Overview of the Integration of the General Broken Lines Algorithm within EUTelescope

Beam Telescopes and Test Beams Workshop
Tutorial
2015

Alexander Morton

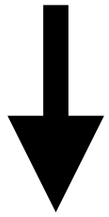
```
[amorton@ppepc181 noDUTExample]$ ls
config  geometry  noDUTIterativeAlignment.sh  output  patRecAndTrackFitUsingConfigurationFile.sh  runlist  steering
[amorton@ppepc181 noDUTExample]$
```

```
#We have a bash script for each example to store the setting to align. Note these will be different from what is stored in configuration file. Since the configuration file will stor
e the settings to fit tracks.
#!/bin/bash
while getopts n:i:r: option
do
        case "${option}"
        in
                n) numberOfIterations=${OPTARG};;
                i) identifier=${OPTARG};;
                r) RUN=${OPTARG};;
       esac
done

if [  -z "$numberOfIterations" ]; then
  echo "The number of iterations is empty in iterativeAlignment bash script!"
  exit
fi
#The location of the example and the scripts
export exampleLocation="/afs/phas.gla.ac.uk/user/a/amorton/ilcsoft/v01-17-05/Eutelescope/trunk/jobsub/examples/GBL/noDUTExample"
export scriptsLocation="/afs/phas.gla.ac.uk/user/a/amorton/ilcsoft/v01-17-05/Eutelescope/trunk/GBL/iterativeAlignmentScripts"

#VARIABLES WHICH VARY THROUGH ALIGNMENT ARE SET HERE. IMPORTANT NOT ALL VARIABLES ARE HERE LOOK IN CONFIG-ALIGNMENT.
export r="1"; #Make resolution large so we begin with small chi2 and factor improvement to get to chi2/ndf=1 on next iteration.
export dutXs="" #This is the resolution of the DUT in the x LOCAL direction taking into account the misalignment
export dutYs="" #This is the resolution of the DUT in the y LOCAL direction taking into account the misalignment
export minTracksPerEventAcceptance=0.001 #This is the number of tracks that is needed per event before we stop pattern recognition. Note value should depend on other cuts.
export ResidualsRMax="0.5" #This is the window size on the next plane that we will accept hits from. This will increase if less than 1 track per event is found.
export inputGearInitial="gear-1T.xml" #Note the gear file changes through the process so must be placed here.
export dutPlanes=""
export allPlanesFixed="0 3 5"


#Export input variables to other bash files.
export RUN="$RUN"
export numberOfIterations="$numberOfIterations"
export outputIdentifier="$identifier" #Use this string to identify final gear/histogram and all iterations before.


$scriptsLocation/initialiseAndRun.sh
```

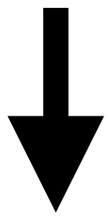
Must change for your location. Same in config file.

./noDUTIterativeAlignment.sh -n 1 -r 97 -i After-Alignment

```
[amorton@ppepc181 noDUTExample]$ ls
config  geometry  noDUTIterativeAlignment.sh  output  patRecAndTrackFitUsingConfigurationFile.sh  runlist  steering
[amorton@ppepc181 noDUTExample]$
```



```
#!/bin/bash
#This file does a quick pat->track using the gear specified in config.
while getopts r:h: option
do
        case "${option}"
        in
                r) RUN=${OPTARG};;
                h) histoNameInput=${OPTARG};;
        esac
done

echo "These are the input parameters to patRec and TrackFit."
echo "Run number $RUN"
echo "HistoNameInput $histoNameInput"

PatRec="patternRecognition"
DRY="--dry-run"
#THIS PART WILL RUN PATTERN RECOGNTION AND TRACKFITTING AGAIN WITH THE FINAL GEAR FILE.
$do jobsub.py -c config/config.cfg -csv runlist/runlist.csv $PatRec $RUN
TrackFit="GBLTrackFit"
#TrackFit="GBLTrackFit-Multi"
#Then we create the first tracks using pattern recognition tracks.
$do jobsub.py  -c config/config.cfg -csv runlist/runlist.csv -o histoName="$histoNameInput" $TrackFit  $RUN
```



./patRecAndTrackFitUsingConfigurationFile.sh -r 97 -h before-alignment

- Steps:
- 1st) Go to noDUTExample
- 2nd) Have a look in config file
  - Change what setting you see fit
- 3rd)Run 97 before alignment with misaligned gear
  - ./patRecAndTrackFitUsingConfigurationFile.sh -r 97 -h before-alignment
- 4th) Look at results.
  - Misaligned as expected
- 5th) Open alignment script and check parameters
- 6th) Run alignment script
  - ./noDUTIterativeAlignment.sh -n 1 -r 97 -i After-Alignment

Takes about 5 minutes in general. Can be made faster.

Link

Contains

EUTelTrack and EUTelState are new objects within EUTel

Track → **Chi2, number of hits ……**

States → **Kink angles, incidence angles……**

Hits → **Position, Covariance matrix……**

Cluster → **Charge, Cluster size……**

We want to easily access all information from a single set of objects

We want the format to be the same for as many setups as possible.

Makes sense to begin at the track level to connect all this information

Link

Contains

Track → Chi2, number of hits ……

States → Kink angles, incidence angles…….

Hits → Position, Covariance matrix…….

Cluster → Charge, Cluster size……

We can easily look at the scattering with position

Link

Contains

Track → Chi2, number of hits ……

States → Kink angles, incidence angles…….

Hits → Position, Covariance matrix…….

Cluster → Charge, Cluster size……

Cluster size with incidence angle

How this is done is shown in EUTelProcessorTrackAnalysis

Thank You!