Pamcomp User Manual

pamcomp

Updated: 17 July 2004
Table Of Contents

NAME

pamcomp - composite (overlay) two Netpbm images together

SYNOPSIS

pamcomp [**-align=**{**left**|**center**|**right**| **beyondleft**|**beyondright**}]
[**-valign=**{**top**|**middle**|**bottom**| **above**|**below**}]
[**-xoff=***X*] [**-yoff=***Y*]
[**-alpha=***alpha-pgmfile*] [**-invert**] [**-opacity=***opacity*] [**-linear**]
overlay_file [*underlying_file* [*output_file*]]

Minimum unique abbreviation of option is acceptable. You may use double hyphens instead of single hyphen to denote options. You may use white space in place of the equals sign to separate an option name from its value.

DESCRIPTION

This program is part of Netpbm. **pamcomp** reads two images and produces a composite image with one of the images overlayed on top of the other, possible translucently. The images need not be the same size. The input and outputs are Netpbm format image files.

In its simplest use, **pamcomp** simply places the image in the file *overlay_file* on top of the image in the file *underlying_file*, blocking out the part of *underlying_file* beneath it.

If you add the **-alpha** option, then **pamcomp** uses the image in file *alpha-pgmfile* as an alpha mask, which means it determines the level of transparency of each point in the overlay image. The alpha mask must have the same dimensions as the overlay image. In places where the alpha mask defines the overlay image to be opaque, the composite output contains only the contents of the overlay image; the underlying image is totally blocked out. In places where the alpha mask defines the overlay image to be transparent, the composite output contains none of the overlay image; the underlying image shows through completely. In places where the alpha mask shows a value in between opaque and transparent (translucence), the composite image contains a mixture of the overlay image and the underlying image and the level of translucence determines how much of each.

The alpha mask is a PGM file in which a white pixel represents opaqueness and a black pixel transparency. Anything in between is translucent. (Like any Netpbm program, **pamcomp** will see a PBM file as if it is PGM).

If the overlay image is a PAM image of tuple type RGB_ALPHA or GRAYSCALE_ALPHA, then the overlay image contains transparency information itself and **pamcomp** uses it the same way as the alpha mask described above. If you supply both an overlay image that has transparency information and an alpha mask, **pamcomp** multiplies the two opacities to get the opacity of the overlay pixel.

Before Netpbm 10.25 (October 2004), **pamcomp** did not recognize the transparency information in a PAM image -- it just ignored it. So people had to make appropriate alpha masks in order to have a non-opaque overlay. Some Netpbm programs that convert from image formats such as PNG that contain transparency

information are not able to create RGB_ALPHA or GRAYSCALE_ALPHA PAM output, so you have to use the old method -- extract the transparency information from the original into a separate alpha mask and use that as input to **pamcomp**.

The output image is always of the same dimensions as the underlying image. **pamcomp** uses only parts of the overlay image that fit within the underlying image.

To specify where on the underlying image to place the overlay image, use the **-align**, **-valign**, **-xoff**, and **-yoff** options. Without these options, the default horizontal position is flush left and the default vertical position is flush top.

The overlay image, in the position you specify, need not fit entirely within the underlying image. **pamcomp** uses only the parts of the overlay image that appear above the underlying image. It is possible to specify positioning such that none of the overlay image is over the underlying image -- i.e. the overlay is out of frame. If you do that, **pamcomp** issues a warning.

The overlay and underlying images may be of different formats (e.g. overlaying a PBM text image over a full color PPM image) and have different maxvals. The output image has the more general of the two input formats and a maxval that is the least common multiple the two maxvals (or the maximum maxval allowable by the format, if the LCM is more than that).

OPTIONS

-align=*alignment*This option selects the basic horizontal position of the overlay image with respect to the underlying image, in syntax reminiscent of HTML. **left** means flush left, **center** means centered, and **right** means flush right.

The **-xoff** option modifies this position. **beyondleft** means just out of frame to the left -- the right edge of the overlay is flush with the left edge of the underlying image. **beyondright** means just out of frame to the right. These alignments are useful only if you add a **-xoff** option. These two values were added in Netpbm 10.10 (October 2002).

The default is **left**.

-valign=*alignment*This option selects the basic vertical position of the overlay image with respect to the underlying image, in syntax reminiscent of HTML. **top** means flush top, **middle** means centered, and **bottom** means flush bottom.

The **-yoff** option modifies this position. **above** means just out of frame to the top -- the bottom edge of the overlay is flush with the top edge of the underlying image. **below** means just out of frame to the bottom. These alignments are useful only if you add a **-yoff** option. These two values were added in Netpbm 10.10 (October 2002).

The default is **top**.

-xoff=*x*This option modifies the horizontal positioning of the overlay image with respect to the underlying image as selected by the **-align** option. **pamcomp** shifts the overlay image from that basic position *x* pixels to the right. *x* can be negative to indicate shifting to the left.

The overlay need not fit entirely (or at all) on the underlying image. **pamcomp** uses only the parts that lie over the underlying image.

Before Netpbm 10.10 (October 2002), **-xoff** was mutually exclusive with **-align** and always measured from the left edge.

-yoff=*y*This option modifies the vertical positioning of the overlay image with respect to the underlying image as selected by the **-valign** option. **pamcomp** shifts the overlay image from that basic position *y* pixels downward. *y* can be negative to indicate shifting upward.

The overlay need not fit entirely (or at all) on the underlying image. **pamcomp** uses only the parts that lie over the underlying image.

Before Netpbm 10.10 (October 2002), **-xoff** was mutually exclusive with **-valign** and always measured from the top edge.

-alpha=*alpha-pgmfile*This option names a file that contains the alpha mask. If you don't specify this

option, there is no alpha mask, which is equivalent to having an alpha mask specify total opaqueness everywhere.

You can specify **-** as the value of this option and the alpha mask will come from Standard Input. If you do this, don't specify Standard Input as the source of any other input image.

-invert This option inverts the sense of the values in the alpha mask, which effectively switches the roles of the overlay image and the underlying image in places where the two intersect.

-opacity=*opacity*This option tells how opaque the overlay image is to be, i.e. how much of the composite image should be from the overlay image, as opposed to the underlying image. *opacity* is a floating point number, with 1.0 meaning the overlay image is totally opaque and 0.0 meaning it is totally transparent. The default is 1.0.

If you specify an alpha mask (the **-alpha** option), **pamcomp** uses the product of the opacity indicated by the alpha mask (as modified by the **-invert** option, as a fraction, and this opacity value. The **-invert** option does not apply to this opacity value.

As a simple opacity value, the value makes sense only if it is between 0 and 1, inclusive. However, **pamcomp** accepts all values and performs the same arithmetic computation using whatever value you provide. An opacity value less than zero means the underlay image is intensified and then the overlay image is "subtracted" from it. An opacity value greater than unity means the overlay image is intensified and the underlaying image subtracted from it. In either case, **pamcomp** clips the resulting color component intensities so they are nonnegative and don't exceed the output image's maxval.

This may seem like a strange thing to do, but it has uses. You can use it to brighten or darken or saturate or desaturate areas of the underlaying image. See this description of the technique.

This option was added in Netpbm 10.6 (July 2002). Before Netpbm 10.15 (April 2003), values less than zero or greater than unity were not allowed.

-linear This option indicates that the inputs are not true Netpbm images but rather a non-gamma-adjusted variation. This is relevant only when you mix pixels, using the **-opacity** option or an alpha mask (the **-alpha** option).

The alpha mask and **-opacity** values indicate a fraction of the light intensity of a pixel. But the PNM and PNM-equivalent PAM image formats represent intensities with gamma-adjusted numbers that are not linearly proportional to intensity. So **pamcomp**, by default, performs a calculation on each sample read from its input and each sample written to its output to convert between these gamma-adjusted numbers and internal intensity-proportional numbers.

Sometimes you are not working with true PNM or PAM images, but rather a variation in which the sample values are in fact directly proportional to intensity. If so, use the **-linear** option to tell **pamcomp** this. **pamcomp** then will skip the conversions.

The conversion takes time. And the difference between intensity-proportional values and gamma-adjusted values may be small enough that you would barely see a difference in the result if you just pretended that the gamma-adjusted values were in fact intensity-proportional. So just to save time, at the expense of some image quality, you can specify **-linear** even when you have true PPM input and expect true PPM output.

For the first 13 years of Netpbm's life, until Netpbm 10.20 (January 2004), **pamcomp**'s predecessor **pnmcomp** always treated the PPM samples as intensity-proportional even though they were not, and drew few complaints. So using **-linear** as a lie is a reasonable thing to do if speed is important to you.

Another technique to consider is to convert your PNM image to the linear variation with **pnmgamma**, run **pamcomp** on it and other transformations that like linear PNM, and then convert it back to true PNM with **pnmgamma -ungamma**. **pnmgamma** is often faster than **pamcomp** in doing the conversion.

SEE ALSO

ppmmix and **pnmpaste** are simpler, less general versions of the same tool.

ppmcolormask and **pbmmask** can help with generating an alpha mask.

pnmcomp is an older program that runs faster, but has less function.

pnm

HISTORY

pamcomp was new in Netpbm 10.21 (March 2004). Its predecessor, **pnmcomp**, was one of the first programs added to Netpbm when the project went global in 1993.

AUTHOR

Copyright (C) 1992 by David Koblas (koblas@mips.com).

Table Of Contents

SYNOPSIS
DESCRIPTION
OPTIONS
SEE ALSO
HISTORY
AUTHOR