Pamlookup User Manual

pamlookup

Updated: 10 November 2002
Table Of Contents

NAME

pamlookup - map an image to a new image by using it as indices into a table

SYNOPSIS

pamlookup -lookupfile=*lookupfile* **-missingcolor=***color* [**-fit**] *indexfile*

All options can be abbreviated to their shortest unique prefix. You may use two hyphens instead of one. You may separate an option name and its value with white space instead of an equals sign.

DESCRIPTION

This program is part of Netpbm.

pamlookup takes a two dimensional array of indices and a lookup table as input. For each position in the index array, it looks up the index in the lookup table and places the result of the lookup in the output image. The output thus has the same width and height as the index image, and tuple types determined by the lookup table.

An index is either a whole number or an ordered pair of whole numbers. If the index image has a depth of one, each index in it is a whole number: the value of the one sample. If the index image has a depth greater than one, each index in it is an ordered pair of the first and second samples in the relevant tuple.

The lookup table is a PAM or PNM image. If the index image contains whole number indices, the lookup image is a single row and the index is a column number. The lookup result is the value of the tuple or pixel at the indicated column in the one row in the lookup table. If the index image contains ordered pair indices, the first element of the ordered pair is a row number and the second element of the ordered pair is a column number. The lookup result is the value of the tuple or pixel at the indicated row and column in the lookup table.

For example: Consider an index image consisting of a 3x2x1 PAM as follows:

center ; <?makeman l l l. ?> <tr>
<td> 0</td>
<td> 1</td>
<td> 0</td>
</tr>

<tr>
<td> 2</td>
<td> 2</td>
<td> 2</td>
</tr>

and a lookup table consisting of a 3x1 PPM image as follows:

```
center ; <?makeman l l l. ?> <tr>
 <td> red</td>
 <td> yellow</td>
 <td> beige</td>
 </tr>
```

The lookup table above says Index 0 corresponds to the color red, Index 1 corresponds to yellow, and Index 2 corresponds to beige. The output of **pamlookup** is the following PPM image:

```
center ; <?makeman l l l. ?> <tr>
 <td> red</td>
 <td> yellow</td>
 <td> red</td>
 </tr>

<tr>
 <td> beige</td>
 <td> beige</td>
 <td> beige</td>
 </tr>
```

Now let's look at an example of the more complex case where the indices are ordered pairs of whole numbers instead of whole numbers. Our index image will be this 3x2x2 PAM image:

```
center ; <?makeman l l l. ?> <tr>
 <td> (0,0)</td>
 <td> (0,1)</td>
 <td> (0,0)</td>
 </tr>

<tr>
 <td> (1,1)</td>
 <td> (1,0)</td>
 <td> (0,0)</td>
 </tr>
```

Our lookup table for the example will be this two dimensional PPM:

```
center ; <?makeman l l l. ?> <tr>
 <td> red</td>
   <td> yellow</td>
 </tr>

<tr>
 <td> green</td>
 <td> black</td>
 </tr>
```

This lookup table says Index (0,0) corresponds to the color red, Index (0,1) corresponds to yellow, Index (1,0) corresponds to green, and Index (1,1) corresponds to black. The output of **pamlookup** is the following PPM image:

```
center ; <?makeman l l l. ?> <tr>
 <td> red</td>
```

```
  <td> yellow</td>
 <td> red</td>
 </tr>

<tr>
 <td> black</td>
 <td> green</td>
  <td> red</td>
 </tr>
```

If an index specifies a row or column that exceeds the dimensions of the lookup table image, **pamlookup** uses the value from the top left corner of the lookup image, or the value you specify with the **-missingcolor** option.

The *indexfile* argument identifies the file containing the index PAM or PNM image. **-** means Standard Input. The mandatory **-lookupfile** option identifies the file containing the lookup table image. Again, **-** means Standard Input. It won't work if both the index image file and lookup table file are Standard Input. The output image goes to Standard Output.

You can use **ppmmake** and **pnmcat** to create a lookup table file.

If you want to use two separate 1-plane images as indices (so that your output reflects the combination of both inputs), use **pamstack** to combine the two into one two-plane image (and use a 2-dimensional lookup table image).

OPTIONS

-lookupfile=*lookupfilelookupfile* names the file that contains the PAM or PNM image that is the lookup table. This option is mandatory.

-missingcolor=*color*This option is meaningful only if the lookup image (and therefore the output) is a PNM image. *color* specifies the color that is to go in the output wherever the index from the input is not present in the lookup table (not present means the index exceeds the dimensions of the lookup image -- e.g. index is 100 but the lookup image is a 50 x 1 PPM).

If you don't specify this option of **-fit**, **pamlookup** uses the value from the top left corner of the lookup image whenever an index exceeds the dimensions of the lookup image.

Specify the color (*color*) as described for the argument of the **ppm_parsecolor()** library routine.

Another way to deal with a too-small lookup image is to use the **-fit** option.

-fit This option says to shrink or expand the lookup image as necessary to fit the indices present in the index image, per the index image's maxval. For example, if your index image has a single plane and a maxval of 255 and your lookup image is 1 row of 10 columns, **pamlookup** stretches your lookup image to 255 columns before doing the lookups. **pamlookup** does the stretching (or shrinking) with the **pamscale** program.

When you use **-fit**, **pamlookup** never fails or warns you due to invalid lookup image dimensions, and the **-missingcolor** option has no effect.

EXAMPLES

Example: rainfall map

Say you have a set of rainfall data in a single plane PAM image. The rows and columns of the PAM indicate lattitude and longitude. The sample values are the annual rainfall in (whole) centimeters. The highest rainfall value in the image is 199 centimeters. The image is in the file rainfall.pam.

You want to produce a PPM rainfall map with green for the wettest places, red for the driest, and other colors in between.

First, compose a lookup table image, probably with a graphical editor and the image blown way up so you can work with individual pixels. The image must have a single row and 200 columns. Make the leftmost pixel red and the rightmost pixel green and choose appropriate colors in between. Call it colorkey.ppm.

```
pamlookup rainfall.ppm -lookupfile=colorkey.ppm >rainfallmap.ppm
```

Now lets say you're too lazy to type in 200 color values and nobody really cares about the places that have more than 99 centimeters of annual rainfall. In that case, just make colorkey.ppm 100 columns wide and do this:

```
pamlookup rainfall.ppm -lookupfile=colorkey.ppm -missingcolor=black         >rainfallma
```

Now if there are areas that get more than 100 centimeters of rainfall, they will just show up black in the output.

Example: graphical diff

Say you want to compare two PBM (black and white) images visually. Each consists of black foreground pixels on a white background. You want to create an image that contains background where both images contain background and foreground where both images contain foreground. But where Image 1 has a foreground pixel and Image 2 does not, you want red in the output; where Image 2 has a foreground pixel and Image 1 does not, you want green.

First, we create a single image that contains the information from both input PBMs:

```
pamstack image1.pbm image2.pbm >bothimages.pam
```

Note that this image has 1 of 4 possible tuple values at each location: (0,0), (0,1), (1,0), or (1,1).

Now, we create a lookup table that we can index with those 4 values:

```
ppmmake white 1 1 >white.ppm
ppmmake black 1 1 >black.ppm
ppmmake red   1 1 >red.ppm
ppmmake green 1 1 >green.ppm
pnmcat -leftright black.ppm red.ppm   >blackred.ppm
pnmcat -leftright green.ppm white.ppm >greenwhite.ppm
pnmcat -topbottom blackred.ppm greenwhite.ppm >lookup.ppm
```

Finally, we look up the indices from our index in our lookup table and produce the output:

```
pamlookup bothimages.ppm -lookupfile=lookup.ppm >imagediff.ppm
```

SEE ALSO

pnmremap, ppmmake, pnmcat, pamstack, pnm, pam

HISTORY

pamlookup was new in Netpbm 10.13 (December 2002).

Table Of Contents

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
EXAMPLES
HISTORY

SEE ALSO