

NAME

imscale - Scales an image up or down and saves it in a new file

SYNOPSIS

imscale [options] infilename outfile

DESCRIPTION

imscale scales an input image up or down to a new size and saves the result in a new file. If the input file contains multiple images, each input image is scaled in the same way and is written to the output file. The input and output image file formats may be different.

OPTIONS

imscale has a variety of options in the following five categories:

File Selection	What input and output files to use
Format Selection	What image file format to use
Format Control	What variant of a file format to generate
Standard	Standard generic options on all SDSC tools
Manipulation	How the image can be manipulated

File Selection, Format Selection, Format Control, 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.

Manipulation Options

Images may be scaled in the X (horizontal) and Y (vertical) directions by specifying a scale factor or the new image size:

-scale x yf	Scale horizontally and vertically by factor
-xscale xf	Scale horizontally by factor
-yscale yf	Scale vertically by factor
-xsize w	Scale the image to a new size horizontally
-ysize h	Scale the image to a new size vertically

-scale, -xscale and -yscale each take a positive floating-point scale factor. To increase the size of an image, use a scale factor greater than 1.0. To decrease an image's size, use a scale factor less than 1.0. The default scale factor is 1.0 (no change).

-xsize and -ysize each take a positive integer image width or height in pixels, respectively. The incoming image is scaled up or down, as necessary, to make it the desired size. The default is to leave the image size unchanged.

These four arguments may be given alone, or in combination with the others as long as -xscale is not given with -xsize, and -yscale is not given with -ysize. -scale cannot be used in combination with any of the other scale and size arguments.

The algorithm used to resize the image may be controlled by:

-bilinear	Use bilinear interpolation
-pixelrep	Use pixel replication

If no algorithm is selected, imscale defaults to bilinear interpolation.

NOTES

Increasing Image Size

Increasing the size of an image increases the number of pixels in the image. The colors used by the new additional pixels are determined by the resolution change algorithm chosen.

-pixelrep performs the pixel replication, a simple form of resolution change. Using pixel replication, doubling an image's size fills in new pixels by copying, or replicating, adjacent pixels. For instance, an input image has the following scanline:

A C E G I K M

Each letter is a different color in the image. When this image is doubled in size using pixel replication, each pixel on the scanline becomes two pixels, both with the same color. The resized scanline will be:

A A C C E E G G I I K K M M

This has an effect similar to looking at the image with a magnifying glass. Small pixels become big pixels; small jaggies become big jaggies.

-pixelrep may only be used when the new image width and height are each multiples of the input image width and height. If they are not, imscale will exit with an error.

-bilinear performs bilinear interpolation to increase the image size. This method resamples the input image by overlaying atop the input image a mathematical grid with width x height points. This gives one grid point for each pixel in the output image. The input image is then sampled at each grid point to determine the output image's pixel colors. Where grid points lie directly over the center of a source pixel, the grid point's color is that of the source pixel. Where grid points lie between input pixel centers, the grid point's color is determined by linearly interpolating between adjacent pixel colors (vertically and horizontally). For instance, an input image has the following scanline:

A C E G I K M

Each letter is a different color in the image. To double this image's size, a mathematical grid will be overlaid. Gridpoints will fall on the center of pixel 'A,' halfway between pixel 'A' and 'C,' on the center of pixel 'C,' and so on. Output image pixels for gridpoints centered on 'A,' 'C,' 'E,' etc., will have those pixel's colors. Output image pixels for gridpoints in-between 'A,' 'C,' 'E,' etc., will have colors interpolated between 'A' and 'C,' 'C' and 'E,' and so on. The resized scanline will be:

A B C D E F G H I J K L M N

When using bilinear interpolation, the larger resized image will appear blurry. Hard edges become softer and jaggies less obvious. Bilinear interpolation is the default if neither -bilinear or -pixelrep options are given.

-bilinear may be used for scaling an image up by any arbitrary scaling factor. It is not limited to multiples of the source image size, as is -pixelrep.

Decreasing Image Size

Decreasing the size of an image decreases the number of pixels in the image. The choice of how to color the pixels that remain depends upon the resizing algorithm chosen.

-pixelrep is only applicable to increasing image size. The algorithm may not be used when the image size is being decreased. imscale will exit with an error if -pixelrep is used when decreasing an image's size.

-bilinear may be used to decrease an image's size using bilinear interpolation. Pixel values in the resulting smaller image will be determined by interpolating between pixel values in the input image in much the same way used when increasing an image's size.

Bilinear interpolation is the default if neither `-bilinear` or `-pixelrep` options are given.

Misc

Image scaling can take awhile, depending upon the size of the input image, the size of the output image, and the resizing algorithm used.

For notes regarding file format conversion and standard image tool options, see the man page for `imconv(1IM)`.

Error messages are reported to `stderr`.

EXAMPLES

To scale a 640x480 image up to 1280x1024 (non-uniform scaling) using bilinear interpolation, use any of the following:

```
imscale small.pix -xscale 2.0 -yscale 2.133 large.pix
or
imscale small.pix -xsize 1280 -ysize 1024 large.pix
or
imscale small.pix -xsize 1280 -ysize 1024 -bilinear large.pix
```

To scale a 640x480 image up to 1280x960 (uniform scaling) using pixel replication, use:

```
imscale small.pix -scale 2.0 -pixelrep large.pix
or
imscale small.pix -xscale 2.0 -yscale 2.0 -pixelrep large.pix
```

To scale the same image up using bilinear interpolation instead, use any of the following:

```
imscale small.pix -scale 2.0 -bilinear large.pix
or
imscale small.pix -scale 2.0 large.pix
or
imscale small.pix -xscale 2.0 -yscale 2.0 -bilinear large.pix
or
imscale small.pix -xscale 2.0 -yscale 2.0 large.pix
```

To reduce an image to 1/3 its original size in X, and scale it up or down to 500 pixels high in Y, using bilinear interpolation, use the following:

```
imscale original.rla -xscale 0.333 -ysize 500 sized.tiff
```

To copy an entire input GIF image to a PCX file, without scaling, use the following:

```
imscale incoming.gif outgoing.pcx
```

With no scaling arguments, the actions of `imscale` default to the same as the SDSC image format conversion tool `imconv(1IM)`, but they take a little longer.

SEE ALSO

`imcopy (1IM)`, `imflip (1IM)`, `imrotate (1IM)`, `imroll (1IM)`, `imshear (1IM)`, `ImVfbResize (3IM)`

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

AUTHOR

Dave Nadeau, Chris Groening, and Toan-Vinh Le
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