NAME

imhist - compute a histogram of an image

SYNOPSIS

imhist [*options*] *infilename* [*outfilename*]

DESCRIPTION

imhist reads each image in the input file and computes a histogram by counting the number of occurrences of each unique color in the image. A table of the computed data, sorted from high to low, plus a header of general image statistics is written to the output file, or stdout if no output file is given.

OPTIONS

imhist has a variety of options in the following categories:

File Selection	What input file to use
Format Selection	What image file format to use
Standard	Standard generic options on all SDSC tools
Output Control	What to output
Fields	What fields to use in computing the histogram

File Selection, *Format Selection*, and *Standard* options are common to all SDSC image tools and are discussed in depth in the man page for imconv(1IM).

All options can be abbreviated to the first few unique characters.

Output Options

imhist computes a variety of image statistics, as well as a histogram of image colors. When complete, a report is written to the output file or stdout that contains a header followed by the histogram occurrence table.

The output report's header contains the following information:

File:	image file name
Resolution:	image resolution
Number of Pixels:	number of pixels in image
Number of Combinations:	number of unique color combinations
Maximum:	maximum field value for each histogram field
Minimum:	minimum field value for each histogram field
Unique Values:	number of unique field values for each histogram field
Column headings	headings for columns in the histogram occurrence table

Printing of the header information may be suppressed by giving the -outnoheader option. By default the header information is output.

The histogram occurrence table gives a list of the unique color combinations that occurred in the image, and a count of the number of times they occurred. The table is sorted to place the most-often occurring colors at the top of the table. Table values are printed one color to a line starting with the occurrence value, followed by the value of each histogram pixel field. For example, the following are the first few lines of a histogram table generated on a simple RGB image:

```
...        ...     ...       ...
Number     RED     GREEN     BLUE
1004       255     0         0
```

842	128	255	0
712	0	0	255
500	0	0	0
...	...	...	...

In this example, the histogram data shows that there were 1004 occurrences of pixels with the RGB combination (255,0,0), and 842 occurrences of pixels with an RGB value of (128,255,0). And so on.

Calculation and printing of the histogram table may be suppressed by giving the -outnohistogram option. By default the histogram information is output.

Sorting of the histogram occurrence data on pathological cases can take as long as collecting the data in the first place. If sorting is not important to your use of the data, you may suppress sorting by giving the -outnosort option. By default the histogram occurrence data is sorted from most-often occurring to least-often occurring.

Fields

By default, imhist computes a histogram on all fields (red, green, blue, alpha, color index, etc) in the input image. Alternatively, the user may direct imhist to specifically compute a histogram on selected image fields, or virtual fields (such as hue, saturation, and intensity) by selecting the fields of interest on the command-line.

Field	Use
-red	Compute on red values
-green	Compute on green values
-blue	Compute on blue values
-index	Compute on color indexes or grayscale values
-mono	Compute on monochrome values
-alpha	Compute on alpha values
-hue	Compute on hue values
-saturation	Compute on saturation values
-intensity	Compute on intensity values

Legal field selections depend upon the image type. Color indexed images (images that use a color lookup table) may have histograms computed on their color indexes, RGB values (as looked up in their color lookup tables), and HSI values (derived from RGB values looked up in the color lookup table).

RGB images may have histograms computed on RGB values or on HSI values (derived from the RGB values).

Images with alpha channels may have histograms computed on alpha values.

Histograms may not be computed that mix color spaces/representations. For instance, histograms on -red -index or -green -hue don't make sense.

NOTES

imhist computes correlated field histograms. Each color in the output histogram is a unique combination of the selected input fields. For instance, say an image has the following RGB colors:

RED	GREEN	BLUE
0	0	0
255	0	0

0	255	255
128	128	255
255	255	255

A histogram on the red field alone would report that there are 3 different unique values: 0, 128, and 255. Similarly, independent single-field histograms on the green and blue fields would report that green has 0, 128, and 255 as unique values, and blue has just 0 and 255 as unique values.

A correlated histogram on all three fields, however, would report each unique combination of red, green, and blue field values that occur in the image. The answer would be the table listed above. In the table the second and fifth colors both have red values of 255, but differ in their green and blue components. This makes them different colors, and thus different entries in the histogram.

Non-correlated single-field histograms are handy if you just want to see how many pixels have some amount of red in them. However, what if you want to know how many have yellow in them? For that you need to correlate a histogram showing red and green values. Independent red and green histograms wouldn't show you anything useful.

imhist may be used to compute either single-field histograms or multiple-field correlated histograms. To compute single-field histograms on red and green, for instance, run imhist twice, once with the -red option, and once with the -green option. Alternatively, to compute a red-green correlated histogram, run imhist with both the -red and the -green options at the same time.

Computing single-field histograms is very quick and doesn't use much memory. Computing multiple-field correlated histograms can take quite a bit longer and use a lot more memory, depending upon the number of correlated fields, the size of the image, and the number of unique color combinations in the image. The following table provides an idea of the memory explosion that can result when computing correlated histograms. The table's values are based upon a worst case 1024 x 1280 image with every pixel a different color.

Number of Fields	Color Space	Internal Memory Use
1	RGBA	2K Bytes
1	HSI	10K Bytes
2	RGBA	328K Bytes
2	HSIA	4,000K Bytes
3	RGBA	9,175K Bytes
3	HSI	20,972K Bytes
4	RGBA	10,486K Bytes
4	HSIA	22,283K Bytes

The memory use reported above is for intermediate data structures internal to imhist. This does not include storage in memory of the image itself, nor any of the other administrative data structures needed.

Of note on this table is the use of some 22Mb for internal data when computing an HSI-Alpha correlated histogram on a pathological case 1280 x 1024 image. If your host doesn't have 22Mb (plus overhead) of memory, or the equivalent in swap space, imhist will be unabled to compute the histogram. Every effort has been made in imhist to keep memory use to a minimum. This high use of memory for correlated histograms cannot be avoided without resorting to temporary files for data storage, and the attendant sever execution speed impact.

Please note again that this table was produced for a pathological image that had a different color for every pixel. Most images have considerably fewer colors. In typical use, memory will not be a problem.

Computation time for correlated histograms also goes up with the number of correlated fields, the size of the image, and the number of unique colors. The following table shows compute times running imhist on local disk on a Silicon Graphics Indigo using uncompressed RGB images.

Size (pixels)	Field Choice	# of Unique Colors	Read (sec)	Compute (sec)	Write (sec)	Total (sec)
64,000	R	7	<1	<1	<1	<1
64,000	RGB	14	<1	<1	<1	<1
64,000	H	10	<1	<1	<1	<1
64,000	HSI	14	1	1	1	1
90,000	R	140	<1	<1	<1	<1
90,000	RGB	256	<1	1	1	2
90,000	H	85	<1	<1	<1	<1
90,000	HSI	256	<1	5	5	10
490,000	R	254	4	1	1	5
490,000	RGB	149,942	4	210	20	234
490,000	H	355	4	5	2	11
490,000	HSI	149,942	4	345	68	417
1,000,000	R	217	7	2	1	10
1,000,000	RGB	7,256	7	9	3	19
1,000,000	H	284	7	7	1	15
1,000,000	HSI	7,256	7	32	7	46

Column one shows the number of pixels in the test image. Column two gives the fields used in computing the histogram. Column three shows the number of unique colors found. Columns four, five, and six show execution times for reading in the image, computing the histogram, and writing out the data respectively. The final column sums these execution times.

This table illustrates that larger images with larger numbers of unique colors drastically increase the time it takes to compute a correlated histogram. This is to be expected. The table also shows that computing in the HSI domain, instead of RGB, takes longer. This is due to the use of floating point computations and RGB-to-HSI conversions everywhere.

Error messages are reported to stderr.

EXAMPLES

Create a histogram of an image and output it to stdout:

```
imhist myimage.ras
```

Create a histogram of just the red field:

```
imhist myimage.rgb -red
```

Create a histogram of color indexes and write it to an output file:

```
imhist image.rgb -index report.text
```

Create a histogram of the hue, saturation, and intensity fields:

imhist image.rgb -hue -saturation -intensity report.text

Only output the header:

imhist image.rgb -outnohistogram header.text

Only output the histogram data:

imhist image.rgb -outnoheader report.text

Skip sorting of the histogram data:

imhist image.rgb -outnosort report.text

SEE ALSO

ImVfbHist(3IM), ImVfbStat(3IM)

For information on SDSC's image library, see imintro(3IM).

AUTHOR

Dave Nadeau and Chris Groening
San Diego Supercomputer Center

See the individual file format man pages for the authors of the underlying format read and write code. The names of these man pages begin with the letters "im" followed by the format name. For example, the name of the TIFF man page is imtiff. To display it, enter man imtiff.

CONTACT

SDSC consultants, (619)534-5100, consult@y1.sdsc.edu