Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

Module 4E
Using Color in Computer Graphics

Using Color in Computer Graphics

- Human Factors Guidelines for Information Displays
- Many graphics systems suffer from poor color selections
  - More important when dealing with text and information display
    - As opposed to realistic rendering
- Several basic "human factors" rules for information displays
  - Use of color should be a secondary or redundant coding
    - Brightness and contrast can be more useful
    - Design to support monochrome then color deficient users are supported as well
  - Need to consider the perceptual difference between colors
    - Especially between foreground and background colors
    - Yellow text on white background (or vice versa) is bad
    - As are other color combinations too close together
      - Blue-black

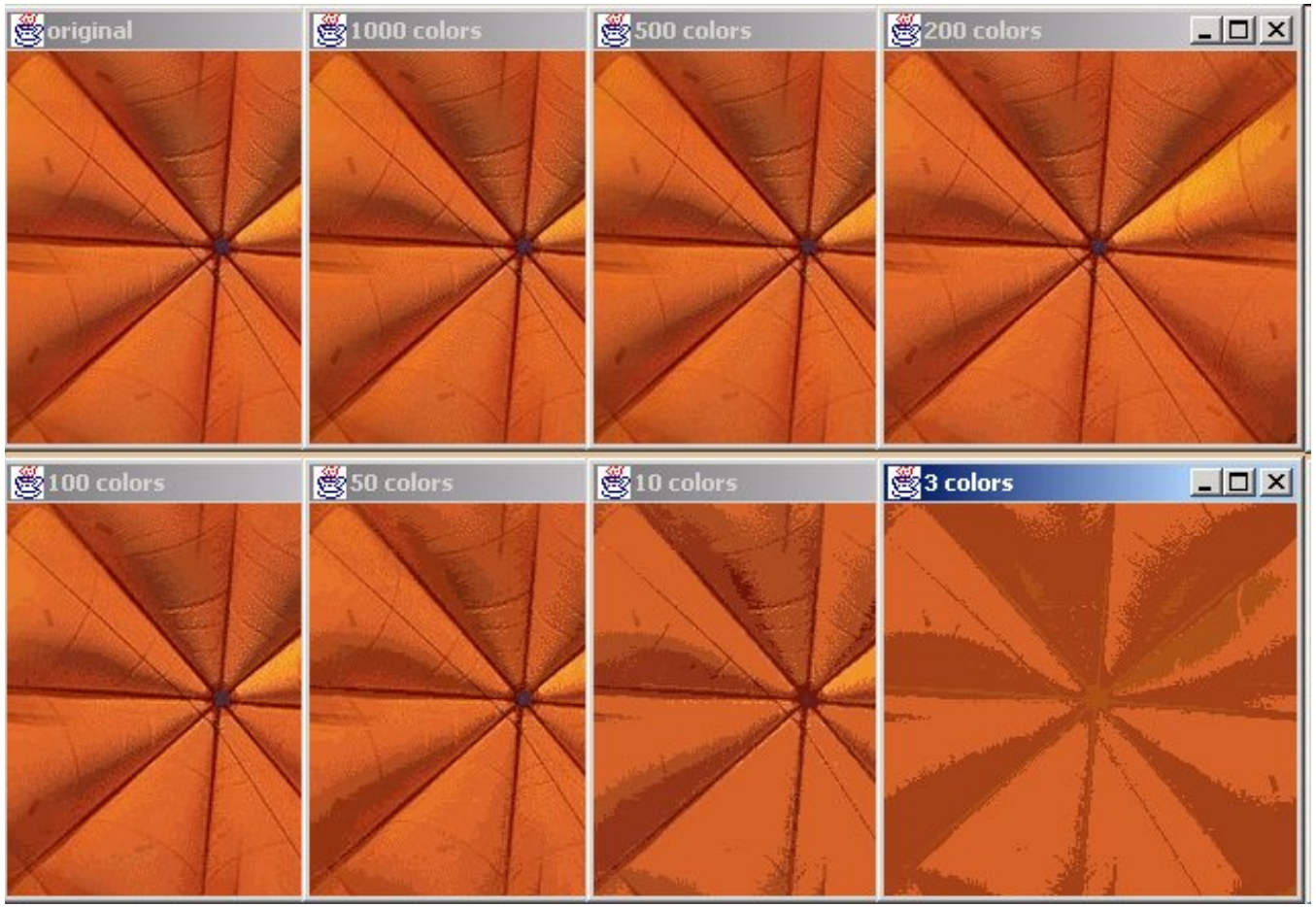
Color Usage Guidelines (cont)

- Color usage guidelines (continued)
  - Blue is a poor choice for lines and text -
    - Eye has trouble focusing on blue
    - Is a good choice for background color though
  - Red on a blue background or vice-versa is bad because the eyes focus at different distance
    - Produces the effect of chromostereopsis -
      - One color seems to float above the other
  - Best to minimize number of colors used and apply them conservatively
  - Gray or desaturated color background works best when applying several different colors in foreground
  - Eye cannot differentiate color of small objects very well
    - Color coding alone is not sufficient

Color Quantization

- Often easiest to generate true color images
  - Easiest to generate scenes / computer graphics images in true color
  - Scanning photographs generally produces true color images
- True color images take large amount of storage
  - An issue for Web based applications
- Need a method of reducing the size of color images
  - Common: reduce to the best 256 colors
    - Map image colors to “closest” of the 256 colors
  - Allows storage in bitmap format with color table and 8 bit pixel values
- Methods include:
  - Uniform quantization
  - Popularity algorithm
  - Median-cut algorithm
  - Octree quantization
- http://en.wikipedia.org/wiki/Color_quantization

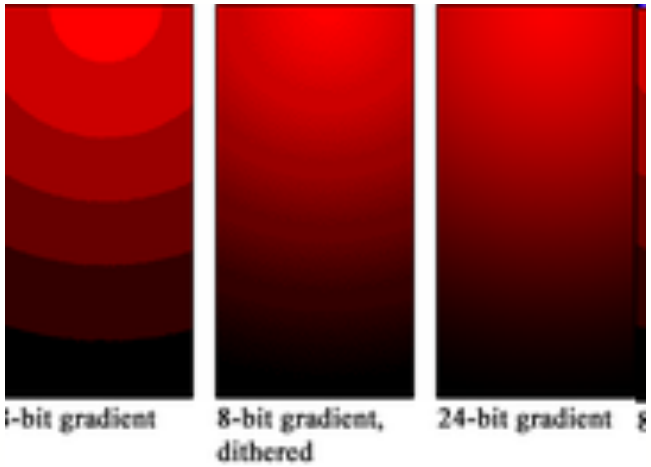
Color Quantization



Example of color quantization using octree method:
[link no longer active]

Color Block and Color Banding

- Color block is the range of RGB values within the image
  - Extent (min, max) of red, green, blue
  - Forms a parallelepiped (box)
  - **Color population** of the image lies within this block
- Useful first step in color quantization is to form the color block for the image
- Color **banding** can result during color quantization
  - Many slightly different colors are replaced by a single color



Color Banding
http://en.wikipedia.org/wiki/Colour_banding

Uniform Quantization

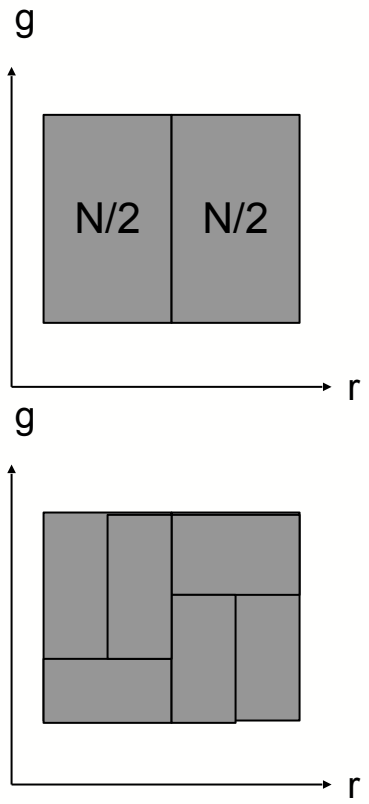
- Subdivides the color block into non-overlapping sub blocks
  - Color at center is a representative for entire block
  - 8 green, 8 red, 4 blue yields 256 sub blocks
  - 6 red, 6 blue, 7 green yields 252 sub blocks
- How to space over the Lookup Table entries?
  - Sometimes constrained to 16 values
- F.S. Hill: Figure 11.21 illustrates setting LUT values for 6R, 7G, 6B
  - Easy to implement but is not very effective
  - Does not consider the distribution of colors within the image

Popularity Algorithm

- Popularity algorithm (Heckbert 1982)
  - http://www.cs.cmu.edu/~ph/ciq_thesis
  - Determines which colors occur often in a file
    - Gives these colors greater priority
- Basic method:
  - Scan the image file to count the occurrences of each color
  - Sort
    - First K colors are the K most popular
  - Method is simplified by truncating each red, green, blue byte to 5 bits
    - Leaves 2^{15}=32K possible colors
- When the image file is rescanned replace each color by the closest representative in the popularity list
  - Minimum mean squared distance
- Can be an expensive

Median-Cut Algorithm

- Also developed by Heckbert (1982)
- Subdivides the color block into K sub blocks
  - Each sub block has approximately the same number of color dots
- Color block is subdivided along its longest dimension
  - Median value found such that N/2 dots occur to each side
  - Each sub block is processed similarly
    - Sliced at the median along its longest dimension
  - Continues until there are K sub blocks
  - Color at center of sub block is take as the representative
- F.S. Hill includes pseudo code
  - Uses a queue of sub blocks
  - Breadth first creation of a tree
- Image file is read again: each color tested to see which sub block it lies in
- http://www.drdobbs.com/184409309?pgno=24



Octree Quantization

- F.S. Hill also discusses a method based on octrees
- Method reads the color file and builds an octree of colors
  - Represents colors perfectly until K different colors have been encountered
  - As each additional color is added it is added to the octree and the octree is reduced to contain no more than K colors
    - Colors are force together
    - Replaced colors by a nearby approximation
- Image file is read again
  - Each color encountered finds its place
    - Perfect representation at the bottom of the tree
    - Some intermediate node where its best color representative lies
- Claim results are comparable to quality obtained by median cut
  - Less processing time and memory required
- http://www.leptonica.org/color-quantization.html