Image Enhancement

Prof. Eric Miller
elmiller@ece.tufts.edu
Overview

• Basic problem in image processing: making images look better

• All sorts of degradations that can cause problems
  – Blue
  – Noise
  – Bad contrast

• Here we deal with the last
  – Pixel-by-pixel modifications to bring out the “detail” in images with poor contrast
  – Technical term: “Contrast enhancement”
Some examples

Input

Output
Basic Program

• Everything we do here amounts to pixel-by-pixel transformation of gray value

\[ s = T(r) \]

Input gray value \( r \)

Transformation

Output gray value \( s \)

Qualities of \( T \)
• Domain?
• Range?
Image Negative

- Image negative means we make black white and white black
- Black = 0 and white = 255
- What about in between? Maybe just a line

\[
s - 255 = \frac{0 - 255}{255 - 0} (r - 0) \Rightarrow s = 255 - r
\]
Stretching contrasts

- Narrow range or dark shades get stretched out to fill a much broader range of gray values
- If image is very dark so small variations are tough to tell apart then this will improve the contrast
- Common choice for this function

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

- Choose \( c \) so that \( s=0 \) at \( r=0 \) and \( s=255 \) at \( r=255 \). What should it be?
Piecewise linear stretch

- Different gray values get stretched in different ways.
- Transform completely specific by the "breakpoints" \((r_j, s_j)\) for \(j=1,2,\ldots,N\).
- Restrictions on these?
- Idea would be to choose these breakpoints in some "optimal" way.

\[ s = s_3 + \frac{s_3 - s_2}{r_3 - r_2} (r - r_3) \]
Histogram equalization

- Dark image means lots of gray levels near zero
- Histogram skewed to the left
- Wouldn’t it be better if we could use all of the gray levels sort of equally?
- In that case we would have a flat histogram
- So, let’s find a $T$ that flattens the histogram!

$$s_k = \frac{255}{n} \sum_{j=0}^{k} n_k \quad k = 0, 1, 2, \ldots, 255 \quad n = \text{total number of pixels}$$
$$n_k = \text{number of pixels} = k$$

Image histogram

Gray levels (0 - 255)
Why does it work?

$$s_k = \frac{255}{n} \sum_{j=0}^{k} n_k \quad k = 0,1,2,K \ 255$$

\[ n = \text{total number of pixels} \]
\[ n_k = \text{number of pixels} = k \]

- Kind of like the log transformation
- Rapid rise to stretch out the dark
- Plateau to deal with the light
- For continuous images this will exactly give us flat histogram
- For discrete, only approximate
Matlab Concepts Covered

• Use of a `switch` statement to easily choose among alternatives
• Simple string function `lower` to make sure we have all lower cases when we use the `switch`
• Use of well commented functions to make programming easier