
EVNDISP Manual

IACT event analysis and display

v4.00

Gernot Maier (DESY)

.....

SVN *Rev* : 361

SVN *Date* : 2011 – 09 – 29 16 : 25 : 12 + 0200(*Thu, 29 Sep 2011*)

SVN *Author* : *maierg*

Abstract

eventdisplay is a complete package for VERITAS and CTA analysis (whatever 'complete' means...) The package consists of several analysis steps and tools:

1. evndisp (calibrate and parametrize images, event reconstruction, stereo analysis)
2. mscw_energy (use lookup tables to produce msw, msl, and energies)
3. anasum (produce maps and calculate analysis results)
4. shared library tools and macros (produce the energy spectrum and integral fluxes, plot maps, plot sensitivities, etc.)
5. makeEffectiveArea (calculate effective areas)
6. trainTMVAforGammaHadronSeparation (tools to train MVA and optimize cuts)
7. ...

This is a very incomplete manual, started in September 2009. Please help updating it.

Contents

1	Documentation	1
2	Introduction	3
3	Installation and auxiliary data files	4
3.1	Environmental variables	4
3.2	Auxiliary data files	4
3.2.1	Detector description	4
4	eventdisplay - calibration, image analysis and stereo reconstruction	5
5	mscw_energy - using lookup tables	6
5.1	Energy reconstruction	6
6	Gamma/hadron separation	7
6.1	Cut parameters	7
7	makeEffectiveArea - instrument response functions	9
8	CTA analysis	11
8.1	General concept	11
8.2	Analysis steps for IRF and sensitivity calculation	11
8.2.1	Step 1: Converter	11
8.2.2	Step 2: Display (event-by-event)	12
8.2.3	Step 3: Calibration, image analysis and stereo reconstruction	13
8.2.4	Step 1 & 3 combined [USE]	13
8.2.5	Step 4: mscw_energy	13
8.2.6	Step 5: Optimize cuts or train MVA	13
8.2.7	Step 6: Effective Areas	13
9	Input data format	14
9.1	VBF - VERITAS Bank Format	14
9.2	DST - data summary tree	14
9.2.1	Limitations	14
10	Detector Setup	15
10.1	Telescope types	15

Chapter 1

Documentation

EVNDISP is work-in-progress and the documentation is not in the state it is supposed to be. Apart from the information in this manual, other sources for help are:

README files

INSTALL: information on installation the analysis package, dependencies, environmental variables

README.CTA : short description of a typical CTA analysis

README.VTS: description of a typical VERITAS analysis

AUTHORS: author description

Description and command line options for the different software parts:

README.EVNDISP:

README.MSCW_ENERGY:

README.ANASUM:

README.EFFECTIVEAREA:

README.ANALYSISLIBRARY:

README.SCRIPTS:

README.MACROS:

WIKI pages

The EVNDISP manual for VERITAS users:

http://veritas.sao.arizona.edu/wiki/index.php/Eventdisplay_Manual

Chapter 2

Introduction

Chapter 3

Installation and auxiliary data files

3.1 Environmental variables

3.2 Auxiliary data files

The auxiliary data files contain information needed for the analysis. This might be files describing the detector geometry, lookup tables, effective areas file, etc. We assume in the following that these files are located in the directory *\$OBS_EVNDISP_ANA_DIR* where *OBS* is your observatory (i.e. *VERITAS* or *CTA*).

3.2.1 Detector description

VERITAS

CTA

No detector description is needed, since the telescope and array description is read directly from the DST file (*telconfig* tree). The converter from hessio to EVNDISP DST format needs a subarray file, a simple list of telescopes to be selected from the corresponding hyper array. The subarray files for prod1 can be found in *\$OBS_EVNDISP_ANA_DIR/DetectorGeometry/*.lis*.

Chapter 4

**eventdisplay - calibration, image
analysis and stereo reconstruction**

Chapter 5

mscw_energy - using lookup tables

5.1 Energy reconstruction

Chapter 6

Gamma/hadron separation

6.1 Cut parameters

Option	Number of parameters	Allowed value(s)	Default value(s)	Description
cutselection	2			type of gamma/hadron and direction cut
	Parameter #1	?	0	gamma/hadron cut id
	Parameter #2	0-5	0	direction cut id: fixed Θ^2 cut (0, needs parameter <i>theta2cut</i>), energy dependent Θ^2 from a function read from a IRF file (1, needs parameter <i>theta2file</i>), from a IRF graph (2, needs parameter <i>theta2file</i> and option <i>IRF</i>), all other values: experimental (3-5, TMVA)
angres	1]0, 100]	0	containment probability for energy dependent Θ^2 cut (in %)

Table 6.1: *Parameter definition and range for gamma/hadron cut files. This is used for example in the effective area calculation or for data analysis.*

Table 6.2: *Gamma/hadron cut selector values. They consist of two digits: ID1+ID2*10*

gamma/hadron cut selector	Description
ID2	
0	apply gamma/hadron cuts on parameters in given data tree
1	apply gamma/hadron cuts on probabilities given by a friend to the data tree (e.g. random forest analysis)
2	same as 2
3	apply cuts on probabilities given by a friend to the data tree already at the level of the event quality level (e.g. of use for analysis of certain binary phases only)
4	TMVA gamma/hadron separation
ID1	
0	apply cuts on MSCW/MSCL (mean reduced scaled width/length)
1	apply cuts on mean width/length (no lookup tables necessary)
2	no cut applied (always passed)
3	apply cuts on MWR/MLR (mean scaled width/length)
Example:	
0	apply MSCW/MSCL cuts (default)
22	apply event probability cuts
10	apply cuts from a tree AND apply MSCW/MSCL cuts
40	use TMVA AND apply MSCW/MSCL cuts

Chapter 7

makeEffectiveArea - instrument response functions

Instrument response functions (IRF), i.e. effective areas and angular, core and energy resolution and bias curve can be calculated using *makeEffectiveArea*.

Option	Number of parameters	Allowed value(s)	Default value(s)	Description
FILLINGMODE	1	0,1,2,3	0	filling of IRFs: fill all IRFs (0), resolution plots only (1), angular resolution plot only (2), effective areas only (3)
ENERGYRECONSTRUCTIONMETHOD	1	0,1	0	energy reconstruction method (see 5.1)
ENERGYAXISBINS	1	> 0	60	number of bins on log ₁₀ energy axis
ENERGYRECONSTRUCTIONQUALITY	1	0,1	0	
AZIMUTHBINS	1	0,1	1	define azimuth bins and calculate IRFs in each azimuth bin. Bins are hardwired with a bin width of 22.5° (16 bins), bin 17 contains the full azimuth range
ISOTROPICARRIVALDIRECTIONS	1	0,1	0	input MC are simulated with random direction (wobble) offsets (use for gamma rays only)
TELESCOPETYPECUTS	1	0,1	0	apply telescope type dependent cuts CHECK! STILL USEFUL?
FILLMONTECARLOHISTOS	1	0,1	0	fill histograms with MC spectra only (no IRF calculation)

ENERGYSPECTRUMINDEX	3				reweight events to this set of spectral indexes
	#1	> 0	1		number of different spectral indexes
	#2	> 0.	2.0		lower value
	#3	> 0.	0.1		step size
CUTFILE	1				cut file (full path, see 6.1)
SIMULATIONFILE_DATA	1				simulation data file (mscw file)
SIMULATIONFILE_MCHISTO	1				data file with thrown events. This can be either a full mscw file (slow) or a file with the filled MC histograms

Table 7.1: *Parameters in the run parameter file for the IRF calculation.*

Chapter 8

CTA analysis

8.1 General concept

To use Eventdisplay for CTA analyses, the *simtel.gz* files have first to be converted into ROOT format (called in the following DST format). This must be done for each subarray separately. The following list shows the binaries and macros needed to produce CTA sensitivity files:

CTA.convert_hessio_to_VDST convert simtel.gz files into ROOT format

eventdisplay image cleaning & calculation of telescope parameters & reconstruction of the direction and core; display

mscw_energy train and use lookup tables to estimate the energy and mean scaled parameters

trainTMVAforGammaHadronSeparation optimize cuts or train MVA

makeEffectiveArea make effective areas

\$EVNDISPSYS/macros/sensitivity.C calculate and plot sensitivity curves

writeCTAWPPPhysSensitivityFiles write sensitivity files (WP-Phys style root file)

There are scripts to simplify these steps in the `$EVNDISP/scripts/CTA` directory. It is easy with their help to analyze many *simtel.gz* files for several subarrays, offsets, and so on. The scripts expect several environmental variables to be set (see 3.1). Most of the scripts work fine on the DESY batch system, they might need some adjustment for other computing environments.

8.2 Analysis steps for IRF and sensitivity calculation

8.2.1 Step 1: Converter

In this step a simulation data file in hessio format is converted to the EVNDISP DST format (see 9.2). These are the possible options to run the converter:

```
$EVNDISPSYS/bin/CTA.convert_hessio_to_VDST
```

```
c_DST: A program to convert hessio data to EVNDISP DST files (v.4.00)
```

```
Syntax: ./bin/CTA.convert_hessio_to_VDST [ options ] [ - | input_fname ... ]
```

```
Options:
```

```

-v          (More verbose output)
-q          (Much more quiet output)
-s          (Show data explained)
-S          (Show data explained, including raw data)
--history (-h) (Show contents of history data block)
-i          (Ignore unknown data block types)
--max-events n (Skip remaining data after so many triggered events.)
-a subarray file (list of telescopes to read with FOV.)
-o dst filename (name of dst output file)
-f on=1/off=0 (write FADC samples to DST file; default=0)
```

The following options are necessary: -a and -o. Note the limitations of the DST format (9.2.1).

Many of the CTA simulation files contain data for a so called hyper array. To select the actual array, the corresponding subarray has to be specified in a ASCII text file (called *subarray* file in the following). The following example file selects an array consisting of telescopes 63, 19, 67, and 33, each telescope with a field of view of 8 degrees:

```
63 8
19 8
67 8
33 8
```

Subarray files for most of the typical arrays used in the CTA sensitivities studies are part of the analysis file package (see 3.2). Note that a subarray file is always needed, even if all telescopes from the hessio file are read out and analyzed.

For a typical run, the following command line should be used:

```
./CTA.convert_hessio_to_VDST -a subArray.list -o dstfile.dst.root \
    gamma_run12241.simhess.gz
```

For FADC analysis, add the option *-f 1*. Note again the limitations of the DST format (9.2.1).

8.2.2 Step 2: Display (event-by-event)

It is always useful to look at events in the display. To do this, it is best to select a small subarray with the *-teltoana* option in *evndisp*. The display might otherwise be quite slow in responding to your input due to the large number of objects to be drawn.

A typical command line to look at events might be (remove the *-traceanalysis=1* command for dst files without trace information):

```
$EVNDISPSYS/bin/evndisp -display=1 \
-reconstructionparameter ./EVNDISP.reconstruction.runparameter \
-useFixedThresholds -imagethresh=10.0 -borderthresh=5.0 \
-l2setspecialchannels nofile \
-sourcefile tt.v2.root -traceanalysis=1
```

8.2.3 Step 3: Calibration, image analysis and stereo reconstruction

8.2.4 Step 1 & 3 combined [USE]

Run the converter and Eventdisplay for a specific subarray: use script `$EVNDISPSYS/scripts/CTA/CTA.EVNDISP.sub_convert_and_analyse_MC_VDST.sh`

NOTE: The image cleaning thresholds can be specified in the file `$OBS_EVNDISP_ANA_DIR/ParameterFiles` file either for all telescopes to the same values or for each telescope type separately.

8.2.5 Step 4: mscw_energy

If you use standard configurations maybe some lookup tables already exist (ask Gernot or Heike where you could find them).

If not, you have to create them yourself with

```
$EVNDISPSYS/scripts/CTA/CTA.MSCW_ENERGY.sub_make_tables.sh
```

If you have your lookup tables you have to run `mscw_energy` for estimating the energy of each event:

```
$EVNDISPSYS/scripts/CTA/CTA.MSCW_ENERGY.sub_analyse_MC.sh
```

8.2.6 Step 5: Optimize cuts or train MVA

8.2.7 Step 6: Effective Areas

To calculate effective areas, look at:

```
$EVNDISPSYS/scripts/CTA/CTA.EFFAREA.sub_analyse.sh
```

For the first time you have to use `mscw` files as input since `MCpars` has to be analysed once.

If you change your cuts afterwards (but not the number of input files) you can use the faster version by using the effective area file as input (helps a lot for protons).

Chapter 9

Input data format

9.1 VBF - VERITAS Bank Format

9.2 DST - data summary tree

The DST format is a simple ROOT tree containing standard C++ variables only (no class data).

9.2.1 Limitations

The implementation requires the hardwiring of the maximum number of telescopes, channels, etc. These values can be found in `inc/VGlobalRunParameter`, for example:

```
// HARDWIRED MAXIMUM NUMBER OF TELESCOPES AND CHANNELS, etc.
// maximum number of telescopes
#define VDST.MAXTELESCOPES 100
// maximum number of channels per telescopes
#define VDST.MAXCHANNELS 12000
// maximum number of summation windows
// (=maximum number of samples per FADC trace)
#define VDST.MAXSUMWINDOW 64
// maximum number of time slices for pedestal calculation
#define VDST.PEDTIMESLICES 5000
// maximum number of arrayreconstruction method
#define VDST.MAXRECMETHODS 100
// maximum number of timing levels
#define VDST.MAXTIMINGLEVELS 10
```

NOTE: These numbers determine the memory requirements of `evndisp` and `CTA.convert_hessio_to_VDST`.

NOTE: `evndisp` must be compiled with the same settings as the writing program.

Chapter 10

Detector Setup

10.1 Telescope types

Different telescope types (e.g. mid-size and small-size telescopes, telescopes with different FOV, etc) are assigned a telescope type number in the code, this number is as well written to the data trees. The telescope type contains the mirror shape (DC, Parabolic, SC), the mirror area (m^2), the field of view ([deg]) and the pixel size ([deg]). For VERITAS, the telescope type correspond simply to the different telescope numbers (and are therefore 0,1,2,3).

For clarification, this is the corresponding code bit from `src/CTA.convert_hessio_to_VDST.cpp`:

```
fTelescope_type = TMath::Nint(pix_size*100.);
fTelescope_type += TMath::Nint(fFOV*10.)*100;
fTelescope_type += TMath::Nint(fMirrorArea)*100*10*100;
// all large telescopes are parabolic, all others are Davies-Cotton (hardwired)
if( fMirrorArea > fParabolic_mirrorArea ) fTelescope_type += 100000000;
// Schwarzschild-Couder: check number of mirrors
else if( fNMirrors == fSC_number_of_mirrors ) fTelescope_type += 200000000;
```

Note: There is currently no way to determine the mirror/telescope shape (parabolic, Davies-Cotton, etc) from the hessio file. This is why the mirror area and the number of mirrors is used. The parabolic shape is assigned to all telescopes with a mirror area $> 400 \text{ m}^2$. Schwarzschild-Couder Design are all telescopes with 2 mirrors only.