NAME

imgBathy – create a bathymetry SDS

SYNOPSIS

imgBathy inFile [ouFile]

DESCRIPTION

This program is used to create a bathymetery SDS for the given input file. The resulting bathymetry will be written as a 2-D float32 array. It will be named ‘‘bathymetery’’. Normally, if the user calls this program with a single filename, then the ‘‘bathymetry’’ array will be appended to that file. The user may also select write the ‘‘bathymetry’’ SDS to a second file, using the first one for navigation only.

By default, the input bathymetery file is `$AUTO_DATA/BATHY.DAT` file.

OPTIONS

−B Bathymetery File
: This option is used to specify the input landmask file.

−n name
: This option is used to specify another name for the output SDS. The default is ‘‘bathymetery’’

--help Print out a small help guide.

--version
: Print out version of software and quit.

ENVIRONMENT VARIABLES

AUTO_DATA
: The directory where all the data files exist. Defaults to $AUTO_DIR/data.

FILES

BATHY.DAT
: The bathymetery file. It is the ETOP5 5-minute gridded product created by NGDC. Each value is in whole meters.

EXAMPLES

In this example, the ‘‘bathymetry’’ is appended to the given level-3 data file.

```
$ imgBathy S2000065175121.N3_HNAV_MSB
```

Now, suppose that instead we want to put the ‘‘bathymetry’’ in a second file called GOM_BATHY.hdf and we want to call the array ‘‘depth’’ instead of ‘‘bathymetry’’.

```
$ imgBathy -n depth S2000065175121.N3_HNAV_GOM GOM_BATHY.hdf
```

SEE ALSO

aps(1) **bathymetery**(5)

NAME

imgBrowse – create a quick-look image

SYNOPSIS

imgBrowse inFile sdsName ouFile

DESCRIPTION

This program creates a reduced image of the given product in a standard graphic format. The exact formats available depend on the software build process as it uses external libraries. Use the `--help` option to see which formats are known.

See the EXAMPLES section below for many different calling ideas. Also see the CAVEATS section below for current problems.

OPTIONS

-1 Build the image at 1-to-1 resolution. A built-in limit of 3072x3072 cannot be exceeded.

a fmt Use the given format for the colorbar labels. Use a C-printf style format.

A fontpath
Use the given True Type font for text.

b filename
Reads in the given blotch file and overlays in on the image.

-B [isp=isp,iep=iep,isl=isl,iel=iel,irp=irp,irl=irl]
Define a subsection of the original image for output. Cannot be used with the -s option.

isp is the starting sample number iep is the ending sample number isl is the starting line number iel is the ending line number irp is the replication factor along the samples dimension irl is the replication factor along the lines dimension

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-c file Use the following colortable file. If not set, then a black and white linear colortable will be use. If a file name is given and it contains ".lut", it is assumed to be a SeaDAS ASCII colortable. If the name contains ".ct", then it is assumed to be an NSIPS direct (binary) colortable.

-C num Use the internal (compiled in) color table. The available palettes are: 0 - Sherwin's, 1 - BioSphere, 2 - NDVI, 3 - Blue to Red, 4 - Negative Radiance, 5 - Goddard, 6 - Visibility, 7 - SST, 8 - Johns Hopkins Univ. APL SST, 9 - Hue 2, 10 - Rainbow, 11 - Purple Red Strips.

-D date Use given string for the date field

-E Add text indicating this product is "EXPERIMENTAL". Additionally if there is a file named experimental.png in $AUTO_DATA, it will use that file as the watermark(-w overrides this behavior).

-f [linear|log|log10]
Select Function used for scaling. Defaults to linear.

-F name
Name of the SDS containing the flags. Defaults to "l2_flags".

-g Prevents the drawing of gridlines on the image.

-i Create the image only. In this case the border, the color table, and annotations, etc. will not be drawn. Logo and watermark will still be drawn though.

-k Defines the location (if any) of the classification of the image. May be set to 0 (no classification) or 1 (classification written to top and bottom on image).

-K options
This option is used to create a colorbar consisting of breakpoints. That is data between breakpoints will be all set to the same color. Each break point is separated by semicolons (;) and the color is set after the break point with a colon (:). Each component of the color (red,green,blue) are separated by commas (,). For example,

-K 0:0,0,0;20:255,0,0;40:0,255,0 -r 0,40 ifile sst sst.jpg

will create an image of sst such that all pixels less than 0 will be black. Those from 0 to 20 will be red, those from 20 to 40 will be green.

-l Prevents the drawing of the landmask.

-L file Logo file. This should be an image approximately 48x48 in JPEG, PNG, or raw PPM format.

-m mapFile:mapName
Use the following mapFile and mapName.

-M NAME1;NAME2;NAME3
Name of masks to use. Multiple mask names must be separated by a semi-colon ";". A color may be assigned for each mask by following the mask name with a colon ":" and an RGB triplet separated by commas. For example,

-M CLDICE;ATMFAIL

or

-M CLDICE:220,220,220;ATMFAIL:127,127,127

-o type Sets the output file type. Can be "jpg", "png", "tiff", "rgb", "pnm".

-O x Sets the offset in the log10 function.

-p file Place points or symbols from the file on the image.

-P Get political boundaries from $AUTO_DATA/polbnd.dat in Piskor's format and overlay them on the image.

-r min,max
Select range of input data for scaling. Defaults to min/max of each SDS's validRange parameters. Failing that it uses the min/max of SDS. Failing that it is set to min/max of all reals.

-R min,max
Range of output data. Defaults to 0, 255 for byte scaled imagery.

-s w,h Sets the size of output image. Used to reduce or enlarge the size of the output. Defaults to size of original image. Cannot be used with the -B option.

-S slope,intp
Set the slope and intercept to use for

-t track Overlay a ship track

-T type Type of sensor, for example, 'SeaWiFS', 'AVHRR', etc.

-u units The units for the given product.

-U Replaces filename with the text "UNCLASSIFIED".

-v Verbose output.

-W file Defines the input coastline file. Defaults to `$AUTO_DATA/world_01.dat`.

-w file Use file as watermark image. The image will be scaled to fit the product image and overlayed on top of it. This should be an image created with transparency so that when it is overlayed it will give the illusion of a watermark(recommend creating a white image with alpha channel set to 20-30% opacity).

-x Prevents the drawing of masks on the image.

-X Prevents the drawing of the coastline on the image.

-y Overlay true color image based on mask(-z option). If used more than once (as in -y -y) will overlay true color image even for pixels with the invalid value. This is useful so that a static true color image can be created and used for the landmask or so that the land that falls outside of the image coverage will be drawn black instead of given the land mask color(currently brown).

-Y name
Name of SDS containing true color image. (default: true_color) The SDS may be in another file. In that case this option is used like -Y filename:SDSname.

-z num Set mask for true color overlay. (-y must also be used). default: 514 = (512 + 2 = CLDICE + LAND) meaning that pixels flagged as LAND or CLDICE will be replaced with the value in the

true color image.

--help Print out a small help guide.

--version
Print out version of software and quit.

FILES

`$AUTO_DATA/logo.jpg`
This is the default logo file which contains the default logo to apply to the image.

symbols.dat
This file allows the user to define and apply symbols to the image. The format is white space delimetered UNIX text file in columnar format. The first two columns are the latitude and longitude in decimal degrees with negative being South or West. Next column contains the symbol type, 0 for filled box, 1 for outline of box. The next three columns describe the color as an RGB triplet. Starting at text column 32 the rest of the line is used as a text string which will be printed next to the symbol. A '#' as the first character of a line indicates a comment and the line will be skipped.

track.dat
The track file allows the user to define and apply ship track or other transect over the image. The file format is a white space delimetered UNIX text file in columnar format. The only two columns are the latitude and longitude in decimal degrees with negative being South or West. A line will be skipped.

ENVIRONMENT VARIABLES

$AUTO_DATA
This environmental variable should point to the APS's data directory. It is used to find the default maps file. It not set, the –M option can be used to specify the user's map file.

EXAMPLES

To create a true color image with coastline overlays and other proper annotation from the file S20000001175612.L3_HNAV.

```
% setenv AUTO_DATA /people/aps/aps_v2.6/data
% imgBrowse S20000001175612.L3_HNAV true_color S20000001175612_true_color.tiff
```

CAVEATS

These are the known problems/bugs with the software.

The maximum image size is limited to 3072 X 3072. This limit is imposed by the off-screen rendering functions of the Mesa 3-D graphics library. This limit can be increased by changing MAX_WIDTH and MAX_HEIGHT in Mesa/src/config.h and recompiling the Mesa library(in fact we have increased these two values from 1280 X 1024), and then recompiling this program with the new library.

This program can not navigate on images which are not warped (like the Level-2 files). Additionally, the warped files must have been created by the program **imgMap**(1).

All symbols are drawn in white. There is no option to change these.

SEE ALSO

imgMap(1)

NAME

imgDiff – compute difference between product(s) in two files.

SYNOPSIS

imgDiff [options] ifil1 ifil2 ofile [sds1 sds2 ...]

DESCRIPTION

This program computes a simple difference for all products in two input files writing the result to a third file. The user may select the products, which must exist in both input files **(not implemented)** , on the command line after the output filename. If no products are given on the command line, then imgDiff will use the file attribute 'prodList to obtain the list of products if it exists. **(not implemented)**

If the output file is actually one of the input files, then the product name for the difference image will be the product name plus the term "_diff". **(not implemented)**

Besides the simple difference, imgDiff can compute the percent change and percent difference of the two input images. These products will be added to the simple difference (unless the -d option is used). The names of these products will be the product name plus "_per_chg" and "_per_diff".

The percent change is defined as: $I_1 - I_2/I_2$

The percent difference is define as: $\frac{|I_1 - I_2|}{\left|\frac{I_1 + I_2}{2}\right|}$

OPTIONS

-c Additionally compute the percent change.

-k Count up the number of images that have differences and return this number as an exit status.

-n Do not write output file. If this option is used, then the command line should not contain the output file name. Normally used with the -k and -v options.

-s slope,int
: This option allows the user to specify the scaling slope and intercept to use for the output product. By default, the scaling of the product in the first file is used. Care must be taken when using this option as it applies to ALL output products.

-r min,max
: This option allows the user to specify the minimum and maximum values that are used to determine the difference. If for any pixel, either input image fails to fall within the given range the output pixel will be set to INVALID. Care must be taken when using this option as it applies to ALL output products.

-v Forces imgDiff to run in verbose mode.

--help Print out a small help guide.

--version
: Print out version of software and quit.

EXAMPLES

In the first example, all the products in the 2.3 file (stored in the file attribure 'prodList') will be subtracted from the 2.4 file with the differences stored in the DIFF file. The product names will be consistant across all files. That is, the product 'rrs_412' found in both the 2.3 and 2.4 files, will yield a product called 'rrs_412' in the DIFF file.

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182716.L3_HNAV_2.3 \
         S2000208182716.L3_HNAV_DIFF
```

In this case, we only compute the difference images for the remote sensing reflectances.

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182717.N3_HNAV_2.3 \
         S2000208182716.L3_HNAV_DIFF "rrs_*"
```

Below is an example of using the same output file as input file. Note, that in this case, the output products will be stored back in the 2.4 file with the names: "rrs_412_diff", "rrs_443_diff", etc.

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182717.N3_HNAV_2.3 \
         S2000208182716.L3_HNAV_2.4 "rrs_*"
```

If the user is only interested in whether there are differences and not in the differences themselfs, then the -k option will be useful. Thus it falls that we are normally not interested in the output file, so the -n option is normally selected. Here is an example, using a shell script:

```
dir1=/rs/lvl3/seawifs/2.3/MissBight/2001/jan
dir2=/rs/lvl3/seawifs/2.4/MissBight/2001/jan
find $dir1 -type f > /tmp/a.list
find $dir2 -type f > /tmp/b.list
list=`cat /tmp/a.list /tmp/b.list | sort | uniq`
for f in $list
    if imgDiff -kn $dir1/$f $dir2/$f K_532
    then
        echo $f differs
    fi
done
```

SEE ALSO

NAME

imgLandMask – create a land mask SDS

SYNOPSIS

imgLandMask inFile [ouFile]

DESCRIPTION

This program is used to create a landmask SDS for the given image map provided in the first two arguments. The resulting landmask will be written as 2-D (or 3-D) byte SDS with ''land'' pixels represented by the value 255 and the ''water'' pixels represented by the value 0. If the ''land'' and ''water'' pixels are given has RGB triplets, then the output SDS will have three dimensions. The output SDS will be named ''land_mask''.

By default, the input landmask file is `$AUTO_DATA/landmask.dat` file.

OPTIONS

–l # –or– -l r,g,b
: Output ''land'' pixels using the given value, which must between 0 and 255. Note: ''land'' and ''water'' pixels must have seperate values. The second option will set the ''land'' pixels to the given RGB triplet and produce a 3-D SDS.

–L landmask file
: This option is used to specify the input landmask file.

–n name
: This option is used to specify another name for the output SDS. The default is ''land_mask''

–w # –or– -w r,g,b
: Output 'water' pixels using the given value, which must between 0 and 255. Note: ''land'' and ''water'' pixels must have seperate values. The second option will set the ''land'' pixels to the given RGB triplet and produce a 3-D SDS.

--help Print out a small help guide.

--version
: Print out version of software and quit.

ENVIRONMENT VARIABLES

AUTO_DATA
: The directory where all the data files exist. Defaults to $AUTO_DIR/data.

FILES

landmask.dat
: The landmask file.

SEE ALSO

aps(1) **landmask**(5)

NAME

imgMakeLatLon – create ‘‘latitudes’’ and ‘‘longitudes’’ SDS for image

SYNOPSIS

imgMakeLatLon inFile [ouFile]

DESCRIPTION

This program is used to create a latitude/longitude SDSs for the given input file. The resulting SDS are stored as 32-bit floating point numbers. They are the same size as the arrays in the input file and are named ‘‘latitudes’’ and ‘‘longitudes’’ by default. This program is useful when a latitude/longitude is needed for every point in the image.

OPTIONS

−l name

Rename the ‘‘latitudes’’ SDS to ‘‘name’’.

−L name

Rename the ‘‘longitudes’’ SDS to ‘‘name’’.

−m mapfile:mapname

This option allows one to define the input navigation for a specified mapfile and map name. This is useful when the input file does not contain the navigation information impedded in it.

--help Print out a small help guide.

--version

Print out version of software and quit.

SEE ALSO

aps(1)

NAME

imgMap − project satellite images to map projection

SYNOPSIS

imgMap mapName inFile ouFile sds1 sds2 ...

DESCRIPTION

This program is used to project an image stored as an HDF SDS given a series of control points to a map projection specified by the user. The user should use the program **maps**(1) to create an "image map", that is, an image with a defined number of samples and lines and projection system. Each image map is usually stored in a single file called "maps.hdf". This file contains a series of user-defined image maps given a unique name, since they are stored as HDF SDS's.

The input file must contain the following SDS's "CP_Pixels", "CP_Lines", "CP_Latitudes", and "CP_Longitudes" which define a control points array over the entire input image. These are created by default by **swfCase2**(1) or **spkToSDS**(1) for the seawifs or avhrr data streams, respectively. See **swfScripts**(1) or **avhScripts**(1). Using these points to navigate over the input image (usually in the satellite sensor projection), **imgMap** will fill in the output array from the nearest pixel in the input image. This program does not perform any type of interpolation.

The list of SDS can use regular expressions. The user should quote them, however, so that they are not interpreted by the UNIX shell.

This program can handle 2−D and 3−D input arrays. It is assumed that the 3−D data is stored in BIP format.

The program will append the folling file attribute to the output data file: ‘‘mapProjection’’. This attribute will point to the SDS which contains the mapName. The mapName SDS will automatically be appended to the output file.

OPTIONS

-M mapFile
: Use the given mapFile rather than the default version.

-v
: Forces imgMap to run in verbose mode.

--help
: Print out a small help guide.

--version
: Print out version of software and quit.

FILES

$AUTO_DATA/maps.hdf
: This is the default mapFile which contains the map provided on the command line.

ENVIRONMENT VARIABLES

$AUTO_DATA
: This environmental variable should point to the APS's data directory. It is used to find the default maps file. It not set, the −M option can be used to specify the user's map file.

EXAMPLES

This example will use warp all the remote sensing reflectance images located in S20000001175612.L2_HNAV file using the ‘‘MissBight’’ image map located in the file ~/ladner/maps.hdf.

```
% imgMap -M /people/ladner/maps.hdf MissBight \
S20000001175612.L2_HNAV S20000001175612.L3_HNAV "rrs_*"
```

If $AUTO_DATA is set then, then user does not have to use the –M option.

```
% setenv AUTO_DATA /people/aps/aps_v2.6/data
% imgMap GulfOfMexico S20000001175612.L2_HNAV S20000001175612.L3_HNAV "rrs_*"
```

SEE ALSO

maps(1)

NAME

imgMean – calculate Mean/Min/Max/StDev of a series of images.

SYNOPSIS

imgMean ofile ifil1 ifil2 ...

DESCRIPTION

This program will produce an image of the mean for each pixel in a series of images. The program can also produce an image of the minimum, maximum, standard deviation and count. Additionally to aid in "running" averages the user can request that the sum and sum of the squares (used for standard deviation) are also output.

OPTIONS

-a Do not create the average (mean) image.

-c Produces an image of the number of pixels at each corresponding (X,Y) coordinates were used to determine the mean. The minimum value is zero and the maximum will be the number of input files used to produce the mean.

-f # Apply the given function to the input data before determining any statistics with the data. The number represents the available functions which are: 1 for log10, 2 for alog10, 3 for ln, 4 for exp. This filtering is applied after any range checks are performed.

-F names
Allows the user to specify the full output file names, rather than supplying a base output file name. **[currently unimplemented]**

-H sdsName
Designates that the files to be used in making composites will be in HDF format. The user must supply the name of which SDS to use.

-i Read in the sum and sum squared images. **[currently unimplemented]**

-I # Used to specify the value to be used in replacing invalid data.

-m # The mask value used to filter out pixels from the compositing. **[currently unimplemented]**

-M sdsName
The name of the mask array to use for masking. The default is "l2_flags". **[currently unimplemented]**

-n Produces an image of the minimum value of all input images at each corresponding pixel location.

-N Designates that the files to be used in makeing composites will be in NSIPS **[currently unimplemented]**

-o Output the sum and sumsq images.

-r #,# Set the lower and upper bounds for range checking.

-s Produces an image of the standard deviation for each corresponding pixel for all input images.

-S Designates that the files to be used in making composites will be in SeaPack format. **[currently unimplemented]**

-v Forces imgMean to run in verbose mode.

-x Produces an image of the maximum value of all input images at each corresponding pixel location.

-W weights

Creates a weighted average based upon the weight table given in weights. The output SDS for the weighted average will be SDS_NAME_weight.

NOTE weighted tables may only be generated from HDF input !!!

weights is defined as a string of colon ":" seperated floating point numbers which define your weight table.

ex: -W .6:.2:.1:.05:.04:.01

would define a weight table as follows

first valid pixel .6
second valid pixel .2
third valid pixel .1
etc

The weight table is sliding, that is the weights get used in the order that they are given, if a pixel does not exist for a particular weight the next valid pixel will have that weight applied.

One should be careful in choosing the weight table as data can be manipulated via the use of incorrect tables. A suggestion would be to use values 0 - 1 where the weights add to 1 or 0 - 100 where the values add to 100. The STD dev of your weight table should NOT be higher than the median. The program will generate a warning in this circumstance.

--help Print out a small help guide.

--version

Print out version of software and quit.

EXAMPLES

This example will composite together all sea surface temperature values for NOAA-14 for day 364 of 1999.

```
$ imgMean -H sst day364.hdf /rs/lvl3/avhrr/2.4/Gulf*/1999/dec/ND1999360*
$ hdf day364.hdf list
File: /people/aps/day364.hdf
File Attributes: (6)

createTime = "Thu Jan  6 09:34:16 2000"
```

```
createAgency = "Naval Research Laboratory, Stennis Space Center"
createSoftware = "APS v2.4"
createUser = "aps"
createPlatform = "mips-sgi-irix6.5"
inputFiles = "/rs/lvl3/avhrr/2.4/GulfOfMexico/1999/dec/ND1999360
  112608.L3_HNAV_GOM,/rs/lvl3/avhrr/2.4/GulfOfMexico/1999/dec/ND
  999360130716.L3_HNAV_GOM,/rs/lvl3/avhrr/2.4/GulfOfMexico/1999/
  dec/ND1999360224130.L3_HNAV_GOM"

Data Sets: (2)

char8   History [0]

int16   sst_mean [1810,2430]
    createTime = "Thu Jan  6 09:34:16 2000"
    createUser = "aps"
    createPlatform = "mips-sgi-irix6.5"
    productScaling = "Linear"
    scalingSlope = 0.001
    scalingIntercept = 20
$
```

Notice that the list of files used in the composite are given in the file attribute "inputFiles". Also, notice that the mean product is defined by appending the sdsName with "_mean".

Currently, the imgMean program is navigationally-impaired, that is it does not copy over the navigation information from the files. We can fix that by using the **hdf** program to copy over the navigation data. Each of these input files contains four SDS's which formulate a control points grid over the image. They are CP_Lines, CP_Pixels, CP_Latitudes, and CP_Longitudes.

```
$ cd /rs/lvl3/avhrr/2.4/GulfOfMexico/1999/dec/
$ hdf ND1999360112608.L3_HNAV_GOM copy ~/day364.hdf CP_Pixels CP_Lines
$ hdf ND1999360112608.L3_HNAV_GOM copy ~/day364.hdf CP_Latitudes CP_Longitudes
```

At this point the image can be read into SeaDAS and properly navigated.

SEE ALSO

hdf(1)

NAME

imgMean2 – calculate Mean/Min/Max/StDev of a series of images.

SYNOPSIS

imgMean2 ifil1 ifil2 ...

DESCRIPTION

This program will produce an image of the mean for each pixel in a series of images. The program can also produce an image of the minimum, maximum, standard deviation. Currently the program is limited by the HDF libraries to composites of at most 31 files. A shell script has been created that when used with the -F option can work around this limit.

In addition, this program can make incrememtal composites(the addition of one or more files to an already created composite file) and can merge two or more composite files. Names of composite files may be placed on the command line or in an input file(when using -F) just like regular Level-3 files, they will be recognized as NRL Level-4 files.

OPTIONS

-a Do not create the average (mean) image.

-c Output a "count" product.

-C sdsName
The name of the SDS containing the cloud albedo for each pixel. Default: "cloud_albedo".

-F filename
Get files to composite from "filename". The format of this file is one file name per line, the whole line is used, and may contain spaces or any other character. DO NOT QUOTE FILENAMES. Any combination of Level-3 and NRL Level-4 files may be used.

-f # Apply the given function to the input data before determining any statistics with the data. The number represents the available functions which are: 0 for none, 1 for log10, 2 for alog10, 3 for ln, 4 for exp. The function is applied after any range checks are performed. Multiple functions may be specified by separating the functions by a comma. If fewer functions are specified than products, no function will be used with the remaining products. The -f option is only valid when -H is also specified.

ex.
-H K_532,chl_oc4 -f 0,1

K_532 function none
chl_oc4 function log10

or

-H K_532,chl_oc4,bb_555_arnone -f 0,1

K_532 function none
chl_oc4 function log10
bb_555_arnone function none

-H sdsName

Designates that the files to be used in making composites will be in HDF format. The user must supply the name of which SDS to use. Multiple SDS's may be specified by separating them with comments.

ex.
-H K_532,chl_oc4

-I # Used to specify the value to be used in replacing invalid data.

-m # The mask value used to filter out pixels from the compositing. The mask may be specified as an integer or as a comma separated string of flag names.

ex. To mask out the ATMFAIL and LAND flags, use either of the following:
-m 3
or
-m ATMFAIL,LAND

-M sdsName

The name of the mask array to use for masking. The default is "l2_flags".

-n Produces an image of the minimum value of all input images at each corresponding pixel location.

-o outfile

Set the output filename to "outfile". If this option is not used then an output filename will be created based on the start and end times of the input files. The name will be of the form SYYYYDDDYYYYDDD.L4_TT_REGION where the first group of YYYYDDD is the earliest start year and julian day and the second group is the latest end year and julian day, the TT is the composite type as set by the -T option. TT will be WE for weekly, MO for monthly, YR for yearly, and RO for Rolling composites. If a daily composite was specified then the name will be SYYYYDDD.L4_REGION. REGION is a short string describing the region. (e.g. GOM for GulfOfMexico)

-r #,# Set the lower and upper bounds for range checking. A lower and upper bounds may be specified and must be separated by a comma. Ranges for multiple products may be specified by separating the ranges by a colon. The -r option is only valid when the -H option is also specified.

ex.
-H K_532 -r .01,6
or
-H K_532,chl_oc4 -r .01,6:.01,64

If fewer ranges are specified using -r than products using -H then the last value in the -r list will be for the rest of the -H products.

ex.
-H K_532,chl_oc4, chl_stumpf,... -r .1,5:.1,64

K_532 range .1 - 5
chl_oc4 range .1 - 64
chl_stumpf range .1 - 64
... range .1 - 64

If no ranges are specified, then the ranges from the 'validRange' attribute are used for range checking.

-s Produces an image of the standard deviation for each corresponding pixel for all input images.

-T # Specify type of composite. 0 for daily, 1 for weekly, 2 for monthly, 3 for yearly, 4 for rolling composites. This is mainly an informational option to describe the composite. It is also used when automatically creating the filename(see -o).

-v Forces imgMean to run in verbose mode.

-x Produces an image of the maximum value of all input images at each corresponding pixel location.

-W weights

Creates a weighted average based upon the weight table given in weights. The output SDS for the weighted average will be SDS_NAME_weight.

NOTE Currently incremental composites are not possible with this option.

weights is defined as a string of comma "," seperated floating point numbers which define your weight table.

ex: -W .6,.2,.1,.05,.04,.01

would define a weight table as follows

[all valid pixels in the first input file] * .6
[all valid pixels in the second input file] * .2
[all valid pixels in the third input file] * .1
etc.

The sum of the valid pixels * their respective weights is divided by the addition of those weights that were used for each pixel location.

ex. 5 files are input, using a weight table of .5,.4,.3,.2,.1
for pixel location p files 1,2, and 5 have valid data, the weighted mean would be calculated as follows:
([pixel 1] * .5 + [pixel 2] * .4 + [pixel 5] * .1) / (.5 + .4 + .1)

-z # Set the sensor zenith angle threshold. imgMean2 will ignore pixels whose sensor zenith angle(degrees) is above this threshold.

-Z sdsName

The name of the SDS containing the sensor zenith angles(degrees) for each pixel. Default: "senz".

--help Print out a small help guide.

--version

Print out version of software and quit.

EXAMPLES

This example will composite together all sea surface temperature values for NOAA-14 for day 29 of 2002.

```
$ imgMean -H sst day029.hdf /rs/lvl3/avhrr/2.4/GulfOfMexico/2002/jan/ND2002029*
$ hdf day029.hdf list
File: day029.hdf

File Attributes: (39)

createTime = "Fri Feb  1 10:54:29 2002"
createAgency = "Naval Research Laboratory, Stennis Space Center"
createSoftware = "APS v2.4"
createUser = "aps"
createPlatform = "mips-sgi-irix6.5"
file = "day029.hdf"
fileTitle = "NRL Level-4 Data"
fileVersion = "2.4"
fileClassification = "UNCLASSIFIED"
fileStatus = "Operational"
sensor = "AVHRR"
sensorAgency = "NOAA"
sensorType = "scanner"
sensorSpectrum = "Visible/Thermal"
sensorNumberOfBands = 5
sensorPlatform = "NOAA-12"
sensorPlatformType = "Polar-orbiting Satellite"
navType = "mapped"
mapProjectionSystem = "NRL(USGS)"
mapProjection = "GulfOfMexico"
mapUpperLeft = [31.0064,-98.0074]
mapUpperRight = [31.0064,-80.0074]
mapLowerLeft = [18.811,-98.0074]
mapLowerRight = [18.811,-80.0074]
inputParameters = "-v -H sst day029.hdf /rs/lvl3/avhrr/2.4/GulfOfMexico/2002/jan/ND2
inputMasks = ""
inputMasksInt = 0
timeStart = "Tue Jan 29 09:49:13 2002"
timeStartYear = 2002
timeStartDay = 29
timeStartTime = 35353313
timeDayNight = "Day/Night"
timeEnd = "Tue Jan 29 22:46:49 2002"
timeEndYear = 2002
timeEndDay = 29
timeEndTime = 82009657
prodList = "sst_mean"
Composition Type = "Weekly Composite"
inputFiles = "/rs/lvl3/avhrr/2.4/GulfOfMexico/2002/jan/ND2002029094913.L3_HNAV_GOM,/

Data Sets: (6)

float64 CP_Pixels [122]
    createTime = "Tue Jan 29 04:03:28 2002"
    createSoftware = "imgReformat"
```

```
        createUser = "aps"
        createPlatform = "i686-pc-linux-gnu"
        productName = "Sample Locations"
        productUnits = "pixels"

    float64 CP_Lines [91]
        createTime = "Tue Jan 29 04:03:28 2002"
        createSoftware = "imgReformat"
        createUser = "aps"
        createPlatform = "i686-pc-linux-gnu"
        productName = "Line Locations"
        productUnits = "pixels"

    float64 CP_Longitudes [91,122]
        createTime = "Tue Jan 29 04:03:28 2002"
        createSoftware = "imgReformat"
        createUser = "aps"
        createPlatform = "i686-pc-linux-gnu"
        productName = "Longitudes"
        productUnits = "decimal degrees"
        validRange = [-180,180]
        dataRange = [-98,-80.0074]

    float64 CP_Latitudes [91,122]
        createTime = "Tue Jan 29 04:03:28 2002"
        createSoftware = "imgReformat"
        createUser = "aps"
        createPlatform = "i686-pc-linux-gnu"
        productName = "Latitudes"
        productUnits = "decimal degrees"
        validRange = [-90,90]
        dataRange = [18.811,31]

    float64 GulfOfMexico [29]
        createTime = "Tue Jan 29 04:03:27 2002"
        createSoftware = "imgReformat"
        createUser = "aps"
        createPlatform = "i686-pc-linux-gnu"
        productName = "Map Projection Parameters"

    int16   sst_mean [1810,2430]
        createTime = "Fri Feb  1 10:54:29 2002"
        createSoftware = "imgMean2"
        createUser = "casey"
        createPlatform = "mips-sgi-irix6.5"
        productName = "Sea Surface Temperature"
        productUnits = "Celsius"
        productScaling = "Linear"
        scalingSlope = 0.001
        scalingIntercept = 20
        validRange = [-1.79769E+308,1.79769E+308]
    $
```

The file name given to the file is in the attribute "file". The "fileTitle" attribute has been set to "NRL Level-4

Data". "inputParameters", "inputMasks", and "inputMasksInt" contain their respective values. The "timeStart*" and "timeEnd*" attributes contain the earliest start times and latest end times respectively of the input files. "inputFiles" contains a list of the files used to create the products. Also, notice that the mean product is defined by appending the sdsName with "_mean", as would be true for the std. dev. ("_dev"),max ("_max") and min ("_min").

SEE ALSO

hdf(1)

NAME

imgRead − dump information from images

SYNOPSIS

imgRead [options] ifil sds1 sds2 ...

DESCRIPTION

The program imgRead allows the user to retrieve data from an image at any desired position specified either by (latitude,longitude) pair or (line,sample) pair. The values are read as geophysical values and dumped to stdout. The user may select a single point or a square around the specified position. Eight different box sizes specify an area from 3x3 to 17x17 pixels.

CAVEAT

The latitude/longitude options can be used only with map projected files and must use the −m option. All others (Level−1, Level−2, etc.) can only use line,sample option, i.e. −x option.

OPTIONS

-b size Use a box around the point of interest. Should be one of 3, 5, 7, 9, 11, 13, 15, 17.

-c Output data in columns. If used with the −b option, this will output the average of the box in the column. Cannot be used with the -S option.

-f %#.#f

Used to control output formatting with the -b option. The number is the number of spaces for the entire number and the second number reprsents the number of decimal places. For example: −f %10.5f will output: xxxx.xxxxx Default format is %10.4f.

-F flagName

Show the flag specified as *flagName* as a 16 bit binary number. The output will be a string of 16 one's or zeros.

-g outputFile

Used to output data in a format acceptable by the GNU program graph. The argument outputFile should be the name of the output file to be created. The name of the SDS extracted will be appended to the outputFile name.

-h This option is used to suppress the headers.

-m mapFile:mapName

This option is used when a mapped file will be read by the program. The first string should be that of the maps file followed by a colon and the name of the map. For example, −m maps.hdf:ChesepeakeBay.

-p [x|y] This option outputs a row (y) or a column (x) profile of the data.

-r min,max

Set the minimum and maximum range for the data used to calculate the output statistics when using the box option.

-S This will force the output to look similar to the Seadas output. Cannot be used with the −c option.

-t Insert tabs between columns when using columnar output.

-x Treat input values as samples and lines.

--help Print out a small help guide.

--version
Print out version of software and quit.

EXAMPLES

To dump a series of points read in from file stations.dat and output to data.dat.

```
$ more stations.dat
37.4502 -89.3403
37.5320 -89.3403
-99.0 -99.0
$ imgRead -m maps.hdf:MissBight S1998100175129.N3_HNAV_MSB nLw_412 nLw_443 < station
$ more data.dat
  Latitude   Longitude Pixel  Line     nLw_412     nLw_443
   37.4502    -89.3403   302   142     -0.2040      0.0300
   37.5320    -89.3403   303   141     -0.0010      0.2160
```

NAME

imgReformat – tile and compress an HDF SDS.

SYNOPSIS

imgReformat inFile ouFile xnumchunks ynumchunks sds1 sds2 ...

DESCRIPTION

This program is used to convert an unchunked SDS into a chunked and compressed SDS. The number of chunks across the array are given on the command line as xnumchunks and ynumchunks. Each chunk will be compressed using the gzip deflate compression scheme at a level of 6 by default. To specify an encryption method the user may use command line options as defined below. Three dimensioned chunks are supported but they will be chunked along the XY with Z being constant to it size.

OPTIONS

--help Print out a small help guide.

--version

Print out version of software and quit.

CAVEATS

The input SDS's must all be 2-D. This program can not handle other dimension sizes.

SEE ALSO

HDF User Guide

NAME

imgScale − scales an HDF SDS to a byte SDS

SYNOPSIS

imgScale [options] inFile ouFile sds1 sds2 ...

DESCRIPTION

imgScale is used to scale an input product SDS to byte SDS using one of several methods. The program may also subsection and subsample the input array.

OPTIONS

-B [isp=isp,iep=iep,isl=isl,iel=iel,irp=irp,irl=irl]

Define a subsection of the original image for output. Cannot be used with the -s option.

isp is the starting sample number
iep is the ending sample number
isl is the starting line number
iel is the ending line number
irp is the replication factor along the samples dimension

The value indicates the number of samples to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

irl is the replication factor along the lines dimension

-f [linear|log|log10]

Select Function used for scaling. Defaults to linear.

-o offset

Used to shift the input pixel before taking the log. This can be used if the input SDS may contain negative values. Used only if −f is set to *log* or *log10*.

-r min,max

Select range of input data for scaling. Defaults to min/max of each SDS's validRange parameters. Failing that it uses the min/max of SDS. Failing that it is set to min/max of all reals. Cannot be used with the −S option.

-R min,max

Range of output data. Defaults to 0, 255 for byte scaled imagery.

-S slope,intp

Set the slope and intercept to be used for scaling the data. Cannot be used with the −r option.

-s w,h Sets the size of output image. Used to reduce or enlarge the size of the output. Defaults to size of original image. Cannot be used with the −B option.

-v Forces imgScale to run in verbose mode.

--help Print out a small help guide.

--version
Print out version of software and quit.

EXAMPLES

To rescale a Gulf of Mexico chlorophyll–a image to one third its original size (assuming the original data is 2430 samples by 1810 lines) with a data range from 0.01 to 64.0 and a log10 scale:

```
$ imgScale -f log10 -r 0.01,45.0 -s 810,603 S1999240175128.N3_HNAV_GOM chlor_a.hdf c
```

```
To subsection out the MissBight region from a Gulf of Mexico chlorophyll-a
using the known sample/line ranges with a data range from 0.01 to 64.0
and a log10 scale:
```

```
$ imgScale -v -r 0.01,64.0 -f log10 -B isp=1000,iep=1749,isl=50,iel=399 S19992401751
```

NAME

imgSDStoImg − convert HDF SDSs to another output image format.

SYNOPSIS

imgSDStoImg [options] inFile baseName sds1 sds2 ...

DESCRIPTION

By default, the program **imgSDStoImg** will output each SDS specified on the command line to its own output file. The output file can be one of: (1) a simple binary file; (2) an ENVI formatted file; (3) a NSIPS image file; (4) an NRL−type PC−SEAPAK formatted file; or (5) a TIFF image file (possibly with GeoTiFF tags). In some cases, much of the navigation and ancilliary data will not be present in the resulting file as some of these formats are not tuned for this type of information.

For binary files, the image is simply written in binary format. The −r option can be used to output the SDS in geophysical units. The output files will be named by appending the name of each SDS with the string "`.bin`".

For ENVI output, the result is actually two files. One is a binary dump of the input data which may be converted to floating point geophysical data using the −r option. The second file is an ENVI header which describes the data in the binary file. If the input file contains navigation information in the NRL format (see **maps**(1)), then the map projection information is written to the ENVI header. *Note*: The NRL projection software is based on the USGS projection code and contains over 30 different projections. Currently, **imgSDStoImg** will only handle the Mercator map projection. The ENVI format may also be written in multibanded format (see −M). For each single−banded output ENVI file (default), the output file name will consist of the baseName with the name of each SDS product and the string "`.envi`". The ENVI header file will have the string "`.hdr`" further appended to it. For a multi−banded ENVI file, the baseName will *be* the output filename. The ENVI header file in this case will be the baseName with the string "`.hdr`".

For NSIPS files, only the image data is converted. Since the format cannot handle attributes, including scaling parameters, none of these are output. However, the user can output the data in "geophysical" values by applying the slope and intercept attributes to the data before it is written. In this case, the output file will be a floating point type NSIPS file. See option −r. The output files will be named by appending the name of each SDS with the string "`.img`".

For SEAPAK files, the scaling parameters are stored in the output file along with the imagery. If the input SDS file contains a control points grid, it is written out in a SEAPAK control points file − so navigation is possible with SEAPAK files. The output files will be name by appending the name of each SDS with the string "`.spk`". If the control point file is created, its name is generated by appending the baseName with the string "`.ctl`" without the SDS name.

For TIFF files, the image data will be written to a TIFF formatted file. If the input HDF file contains navigation, then GeoTIFF tags will be appended to the TIFF file. The output file will be named by appending the name of each SDS with the string "`.tiff`".

The user may specify the Scientific Data Sets for output by using regular expressions. These must be put in quotes to protect them from expansion by the shell, however.

OPTIONS

-b Output file should be a flat binary file.

-e Output file should be a ENVI Standard type file.

-m mapFile:mapName
: Read the map projection information from the given file and named SDS. This is usually required if the map projection information is located in another file and not supplied in the input file.

-M Put all bands into a single file (available only for ENVI files).

-n Output file should be an NSIPS file.

-r Convert data to floating point if the data calibration is available. This is only applied to data being written out in the binary, ENVI, and NSIPS formats.

-s Output file should be a SEAPAK file.

-S Output each SDS into it own single output file.

-t Output file should be a TIFF file. If the input file has navigation, then the GeoTIFF tags will be written to the output file **(currently unimplemented)**.

-v Verbose mode

--help Print out a small help guide.

--version
: Print out version of software and quit.

EXAMPLES

To create a series of NSIPS ‘‘geophysical’’ files of all the remote sensing reflectance Scientific Data Sets in the file S2000001175134.N3_HNAV.

```
$ imgSDStoImg -nr S2000001175134.N3_HNAV S2000001175134_ "rrs_*"
$ ls S2000001175134*
S2000001175134_rrs_412.img     S2000001175134_rrs_510.img
S2000001175134_rrs_443.img     S2000001175134_rrs_555.img
S2000001175134_rrs_490.img     S2000001175134_rrs_670.img
```

To create an ENVI file of remote sensing reflectances from the file S2000001175134.N3_HNAV:

```
$ imgSDStoImg -erM S2000001175134.N3_HNAV S2000001175134.rrs "rrs_*"
$ ls S2000001175134*
S2000001175134.rrs             S2000001175134.rrs.hdr
```

To create an ENVI file of remote sensing reflectances from the file S2000001175134.N3_HNAV out of the database with map projection information in the file:

```
$ imgSDStoImg -erMm /people/aps/aps_v2.6/maps.hdf:GulfOfMexico
/rs/lvl3/seawifs/2.3/GulfOfMexico/2000/mar/S2000079181416.N3_HNAV_GOM
S2000079181416.envi "rrs_*"
$ ls S2000079181416*
S2000079181416.envi            S2000079181416.envi.hdr
```

NOTES

The binary files produced by imgSDStoImg are in native format. If the file is transferred across platforms, the user will have to handle any all byte swapping. For example, if writing out a binary file on an SGI and reading the image into MATLAB on a PC. Some formats (like TIFF) and software (like APS) handle the platform conversion automatically.

NAME

imgFlags – produce an image SDS from a flag SDS.

SYNOPSIS

imgFlags [options] inFile.hdf ouFile.hdf

DESCRIPTION

This program will extract a flag out of an SDS of flags and make an image SDS out of it. Given an input array that must be an integer of either 8, 16, or 32 bits (signed or unsigned), this program will create a byte image SDS for each pixel whose desired bit is set to some user defined value. The user can also select a "true" color image by setting all three colors of an RGB triplet.

OPTIONS

-F flag Name of the flag to extract. A ‘‘proper’’ SDS flags array will contain attributes of the format ‘‘fXX_name’’. These attributes contain the name of the flag for each bit. For example, the SDS ‘‘l2_flags’’ may contain the attribute ‘‘f01_name’’ and its value man be ‘‘LAND’’. Then, the user may add ‘‘-F LAND’’ to make an image of all pixels for which bit 1 is turned on.

-N name

Name of SDS in the input file containing a integer SDS flags array. Defaults to "l2_flags".

-s n –or– -s r,g,b

Color index or RGB triplet output for all "set" values in output image. Defaults to 255.

-S samples,lines

Used to set size of output image, if user wants to resize the image.

-u n –or– -u r,g,b

Color index or RGB triplet output for all "unset" values in output image. Defaults to zero.

P --help Print out a small help guide.

--version

Print out version of software and quit.

EXAMPLES

This example uses the SDS named "l2_flags" located in the file S1999341014937.N3_HJMS and pulls out the LAND flag. The output image is a byte SDS formatted file that has 750 samples and 750 lines. It contains only zero’s and one’s with one’s indicating land as defined by the SeaWiFS flag LAND1.

```
$ imgFlags -F LAND -s 1 -S 750,750 S1999341014937.N3_HJMS land.hdf
```

In this example, the land pixels are set to green and the water pixels (technically, NOT land) are set to blue. The resulting SDS image has 384 lines and 384 samples and 3 bands. (This example is using a MOS cube).

```
$ imgFlags -s 0,255,0 -u 0,0,255 -F LAND M2000109164401.\
L2_HWFF_00280B18 true.hdf
```

NAME

imgTSeries – compute difference between product(s) in two files.

SYNOPSIS

imgTSeries [options] ifile roi ofile sds

DESCRIPTION

imgTSeries is used to perform stastics on a region of interest and output the results in an ASCII file.

OPTIONS

-F name

Name of the mask SDS. Defaults to 'l2_flags'.

-L name

Name of the input land mask file SDS. Defaults to $AUTO_DATA/landmask.dat

-r min,max

Set the minimum and maximum of the range of data to consider. Defaults to values set in 'validRange' attribute.

-v Forces imgMedian to run in verbose mode.

--help Print out a small help guide.

--version

Print out version of software and quit.

SEE ALSO

roi(1)

NAME

spkToSDS − convert SEAPAK files to HDF SDSs

SYNOPSIS

spkToSDS ofile.hdf ifil1 ifil2 ..

DESCRIPTION

This program is used to convert SEAPAK files to an HDF format. The program will read in each file and create an SDS based on the data product type.

EXAMPLES

The following examples takes a series of image files called levels1.spk, levels2.spk, up to levels9.spk and a control points file and stores the results in the HDF file called ofile.hdf

```
$ spkToSDS ofile.hdf levels?.spk levels.ctl
```