Outline

• Extracting useful information from Images
  – Continue Filtering Applications
  – Edge Models & Edge Detection
Recap: Filtering Overview
Revisiting Median Filtering

- Original image “Eight.tif” with added ‘salt-and-pepper’ noise then filtered with a (3-by-3) averaging filter and a (3-by-3) median filter.

Observation:
The median filter does a better job of removing ‘salt-and-pepper’ noise, with less blurring of edges.
Recap: Image Histogram

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

image

histogram
Histogram and PDF

• Assume a scalar image, A, and its histogram H.

\[ h(u) = \frac{H(u)}{|\Omega|} \]

• Denominator is the size of histogram (num. of pixels)

• \( h \rightarrow \text{PDF, relative frequencies are set between 0 and 1.} \)
Cumulative Distribution Function (CDF)

- If $h$ is the normalized histogram of an image, the CDF of $h$ is defined as:

$$CDF_h(i) = \sum_{j=0}^{i} h(i)$$
Image Gradation Functions

• When recording image data, there are often particular problems: lighting, motion blur, noise, ...
  – Uniform illumination is desired
  – Smoothing (denoising)
  – Sharpening
Gradation Functions

• We transform an image $A$ into a new image $A_{new}$ of the same size, by mapping a grey level $u$ at pixel location $p$ in $A$ by a gradation function $g$ onto a grey level $v=g(u)$.

$$y = x$$

No influence on visual quality at all
Gradation Functions

• We transform an image $A$ into a new image $A_{new}$ of the same size, by mapping a grey level $u$ at pixel location $p$ in $A$ by a gradation function $g$ onto a grey level $v=g(u)$.

• Digital negative

$y = L - x$
Gradation Functions - Contrast Stretching

\[ y = \begin{cases} 
\alpha x & 0 \leq x < a \\
\beta(x - a) + y_a & a \leq x < b \\
\gamma(x - b) + y_b & b \leq x < L 
\end{cases} \]

\[ a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200 \]
Gradation Functions - Clipping

\[ y = \begin{cases} 
0 & 0 \leq x < a \\
\beta(x-a) & a \leq x < b \\
\beta(b-a) & b \leq x < L 
\end{cases} \]

\[ a = 50, b = 150, \beta = 2 \]
Gradation Functions - Range Compression

\[ y = c \log_{10}(1 + x) \]

\[ c = 100 \]
Histogram Equalization

- Histogram equalization is a non-linear process aimed to highlight image brightness in a way particularly suited to human visual analysis.
- Histogram equalization aims to change a picture in such a way as to produce a picture with a [flatter histogram](#), where all levels are equiprobable.
Histogram Equalization

Over-exposed image
Histogram Transform
Any idea how to do this simply?
Any idea how to do this simply?

Use CDF and remap Intensities into 0-L-1 interval!

(L: gray level)

Credit: scratchapixel.com
Any idea how to do this?

Steps:
1. Compute normalized histogram (PDF)
   ```
   int hist[256] = 0...0
   float norm[256]
   for x, hist[ I(x) ]++
   for g in [0,255],
       norm[g] = hist[g] / npixels
   ```
2. Compute cumulative histogram (CDF)
   ```
   for g in [0,255],
       cum[g] = cum[g-1] + norm[g]
   ```
3. Transform
   ```
   for x, I(x) = 255*cum[ I(x) ]
   ```

Credit: Stan Birchfield
Algorithm for Histogram Equalization

Normalized histogram $p$ (PDF) of an image $f$, whose intensity values span from 0 to $L-1$

\[
p_n = \frac{\text{number of pixels with intensity } n}{\text{total number of pixels}} \quad n = 0, 1, ..., L - 1.
\]

The histogram equalized image $g$ will be defined by

\[
g_{i,j} = \text{floor}\left( (L - 1) \sum_{n=0}^{f_{i,j}} p_n \right),
\]
Algorithm for Histogram Equalization

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('wiki.jpg', 0)

hist, bins = np.histogram(img.flatten(), 256, [0, 256])

cdf = hist.cumsum()
cdf_normalized = cdf * hist.max() / cdf.max()

plt.plot(cdf_normalized, color='b')
plt.hist(img.flatten(), 256, [0, 256], color='r')
plt.xlim([0, 256])
plt.legend(('cdf', 'histogram'), loc='upper left')
plt.show()
```
Histogram Equalization Example 1

before

after
Histogram Equalization Example 2
Histogram Equalization Example 3
Histogram Equalization Example 4
Histogram Equalization Example 5
Histogram Equalization Example 6
Edges

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
  - Marks the border of an object
Origins of Edges

- Edges are caused by a variety of factors

- Surface normal discontinuity
- Depth discontinuity
- Surface color discontinuity
- Illumination discontinuity
Close-up Edges
Close-up Edges
Characterizing Edges

• An edge is a place of rapid change in the image intensity function.
Gradient Operations ($G_x$, $G_y$)

\[
G_x = \begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix}
\]

\[
G_y = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
\]
Revisit: Image Gradient

- The gradient of an image: \( \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \)

\[ \nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right] \]

\[ \nabla f = \left[ 0, \frac{\partial f}{\partial y} \right] \]

The gradient points in the direction of most rapid increase in intensity.

The gradient direction is given by \( \theta = \tan^{-1} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \)

- how does this relate to the direction of the edge?

The edge strength is given by the gradient magnitude:

\[ \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]

Source: Steve Seitz
Effects of Noise on Gradient Computation

- Zero mean additive Gaussian noise

Increasing noise
Effects of Noise on Gradient Computation

Zero mean additive Gaussian noise

Take Home Message: Gradient computation is very sensitive to noise
Effects of Noise

Where is the edge (solution of derivative) ?

Credit to Seitz.
Solution: Smoothing

- Smoothing removes noise, but **blurs** edge.
Edge Detection

• Requires **difference** filters
• What do we expect from these filters? What should be the main properties of such filters?
Design Criteria for Edge Detection Problems

- **Good Detection:** minimize prob. of FP (detecting spurious edges) and FN (missing real edges)
- **Good Localization:** must be as close as possible to the true edges
- **Single Response:** must return one point only for each true edge point

True edge  | Poor robustness to noise  | Poor localization  | Too many responses
Basic Comparisons of Edge Operators

Gradient:
\[ \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \]

Roberts (2 x 2):
\[
\begin{array}{cc}
0 & 1 \\
-1 & 0 \\
\end{array}
\]
\[
\begin{array}{cc}
1 & 0 \\
0 & -1 \\
\end{array}
\]
- Good Localization
- Noise Sensitive
- Poor Detection

Sobel (3 x 3):
\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1 \\
\end{array}
\]
\[
\begin{array}{ccc}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & 1 \\
\end{array}
\]
- Poor Localization
- Less Noise Sensitive
- Good Detection

Sobel (5 x 5):
\[
\begin{array}{cccc}
-1 & -2 & 0 & 2 \\
-2 & -3 & 0 & 3 \\
-3 & -5 & 0 & 5 \\
-2 & -3 & 0 & 3 \\
-1 & -2 & 0 & 2 \\
\end{array}
\]
\[
\begin{array}{cccc}
1 & 2 & 3 & 2 \\
2 & 3 & 5 & 3 \\
0 & 0 & 0 & 0 \\
-2 & -3 & -5 & -3 \\
-1 & -2 & -3 & -2 \\
\end{array}
\]
- Poor Localization
- Less Noise Sensitive
- Good Detection
Example: Laplacian of Gaussian (LoG) and Canny Edge Detector

*Marr and Hildreth Filtering, 1980.*
- Smooth Image with Gaussian Filter
- Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
- Find zero-crossings
- Find slope of zero-crossings
- Apply threshold to slope and mark edges

*J. Canny. 1986*
- Smooth Image with Gaussian filter
- Compute Derivative of filtered image
- Find Magnitude and Orientation of gradient
- Apply Non-max suppression
- Apply Thresholding (Hysteresis)
Example: Canny-Gradients

X-Derivative of Gaussian  Y-Derivative of Gaussian  Gradient Magnitude
Next Lecture

- We will go deeper into Canny and Laplacian of Gaussian (LoG) edge detection.
- PA2 is about Canny edge detection (from scratch)
What is Entropy? (For PA1)

If the particles represent gas molecules at normal temperatures inside a closed container, which of the illustrated configurations came first?

If you tossed bricks off a truck, which kind of pile of bricks would you more likely produce?

Disorder is more probable than order.
What is Entropy? (For PA1)

Image entropy is a quantity which is used to describe the `business' of an image, i.e. the amount of information which must be coded for by a compression algorithm.

Low entropy images, such as those containing a lot of black sky, have very little contrast and large runs of pixels with the same or similar DN values.

An image that is perfectly flat will have an entropy of zero.

\[
\text{Entropy} = - \sum_{i} P_i \log_2 P_i
\]
Questions?