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\textsuperscript{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

Quantization Error

- If pixel \( p \) with color \( (r,g,b) \) is coded by \( k \), a possible quantization error for \( p \) is:
  \[
  E_p^2 = (r - R_k)^2 + (g - G_k)^2 + (b - B_k)^2
  \]
- The total error introduced by a LUT is:
  \[
  E^2 = \sum_p E_p^2
  \]
- Unknowns: \( (r,g,b) \rightarrow k \)

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 \( q_i \)
- Uniform Quantization:
  \[
  q_i = \frac{z_i + z_{i+1}}{2} \\
  z_{i+1} - z_i = (z_k - z_0) / k
  \]
- Prior Mappings (e.g. Gamma Correction)

Optimal Quantization (6.5.1)

- Minimize the error:
  \[
  \sum_{i=0}^{k-1} \int (q_i - z)^2 p(z) dz
  \]
- Solution (Prove!):
  \[
  q_i = \frac{\int_{z_i}^{z_{i+1}} p(z) dz}{\int_{z_i}^{z_{i+1}} p(z) dz}
  \\
  z_i = \frac{q_{i-1} + q_i}{2}
  \]
Operation with LUT (4.2)

- **Stretch**
  \[
  \begin{bmatrix}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  0 & 0 & 0 & 2 & 4 & 6 & 7 & 7 \\
  \end{bmatrix}
  \]

- **Threshold**
  \[
  \begin{bmatrix}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  0 & 0 & 0 & 0 & 7 & 7 & 7 & 7 \\
  \end{bmatrix}
  \]

- **Negative**
  \[
  \begin{bmatrix}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
  \end{bmatrix}
  \]

### 1-D Discrete Convolution

\[
 h(i) = (f \ast g)(i) = \sum_{k=1}^{n} f(k)g(i-k)
\]

\[
 f = (0 0 1 0 0) \quad g = (0 0 1 \ -1 \ 0) 
\]

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

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

### 2D Discrete Convolution

\[
 h = f \ast g
\]

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

**Question**: What is the complexity of convolution?

### Convolutions: Smoothing

- **Q**: What is the average gray level after convolution?
  - **Smoothing**
    - \[
    \begin{bmatrix}
    1 & 1 & 1 \\
    1 & 1 & 1 \\
    1 & 1 & 1 \\
    \end{bmatrix}
    \]
    - \[
    \begin{bmatrix}
    1 & 2 & 1 \\
    2 & 4 & 2 \\
    1 & 2 & 1 \\
    \end{bmatrix}
    \]

- **Original Image**
- **Corrupted Image**
- **Filtered Image**

### Convolutions: Edge-Detection

- **Q**: What is the average gray level after convolution?
  - **Edge Detection**
    - \[
    \begin{bmatrix}
    1 & 0 & -1 \\
    1 & 0 & -1 \\
    1 & 0 & -1 \\
    \end{bmatrix}
    \]
    - \[
    \begin{bmatrix}
    -1 & 1 \\
    -1 & 1 \\
    \end{bmatrix}
    \]

- **Original Blood Image**
- **Edge Map**