

```

(* Pseudocoloring Ordinary Photographs to Highlight Grayscale Changes *)
(* Copyright May 2, 2006, Doug Youvan www.youvan.com & www.pseudocolor.com *)

(* Make, show,
and export linear grayscale list of 256 elements with rgb values ramped from 0 to
1 in increments of 1/256 to be appended to images outside of this program *)

graybar = Table[{j/256, j/256, j/256}, {i, 1, 1}, {j, 1, 256}];
gbar = Graphics[RasterArray[Apply[RGBColor, graybar, {2}]],
  ImageSize -> {256, 20}, AspectRatio -> Automatic];

Export["D:\\Mathematica\\Data\\gbar.jpg", gbar, ImageSize -> {256, 1}];
Show[gbar];

(* Make a pseudocolor 'hotmap' with 256 RGB elements *)

hotmap = Table[{
  Piecewise[{
    {1, 1 <= i < 42},
    {1, 42 <= i < 84},
    {Floor[5.8 * (i - 83)], 84 <= i < 128},
    {256, 128 <= i < 170},
    {256, 170 <= i < 212},
    {256, 212 <= i <= 256}
  ]},
  Piecewise[{
    {1, 1 <= i < 42},
    {(i - 41) * 6, 42 <= i < 84},
    {256, 84 <= i < 128},
    {256 - (i - 128) * 6, 128 <= i < 170},
    {1, 170 <= i < 212},
    {Floor[(i - 211) * 5.7], 212 <= i <= 256}
  ]},
  Piecewise[{
    {Floor[i * 5.8], 1 <= i < 42},
    {256 - ((i - 42) * 6), 42 <= i < 84},
    {1, 84 <= i < 128},
    {1, 128 <= i < 170},
    {(i - 169) * 6, 170 <= i < 212},
    {256, 212 <= i <= 256}
  ]}
  ],
  {i, 1, 256, 1}];

hotmap[[1, All]] = 1;
(* This changes 1,1,5 to 1,1,1 at i=1 for the blue channel *)

(* Export hotmap for other programs *)
Export["D:\\Mathematica\\Data\\hotmap.dat", hotmap];

```

```
In[8]:=
(* Import image with gbar appended *)
church = Import ["D:\\Mathematica\\Data\\gbar_church_256.jpg"];

(* Note Mathematica 'ImageSize' is given in {width, height} ,
unlike a matrix which is row-by-column;
note that a displayed matrix row 1 is the bottom row in an image *)

(* hardcoded size *)
Show[Graphics[church], ImageSize -> {368, 256}];

(* Convert the graphic 'church' to a 'List' and determine its width, height,
and depth, i.e., there are 3 RGB values so it is a rank three tensor *)

church = church /. Graphics -> List;
pv = church[[1, 1]]; (* extracts all the pixels' values *)
size = Dimensions[pv];
height = size[[1]];
width = size[[2]];
depth = size[[3]];

(* pv is an RGB tensor; make pv1, a matrix with only R values *)

pv1 = Table[{Part[pv, i, j, 1]}, {i, 1, height}, {j, 1, width}];

(* Flatten pv1 to a list of width* height single numbers *)

pv2 = Flatten[pv1];

(* index grayscale values into hotmap *)

numpixs = height*width;
pv3 = Table[{i}, {i, 1, numpixs}]; pv3[[All]] = {0, 0, 0};

(* +1 because some grays are zero and hotmap zero index is its Head,
not a triplet *)

For[i = 1, i <= numpixs, i++, pv3[[i, All]] = hotmap[[(pv2[[i]] + 1)]] / 256];

(* reform matrix *)
pv4 = Partition[pv3, width];

pseudochurch = Graphics[RasterArray[Apply[RGBColor, pv4, {2}]],
ImageSize -> {width, height}, AspectRatio -> Automatic];
Show[pseudochurch];

Export ["D:\\Mathematica\\Data\\pseudochurch.tif",
pseudochurch, ImageSize -> {height, width}];
Export ["D:\\Mathematica\\Data\\pseudochurch.jpg",
pseudochurch, ImageSize -> {height, width}];
```

