URL: http://cxc.harvard.edu/ciao3.4/dmimgcalc.html
Last modified: December 2006

AHELP for CIAO 3.4

dmimgcalc

Context: tools

Synopsis

Perform arithmetic on images

Syntax

```
dmimgcalc  infile infile2 outfile operation [weight] [weight2]
[lookupTab] [clobber] [verbose]
```

Description

`dmimgcalc' is used to combine multiple images using a wide range of mathematical operations. Rather than being restricted to simple operations, (almost) arbitrarily complex expressions can be used. The tool can also be used to compare two images on a pixel−by−pixel basis. The input images must have the same sizes and are combined using logical coordinates (see "ahelp dmimages" for information on image coordinate systems).

Simple combinations of two images

Using the operation, weight, and weight2 parameters, one or two images can be combined − using addition, subtraction, multiplication, or division − with optional weights. The output image is defined as

```
image1*weight <operation> image2*weight2
```

when <operation> is one of "add", "sub", "mul", or "div". The data type of the output image will be set to double if weighting is used (either weight or weight2 is not equal to 1), one integer image is dividing another integer image, or the two images have different data types, otherwise the data type will match that of the input images.

Comparing two images

If operation is set to "tst" then the two images are checked for equality − i.e. that matching pixels in the two images are the same − and the exit status value set accordingly (0 if equal, 1 if not). Note that in this case there is no output image so outfile can be set to "none".

Multiplying an image by a constant

If infile2 is set to "none", there is only one file in the infile parameter (so it is not a stack), and the operation parameter is set to one of "add", "sub", "mul", or "div" then the output image will the input image multiplied

by the value of the weight parameter. This functionality is superceeded by the ability to use complex mathematical expressions for the operation parameter, as discussed below.

General arithmetic combinations of many images

The previous syntax is adequate for simple image combinations. More complex operations − such as combining more than two images, converting the data type of an image, or applying mathematical functions to images − can also be specified. In this case the list of input images is given as a stack to the infile parameter (see "ahelp stack") and the individual images are referenced by the labels "img<n>", where "img1" refers to the first image and "imgn" the nth image in the stack.

In this mode the operation parameter is set to

```
imgout=...
```

where the allowed syntax for everything to the right of "imgout=" is described in "ahelp syntax". As an example, setting

```
imgout=(sqrt(img1)+sqrt(img2))/img3
```

will set the output image to be the sum of the square roots of the first two images divided by the third image. Further examples are given in the Examples section below.

In this mode the weight and weight2 parameters are ignored.

Image alignment

The tool will operate on corresponding pixels from the two images regardless of any (conflicting) WCS information in the header. Thus the user must ensure that the two images must be aligned in logical array coordinates prior to using the tool. If infile has WCS information, that information will be propagated to the output file. If the WCS in infile does not match the WCS in infile2, a warning will be printed.

Example 1

```
dmimgcalc acis.fits exposure.fits normalized.fits div
```

The file normalized.fits is set to the results of dividing acis.fits by exposure.fits.

Example 2

```
dmimgcalc acis1.fits acis2.fits output.fits add
```

Add images acis1.fits and acis2.fits and store in output.fits.

Example 3

```
dmimgcalc acis1.fits acis2.fits output.fits add weight=2.5 weight2=3
```

Add acis1.fits*2.5 to acis2.fits*3 and store in output.fits.

Example 4

```
dmimgcalc acis1.fits none output.fits add weight=3.2
```

Multiply acis1 by 3.2 and store in output.fits. Note that the value for the operation parameter is ignored (as long as it is set to a legal value).

dmimgcalc can also be used for more complicated image combinations, as shown in the following examples.

Example 5

```
dmimgcalc a.fits none sa.fits op="imgout=sqrt(fabs(img1))"
```

Here we use the ability to specify aribtrary expressions (see "ahelp syntax" for further details on the allowed format) to create sa.fits which contains the square root of the absolute pixel values in the input image (a.fits).

Since the value of the operation parameter begins with 'imgout=' we have to include the parameter name − otherwise the tool would think you were setting the parameter imgout to a value which would result in an error.

Example 6

```
dmimgcalc img.fits none dimg.fits op="imgout=(double)img1"
```

Here we use dmimgcalc to convert the input image (img.fits) into double format (dimg.fits).

Example 7

```
dmimgcalc a.fits,b.fits,c.fits none sum.fits op="imgout=img1+img2+img3"
```

Here we use dmimgcalc to add the three images together to form sum.fits.

Example 8

```
dmimgcalc "a.fits,b.fits" none combined.fits
op="imgout=(((img1*img1_exposure)+(img2*img2_exposure))/(img1_exposure+i
mg2_exposure))"
```

Here we combine a stack of input files (a.fits and b.fits) based on the expression given to the operation parameter to create combined.fits. The expression used is the weighted sum of two images, using the exposure times of the two images as the weights.

In this mode, the stack of files input as the infile parameter (so a.fits and b.fits in the example) are referred to be "img<n>" in the expression (<n> starts from 1, so "img1" refers to a.fits in the example). The output image is referenced by "imgout", and the "img1_exposure" terms are replaced by the value of the "EXPOSURE" keyword from the header of the images (in this case a.fits). So, if the EXPOSURE values of a.fits and b.fits were 12012.45 and 24034.9 respectively, the output image (combined.fits) would be equal to

```
( (a.fits*12012.45)+(b.fits*24034.9) ) / (12012.45+24034.9)
```

Example 9

```
dmimgcalc acis1.fits acis2.fits none tst
```

If the two images (here acis1.fits and acis2.fits) are equal − in that corresponding pixels have the same value in the two images − then the status/exit value is set to 0, otherwise it is set to 1. The way to check this value depends on what shell you are using; two common ways are:

echo $status	**for csh/tcsh users**
echo $?	for bash/sh users

Parameters

name	type	ftype	def	min	max	reqd	stacks
infile	file	input				yes	yes
infile2	file	input				yes	
outfile	file	output				yes	
operation	string					yes	
weight	real		1				
weight2	real		1				
lookupTab	file	input					
clobber	boolean		no				
verbose	integer		0	0	5		

Detailed Parameter Descriptions

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

The input file (or files)

If a stack of files is used then the operation parameter should be of the form "imgout=...".

Parameter=infile2 (file required filetype=input)

The second input virtual file specification, or "none". If the former, image must be same size as infile.

Parameter=outfile (file required filetype=output)

The output file name

Parameter=operation (string required)

Arithmetic operation to be performed (add/sub/mul/div/tst/expression).

This parameter can be one of the following:

Valid arithmetic operations

Value	Description
add	add images
sub	subtract infile2 image from infile image
mul	multiply images
div	divide infile image by infile2 image
tst	if each pixel in infile2 equals the same pixel in infile then set the status flag to 0, otherwise set it to 1. The outfile parameter can be set to "none".
imgout=.....	Used to specify an arbitrary expression combining a number of files. See "ahelp syntax" for the syntax that can be used here. The infile2 parameter should be set to "none" when using this form and the values of the weight and weight2 parameters are ignored.

Parameter=weight `(real default=1)`

Value by which to multiply infile before operation

Parameter=weight2 `(real default=1)`

Value by which to multiply infile2 before operation

Parameter=lookupTab `(file filetype=input)`

Lookup table for header merging

The look−up table defines the rules used to combine header keywords from multiple files, as described in "ahelp merging_rules". Set to "none" to stop the header files from being merged (so the output file will have a copy of the first input file).

Parameter=clobber `(boolean default=no)`

Overwrite existing file with same output filename?

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

Determines the level of descriptive output as the tool runs.

Set to "0" for none, up to "5" for maximum description.

CHANGES IN CIAO 3.0

Combining images using complex mathematical expressions

The operation parameter can now accept complex expressions for defining how images should be combined. See "ahelp syntax" and the examples above. Note that when using this mode the number of input images is not restricted to 1 or 2.

Header merges are now optional

If the lookupTab parameter is set to NONE or "" then no header merge (other than for WCS information) occurs.

HANDLING INVALID DATA

If a division by zero occurs then the output value is set to 0. Attempts to perform any operation on a pixel value of NaN are also set to 0. If these changes occur and the operation was a "simple" one − namely "add", "sub", "mul", or "div" − then a summary message will be displayed at the end of the program.

HEADER KEYWORDS

When images are combined the header keywords from the different files to be merged together following the rules given in the file pointed to in the lookupTab parameter. The rules for the merging − and the syntax for this file − are given in "ahelp merging_rules".

One common occurrence − when using the default set of rules − is for the DETNAM keyword to be changed to the value "Merged", which may cause issues when processing the resulting image with other tools. This tends to happen when combining an observation with an exposure map since the data will likely have a DETNAM field that is set to the value from the event file − e.g. "ACIS−012367" − whilst the exposure map just has the individual chipe name − e.g. "ACIS−7". Since the two values are different the output is set to "Merged". You can set the lookupTab parameter to "NONE" to avoid this merging − in which case the header is taken from the first input file, use a modified copy of the table used to define the merging rules, or use dmhedit to manually set the header keyword to the desired value.

Bugs

See the bugs page for this tool and for the mathematical syntax support on the CIAO website for an up−to−date listing of known bugs.

See Also

concept

merging_rules

dm

dmimages, dmimfiltering

tools

dmappend, dmcontour, dmfilth, dmgti, dmimg2jpg, dmimghist, dmimgpick, dmimgthresh, dmmerge, dmregrid, dmtcalc, get_sky_limits, mtl_build_gti, syntax

---

URL: http://cxc.harvard.edu/ciao3.4/dmimgcalc.html
Last modified: December 2006