# Pamperspective User Manual

**pamperspective**

Updated: 2 September 2004
Table Of Contents

## NAME

pamperspective - a reverse scanline renderer for Netpbm images

## SYNOPSIS

**pamperspective**
[**--bottom_margin=***num*]
[**--detail=***num*]
[**--frame_include=***bool*]
[**--height=***num*]
[**--include=**[*x1*,*y1*;*x2*,*y2*; ...]]
[**--input_system=***spec*]
[**--input_unit=***spec*]
[**--interpolation=***spec*]
[**--left_margin=***num*]
[**--margin=***num*]
[**--output_system=***spec*]
[**--proportion=***spec*]
[**--ratio=***num*]
[**--right_margin=***num*]
[**--top_margin=***num*]
[**--width=***num*]
{
{
*upper_left_x upper_left_y upper_right_x upper_right_y*
*lower_left_x lower_left_y lower_right_x lower_right_y*
}
|
{
{**--upper_left_x**|**--ulx**}=*upper_left_x*
{**--upper_left_y**|**--uly**}=*upper_left_y*
{**--upper_right_x**|**--urx**}=*upper_right_x*
{**--upper_right_y**|**--ury**}=*upper_right_y*
{**--lower_left_x**|**--llx**}=*lower_left_x*
{**--lower_left_y**|**--lly**}=*lower_left_y*
{**--lower_right_x**|**--lrx**}=*lower_right_x*
{**--lower_right_y**|**--lry**}=*lower_right_y*
}
}
[*infile*]

<?makeman .SH OPTION USAGE ?>

Minimum unique abbreviation of option is acceptable. (But note that shortest unique prefixes might be longer in future versions of the program.) You may use single hyphens instead of double hyphen to denote options. You may use white space in place of the equals sign to separate an option name from its value. All options starting with hyphens may be given in any order.

## DESCRIPTION

This program is part of Netpbm. **pamperspective** reads a Netpbm image as input and produces a Netpbm image of the same format as output.

The values *upper_left_x ... lower_right_y*, that are required on the command line, specify a quadrilateral in the input image. **pamperspective** interprets the input image as a perspective projection of a three dimensional scene, for example a photograph. **pamperspective** assumes the specified quadrilateral represents a parallelogram in that scene. The output image consists of this parallelogram, sheared to a rectangle. In this way **pamperspective** undoes the effect of a raytracer or scanline renderer.

The input is from *infile*, or from Standard Input, if *infile* is not specified. The output is to Standard Output.

## OPTIONS

For options of the form **--name=***num*, You can specify the value *num* in any of the traditional ways. Additionally, you can specify it as *num1/num2*, where *num1* and *num2* are specified traditionally. This is useful for specifying a width/height ratio of 4/3, without having to write infinitely many digits. Where *num* is supposed to be a natural number, **pamperspective** does not allow this format.

### Quadrilateral specification options

**--upper_left_x=***num*

**--ulx=***num*This specifies the horizontal coordinate of the upper left vertex of the quadrilateral. The meaning of 'upper left' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--upper_left_y=***num*

**--uly=***num*This specifies the vertical coordinate of the upper left vertex of the quadrilateral. The meaning of 'upper left' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--upper_right_x=***num*

**--urx=***num*This specifies the horizontal coordinate of the upper right vertex of the quadrilateral. The meaning of 'upper right' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--upper_right_y=***num*

**--ury=***num*This specifies the vertical coordinate of the upper right vertex of the quadrilateral. The meaning of 'upper right' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--lower_left_x=***num*

**--llx=***num*This specifies the horizontal coordinate of the lower left vertex of the quadrilateral. The meaning of 'lower left' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--lower_left_y=***num*

**--lly=***num*This specifies the vertical coordinate of the lower left vertex of the quadrilateral. The meaning of 'lower left' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--lower_right_x=***num*

**--lrx=***num*This specifies the horizontal coordinate of the lower right vertex of the quadrilateral. The meaning of 'lower right' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--lower_right_y=***num*

**--lry=***num*This specifies the vertical coordinate of the lower right vertex of the quadrilateral. The meaning of 'lower right' is relative to the output image. The interpretation of *num* depends on the values for **--input_system** and **--input_unit**.

**--input_system=***system*

**--input_unit=***unit*The input image consists of pixels, which are, from the point of view of a scanline renderer, solid squares. These options specify how the coordinates are interpreted:

*system*=**lattice**, *unit*=**image**(0,0) refers to the upper left corner of the upper left pixel and (1,1) refers to the lower right corner of the lower right pixel.

*system*=**lattice**, *unit*=**pixel**(0,0) refers to the upper left corner of the upper left pixel and (*width*,*height*) refers to the lower right corner of the lower right pixel. Here *width* and *height* are the width and height of the input image.

*system*=**pixel**, *unit*=**image**(0,0) refers to the centre of the upper left pixel and (1,1) refers to the centre of the lower right pixel.

*system*=**pixel**, *unit*=**pixel**(0,0) refers to the centre of the upper left pixel and (*width*-1,*height*-1) refers to the centre of the lower right pixel. Here *width* and *height* are the width and height of the input image.

The defaults are **--input_system=lattice** and **--input_unit=pixel**. Point-and-click front ends should use **--input_system=pixel**.

### Frame options

By default **pamperspective** outputs exactly the above parallelogram, sheared to a rectangle. With the following options, it is possible to make **pamperspective** output a larger or smaller portion, which we call the "visible part." We refer to the default rectangle as the "frame." The visible part is always a rectangle the axes of which are parallel to those of the frame.

The frame options are additive. All the parts of the image specified by either margin options, **--include_frame**, or **--include** (or their defaults) are in the visible part. The visible part is the smallest possible rectangle that contains the parts specified those three ways.

The visible part must have nonzero size. That means if you specify **--frame-include=no** (overriding the default), you'll need to specify other frame options in order to have something in the visible part.

[**--margin=***num*]This specifies an area surrounding the frame that is to be included in the visible part. The units of *num* are the width of the frame for the horizontal extensions and the height of the frame for vertical extensions.

For example, **--margin=1** makes the visible part 9 times as large, because it makes the visible part extend one frame's worth to the left of the frame, one frame's worth to the right, one frame's worth above the frame, and one frame's worth below the frame, for a total of 3 frames' worth in both dimensions.

A negative value has an effect only if you specify **--frame_include=no**. The default is no margin.

The individual margin options below override this common margin setting.

[**--top_margin=***num*]

[**--left_margin=***num*]

[**--right_margin=***num*]

[**--bottom_margin=***num*]These are like **--margin**, but they specify only one of the 4 sides. The default value for each is the value (or default) of **--margin**.

[**--frame_include=***bool*]Valid values for *bool* are:

**yes**

**true**

**on** The frame itself is in the visible part.

**no**

**false**

**off** The frame itself is not necessarily in the visible part (but it could be if other options cause it to be).

The default value is **yes**

**--include=**[*x1*,*y1*;*x2*,*y2*; ...]The visible part is made large enough such that every point (*x1*,*y1*), (*x2*,*y2*), of the input image is visible. The meaning of *x* and *y* is determined by **--input_system** and **--input_unit**. You can specify any number of semicolon-delimited points, including zero.

If you're supplying these options via a Unix command shell, be sure to use proper quoting, because semicolon (**;**) is usually a shell control character.

The frame options were new in Netpbm 10.25 (October 2004).

### Output size options

**--width=***width*

**--height=***height*These specify the size of the output image in horizontal and vertical direction. The values are numbers of pixels, so only natural numbers are valid. These values override the default means to determine the output size.

**--detail=***num*If you do not specify **--width**, **pamperspective** determines the width of the output image such that moving *num* output pixels horizontally does not change the corresponding pixel coordinates of the input image by more than 1. **pamperspective** determines the height of the output image analogously. The default value is 1.

**--proportion=***prop*

**--ratio=***ratio*Valid values for *prop* are:

**free** In this case **--ratio** does not have any effect.

**fixed**After the width and height are determinedaccording to **--detail**, one of both will be increased, in order to obtain width/height=*ratio*.

The defaults are **--proportion=free** and **--ratio**=1.

### Output options

**--output_system=***spec*The output image consists of pixels, which are, from the point of view of a scanline renderer, solid squares. This option specifies how the four vertices of the quadrilateral correspond to the pixels of the output image. Valid values for *spec* are:

**lattice**The upper left vertex corresponds to the upper left corner of the upper left pixel and The lower right vertex corresponds to the lower right corner of the lower right pixel.

**pixel** The upper left vertex corresponds to the centre of the upper left pixel and The lower right vertex corresponds to the centre of the lower right pixel.

The default value is **lattice**. Point-and-click front ends should use **pixel**.

**--interpolation=***spec*Usually (centres of) output pixels do not exactly correspond to (centres of) input pixels. This option determines how the program will choose the new pixels. Valid values for *spec* are:

**nearest**The output pixel will be identical to the nearest input pixel.

**linear**The output pixel will be a bilinear interpolation of the four surrounding input pixels.

The default value is **nearest**.

## HINTS

It might be tempting always to use the options **--include 0,0;0,1;1,0;1,1** (assuming **--input_system=lattice** and **--input_unit=image**), so that no part of the input image is missing in the output. There are problems with that:

> If the threedimensional plane defined by the quadrilateral has a visible horizon in the input image, then the above asks **pamperspective** to include points that cannot ever be part of the output.
>
> If the horizon is not visible, but close to the border of the input image, this may result in very large output files. Consider a picture of a road. If you ask a point close to the horizon to be included, then this point is far away from the viewer. The output will cover many kilometers of road, while **--detail** perhaps makes a pixel equal to a square centimeter.

When working with large files **pamperspective**'s memory usage might be an issue. In order to keep it small, you should minimize each of the following:

> The vertical range that the top output line consumes in the input image;
>
> The vertical range that the bottom output line consumes in the input image;
>
> The vertical range from the topmost (with respect to the input image) quadrilateral point to the top (with respect to the output image) output line.

For this purpose you can use **pamflip** before and/or after **pamperspective**. Example: Instead of

**pamperspective 10 0 100 50 0 20 95 100 infile** > **outfile**

you can use

**pamflip -rotate90 infile** |<br>
  **pamperspective 50 0 100 5 0 90 20 100** |<br>
  **pamflip -rotate270** > **outfile**

## SEE ALSO

**netpbm**, **pam**, **pnm**, **pamcut**, **pamflip**, **pnmrotate**, **pamscale**, **pnmshear**, **pnmstitch**

## HISTORY

Mark Weyer wrote **pamperspective** in March 2004.

It was new in Netpbm 10.22 (April 2004).

## AUTHOR



## Table Of Contents