NAME

pam - Netpbm common 2-dimensional bitmap format

GENERAL

The PAM image format is a lowest common denominator 2 dimensional map format.

It is designed to be used for any of myriad kinds of graphics, but can theoretically be used for any kind of data that is arranged as a two dimensional rectangular array. Actually, from another perspective it can be seen as a format for data arranged as a three dimensional array.

The name 'PAM' is an acronym derived from 'Portable Arbitrary Map.' This derivation makes more sense if you consider it in the context of the other Netpbm format names: PBM, PGM, and PPM.

This format does not define the meaning of the data at any particular point in the array. It could be red, green, and blue light intensities such that the array represents a visual image, or it could be the same red, green, and blue components plus a transparency component, or it could contain annual rainfalls for places on the surface of the Earth. Any process that uses the PAM format must further define the format to specify the meanings of the data.

A PAM image describes a two dimensional grid of tuples. The tuples are arranged in rows and columns. The width of the image is the number of columns. The height of the image is the number of rows. All rows are the same width and all columns are the same height. The tuples may have any degree, but all tuples have the same degree. The degree of the tuples is called the depth of the image. Each member of a tuple is called a sample. A sample is an unsigned integer which represents a locus along a scale which starts at zero and ends at a certain maximum value greater than zero called the maxval. The maxval is the same for every sample in the image. The two dimensional array of all the Nth samples of each tuple is called the Nth plane or Nth channel of the image.

Though the basic format does not assign any meaning to the tuple values, it does include an optional string that describes that meaning. The contents of this string, called the tuple type, are arbitrary from the point of view of the basic PAM format, but users of the format may assign meaning to it by convention so they can identify their particular implementations of the PAM format. Some tuple types are defined as official sub-formats of PAM. See Defined Tuple Types .

The Confusing Universe of Netpbm Formats

It is easy to get confused about the relationship between the PAM format and PBM, PGM, PPM, and PNM. Here is a little enlightenment:

"PNM" is not really a format. It is a shorthand for the PBM, PGM, and PPM formats collectively. It is also the name of a group of library functions that can each handle all three of those formats.

'PAM' is in fact a fourth format. But it is so general that you can represent the same information in a PAM image as you can in a PBM, PGM, or PPM image. And in fact a program that is designed to read PBM, PGM, or PPM and does so with a recent version of the Netpbm library, will read an equivalent PAM image just fine and the program will never know the difference.

To confuse things more, there is a collection of library routines called the 'pam' functions that read and write the PAM format, but also read and write the PBM, PGM, and PPM formats. They do this because the latter formats are much older and more popular, so even a new program must work with them. Having the library handle all the formats makes it convenient to write programs that use the newer PAM format as well.

THE LAYOUT

A convenient way to read and write the PAM format accurately is via the **libnetpbm**(1)**C**subroutine**library.**

A PAM file consists of a sequence of one or more PAM images. There are no data, delimiters, or padding before, after, or between images.

Each PAM image consists of a header followed immediately by a raster.

Here is an example header:

P7 WIDTH 227 HEIGHT 149 DEPTH 3 MAXVAL 255 TUPLTYPE RGB ENDHDR

The header begins with the ASCII characters 'P7' followed by newline. This is the magic number.

Note: **xv** thumbnail images also start with the "P7" magic number. (This and PAM were independent extensions to the Netpbm formats). The rest of the format makes it easy to distinguish PAM from that format, though).

The header continues with an arbitrary number of lines of ASCII text. Each line ends with and is delimited by a newline character.

Each header line consists of zero or more whitespace-delimited tokens or begins with '#'. If it begins with '#' it is a comment and the rest of this specification does not apply to it.

A header line which has zero tokens is valid but has no meaning.

The type of header line is identified by its first token, which is 8 characters or less:

ENDHDR
: This is the last line in the header. The header must contain exactly one of these header lines.

HEIGHT
: The second token is a decimal number representing the height of the image (number of rows). The header must contain exactly one of these header lines.

WIDTH
: The second token is a decimal number representing the width of the image (number of columns). The header must contain exactly one of these header lines.

DEPTH
: The second token is a decimal number representing the depth of the image (number of planes or channels). The header must contain exactly one of these header lines.

MAXVAL
: The second token is a decimal number representing the maxval of the image. The header must contain exactly one of these header lines.

TUPLTYPE
: The header may contain any number of these header lines, including zero. The rest of the line is part of the tuple type. The rest of the line is not tokenized, but the tuple type does not include any white space immediately following **TUPLTYPE** or at the very end of the line. It does not include a newline. There must be something other than white space after the **TUPLTYPE** token.

  If there are multiple **TUPLTYPE** header lines, the tuple type is the concatenation of the values from each of them, separated by a single blank, in the order in which they appear in the header. If there are no **TUPLTYPE** header lines the tuple type is the null string.

The raster consists of each row of the image, in order from top to bottom, consecutive with no delimiter of any kind between, before, or after, rows.

Each row consists of every tuple in the row, in order from left to right, consecutive with no delimiter of any kind between, before, or after, tuples.

Each tuple consists of every sample in the tuple, in order, consecutive with no delimiter of any kind

between, before, or after, samples.

Each sample consists of an unsigned integer in pure binary format, with the most significant byte first. The number of bytes is the minimum number of bytes required to represent the maxval of the image.

The character referred to as 'newline' herein is the character known in ASCII as Line Feed or LF.

LIMITATIONS

The maxval of an image is never greater than 65535. (The reason it is limited is to make it easier to build an image processor, in which intermediate arithmetic values often have to fit within 31 or 32 bits). There was no specified limitation before October, 2005, but essentially all implementations have always observed it.

Height and width are at least 1.

Height and width have no defined maximum, but processors and generators of images usually have their own limitations.

DEFINED TUPLE TYPES

Some tuple types are defined in this specification to specify official subformats of PAM for especially popular applications of the format. Users of the format may also define their own tuple types, and thus their own subformats.

PAM Used For Visual Images

A common use of PAM images is to represent visual images such as are typically represented by images in the older and more concrete PBM, PGM, and PPM formats.

Black And White (PBM)

A black and white image, such as would be represented by a PBM image, has a tuple type of "BLACKANDWHITE". Such a PAM image has a depth of 1 and maxval 1 where the one sample in each tuple is 0 to represent a black pixel and 1 to represent a white one. The height, width, and raster bear the obvious relationship to those of the equivalent PBM image.

Note that in the PBM format, a zero value means white, but in PAM, zero means black.

Grayscale (PGM)

A grayscale image, such as would be represented by a PGM image, has a tuple type of "GRAYSCALE". Such a PAM image has a depth of 1. The maxval, height, width, and raster bear the obvious relationship to those of the equivalent PGM image.

Color (PPM)

A color image, such as would be represented by a PPM image, has a typle type of "RGB". Such a PAM image has a depth of 3. The maxval, height, width, and raster bear the obvious relationship to those of the PPM image. The first plane represents red, the second blue, and the third green.

Transparent

Each of the visual image formats mentioned above has a variation that contains transparency information. In that variation, the tuple type has '_ALPHA' added to it (e.g. 'RGB_ALPHA') and one more plane. The highest numbered plane is the opacity plane (sometimes called an alpha plane or transparency plane).

In this kind of image, the color represented by a pixel is actually a combination of an explicitly specified foreground color and a background color to be identified later.

The planes other than the opacity plane describe the foreground color. A sample in the opacity plane tells how opaque the pixel is, by telling what fraction of the pixel's light comes from the foreground color. The rest of the pixel's light comes from the (unspecified) background color.

For example, in a GRAYSCALE_ALPHA image, assume Plane 0 indicates a gray tone 60% of white and Plane 1 indicates opacity 25%. The foreground color is the 60% gray, and 25% of that contributes to the ultimate color of the pixel. The other 75% comes from some background color. So let's assume further that the background color of the pixel is full white. Then the color of the pixel is 90% of white: 25% of the foreground 60%, plus 75% of the background 100%.

The sample value is the opacity fraction just described, as a fraction of the maxval. Note that it is *not* gamma-adjusted like the foreground color samples.

SEE ALSO

Netpbm(1), **pbm**(1), **pgm**(1), **ppm**(1), **pnm**(1), **libnetpbm**(1)