Digital imaging
and
image processing
Review

Final Exam Monday June 11th at 8am
Digital Data
Photon creates electron/hole pair. The charges are separated by the electric field.

Outside the depletion volume no electric field exists. Electron/hole pairs created here will recombine because they are not separated.
Digital Sensors

(a) Single sensing element.
(b) Line sensor.
(c) Array sensor.
Grayscale image

<table>
<thead>
<tr>
<th>157</th>
<th>153</th>
<th>174</th>
<th>168</th>
<th>150</th>
<th>152</th>
<th>129</th>
<th>151</th>
<th>172</th>
<th>161</th>
<th>155</th>
<th>156</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>182</td>
<td>163</td>
<td>74</td>
<td>75</td>
<td>62</td>
<td>33</td>
<td>17</td>
<td>110</td>
<td>210</td>
<td>180</td>
<td>154</td>
</tr>
<tr>
<td>180</td>
<td>180</td>
<td>50</td>
<td>14</td>
<td>34</td>
<td>6</td>
<td>10</td>
<td>33</td>
<td>48</td>
<td>106</td>
<td>159</td>
<td>181</td>
</tr>
<tr>
<td>206</td>
<td>109</td>
<td>5</td>
<td>124</td>
<td>131</td>
<td>111</td>
<td>120</td>
<td>204</td>
<td>166</td>
<td>15</td>
<td>56</td>
<td>180</td>
</tr>
<tr>
<td>194</td>
<td>68</td>
<td>137</td>
<td>251</td>
<td>237</td>
<td>239</td>
<td>239</td>
<td>228</td>
<td>227</td>
<td>87</td>
<td>71</td>
<td>201</td>
</tr>
<tr>
<td>172</td>
<td>105</td>
<td>207</td>
<td>233</td>
<td>233</td>
<td>214</td>
<td>220</td>
<td>229</td>
<td>228</td>
<td>98</td>
<td>74</td>
<td>206</td>
</tr>
<tr>
<td>188</td>
<td>88</td>
<td>179</td>
<td>209</td>
<td>185</td>
<td>215</td>
<td>211</td>
<td>158</td>
<td>139</td>
<td>75</td>
<td>20</td>
<td>169</td>
</tr>
<tr>
<td>189</td>
<td>97</td>
<td>156</td>
<td>84</td>
<td>10</td>
<td>168</td>
<td>134</td>
<td>11</td>
<td>31</td>
<td>62</td>
<td>22</td>
<td>148</td>
</tr>
<tr>
<td>199</td>
<td>168</td>
<td>191</td>
<td>193</td>
<td>158</td>
<td>227</td>
<td>178</td>
<td>143</td>
<td>182</td>
<td>106</td>
<td>36</td>
<td>190</td>
</tr>
<tr>
<td>206</td>
<td>174</td>
<td>155</td>
<td>252</td>
<td>236</td>
<td>231</td>
<td>149</td>
<td>178</td>
<td>228</td>
<td>43</td>
<td>95</td>
<td>234</td>
</tr>
<tr>
<td>190</td>
<td>216</td>
<td>116</td>
<td>149</td>
<td>236</td>
<td>187</td>
<td>86</td>
<td>150</td>
<td>79</td>
<td>38</td>
<td>218</td>
<td>241</td>
</tr>
<tr>
<td>190</td>
<td>224</td>
<td>147</td>
<td>108</td>
<td>227</td>
<td>210</td>
<td>127</td>
<td>102</td>
<td>36</td>
<td>101</td>
<td>255</td>
<td>224</td>
</tr>
<tr>
<td>190</td>
<td>214</td>
<td>173</td>
<td>66</td>
<td>103</td>
<td>143</td>
<td>96</td>
<td>80</td>
<td>2</td>
<td>109</td>
<td>249</td>
<td>215</td>
</tr>
<tr>
<td>187</td>
<td>196</td>
<td>235</td>
<td>75</td>
<td>1</td>
<td>81</td>
<td>47</td>
<td>0</td>
<td>6</td>
<td>217</td>
<td>255</td>
<td>211</td>
</tr>
<tr>
<td>183</td>
<td>202</td>
<td>237</td>
<td>145</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>108</td>
<td>200</td>
<td>138</td>
<td>243</td>
<td>236</td>
</tr>
<tr>
<td>196</td>
<td>206</td>
<td>123</td>
<td>207</td>
<td>177</td>
<td>121</td>
<td>123</td>
<td>200</td>
<td>175</td>
<td>13</td>
<td>96</td>
<td>218</td>
</tr>
</tbody>
</table>
Bit depth

- Binary: 0 and 1
- 8 bit: 0 up to \(2^8 = 256\)
- 16 bit: 0 up to \(16^2 = 65,536\)
- 32 bit: 0 up to \(32^2 = 4,294,967,296\)
- Color: RGB contains a red, green and blue matrix of the bit depth specified
Image displayed in 32, 16, 8, 4, and 2 intensity levels.
Saturation
Forming a vector

\[ \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \]

Component image 1 (Red)
Component image 2 (Green)
Component image 3 (Blue)
Color Images

[Images of a puffer fish in color and in different color channels: blue, green, red]
Color sensor: Bayer pattern
We lose light
Dichroic Mirrors, Multiple Cameras
Image Display

Simplest contrast adjustment:
Set Min, Max of display
More powerful methods to improve contrast
In image space

- We distinguish two domains:
  - Spatial or Pixel domain: \( f(x, y) \) or \( f(m, n) \)
  - Frequency Domain: \( F(w_x, w_y) \) or \( F(u, v) \)
Operations

- Element-wise (pixel by pixel) vs. Matrix operations

- Single-pixel vs neighborhood:
  - Single-pixel: grab e.g. the value of the one nearest pixel)
  - Neighborhood (calculate and use e.g. the average / max / median / min or other calculated value of the nearest neighbors)
Simplest form of processing: Point Processing

Pixel

\[ r = f(m, n) \]

\[ s = T(r) \]

Thresholding

Image "negative": \[ s = L - 1 - r \]
(a) 8-bit image. (b) Intensity transformation function used to obtain the digital equivalent of a “photographic” negative of an 8-bit image. The arrows show transformation of an arbitrary input intensity value $z$ into its corresponding output value $s_0$. (c) Negative of (a) obtained using (b)
Better visibility for display / diagnosis
Binary

- Small storage
- Easier to apply some operations
Simplest form of processing: Point Processing

Pixel

\[ r = f(m,n) \]

\[ s = T(r) \]

Common Examples:

- Dynamic Range Compression

\[ T(r) = c \log(1 + r) \]

- Gamma Correction

\[ T(r) = c r^\gamma \]

Narrow range of “dark” gets mapped to broad range of “gray”
Gamma Correction

\[ L = U^\alpha \]

\[ 1.8 \leq \alpha \leq 2.5 \]
Origins of gamma correction:

• Nonlinear response of CRT's and imagers

To correct for this in image display, the images or commands to the CRT are “predistorted”.

Gamma Correction:

Image as viewed on monitor

Monitor

Monitor

Gamma correction

Image as viewed on monitor

Monitor

Monitor
Scaling
Log scale display
Synthetic lookup tables

Chasing the right one can make it easier to see stuff — and to get published…
Histogram Processing:

- Distribution of gray-levels can be judged by measuring a Histogram

**Histogram:**

- Distribution of gray-levels can be judged by measuring a Histogram

- For B-bit image, initialize 2^B counters with 0
- Loop over all pixels x,y
- When encountering gray level \( f(x,y) = i \), increment the counter number i

- With proper normalization, the histogram can be interpreted as an estimate of the probability density function (pdf) of the underlying random variable (the graylevel)

- You can also use fewer, larger bins to trade off amplitude
Histograms
Histogram manipulation

Illustration of the mean and standard deviation as functions of image contrast. (a)-(c) Images with low, medium, and high contrast, respectively. (Original image courtesy of the National Cancer Institute.)
Example:

- A histogram of gray level vs. number of pixels.
- An image with a caption "Cameraman image."
Histogram Equalization

- Make it flat and spread it out
- This is a nonlinear operation
The general idea: map the histogram of the given image to a flat histogram by performing a nonlinear operation on the gray value at each pixel.

Nonlinear Transformation: \( s = T(r) \)

Questions:

– Why do this?
– What is the right transformation?
– How do we find it, given a particular image.

Analysis for the continuous grayscale first

Histogram Equalization Example:

• Consider the histogram of the given (continuous graytone) image as a pdf \( p(r) \), where \( r \) is in the interval \( [0,1] \).

  – Recall that as a pdf we have \( \int_0^1 p(r) \, dr = 1 \).

  – Any pixel operation \( T(r) \) should map \( [0,1] \) to \( [0,1] \).

• Desired properties of \( T(r) \):

  – Keep the back/white order (\( T(r) \) should be monotonic increasing)

  – \( T(r) \) should be single valued (one-to-one), hence invertible.

• Question: Given image with histogram \( p(r) \), what does histogram of \( s = T(r) \), denoted \( q(s) \), look like in general?

\[
q(s) = \int_{T^{-1}(s)}^1 p(r) \, dr
\]
Color Histogram
Spatial filtering

- In image space
- In frequency space
Image size / Sampling
Aliasing
Nyquist sampling = twice the frequency
Aliasing

• (Image downsized around four times)
Re-sampling: Change size by interpolation

(a) Image reduced to 72 dpi and zoomed back to its original 930 dpi using nearest neighbor interpolation.
(b) Image reduced to 72 dpi and zoomed using bilinear interpolation.
(c) Same as (b) but using bicubic interpolation.
Average when downsizing?

- Edges lose contrast if you average but result is smoother
Convolution

Let $f$ be the image and $g$ be the kernel. The output of convolving $f$ with $g$ is denoted $f * g$.

$$(f * g)[m,n] = \sum_{k,l} f[m-k,n-l] g[k,l]$$

- Convention: kernel is “flipped”
- MATLAB: conv2 vs. filter2 (also imfilter)
Convolution

Key properties

• **Linearity:** \( \text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2) \)

• **Shift invariance:** same behavior regardless of pixel location: \( \text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f)) \)

• Theoretical result: any linear shift-invariant operator can be represented as a convolution
Convolve
“Drag-and-Stamp”

Mask dimension = 2M+1

Border dimension = M
Spatial Filtering: Blurring

• Example

Averaging Mask:

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1}{9}
\]

**FIGURE 3.35** (a) Original image, of size 500 x 500 pixels. (b) - (f) Results of smoothing with square averaging filter masks of sizes \( m = 3, 5, 9, 15, \) and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 x 120 pixels.
Image Enhancement: Spatial Filtering Operation

- Idea: Use a “mask” to alter pixel values according to local operation
- Aim: (De)-Emphasize some spatial frequencies in the image.

**Figure 3.37** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a $3 \times 3$ averaging mask. (c) Noise reduction with a $3 \times 3$ median filter. (Original image courtesy of Mr. Joseph E. Pascenti, Lixi, Inc.)
Image Enhancement: Spatial Filtering Operation

• An important point: **Edge Effects (Ex.: 5x5 Mask)**
  – How to fill in a “border”
    • Zeros (Ringing)
    • Replication (Better)
    • Reflection (“Best”)

\[
\begin{array}{cccc}
  d & c & a & b \\
  b & a & a & b \\
  b & a & a & b \\
  d & c & c & d \\
\end{array}
\]

• **Procedure:**
  – Replicate row-wise
  – Replicate column-wise
  – Apply filtering
  – Remove borders
What about near the edge?
- the filter window falls off the edge of the image
- need to extrapolate
- methods:
  - clip filter (black)
  - wrap around
  - copy edge
  - reflect across edge

Source: S. Marschner

Image Enhancement: Spatial Filtering Operation
Gaussian Kernel

\[ G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)

5 x 5, \( \sigma = 1 \)

fspecial('gauss',5,1)

Source: C. Rasmussen

Choosing kernel width

- Gaussian filters have infinite support, but discrete filters use finite kernels

Source: K. Grauman
Choosing kernel width

- Gaussian filters have infinite support, but discrete filters use finite kernels.

Source: C. Rasmussen

\[
\begin{array}{c}
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
0.022 & 0.097 & 0.159 & 0.097 & 0.022 \\
0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
\end{array}
\]

Source: K. Grauman
Example: Smoothing with a Gaussian
Mean vs. Gaussian filtering

Gaussian filters

• Remove "high-frequency" components from the image (low-pass filter)
• Convolution with self is another Gaussian
  – So can smooth with small-width kernel, repeat, and get same result as larger-width kernel would have
  – Convolving two times with Gaussian kernel of width $\sigma$ is same as convolving once with kernel of width $\sigma \sqrt{2}$

• Separable kernel
  – Factors into product of two 1D Gaussians
Gaussian filters

- Remove “high-frequency” components from the image (low-pass filter)
- Convolution with self is another Gaussian
  - So can smooth with small-width kernel, repeat, and get same result as larger-width kernel would have
  - Convolving two times with Gaussian kernel of width $\sigma$ is same as convolving once with kernel of width $\sigma\sqrt{2}$

- **Separable** kernel
  - Factors into product of two 1D Gaussians
Use this to sharpen!

Sharpening revisited

- What does blurring take away?

Let's add it back:

\[ \text{original} + \alpha \text{detail} = \text{sharpened} \]
More on Linear Operations: Sharpening Filters

- Sharpening filters use masks that typically have + and – numbers in them.

- They are useful for highlighting or enhancing details and high-frequency information (e.g. edges)

- They can (and often are) based on derivative-type operations in the image (whereas smoothing operations were based on “integral” type operations)
Derivatives

### Differentiation and convolution

- Recall, for 2D function, $f(x,y)$:
  \[
  \frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \left( \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right)
  \]

  - This is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as
  \[
  \frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}
  \]

  - which is obviously a convolution with kernel

\[-1\quad 1\]
Derivative-type Filters

\[
\begin{bmatrix}
-1 & 1 \\
-1 \\
1
\end{bmatrix}
\rightarrow \frac{\partial f}{\partial x} \approx f(x+1, y) - f(x, y)
\]

\[
\begin{bmatrix}
-1 \\
1
\end{bmatrix}
\rightarrow \frac{\partial f}{\partial y} \approx f(x, y+1) - f(x, y)
\]

\[
\begin{bmatrix}
1 & -2 & 1 \\
1
\end{bmatrix}
\rightarrow \frac{\partial^2 f}{\partial x^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)
\]

\[
\begin{bmatrix}
1 & -2 \\
-2 & 1
\end{bmatrix}
\rightarrow \frac{\partial^2 f}{\partial y^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)
\]

Laplacian: \[\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Rightarrow [1 \ -2 \ 1] + [1 \ -2 \ 1] = [0 \ 1 \ 0 \ 1 \ -4 \ 1 \ 0 \ 1 \ 0] \]
FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)
Sharpening Using the Laplacian Filter

\[ g(x, y) = Af(x, y) - \nabla^2 f(x, y) \]

\[
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & A+8 & -1 \\
-1 & -1 & -1 \\
\end{bmatrix}
\]

**FIGURE 3.43**
(a) Same as Fig. 3.41(c), but darker.
(b) Laplacian of (a) computed with the mask in Fig. 3.42(b) using \( A = 0 \).
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with \( A = 1 \).
(d) Same as (c), but using \( A = 1.7 \).
Laplacian of Gaussian

Gaussian Unsharp Mask Filter

\[ f + \alpha(f - f \ast g) = (1 + \alpha)f - \alpha f \ast g = f \ast ((1 + \alpha)e - g) \]

- unit impulse
- blurred image
- unit impulse (identity)

Edge detection

- Goal: Identify sudden changes (discontinuities) in an image
- Intuitively, most semantic and shape information from the image can be encoded in the edges
- More compact than pixels

- Ideal: artist's line drawing (but artist is also using object-level knowledge)

Source: D. Lowe
Edge detection

Example: Laplacian

\[ \nabla^2 I(x, y) \]

Notes about the Laplacian:

- \( \nabla^2 I(x, y) \) is a SCALAR
  - Can be found using a SINGLE mask
  - Orientation information is lost
- \( \nabla^2 I(x, y) \) is the sum of SECOND-order derivatives
  - But taking derivatives increases noise
  - Very noise sensitive!
- It is always combined with a smoothing operation:
  - Smooth Laplacian

LoG Filter

- First smooth (Gaussian filter),
- Then, find zero-crossings (Laplacian filter):
  - \[ O(x, y) = \nabla^2 (I(x, y) * G(x, y)) \]

LoG of Gaussian-filtered image

Do you see the distinction?

1D Gaussian and Derivatives

\[
\sigma^2 e^{-\frac{x^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} - \frac{x^2}{2\sigma^2} e^{-\frac{x^2}{2\sigma^2}}
\]

Effect of LoG Operator

Band-Pass Filter (suppresses both high and low frequencies)

Why? Easier to explain in a moment.
Blob detection

Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)

LoG sigma = 2, zero-crossing

LoG sigma = 4, zero-crossing

LoG sigma = 8, zero-crossing

Note: Closed Contours

You may have noticed that zero-crossings form closed contours. It is easy to see why…

Think of equal-elevation contours on a topo map. Each is a closed contour. Zero-crossings are contours at elevation = 0.

remember that in our case, the height map is of a LoG filtered image - a surface with both positive and negative “elevations”

Other uses of LoG:

Blob Detection


Pause to Think for a Moment:

How can an edge finder also be used to find blobs in an image?
Sampling in time
Rolling Shutter/Global Shutter and Artifacts
Average multiple images

(a) Noisy image of the Sombrero Galaxy. (b)-(f) Result of averaging 10, 50, 100, 500, and 1,000 noisy images, respectively. All images are size 1548x2238 pixels and all scaled so intensities span the full [0, 255] intensity scale.
Computational Denoising

Denoised using ROF denoise in FIJI
Digital subtraction angiography.
(a) Mask image. (b) A live image.
(c) Difference between (a) and (b).
(d) Enhanced difference image.

Image courtesy of the Image Sciences Institute, University Medical Center, Netherlands
(from our textbook: Digital Image Processing in Matlab)
Filtering in Frequency (Fourier) Space
Time and space

Pressure

Amplitude

We can Fourier Transform back and forth

\[
\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft} dt
\]

\[
\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft} df
\]

\[f = \frac{1}{t}\]
Fourier series

This works exactly the same way for light, where we look at the wavelength vs frequency.
Short-time Fourier spectrogram
1D spatial frequency
2D spatial frequency

\[ y(k_1, k_2) = \int \int f(x_1, x_2)e^{-i2\pi(k_1 x_1 + k_2 x_2)} \, dx_1 \, dx_2 \]
High and low-pass
Filtering out a single spatial frequency