    'killEventForHighOccupancyThreshold': killThreshold,
    'highOccupancyThreshold': hiocThreshold,
    'cleanOverlappingSet': False,
    'filterOverlappingTCs': filterOverlaps,
    'TESTERexpandedTestingRoutines': True,
    'qiSmear': False,
    'smearSigma': 0.000001,
    'GFTrackCandidatesColName': 'caTracks',
```

```
    'tuneCutoffs': cutoffTune,

    'activateDistanceXY': [True],
    'activateDistance3D': [True],
    'activateDistanceZ': [False],
    'activateSlopeRZ': [True],
    'activateNormedDistance3D': [False],

    'activateAngles3D': [True],
    'activateAnglesXY': [False],
    'activateAnglesRZ': [False],
    'activateDeltaSlopeRZ': [True],
    'activateDistance2IP': [False, False, False],
    'activatePT': [False, False, False],
    'activateHelixParameterFit': [False],
    'activateDeltaSlopeZOverS': [False],
    'activateDeltaSOverZ' : [False],

    'activateAngles3DHioC': [True],
    'activateAnglesXYHioC': [True],
    'activateAnglesRZHioC': [False],
    'activateDeltaSlopeRZHioC': [False],
    'activateDistance2IPHioC': [False],
    'activatePTHioC': [False],
    'activateHelixParameterFitHioC': [False],
    'activateDeltaPtHioC': [False],
    'activateDeltaDistance2IPHioC': [False],

    'activateZigZagXY': [False, True, True],
    'activateZigZagRZ': [False],
    'activateDeltaPt': [False, False, False],
    'activateDeltaDistance2IP': [False],
    'activateCircleFit': [False],
    'tuneCircleFit': [0.00000001],
    'displayCollector': activateCollector,
    }
vxdtf.param(param_vxdtf)
```

The meaning of the different parameters can be retrieved when typing `basf2 -m VXDTF` in a shell where the framework is set up. For the PXD cases, most parameters stay the same, except switching to a VXD sectormap and some other thresholds. Here is a list of the parameters which are different from the SVD-only setting:

```
killThreshold = 5500
```

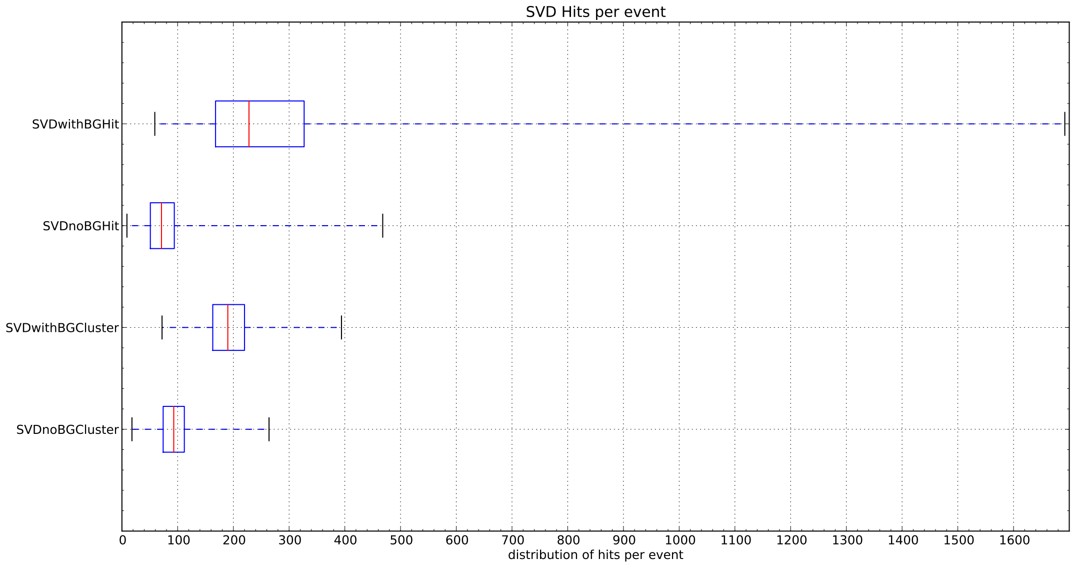


Figure 3.1: Boxplots for the cases of no background and with background for SVD Clusters and Hits

```
hiocThreshold = 400
minState = 2
minLayer = 4
cutoffTune = 0.22
secSetup = ['secMapEvtGenOnR10933June2014VXDStd-moreThan500MeV_PXDSVD', \
  'secMapEvtGenOnR10933June2014VXDStd-125to500MeV_PXDSVD', \
  'secMapEvtGenOnR10933June2014VXDStd-30to125MeV_PXDSVD']
```

the following parameters are changed again for `runVXDfullBG` to meet the extreme circumstances of 10,000 hits per event.

```
killThreshold = 500000
hiocThreshold = 500
minState = 4
cutoffTune = 0.15
```

3.2 Results

3.2.1 Hits, Clusters & Combinatorics

In Fig. 3.1 one can see some boxplots summarizing the distribution of the number of SVD clusters and hits per event, both without and with background. The red line in the box represents the median of the distribution, the box encloses the first and third quartile,

Table 3.1: 2HC means Two-Hit-Combination, qX is the quantile at position X

	q0	q0.25	median/q0.5	q0.75	q1
PXDClusters NOBG	5	17	22	27	78
PXDClusters SMALLBG	641	813	877	945	1141
PXDClusters FULLBG	8807	9217	9461	9546	9757
SVDClusters NOBG	18	74	93	112	264
SVDClusters WITHBG	72	163	190	220	394
SVDHits NOBG	9	51	71	94	468
SVDHits WITHBG	59	168	228	327	1692
2HC SVD NOBG	17	164	282	466	32144
2HC SVD WITHBG	106	707	1206	2483	111462
2HC VXD NOBG	24	240	385	601	14190
2HC VXD SMALLBG	10768	24182	28878	34829	163574
2HC VXD FULLBG	1533240	2252747	2412176	2475277	2704628

while the whiskers mark the largest and the smallest value that occurred during the runs. A small number of clusters per sensor adds up to a smaller number of hits, since one needs always two clusters to form one hit. But this effect is dominated by the creation of ghost hits at higher occupancy. Therefore not only the maximum values increase when switching from clusters to hits, but also the median value, when background is added. The opposite is true when there is no background, here the median of hits is less than the one of the clusters. Table 3.1 summarizes Fig. 3.1, Fig. 3.2 and Fig. 3.3.

In Fig. 3.1 the boxplots show the results for the PXD without background, the smaller time window for PXD-ROFs in `runVXDsmallBG`, and full background.

While the median of SVDHits without background is smaller by only a factor of 3 than the median including SVD background, the case for the PXD is more severe. Even the small ROF size of 1.8 ms already increases the rate by a factor of over 35. The full background with a ROF size of 20 ms increases the total amount of clusters in the PXD by a factor of 430. This means that the signal to noise ratio is about $1 : 430$. The initial assumption for the **`VXDTF`** was a ratio of $1 : 20 - 1 : 100$, which now turns out to be far too optimistic.

The combinatorial problem in track finding grows exponentially with the number of hits. This can be illustrated by counting the number of possible combinations of two compatible hits. Compatible in this case means that the combinatorial problem has already been reduced by using the sector map. The unfiltered number of combinations can not be easily measured since hits are combined not only on neighboring layers, but overlapping regions and missing hits have to be considered too. In order to give at least a rough estimate, the following formulas have been used to calculate the number of combinations when constraining the combinations to hits on neighboring layers:

$$n_{\mathrm{TOT}} = n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 \tag{3.1}$$

$$n_{\mathrm{TOT}} = n\mathrm{Hit}_{\mathrm{PXD}}/2 * n\mathrm{Hit}_{\mathrm{PXD}}/2 * n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 * n\mathrm{Hit}_{\mathrm{SVD}}/4 \tag{3.2}$$

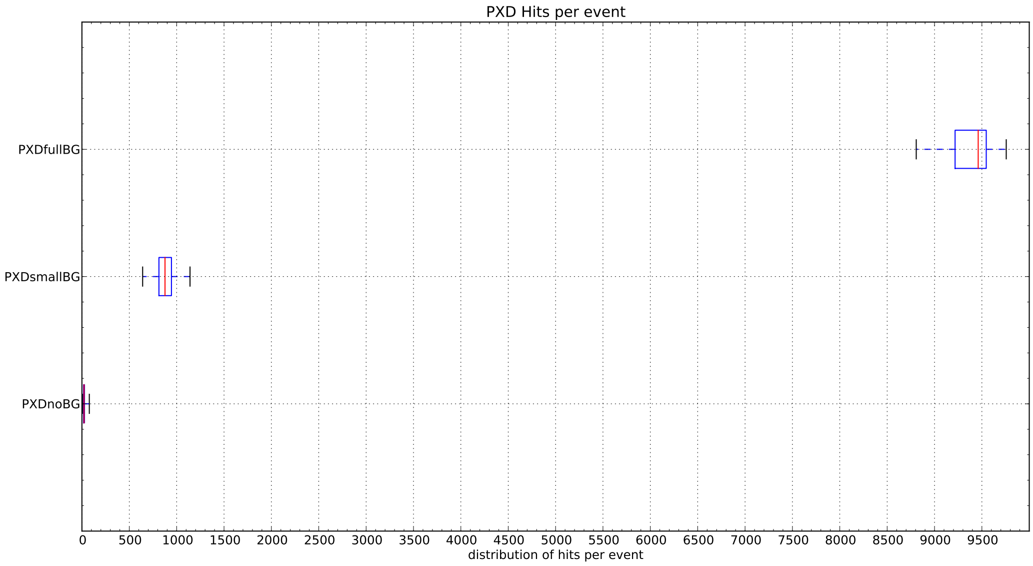


Figure 3.2: Boxplots for the cases of no background, small background and with background for PXD Hits

The following values are calculated by applying (Eq.3.1) for SVD-only and (Eq.3.2) for VXD on the median values of Table 3.1 to illustrate the cases mentioned:

- SVD noBG $n_{\mathrm{TOT}} \sim 40,658$
- SVD withBG $n_{\mathrm{TOT}} \sim 9,318,137$
- VXD noBG $n_{\mathrm{TOT}} \sim 4,919,618$
- VXD withBG $n_{\mathrm{TOT}} \sim 2.085178244 * 10^{14}$

To find out the reduction factor of the sector map, the ratio of n_{TOT} and the number of combinations taken from Table 3.1 can be calculated:

- SVD noBG $f_{\mathrm{red}} = 40,658/282 = 144$
- *SVD withBG* $f_{\mathrm{red}} = 9,318,137/1,206 = 7,726$
- *VXD noBG* $f_{\mathrm{red}} = 4,919,618/385 = 12,778$
- *VXD withBG* $f_{\mathrm{red}} = 2.085178244 * 10^{14}/2412176 = 86,443,868 \sim 8.64 * 10^{7}$

This makes clear that preselection of possible hit combinations via sorting into the sector map is essential for a reasonable speed of the **VXDTF**. Fig. 3.3 shows the number of possible two-hit-combinatins [2HC] for different cases. The value increases from a median

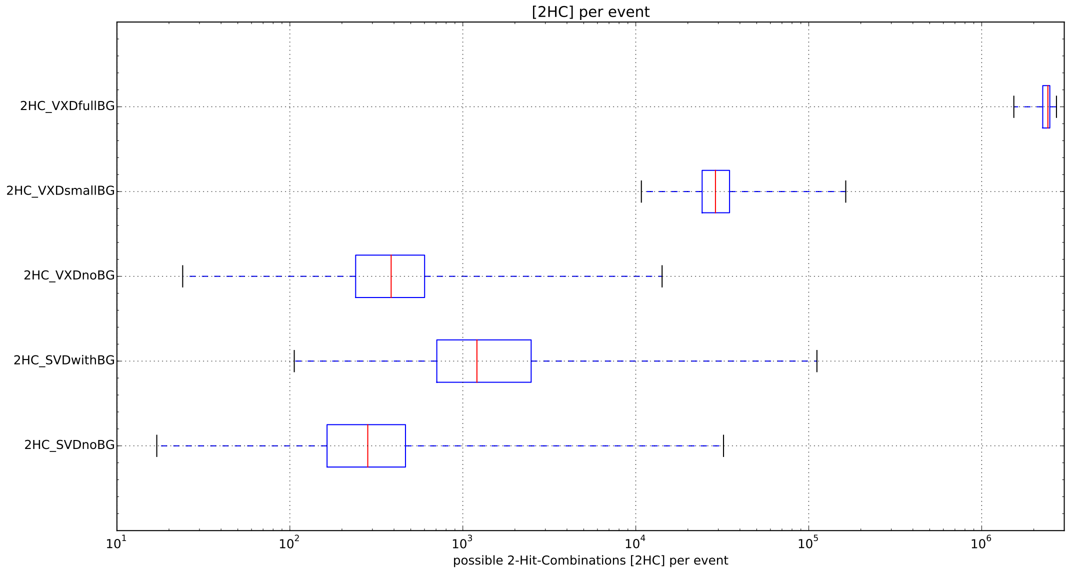


Figure 3.3: Boxplots describing the behavior of the compatible two-hit-combinations [2HC] for different cases.

Table 3.2: Time consumption of the VXDTF in [ms] for different cases. The average time consumption of these samples has always been higher than the third quartile

	q0	q0.25	median/q0.5	q0.75	mean	q1
SVD NOBG	0.38	2.0	2.96	4.78	5.44	353.67
SVD WITHBG	1.12	4.11	6.32	10.61	12.2	395.33
VXD NOBG	0.5	2.11	3.1	5.18	14.2	6209.3
VXD SMALLBG	19.67	32.92	40.24	51.88	80.47	9874.74
VXD FULLBG	1601.83	1968.04	2330.75	7536.67	40599.05	1455094.61

of about 300 to about 2.7 million combinations per event when switching to VXD with full background.
These combinations are those which are actually tested by the segfinder by applying its filters. The accepted combinations (which are now far less than the combinations mentioned above) are then stored as *Cells* for the **VXDTF**. The effect of combinatorics on the time consumption of the **VXDTF** is shown in the next section 3.2.2.

3.2.2 Time consumption

After having a look at the number of hits and their number of possible combinations, the resulting time consumption is the next important parameter. In Table 3.2 one can find an overview of the results of the time consumption measurement. The same values

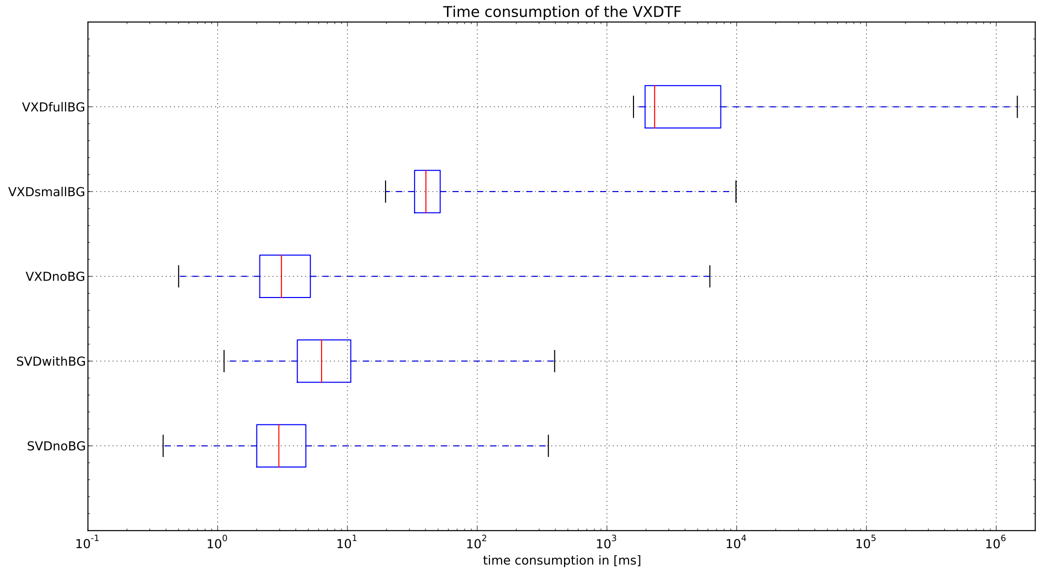


Figure 3.4: Boxplots for the cases of no background, small background and with background for PXD Hits.

are presented as box plots in Fig.3.4. A value which can be found in Table 3.2 but is missing in Fig.3.4 is the average time consumption per event. Since it is an indicator of the total time consumption over all events, it is more important than the median in this context. If there is a considerable deviation between the mean and the median of the same distribution, this is a strong indication of outliers. Therefore a high ratio of the mean over the median of the sample shows that some tuning of the settings is required in order to correctly discard outlying events.

This effect can best be seen by looking at the median (2.33s/event) and at the mean (40.6s/event) time consumption for the worst case (`runVXDfullBG`). The discrepancy is mainly caused by a single event with a duration of 1455s or about 24 minutes. It is clear that such events have to be discarded in order to have enough time for the other ones.

The ratio of the median of the fastest case (SVD noBG, 2.96 ms/event) and the slowest one (VXD fullBG, 2.33 s/event) is a factor of about 1000. This is better than the factor of two-hit-combinations, which is about 8000-10000 for the same example.

Another important fact can be retrieved when looking at the difference in time consumption for events in the range of 1000 hits and for events with ten times more (full PXD background). It increases from a manageable average of 80 ms per event for the 1000 hits per event to extreme 40600 ms per event for 10000 hits, In other words: if the number of hits is of the order of 1000, the speed of the **VXDTF** is within the constraints given by hardware. The current track finder cannot cope, however, with a number of hits of the order of 10 000, as is the case with full BG.

3.2.3 Efficiencies

Having showed the conditions the `VXDTF` is facing, the last question to be answered is the one about the efficiencies. Figures 3.5, 3.6, 3.7, 3.8, 3.9) are plots of the reconstruction efficiency versus the momentum of the reconstructed tracks.

Fig. 3.5 shows the efficiency for SVD-only reconstruction without background and Fig. 3.6 shows SVD-only reconstruction with full background added. Fig. 3.7 shows full VXD reconstruction without background; in Fig. 3.8 there is partial background (added, using 1.8 ms-ROFs for the PXD and full ROFs for the SVD). Finally, Fig. 3.9 shows the results of the full ROFs for SVD and PXD.

The efficiency is best for SVD-only track finding without BG, where it is about 87.3% for the shown momentum range (35 MeV/c — 3500 MeV/c). Adding background reduces the efficiency to about 83.9%, which is a noticeable but not severe effect. A hint of bad parameter optimization for the `VXDTF` can be seen when looking at Fig. 3.7. Although 6 layer (full VXD) tracking is much easier than 4 layer (SVD-only) tracking, the efficiency is worse for the VXD case than for SVD-only. This effect was not there in most of the earlier studies and is a typical issue of bad user settings. Since finding the correct choice for the tuning parameters has many degrees of freedom, this task has to be done by an automated test suite. At the moment, first steps are taken in implementing such a procedure, but it will remain an important topic for the coming months.

As indicated above, the efficiency for VXD tracking without BG is 78.2% and thus a bit lower than the value for SVD-only. If a noticeable fraction of the PXD-Background and the full SVD-background is added to the run, the efficiency stays on the same level with a value of 76%. This means that adding about 900 background hits to the PXD per event does not have severe effects on the result. Adding the full size of PXD background, however, results in a breakdown of the efficiency, as it drops to 39.3%. It is not clear yet whether the `VXDTF` will be able to handle this amount of background after its redesign. Nonetheless it will be part of the considerations for the redesign, since reasonable results seem “only one order of magnitude away”.

We also show the efficiency depending on the Θ angle. The corresponding figures are Figures 3.10, 3.11, 3.12, 3.13, 3.14 for SVD without background, SVD with background, VXD without background, VXD with 1/12th of the full PXD background and VXD with full background, respectively. The pattern is the same for most of the figures: flat efficiency except for a distinctive dent around 90°. Again the presumably easiest case of $\Theta \sim 90°$ is not as expected. But the effect is there since several months and could not be explained by bugs or other design flaws. Current studies concentrate on the choice of filters applied by the CA in the `VXDTF`, where a sensibility with respect to the Θ range is assumed.

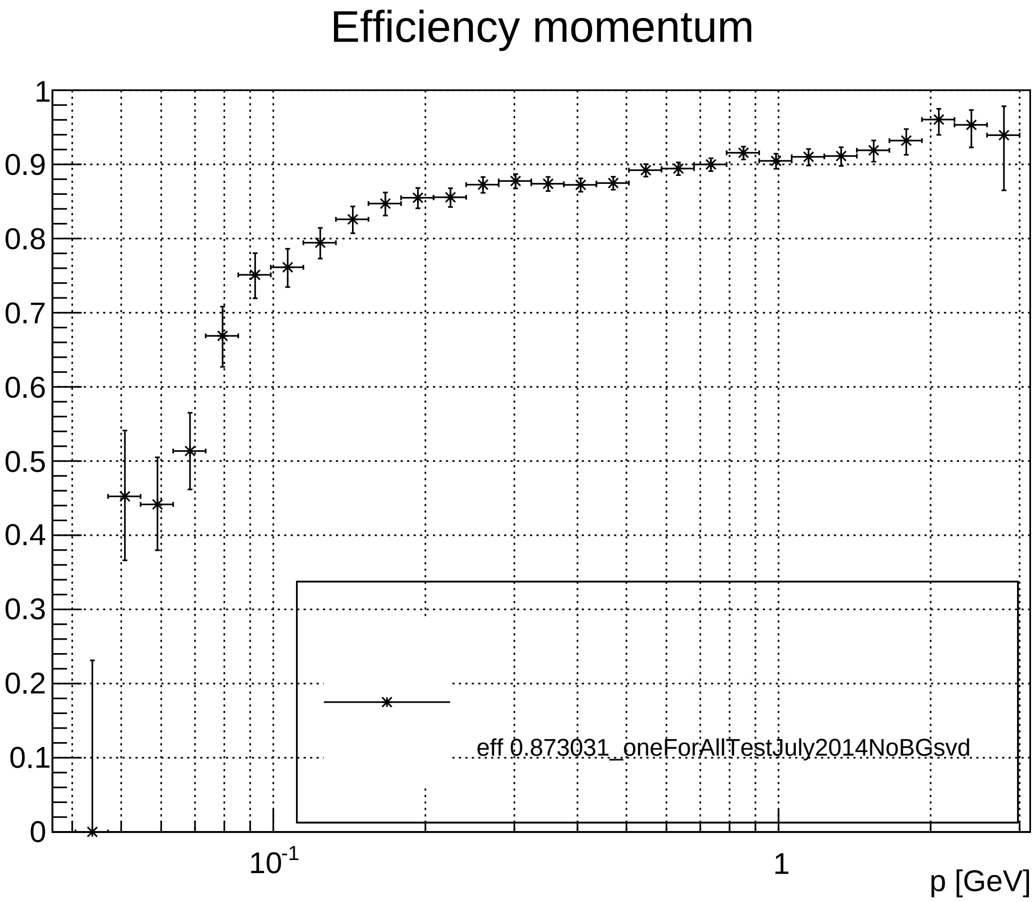

Efficiency momentum
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^-1
1
p [GeV]
eff 0.873031_oneForAllTestJuly2014NoBGsvd

Figure 3.5

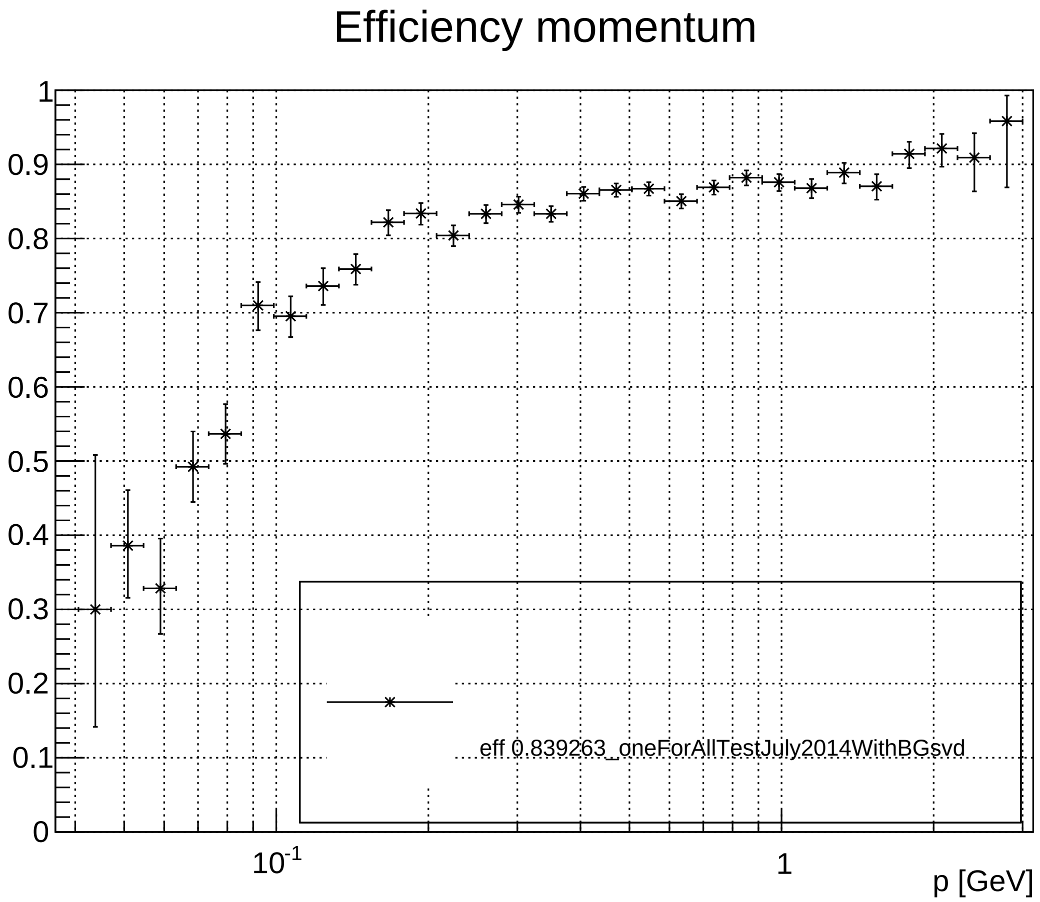

Efficiency momentum
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^{-1}
1
p [GeV]
eff 0.839263_oneForAllTestJuly2014WithBGsvd

Figure 3.6


Efficiency momentum
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^{-1}
1
p [GeV]
eff 0.781638_oneForAllTestJuly2014NoBGvxd

Figure 3.7

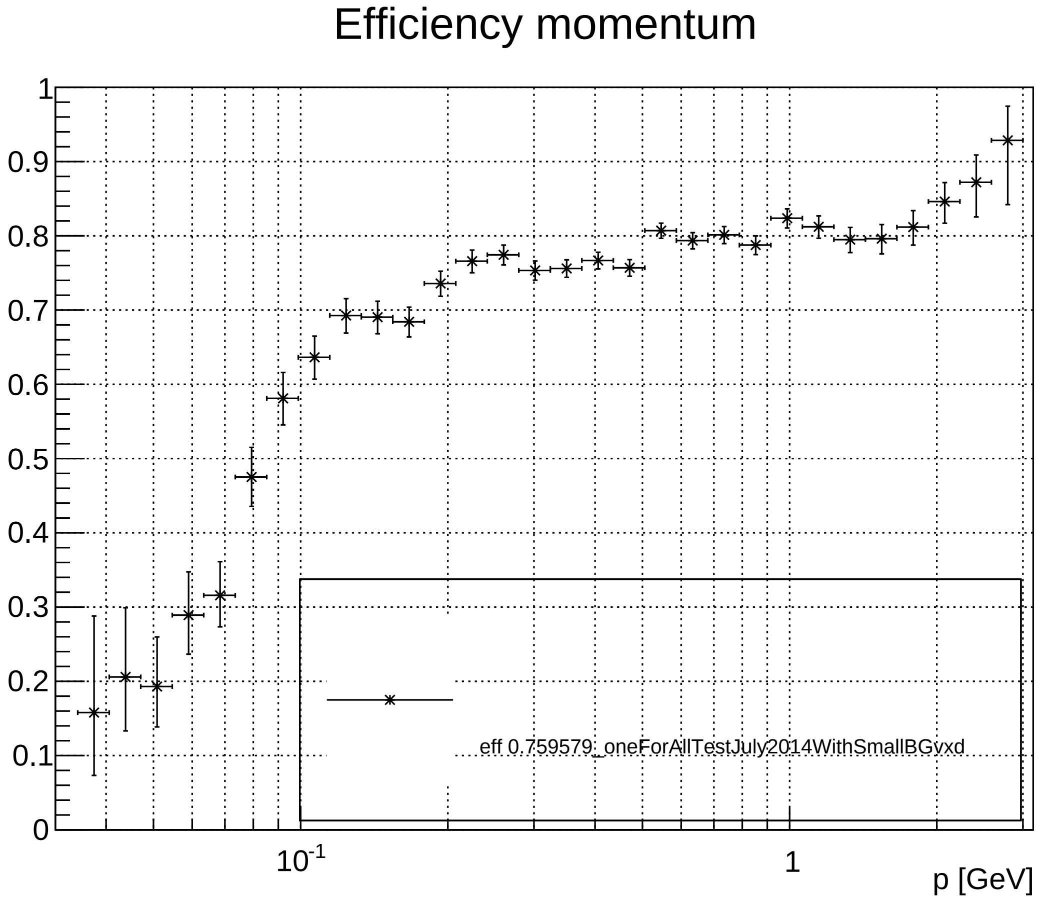

Efficiency momentum
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^{-1}
1
p [GeV]
eff 0.759579_oneForAllTestJuly2014WithSmallBGvxd

Figure 3.8

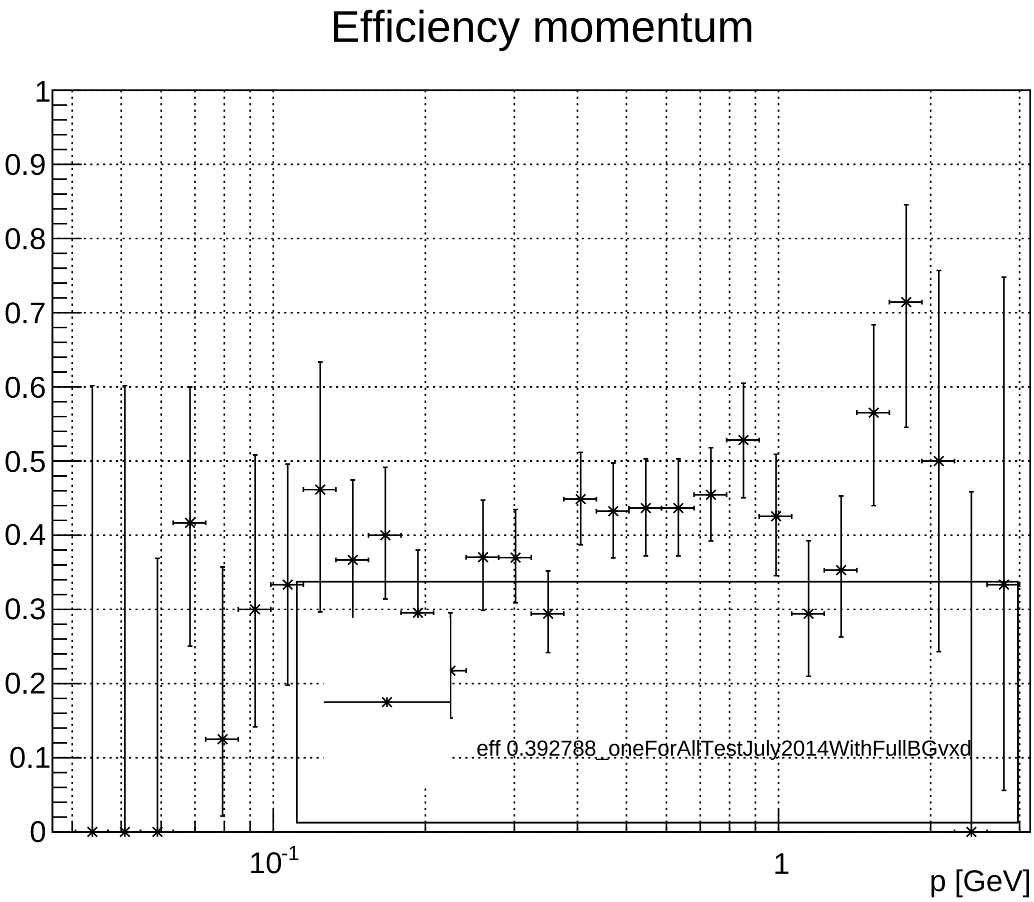

Efficiency momentum
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^{-1}
1
p [GeV]
eff 0.392788_oneForAllTestJuly2014WithFullBGvxd

Figure 3.9

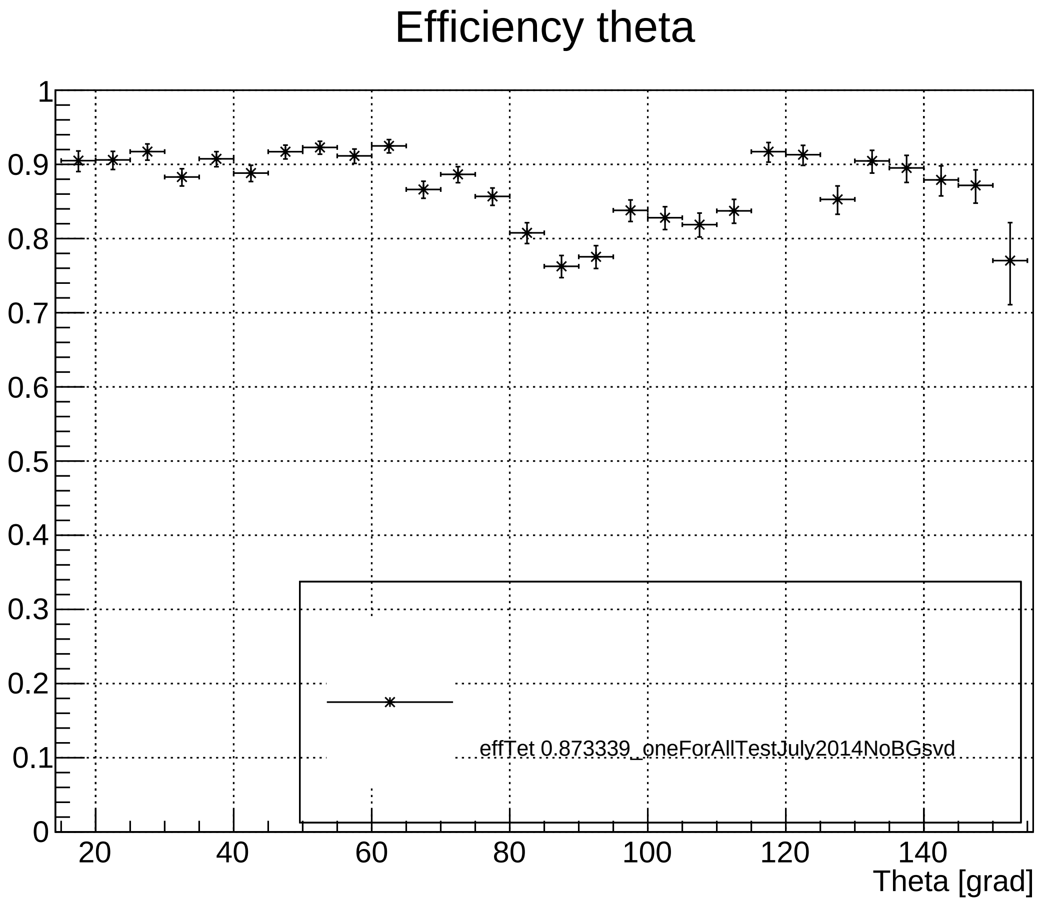

Efficiency theta
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
Theta [grad]
effTet 0.873339_oneForAllTestJuly2014NoBGsvd

Figure 3.10

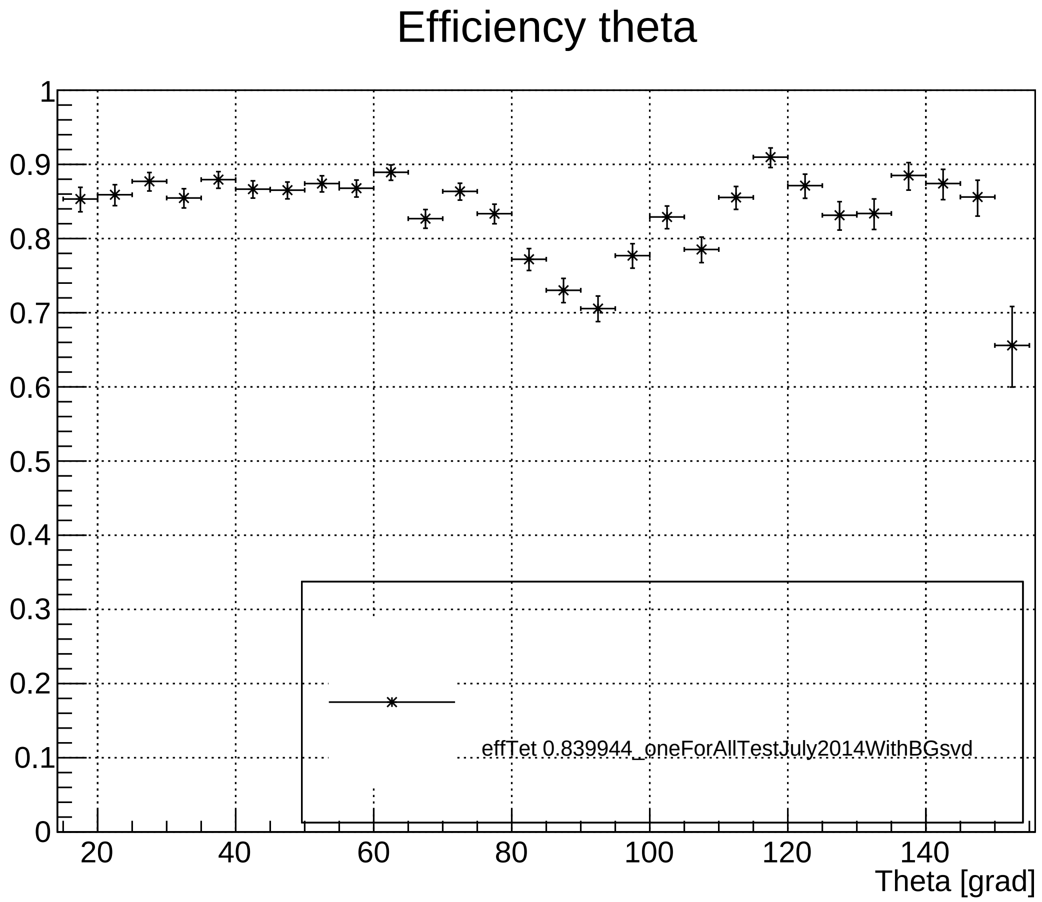

Efficiency theta
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
Theta [grad]
effTet 0.839944_oneForAllTestJuly2014WithBGsvd

Figure 3.11

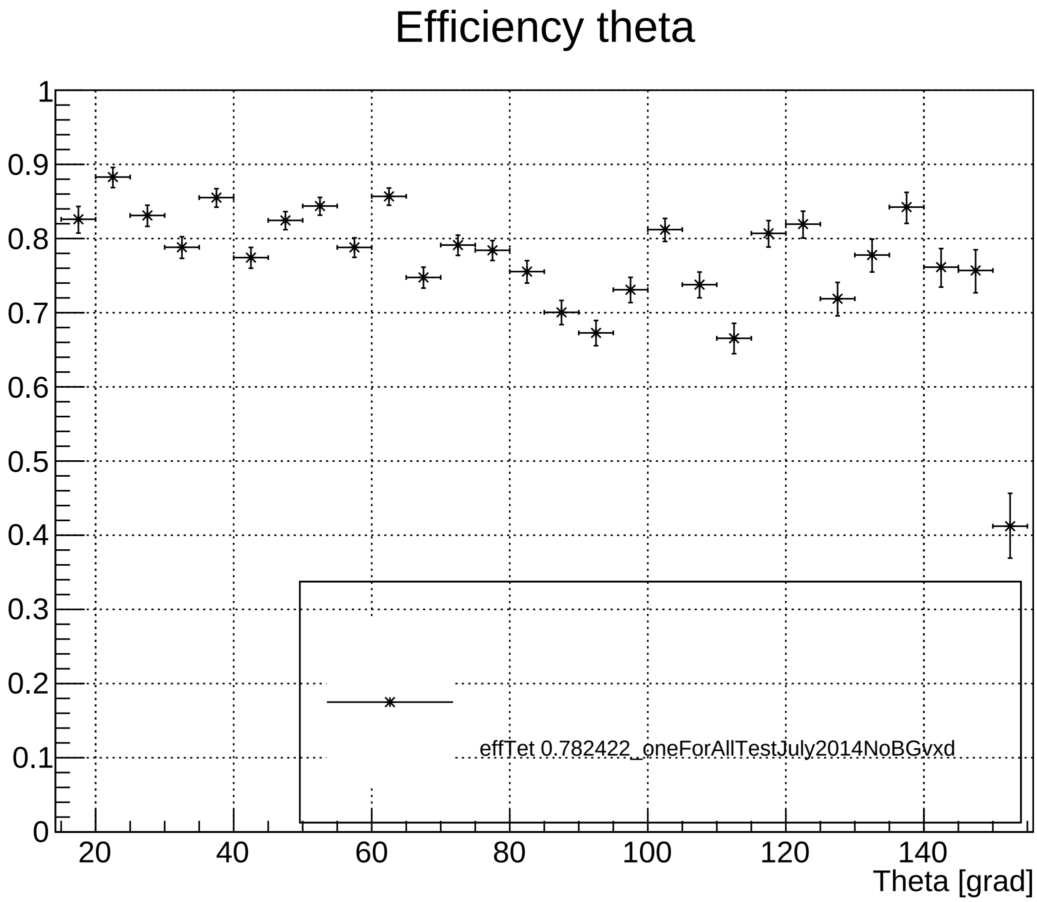

Efficiency theta
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
20
40
60
80
100
120
140
Theta [grad]
effTet 0.782422_oneForAllTestJuly2014NoBGvxd

Figure 3.12

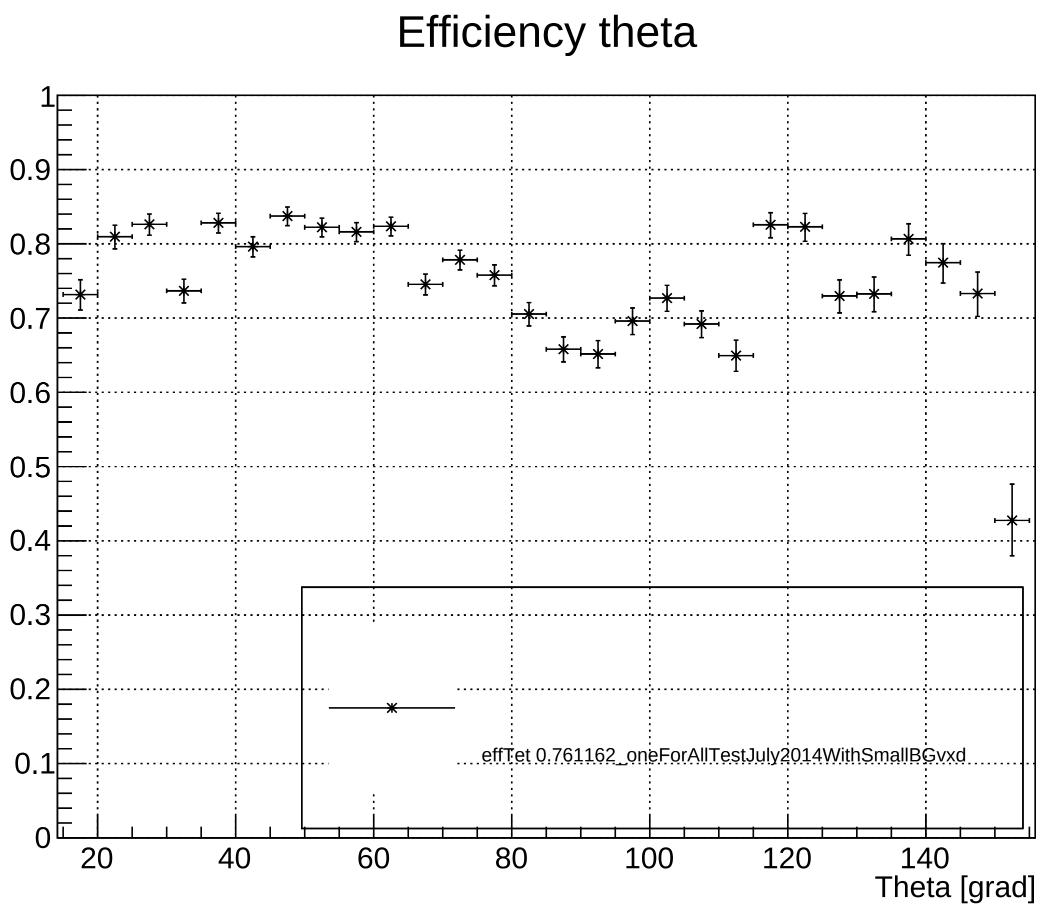

Efficiency theta
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
Theta [grad]
effTet 0.761162_oneForAllTestJuly2014WithSmallBGvxd

Figure 3.13

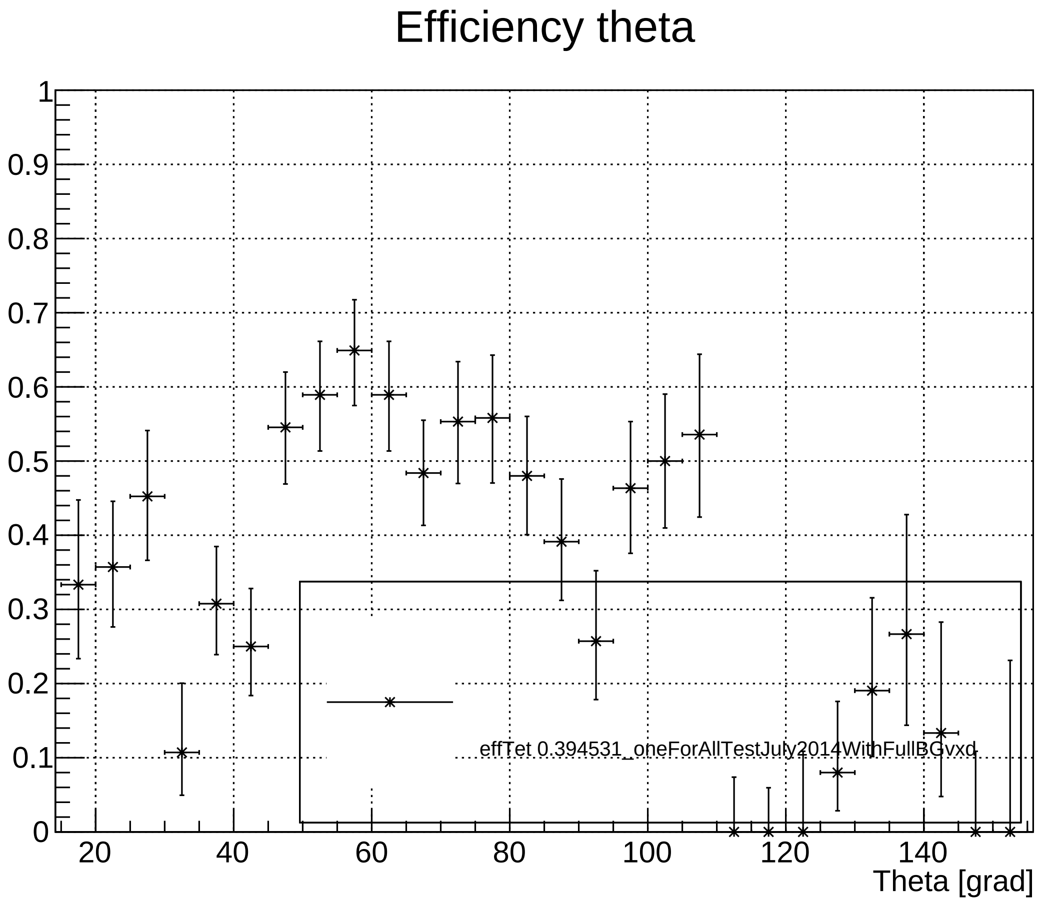

Efficiency theta
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140
Theta [grad]
effTet 0.394531_oneForAllTestJuly2014WithFullBGvxd

Figure 3.14

4 Outlook

goals for the redesign, missing features, rough time schedule for 2014

5 Appendix

5.1 Tutorial

how to use scripts, where to find (including revision) TODO: upload steering files and backgroundCampaign-directory!
complete steering files for basf2 and root