Color Theory
Physical Color

- Visible energy - small portion of the electro-magnetic spectrum
- Pure *monochromatic* colors are found at wavelengths between 380nm (violet) and 780nm (red)
Visible Color

- Eye can perceive other colors as combination of several pure colors.
- Most colors may be obtained as combination of small number of *primaries*.
- Output devices exploits this phenomena.

![Color spectrum diagram with wavelengths for yellow (580 nm), green (520 nm), and red (700 nm).]
Gamma Correction

- The CRT devices are not linear.
- The amount of light emitted from the CRT screen is not linearly proportional to the voltage applied to its tube.
- If \( V_{\text{min}} \) and \( V_{\text{max}} \) are considered black and white intensities, respectively, \( (V_{\text{min}} + V_{\text{max}}) / 2 \) is not 50% gray.
Gamma Correction (cont’d)

- A little bit of electronics: The electronic tube’s (including CRT’s) characteristic graph of the current \( I \) with respect to its voltage \( V \) is a polynomial function of the form \( I = kV^\beta \)
  
  (for semiconductor devices, \( I = k e^{dV} \))

- The power reaching the eye is
  \[ W = IV = V kV^\beta = kV^\gamma \]

- Assume \( n \) intensity levels, \( I_j, 1 \leq j \leq n \) on a linear scale.

- We seek the voltage require for each of these intensity levels, \( I_j \)
Gamma Correction (cont’d)

- We have
  \[ I_j = kV_j^\gamma \]
  or
  \[ V_j = (I_j/k)^{1/\gamma} \]

- This voltage computation that compensates for the nonlinearity of the CRT is denoted *Gamma Correction*.

- Each CRT has its own \( \gamma \) value that the manufacturer provides with the CRT.

- Typically \( \gamma \in [2, 3] \)
Non-Linearity of the Eye

- The human eye is sensitive to intensity of light logarithmically.
- Given four light sources of relative intensities of 1, 2, 9 and 10, the eye will distinguish between 1 and 2 much easier than between 9 and 10.
- It is therefore better to select our intensity levels on an exponential scale: $I_{j+1} = r I_j$ for some unknown constant $r$.
- Assume we have $n = 256$ intensity levels.

**Question**: How can we derive $r$?
Non-Linearity of the Eye (cont’d)

- If $I_0 = L_0$ and $I_{255} = L_1$, then,
  $$L_1 = I_{255} = I_0 r^{255} = L_0 r^{255}$$
  Or
  $$r = (L_1 / L_0)^{1/255}$$

- The factor of $(L_1 / L_0)$ is known as the **dynamic range** of the CRT.

- From practical tests, $r$ should be at most 1.01 for the eye to see the different gray levels continuously.
Mach Banding

- Another (very important to predators) property of animal and human eyes is the ability to perform edge/motion detection.

- Named after Mach who discovered this phenomena in 1865, Mach banding could be seen in poor imagery.

- Mach bands are seen in conjunction with $C^{-1}$ and $C^0$ continuous surfaces.
Dynamic Range Again

- The Human eye has a huge dynamic range
  - See in full day light
  - See in moon light
- 8 Bits has difficulties covering such a large range
- Modern graphics systems supports more bits per color (New NVidia/ATI graphics cards support 32 bits per color or 128 bits per pixel!)
- Algorithms were developed in recent years to support and merge several low range images into one image and display it on contemporary hardware
Dynamic Range Again (cont’d)
Dynamic Range Again (cont’d)
The CIE Diagram (1931 & 1976)

- Universal standard
- Color (ignoring intensity) represented as affine combination of 3 primaries $X$, $Y$, $Z$
  - $x = X / (X + Y + Z)$.
  - Same for $y$ and $z$.
  - Note $x + y + z = 1$
- $Y$ is selected to specifically follow the human-eye sensitivity.
The CIE Diagram (cont’d)

- In 3 space the plot of $x$, $y$, $z$ looks as in the drawing on the right.
- Not all “possible” colors are visible.
- Visible colors contained in horse-shoe region.
- Pure colors (hues) located on boundary of the region.
- Consider the plot of $x$, $y$ only – once $x$, $y$ are known so is $z$. 
The CIE Diagram (cont’d)

- Color “white” is point \( W = (1/3, 1/3, 1/3) \)
- Any visible color \( C \) is blend of a hue \( C' \) and \( W \)
- Purity of color measured by its \textit{saturation}:

\[
\text{saturation (C)} = \frac{d_1}{d_1 + d_2}
\]

- \textit{Complement} of \( C \) is (only) other hue \( D \) on line through \( C' \) and \( W \)
Color enhancement of an image

- increasing the saturation of the colors
- moves them towards the boundary of the visible region

unsaturated  saturated
Color Gamuts

- Not every color output device is capable of generating all visible colors in CIE diagram
- Usually color generated as affine combination of three primaries P, Q, R
- Possible colors bounded by triangle in XYZ space with vertices P, Q, R
- This triangle is called the output device’s *gamut*
Color Gamuts (cont’d)

- Example: Primaries of low quality color monitor:

\[
\begin{bmatrix}
    \text{RED} \\
    \text{GREEN} \\
    \text{BLUE}
\end{bmatrix} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix}
    .628 & .346 & .026 \\
    .286 & .588 & .144 \\
    .150 & .070 & .780
\end{bmatrix}
\]

- Different color displays use different P, Q, R

- Same RGB image data, displayed on two monitors will look different!

- **Question**: Given P,Q,R of two color monitors and image \( I \), what must be done in order for \( I \) to look the same on both monitors?
The RGB Color Model

- Common in describing emissive color displays
- Red, Green and Blue are the primary colors in this model
- Color (including intensity) described as combination of primaries
The RGB Color Model

\[ Col = rR + gG + bB \quad r, g, b \in [0, 1] \]

- Yellow = Red+Green \hspace{1cm} (1, 1, 0)
- Cyan = Green+Blue \hspace{1cm} (0, 1, 1)
- Magenta = Red+Blue \hspace{1cm} (1, 0, 1)
- White = Red+Green+Blue \hspace{1cm} (1, 1, 1)
- Gray = 0.5 Red+0.5 Blue+0.5 Green \hspace{1cm} (0.5, 0.5, 0.5)
- Main diagonal of RGB cube represents shades of gray
The CMY Color Model

- Used mainly in color printing, where the primary colors are subtracted from the background white.
- Cyan, Magenta and Yellow primaries are the complements of Red, Green and Blue.
- Primaries (dyes) subtracted from white paper which absorbs no energy.

\[
\begin{align*}
\text{Red} &= \text{White-Cyan} = \text{White-Green-Blue} \\
&= (0, 1, 1) \\
\text{Green} &= \text{White-Magenta} = \text{White-Red-Blue} \\
&= (1, 0, 1) \\
\text{Blue} &= \text{White-Yellow} = \text{White-Red-Green} \\
&= (1, 1, 0) \\
(r, g, b) &= (1-c, 1-m, 1-y)
\end{align*}
\]
The YIQ Color Model

- Human eye more sensitive to changes in luminance (intensity) than to changes in hue or saturation
- Luminance useful for displaying grayscale version of color signal (e.g. B&W TV)
- Luminance Y – affine combination of R, G & B
- I & Q – null blend (zero sum) of R, G & B
- Conversion

\[(y, i, q) = (r, g, b) \begin{bmatrix} 0.30 & 0.60 & 0.21 \\ 0.59 & -0.28 & -0.52 \\ 0.11 & -0.32 & 0.31 \end{bmatrix}\]

- Green component dominates luminance value
The HSV Color Model

- People naturally mix colors based on Hue, Saturation and Value
- Resulting coordinate system is cylindrical
- H – angle around V axis
- S ∈ [0,1] – distance from V axis
- V ∈ [0,1] - luminance
Color Quantization

- High-quality color resolution for images - 8 bits per primary = 24 bits = 16.7M different colors
- Reducing number of colors – select subset (colormap/pallete) and map all colors to the subset
- Used for devices capable of displaying limited number of different colors simultaneously. E.g. an 8 bit display.
Color Quantization Example

256 colors

64 colors

16 colors

4 colors
Color Quantization Issues

- How are the representative colors chosen?
  - Fixed representatives, image independent - fast
  - Image content dependent - slow

- Which image colors are mapped to which representatives?
  - Nearest representative - slow
  - By space partitioning - fast
Choosing the Representatives

uniform quantization to 4 colors

large quantization error

image-dependent quantization to 4 colors

small quantization error
Uniform Quantization

- Fixed representatives - lattice structure on RGB cube
- Image independent - no need to analyze input image
- Some representatives may be wasted
- Fast mapping to representatives by discarding least significant bits of each component
- Common way for 24 → 8 bit quantization
  - Retain 3 + 3 + 2 most significant bits of R, G and B components

large quantization error
Median-Cut Quantization

- Image colors partitioned into $n$ cells, s.t. each cell contains approximately same number of image colors
- Recursive algorithm
- Image representatives - centroids of image colors in each cell
- Image color mapped to rep. of containing cell
  - not necessarily nearest representative

image-dependent quantization to 4 colors

small quantization error
Quantization

256 colors

uniform

median-cut

8 colors
Binary Dithering

- Improve quality of quantized image by distributing quantization error
Binary Dithering – B&W

- Threshold – map upper half of gray-level scale to white and lower half to black

- Each pixel produces some quantization error

- Distribute quantization error – use local threshold

- Use matrix of thresholds for each $n \times n$ pixels

\[
\begin{align*}
\text{If } I(i, j) &> \frac{1}{2} \\
I(i, j) &= 1; \\
\text{else} \\
I(i, j) &= 0;
\end{align*}
\]

\[
\begin{align*}
\text{If } I(i, j) &> M(i \text{ mod } n, j \text{ mod } n) \\
I(i, j) &= 1 \\
\text{else} \\
I(i, j) &= 0
\end{align*}
\]
Dithering

- For four color input use $2 \times 2$ matrix (gray levels 0 to 255)

\[
\begin{bmatrix}
1 & 1 \\
1 & 1 \\
\end{bmatrix}
\]

- For 16 gray-levels use $4 \times 4$ matrix (gray levels 0 to 255)

\[
\begin{bmatrix}
0 & 8 & 2 & 10 \\
12 & 4 & 14 & 6 \\
3 & 11 & 1 & 9 \\
15 & 7 & 13 & 5 \\
\end{bmatrix}
\]

- How to build $n \times n$ matrix for $n^2$ gray-levels? Build matrix to generate $2^k$ final colors? Build $4 \times 4$ matrix from 256 gray-levels?
Quantization + Dithering

uniform

uniform + dithering

median-cut

8 colors

median-cut + dithering
Error Diffusion

- Reduce quantization error by propagating accumulated error from pixel to (some of) its neighbors
  - in scanline order before thresholding

FloydSteinberg(I)
For x := 1 to XMax do
  For y := 1 to YMax do
    err := I(x,y) - (I(x,y)>128)*255;
    I(x,y) := (I(x,y)>128)*255;
    I(x+1,y) := I(x+1,y)+err*7/16;
    I(x-1,y+1) := I(x-1,y+1)+err*3/16;
    I(x,y+1) := I(x,y+1)+err*5/16;
    I(x+1,y+1) := I(x+1,y+1)+err*1/16;
  end
end

- Note: error propagation weights must sum to one
Error Diffusion (cont’d)

- Here are more examples of FS dithering the Utah teapot:
Halftoning

- Displaying (printing) image on device with lower color range & higher resolution
- Most common in printing (high dpi)
- Pixels grouped together – group pattern used to replace color-scale
- E.g. 2x2 blocks emulating 5 gray-scale levels