Chandra X-ray Center

AHELP for CIAO 3.4 **dmimghist** Context: tools

Jump to: Description Examples Parameters CHANGES IN CIAO 3.2 CHANGES IN CIAO 3.0 Bugs See Also

Synopsis

Create a histogram of the pixel values in a 2–D image

Syntax

```
dmimghist  infile outfile hist [strict] [clobber] [verbose]
```

Description

'dmimghist' creates a histogram of the pixel value distribution in a two–dimensional image and writes it out as a FITS binary table. The table contains columns for bin number, bin minimum, bin maximum, counts in each bin, and the normalized counts in each bin.

The tool accepts an input file, and output file, and a binning specification. The binning specification is of the form

$$[a]:[b][:c] \mid c$$

where a is the low bin, b is the high bin, and c is the bin size. If strict=no, values lower than the lowest specified bin are assigned to the low bin and values greater than the highest specified bin are assigned to the high bin. If strict=yes, out–of–range values are ignored. If the low bin or high bin (or both) is unspecified, it will be automatically determined by scanning the file for the minimum and/or maximum values present.

If the input is a table (e.g. event file), dmimghist will open it as an image, creating on–the–fly the default image for that table (usually the full resolution sky image).

Use of stacks

The input, output, and binning parameters can accept a stack. An equal number of input and output filenames must be specified. Either a single binning specification or a stack of specifications (equal to the number of inputs) may be supplied.

Format of output table

The output table has five columns called BIN, BIN_LOW, BIN_HIGH, COUNTS and NORMALIZED_COUNTS. COUNTS represents the number of pixels in the input image which have pixel values in the range covered by BIN_LOW and BIN_HIGH. BIN_LOW denotes the lower end of the range of pixel values in a BIN. BIN_HIGH indicates the (non−inclusive) upper limit of pixel values in a BIN. The NORMALIZED_COUNTS column is calculated by dividing the COUNTS column by the sum of the COUNTS column.

Example 1

```
dmimghist acis.fits hist.fits 1:100:2
```

Make a histogram with 50 bins of size 2, starting at 1.

BIN will have the values 1,3,5... in succeeding rows, denoting ranges 1 to 2 (first row), 3 to 4 (second row), 5 to 6 (third row).

COUNTS(row) gives the number of pixels in the input image which have a value equal to or more than BIN_LOW(row) but less than BIN_HIGH(row). However, the end bins (first and last row of the table) include counts beyond the ends − if strict=no − so the first row actually contains the number of pixels with any pixel value less than 3.

Example 2

```
dmimghist acis.fits hist.fits 1:100:2 strict=yes
```

As in the previous example, except that the end bins no longer contain the counts for pixels below (for the minimum bin) or above (for the maximum bin) their range. This also affects the values of the NORMALIZED_COUNTS column, since the sum of the COUNTS column is used for the normalization.

Example 3

```
dmimghist acis.fits hist.fits :100:2
```

Make a histogram with bins size 2, maximum bin 100, and minimum bin equal to the smallest value in the file.

Example 4

```
dmimghist @in.lis @out.lis @bins.lis
```

Bin each of the files in in.lis according to the corresponding binning spec in bins.lis, and create the files named in out.lis. There must be the same number of entried in in.lis, out.lis, and bins.lis.

So, if the contents of the files are

```
unix% cat in.lis
img1.fits
img2.fits
unix% cat out.lis
hist1.fits
hist2.fits
unix% cat bins.lis
1::0.5
1:200:10
```

then this is equivalent to running dmimghist on img1.fits to create hist1.fits using a binning specification of 1::0.5, and on img2.fits to create img2.fits using a binning specification of 1:200:10.

Example 5

```
dmimghist @in.lis @out.lis 1::5
```

This is similar to the previous example except that the same binning specification – here "1::5" – is used on each file in in.lis to create the files listed in out.lis. There must be the same number of file in in.lis and out.lis.

Parameters

name	type	ftype	def	min	max	reqd	stacks
infile	file	input				yes	yes
outfile	file	output				yes	yes
hist	string					yes	yes
strict	boolean		no				
clobber	boolean		no				
verbose	integer		0	0	5		

Detailed Parameter Descriptions

Parameter=infile `(file required filetype=input stacks=yes)`

The input virtual file specification.

If the input is a table, it will be opened as an image, using the default image columns for the table. For instance, if an event list is given, the image will be a full–resolution sky image (unless explicitly binned).

The image must have two dimensions.

Parameter=outfile `(file required filetype=output stacks=yes)`

The output file(s).

Example 5 3

Parameter=hist `(string required stacks=yes)`

The binning specification.

The binning specification is of the form

```
[a]:[b][:c] | c
```

where a is the low bin, b is the high bin, and c is the bin size. The behavior for the minimum and maximum bins is controlled by the strict parameter.

Parameter=strict `(boolean default=no)`

Strict binning selection. "Yes" means values below the lowest bin and above the greatest bin are ignored when calculating the COUNTS and NORMALIZED_COUNTS columns.

Parameter=clobber `(boolean default=no)`

If outfile already exists, clobber=yes will allow you to overwrite it.

Parameter=verbose `(integer default=0 min=0 max=5)`

Verbose can be from 0 to 5, generating different amounts of output.

CHANGES IN CIAO 3.2

Null Value and Region Exclusion Checking

`dmimghist' now supports null value checking and region exclusion checking. Pixels with the NULL value (NaN for floating–point images or the BLANK keyword for integer images) are excluded from the calculation. Those pixels which are located outside the region filter in the data subspace of the image are also excluded from the calculation. This means that dmimghist, when used in a manner similar to

```
unix% dmcopy evt2.fits"[sky=circle(4100,4200,100)][bin sky]" src.img
unix% dmimghist src.img src.hist 0:10:1
```

will ignore those pixels in the image src.img that lie outside the region used in the filter applied by dmcopy.

BIN_LOW and BIN_HIGH columns

The output file now contains BIN_LOW and BIN_HIGH columns which list the domain of each row.

CHANGES IN CIAO 3.0

Exit Status

The exit status of the tool is now set to 0 on success and non–zero on error (prior to this release the exit status was not set to 0 on successful completion).

Bugs

See the bugs page for this tool on the CIAO website for an up–to–date listing of known bugs.

See Also

dm

dmbinning, dmimages, dmimfiltering

tools

dmappend, dmcontour, dmcopy, dmextract, dmfilth, dmgroup, dmimg2jpg, dmimgcalc, dmimgpick, dmimgthresh, dmregrid, fullgarf, get_sky_limits, mkacisrmf, mkarf, mkexpmap, mkgarf, mkgrmf, mkinstmap, mkpsf, mkrmf, mkwarf, tgextract

---

Bugs