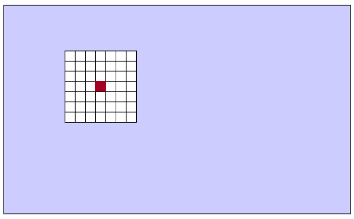
Adaptive Histogram Equalization

- Different regions in a single image
  - Example: Coin on white paper
- Poor result for Histogram Equalization
  - Do the coins and paper separately
  - How to segment?
- Compute histogram in local regions around each pixel

Adaptive Equalization



- For each pixel
  - Compute Histogram in Neighborhood
  - Transform only the center pixel
- Go to next pixel

Color Quantization

- 24 bits/pixel - 8 bits/color -256^3 Colors
- 8 bits/pixel - 256 colors
  - 3-3-2 bits for R-G-B
  - *General* Quantization - Look Up Table (**LUT**)
  - **LUT** can be for **RGB** or for **YUV**

LUT	0	1	2	k	254	255
R	$\cdots$	R_1	$\cdots$	R_k	$\cdots$	R_{255}
G	$\cdots$	G_1	$\cdots$	G_k	$\cdots$	G_{255}
B	$\cdots$	B_1	$\cdots$	B_k	$\cdots$	B_{255}

Quantization Error

LUT	0	1	2	k	254	255
R	$\cdots$	R_1	$\cdots$	R_k	$\cdots$	R_{255}
G	$\cdots$	G_1	$\cdots$	G_k	$\cdots$	G_{255}
B	$\cdots$	B_1	$\cdots$	B_k	$\cdots$	B_{255}

- If pixel ***p*** with color ***(r,g,b)*** is coded by ***k***, a possible quantization error **for *p*** is:

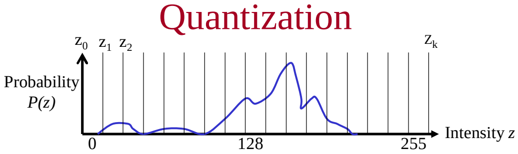
$$\mathrm{E}_p^2 = (r - R_k)^2 + (g - G_k)^2 + (b - B_k)^2$$

- The total error introduced by a LUT is:

$$\mathrm{E}^2 = \sum_p \mathrm{E}_p^2$$

- Unknowns: $(r,g,b) \rightarrow k$, $k \rightarrow (R_k, G_k, B_k)$

Quantization

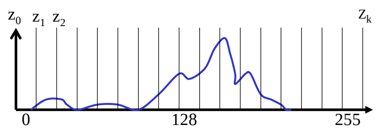


- Map the continuous intensities to $\{q_0, \ldots, q_{k-1}\}$
  - Borders of segments: $z_0, z_1, z_2, \ldots, z_k$
  - Represent each segment $[z_{i-1}, z_i]$ by intensity q_{i-1}
- Uniform Quantization: $q_i = (z_i + z_{i+1})/2$

$$z_{i+1} - z_i = (z_k - z_0)/k$$

- Prior Mappings (e.g. Gamma Correction)

Optimal Quantization (6.5.1)



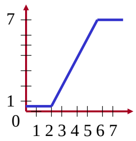
- Minimize the error: $\sum_{i=0}^{k-1} \int_{z_i}^{z_{i+1}} (q_i - z)^2 p(z) dz$
- Solution (Prove!):

$$q_i = \frac{\int_{z_i}^{z_{i+1}} z \cdot p(z) dz}{\int_{z_i}^{z_{i+1}} p(z) dz} \qquad z_i = \frac{q_{i-1} + q_i}{2}$$

Operation with LUT (4.2)

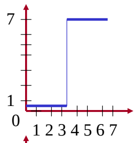
- Stretch

0	1	2	3	4	5	6	7
0	0	0	2	4	6	7	7



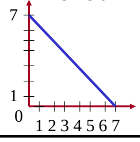
- Threshold

0	1	2	3	4	5	6	7
0	0	0	0	7	7	7	7



- Negative

0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0



1-D Discrete Convolution

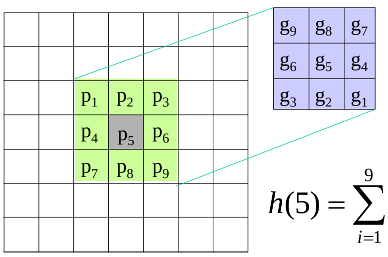
$$h(i) = (f * g)(i) = \sum_{k=1}^{n} f(k) g(i-k)$$

$f = (0 \quad 0 \quad 1 \quad 0 \quad 0)$ $\quad g = (0 \quad 0 \quad 1 \quad -1 \quad 0)$

$$h(6) = \sum_{k=1}^{5} f(k) \cdot g(6-k) = f(3) \cdot g(3) = 1$$

$$h(7) = \sum_{k=1}^{5} f(k) \cdot g(7-k) = f(3) \cdot g(4) = -1$$

2D Discrete Convolution



$$h(5) = \sum_{i=1}^{9} p(i) g(i)$$

2-D Discrete Convolution

$$h = f * g$$

$$h(i,j) = \sum_{k=1}^{n} \sum_{l=1}^{m} f(k,l) g(i-k, j-l)$$

Question: What is the complexity of convolution

Convolutions: Smoothing

Q: What is the average gray level after convolution?

Smoothing $\quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$


Original Image

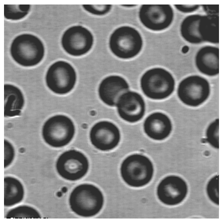

Corrupted Image

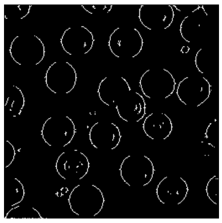

Filtered Image

Convolutions: Edge-Detection

Q: What is the average gray level after convolution?

Edge Detection $\quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$


Original Blood Image


Edge Map