Histogram Segmentation Code

```matlab
%smooth
for i=1:5
    hst=conv2(hst,gauss,'same');\end
end
%compute derivate
dr1=conv2(hst,dr,'same');\end
%find peaks and valleys
pw =dr1(2:254).*dr1(2:255);\end
peaks_valleys(find(pw<0))=1;\end
control=find(peaks_valleys==1);\end
for i=1:size(control,2)\end
    if (dr1(control(i)-2)>0)\end
        valleys(control(i))=1;\end
    end\end
    if (dr1(control(i)-2)<0)\end
        peaks(control(i))=1;\end
    end\end
peaklocs=find(peaks==1);\end
valleylocs=find(valleys==1);\end
current_valley=1;\end
for i=1:size(peaklocs,2)\end
    Va = hst(valleylocs(current_valley));\end
    Vb = hst(valleylocs(current_valley+1));\end
    P  = hst(peaklocs(i));\end
    W  = valleylocs(current_valley+1)-\end
        valleylocs(current_valley);\end
    N  = 0;\end
    for \end
        j=valleylocs(current_valley):valleylocs\end
            (current_valley+1)\end
            N = N + hst(j);\end
    end\end
    val1 = 1-((Va+Vb)/(2*P));\end
    val2 = 1-(N/(W*P));\end
    if (val1>0 && val2>0)\end
        peakiness(peaklocs(i)) = val1*val2;\end
    end;\end
    current_valley = current_valley +1;\end
end\end
```
Plots

Recap
Region Properties

- Area
- Centroid
- Moments
- Perimeter
- Compactness
- Orientation

\[ A = \sum_{x=0}^{m} \sum_{y=0}^{n} B(x, y) \]

\[ \bar{x} = \frac{\sum_{x=0}^{m} \sum_{y=0}^{n} x B(x, y)}{A} \]

\[ \bar{y} = \frac{\sum_{x=0}^{m} \sum_{y=0}^{n} y B(x, y)}{A} \]

\[ \mu_{pq} = \int \int (x - \bar{x})^p (y - \bar{y})^q B(x, y) \, d(x - \bar{x})d(y - \bar{y}) \]

\[ C = \frac{\text{Perimeter}^2}{4\pi A} \]
Recap
Region Orientation

\[ E = \iint r^2 B(x, y) \, dx \, dy \]

Axis of 2nd moment

\[ m_x = \sum_{i} x_i^2 B(x_i, y_i) \]
\[ m_y = \sum_{i} y_i^2 B(x_i, y_i) \]
\[ m_{xy} = \sum_{i} x_i y_i B(x_i, y_i) \]

Closest point to the origin

\[ x_i = -\rho \sin \theta \]
\[ y_i = \rho \cos \theta \]
Recap Region Orientation

\[ x_u = x_i + s \cos \theta \]
\[ y_u = y_i + s \sin \theta \]
\[ x_0 = (-\rho \sin \theta) + s \cos \theta \]
\[ y_0 = (\rho \cos \theta) + s \sin \theta \]

Substitute \( x_0 \) and \( y_0 \).

\[ r^2 = (x - x_0)^2 + (y - y_0)^2 \]
\[ x_0 = (-\rho \sin \theta) + s \cos \theta \]
\[ y_0 = (\rho \cos \theta) + s \sin \theta \]

Substitute \( x_0 \) and \( y_0 \).

\[ r^2 = (x + \rho \sin \theta - s \cos \theta)^2 + (y - \rho \cos \theta - s \sin \theta)^2 \]
\[ r^2 = x^2 + y^2 + \rho^2 + s^2 - 2s(x \cos \theta + y \sin \theta) + 2\rho(x \sin \theta - y \cos \theta) \]
Recap
Region Orientation

\[ r^2 = x^2 + y^2 + \rho^2 + s^2 - 2s(x \cos \theta + y \sin \theta) + 2\rho(x \sin \theta - y \cos \theta) \]

\[ \frac{\partial}{\partial s} r^2 = 2s - 2(x \cos \theta + y \sin \theta) \]

\[ s = x \cos \theta + y \sin \theta \]

Substitute \( s \) back to \( r \)

---

Recap
Region Orientation

\[ r^2 = x^2 + y^2 + \rho^2 + s^2 - 2s(x \cos \theta + y \sin \theta) + 2\rho(x \sin \theta - y \cos \theta) \]

\[ r^2 = x^2 + y^2 + \rho^2 - (x \cos \theta + y \sin \theta)^2 + 2\rho(x \sin \theta - y \cos \theta) \]

\[ r^2 = x^2 + y^2 + \rho^2 - x^2 \cos^2 \theta + 2xy \sin \theta \cos \theta - y^2 \sin^2 \theta + 2\rho(x \sin \theta - y \cos \theta) \]

\[ r^2 = x^2 [1 - \cos^2 \theta] + y^2 [1 - \sin^2 \theta] - 2xy \sin \theta \cos \theta + 2\rho(x \sin \theta - y \cos \theta) + \rho^2 \]

\[ r^2 = x^2 \sin^2 \theta + y^2 \cos^2 \theta - 2xy \sin \theta \cos \theta + 2\rho(x \sin \theta + y \cos \theta) + \rho^2 \]

\[ r^2 = (x \sin \theta - y \cos \theta)^2 + 2\rho(x \sin \theta - y \cos \theta) + \rho^2 \]

\[ r^2 = (x \sin \theta - y \cos \theta + \rho)^2 \]

Alper Yilmaz, Fall 2005 UCF
Recap
Region Orientation

\[ a = x \sin \theta - y \cos \theta \]

\[ \frac{\partial}{\partial \rho} E = \frac{\partial}{\partial \rho} \left( \int \int (x \sin \theta - y \cos \theta + \rho) B(x, y) \, dx \, dy \right) \]

\[ \frac{\partial}{\partial \rho} \rho^2 = \frac{\partial}{\partial \rho} \left( \int \int (a^2 + 2a\rho + \rho^2) B(x, y) \, dx \, dy \right) \]

\[ \int \int 2aB(x, y) \, dx \, dy + \int \int 2\rho B(x, y) \, dx \, dy = 0 \]

\[ \sin \theta \int \int xB(x, y) \, dx \, dy - \cos \theta \int \int yB(x, y) \, dx \, dy + \rho A = 0 \]

\[ \rho = -\sin \theta \int \int xB(x, y) \, dx \, dy + \cos \theta \int \int yB(x, y) \, dx \, dy \]

\[ A \]

\[ \rho = -\bar{x} \sin \theta + \bar{y} \cos \theta \]

Substitute \( \rho \) back to \( r \)

\[ E = \int \int ((x - \bar{x}) \sin \theta - (y - \bar{y}) \cos \theta)^2 B(x, y) \, dx \, dy \]

\[ x' = x - \bar{x}, \quad y' = y - \bar{y} \]

\[ E = \int \int (x' \sin \theta - y' \cos \theta)^2 B(x, y) \, dx \, dy \]

\[ E = \sin^2 \theta \int \int x'^2 B(x, y) \, dx \, dy - 2 \sin \theta \cos \theta \int \int x'y B(x, y) \, dx \, dy + \cos^2 \theta \int \int y^2 B(x, y) \, dx \, dy \]

\[ E = a \sin^2 \theta - 2b \sin \theta \cos \theta + c \cos^2 \theta \]

take derivative wrt \( \theta \) and equating it to 0

\[ \tan 2\theta = \frac{b}{a-c} \]
Homework from last lecture

- Show that following holds \( E_1 = E_2 \)
  where

\[
E_1 = a \sin^2 \theta - b \sin \theta \cos \theta + c \cos^2 \theta
\]

\[
E_2 = \frac{1}{2} (a + c) - \frac{1}{2} (a - c) \cos 2\theta - \frac{1}{2} b \sin 2\theta
\]
Gaussian Pyramid

- Very useful for representing images
- Image Pyramid is built by using multiple copies of image at different scales.
- Each level in the pyramid is ¼ of the size of previous level
- The highest level is of the highest resolution
- The lowest level is of the lowest resolution
A bar in the big images is a hair on the zebra's nose; in smaller images, a stripe; in the smallest, the animal’s nose

Aliasing Problem

- Can't shrink an image by taking every second pixel
- If we do, characteristic errors appear
  - In the next few slides
  - Typically, small phenomena look bigger; fast phenomena can look slower
  - Common phenomenon
    - Wagon wheels rolling the wrong way in movies
    - Checkerboards misrepresented in ray tracing
    - Striped shirts look funny on color television
• Resample the checkerboard by taking one sample at each circle. In the case of the top left board, new representation is reasonable.
• Top right also yields a reasonable representation. Bottom left is all black (dubious) and bottom right has checks that are too big.

Pyramid With Every Other Pixel

Constructing a pyramid by taking every second pixel leads to layers that badly misrepresent the top layer.
Sampling

- 1D

- 2D

Smoothing

- High frequencies (edges) lead to trouble with sampling.
- Solution: suppress edges before sampling
- Common solution: use a Gaussian
  - Convolve image with Gaussian filter
Gaussian Filter

\[ G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \]

- Gaussian * Gaussian = another Gaussian
- Symmetric filter
- Gaussians are low pass filters
  - Removes noise

Scaled images

Without smoothing

With smoothing
  - Sigma 1 pixel
  - Sigma 1.4 pixel
Applications of Scaled Images

- Search for correspondence
  - look at coarse scales, then refine with finer scales
- Edge tracking
  - “Good” edge at a fine scale has parents at a coarser scale
- Control of detail and computational cost in matching
  - Finding stripes
  - Very important in texture representation

Gaussian Pyramid

- Let $w$ be Gaussian filter

$$g_i(i) = \sum_{m=-2}^{2} \hat{w}(m)g_{i+1}(2i+m)$$

$$g_i(2) = \hat{w}(-2)g_{i-1}(4-2) + \hat{w}(-1)g_{i-1}(4-1) + \hat{w}(0)g_{i-1}(4) + \hat{w}(1)g_{i-1}(4+1) + \hat{w}(2)g_{i-1}(4+2)$$

$$g_i(2) = \hat{w}(-2)g_{i-1}(2) + \hat{w}(-1)g_{i-1}(3) + \hat{w}(0)g_{i-1}(4) + \hat{w}(1)g_{i-1}(5) + \hat{w}(2)g_{i-1}(6)$$

$$g_i = \text{REDUCE}[g_{i-1}]$$
Convolution Kernel $w$

$[w(-2), w(-1), w(0), w(1), w(2)]$

- Symmetric
  $$w(i) = w(-i) \Rightarrow [c, b, a, b, c]$$
- Sum must be 1
  $$a + 2b + 2c = 1$$

---

Reduce 1D Convolution Kernel $w$

- All nodes at a given level must contribute the same total weight to the nodes at the next higher level

$$a + 2c = 2b$$
Convolution Kernel (5x1)

\[ w(0) = a \]
\[ w(-1) = w(1) = \frac{1}{4} \quad a=0.4 \text{ GAUSSIAN} \]
\[ w(-2) = w(2) = \frac{1}{4} - \frac{a}{2} \quad a=0.5 \text{ TRIANGULAR} \]

Triangular (5x1)
Approximate Gaussian (5x1)

Two Dimensions?

- Gaussian is separable

\[ \tilde{I}(x,y) = I(x,y) \ast G(x,y) \]

\[ \tilde{I}(x,y) = I(x,y) \ast G(x) \ast G(y) \]

\[ G(x) = G^T(y) \text{ transpose} \]
Gaussian Pyramid Algorithm

- Apply 1D mask to alternate pixels along each row of image.
- Apply 1D mask to alternate pixels along each column of resulting image from previous step.
Expand 1D

\[ g_{l,n}(i) = \sum_{p=-2}^{2} \hat{\omega}(p) g_{l,n-1}(\frac{i-p}{2}) \]

\[ g_{l,n}(4) = \hat{\omega}(-2) g_{l,n-1}(\frac{4-2}{2}) + \hat{\omega}(-1) g_{l,n-1}(\frac{4-1}{2}) + \]
\[ \hat{\omega}(0) g_{l,n-1}(\frac{4}{2}) + \hat{\omega}(1) g_{l,n-1}(\frac{4+1}{2}) + \hat{\omega}(2) g_{l,n-1}(\frac{4+2}{2}) \]

\[ g_{l,n}(4) = \hat{\omega}(-2) g_{l,n-1}(1) + \hat{\omega}(0) g_{l,n-1}(2) + \hat{\omega}(2) g_{l,n-1}(3) \]
The Laplacian Pyramid

- Similar to edge detected images
- Most pixels are zero
- Can be used in image compression
Constructing Laplacian Pyramid

- Compute Gaussian pyramid
  \(g_k, g_{k-1}, g_{k-2}, \ldots, g_2, g_1\)

- Compute Laplacian pyramid as follows:

  \[
  L_k = g_k - \text{EXPAND}(g_{k-1}) \\
  L_{k-1} = g_{k-1} - \text{EXPAND}(g_{k-2}) \\
  L_{k-2} = g_{k-2} - \text{EXPAND}(g_{k-3}) \\
  \vdots \\
  L_1 = g_1
  \]
Reconstructing Image

\[ g_1 = L_1 \]
\[ g_2 = \text{EXPAND}(g_1) + L_2 \]
\[ g_3 = \text{EXPAND}(g_2) + L_3 \]
\[ \vdots \]
\[ g_k = \text{EXPAND}(g_{k-1}) + L_k \]

The Laplacian Pyramid

- **Synthesis (Coding)**
  - Compute Gaussian pyramid
  - Compute Laplacian pyramid

- **Analysis (Decoding)**
  - Compute Gaussian pyramid from Laplacian pyramid
  - \( g_1 \) is reconstructed image

\[ L_1 = g_4 - \text{EXPAND}[g_3] \]
\[ L_2 = g_3 - \text{EXPAND}[g_2] \]
\[ L_3 = g_2 - \text{EXPAND}[g_1] \]
\[ L_4 = g_4 \]
\[ g_4 = L_4 \]
\[ g_1 = \text{EXPAND}[g_4] + L_1 \]
\[ g_2 = \text{EXPAND}[g_1] + L_2 \]
\[ g_1 = \text{EXPAND} [g_2] + L_1 \]
Programming Assignment

- Write the Gaussian Pyramid algorithm
  - The algorithm should be capable of providing any number of resolutions
  - Report should include scaled images of the Lenna image.
  - Due 12 October, 2005