Outline

- Human Vision
- Image Representation
- Reducing Color Quantization Artifacts
- Basic Image Processing
Human Vision

Model of Human Visual System

Vision Components:
- Incoming Light
- The Human Eye
Electromagnetic Spectrum

- Visible light frequencies range between ...
  - Red = $4.3 \times 10^{14}$ hertz (700nm)
  - Violet = $7.5 \times 10^{14}$ hertz (400nm)

Figures 15.1 from H&B
Visible Light

- The human eye can “see” light in the frequency range 400nm – 700nm

White Light

Figure 15.3 from H&B
Human Vision

The human retina contains two types of photoreceptors, cones and rods.

Rods:
- 120 million rods in the retina
- 1000x more light sensitive than cones
- Responsible for scotopic vision
- Short-wavelength sensitive
- Responsible for peripheral vision

Cones:
- 6-7 million cones in the retina
- Responsible for photopic vision
- Color sensitive:
  - 64% red, 32% green, 2% blue
- Distributed in the fovea centralis
Tristimulus Theory of Color

Spectral-response functions of each of the three types of cones on the human retina.

This motivates encoding color as a combination of red, green, and blue (RGB).

Figure 13.18 from FvDFH
Visible Light

• The human eye can “see” light in the frequency range 400nm – 700nm

This does not mean that we can see the difference between the different spectral distributions.

Metamers = Two spectral distributions that look the same

White Light

Figure 15.3 from H&B
Outline

• Human Vision
• Image Representation
• Reducing Color Quantization Artifacts
• Basic Image Processing
Image Representation

What is an image?
Image Representation

An image is a 2D rectilinear array of pixels:

A width x height array where each entry of the array stores a single pixel.
Image Representation

What is a pixel?

Continuous image

Digital image
Image Representation

A pixel is something that captures the notion of color

• Luminance pixels
  ◦ Grey-scale images (aka “Intensity images”)
  ◦ 0 – 1.0 or 0 – 255

• Red, Green, Blue pixels (RGB)
  ◦ Color images
  ◦ 0 – 1.0 or 0 – 255
Resolutions

- Spatial resolution: width x height pixels
- Intensity/Color resolution: \( n \) bits per pixel
- Temporal resolution: \( n \) Hz (fps)

<table>
<thead>
<tr>
<th></th>
<th>width x height</th>
<th>bit depth</th>
<th>Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTSC</td>
<td>640 x 480</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>Handheld</td>
<td>640 x 480</td>
<td>16</td>
<td>45</td>
</tr>
<tr>
<td>Monitor</td>
<td>1920 x 1200</td>
<td>24</td>
<td>120</td>
</tr>
<tr>
<td>CCDs</td>
<td>6000 x 4000</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td>Laser Printer</td>
<td>6600 x 5100</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>
Image Quantization Artifacts

• With only a small number of bits associated to each color channel of a pixel there is a limit to intensity resolutions of an image
  ◦ A black and white image allocates a single bit to the luminance channel of a pixel.
    » The number of different colors that can be represented by a pixel is 2.
  ◦ A 24 bit bitmap image allocates 8 bits to the red, green, and blue channels of a pixel.
    » The number of different colors that can be represented by a pixel is 16,000,000.
Outline

• Human Vision
• Image Representation
• Reducing Color Quantization Artifacts
  ◦ Halftoning and Dithering
• Basic Image Processing
Reducing Color Quantization Artifacts

Key Idea:

For (still) images, the combination of image resolution and intensity/color resolution define the total informational content.

We can trade off between these two to achieve different visual effects at the same cost.
Reducing Color Quantization Artifacts

Disclaimer:

In the next few slides, we assume that the original image has continuous pixel values, $I(x, y) \in [0,1)$.

In practice, all the images you will work with will have integer values, $I(x, y) \in \{0, ..., 255\}$. 
Quantization

- When you have a small number of bits per pixel, you can coarsely represent an image by quantizing the color values:
  \[ P(x, y) = Q_b(I(x, y)) = \text{floor}(I(x, y) \cdot 2^b) \]
  with \( b \) the number of bits per pixel.

- \( I(x, y) \in [0,1) \) : True pixel values
- \( P(x, y) \in \{0,1, \ldots, 2^b - 1\} \) : Discretized pixel values

\[ I(x, y) \quad Q_b(x, y) \]
\[ b = 2 \text{ bits per pixel} \]
Quantization

Image with decreasing bits per pixel

- Note contouring!

\[ b = 8 \text{ bits} \quad b = 4 \text{ bits} \quad b = 2 \text{ bits} \quad b = 1 \text{ bits} \]
Reducing Effects of Quantization

Trade spatial resolution for intensity resolution:

- Half-toning
- Dithering
Classical Half-Toning

- Varying-size dots represent intensities
- Area of dots inversely proportional to intensity

$I(x, y)$

$P(x, y)$
Classical Half-Toning

Newspaper Image

From New York Times, 9/21/99
Digital Half-Toning

• Use cluster of pixels to represent intensity
• Trades spatial resolution for intensity resolution
• Note:
  ◦ Half-toning pattern matters
    » Want to avoid vertical, horizontal lines
  ◦ Loss of information
    » 16 configurations → 5 intensities
Digital Half-Toning

- Use cluster of pixels to represent intensity
- Trades spatial resolution for intensity resolution
- Note:
  - Half-toning pattern matters
    - Want to avoid vertical, horizontal lines
  - Loss of information
    - 16 configurations → 5 intensities

- Original (8 bits)
- Quantized (1 bit)
- Half-toned (1 bit)
Dithering

• Distribute errors among pixels
  ○ Exploit spatial integration in our eye
  ○ Display greater range of *perceptible* intensities
Random Dither

- Randomize quantization errors
- Errors appear as noise

\[
P(x, y) = Q_b \left( I(x, y) + \frac{\text{noise}(x, y)}{2^b} \right)
\]
Random Dither

- Randomize quantization errors
- Errors appear as noise

How much noise should we add?

Enough so that we effect rounding, but not so much that we overshoot:

\[
[-1.0, 1.0]
\]

\[
P(x, y) = Q_b \left( I(x, y) + \frac{\text{noise}(x, y)}{2^b} \right)
\]
Random Dither

Original (8 bits)

Uniform (1 bit)

Random (1 bit)
Ordered Dither

Similar to quantization:

\[
\text{We round the input to a value in the range } \{0, \ldots, 2^b - 1 \}.
\]

Different from quantization:

\[
\text{How we round depends on the pixel’s spatial position.}
\]
Ordered Dither

- Pseudo-random quantization errors
- $n \times n$ matrix stores pattern of thresholds

For Binary Displays

\[
\begin{align*}
  i &= x \mod n \\
  j &= y \mod n \\
  \text{if} \quad &\left( I(x, y) > \frac{D_n(i, j)}{n^2+1} \right) \\
  P(x, y) &= 1 \\
  \text{else} \\
  P(x, y) &= 0 \\
\end{align*}
\]

\[
D_2 = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}
\]
Ordered Dither

- Pseudo-random quantization errors
- $n \times n$ matrix stores pattern of thresholds

For $b$ bit displays

$i = x \mod n$

$j = y \mod n$

$c = I(x, y) \cdot (2^b - 1)$

$e = c - \text{floor}(c)$

If \( e > \frac{D_{n(i,j)}}{n^2 + 1} \)

\[ P(x, y) = \text{ceil}(c) \]

else

\[ P(x, y) = \text{floor}(c) \]

\[
D_2 = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}
\]
Ordered Dither

Original (8 bits)  Uniform (1 bit)  Random (1 bit)  Ordered (1 bit)
Error Diffusion Dither

- Spread quantization error over neighbor pixels
  - Error dispersed to pixels right and below
- Below we see Floyd-Steinberg Method

\[ \alpha + \beta + \gamma + \delta = 1.0 \]
Error Diffusion Dither

for( i=0 ; i<height ; i++ )
  for ( j=0 ; j<width ; j++ )
    Dest_{i,j} = quantize( Source_{i,j} )
    error = Source_{i,j} − Dest_{i,j}
    Source_{i,j+1} = Source_{i,j+1} + \alpha \ast error
    Source_{i+1,j-1} = Source_{i+1,j-1} + \beta \ast error
    Source_{i+1,j} = Source_{i+1,j} + \gamma \ast error
    Source_{i+1,j+1} = Source_{i+1,j+1} + \delta \ast error

\alpha = 7/16 \quad \beta = 3/16 \quad \gamma = 5/16 \quad \delta = 1/16

Floyd-Steinberg Dither
Error Diffusion Dither
Outline

- Human Vision
- Image Representation
- Reducing Color Quantization Artifacts
- Basic Image Processing
  - Single Pixel Operations
Computing Grayscale

- The human retina perceives red, green, and blue as having different levels of brightness.

- To compute the luminance (perceived brightness) of a pixel, we need to take the weighted average of the RGBs:
  \[ L_p = 0.30 \cdot r_p + 0.59 \cdot g_b + 0.11 \cdot b_p \]

Original Grayscale

Figure 13.18 from FvDFH
Adjusting Brightness

- Simply scale pixel components
  - Must clamp to range (e.g., 0 to 255)

\[ I(p) \leftarrow I(p) \cdot \alpha \]
Adjusting Contrast

• Compute mean image luminance $\bar{L}$
  - $\bar{L} = \text{Average}(0.30 \cdot r_p + 0.59 \cdot g_p + 0.11 \cdot b_p)$

• Scale deviation from $\bar{L}$ for each pixel component
  - Must clamp to range (e.g., 0 to 255)

\[ I(p) \leftarrow (I(p) - \bar{L})\alpha + \bar{L} \]
Adjusting Saturation

• Compute per-pixel luminance $L_p$
  \[ L_p = 0.30 \cdot r_p + 0.59 \cdot g_p + 0.11 \cdot b_p \]

• Scale deviation from $L_p$ for each pixel component
  \[ I(p) \leftarrow (I(p) - L_p)\alpha + L_p \]

Original  More Saturation