Pyramids

Lecture-5

Pyramids

• Very useful for representing images.
• Pyramid is built by using multiple copies of image.
• Each level in the pyramid is 1/4 of the size of previous level.
• The lowest level is of the highest resolution.
• The highest level is of the lowest resolution.
Pyramid

Gaussian Pyramids

\[ g_l(i, j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m, n) g_{l-1}(2i+m, 2j+n) \]

\[ g_l = REDUCE[g_{l-1}] \]
Convolution

Reduce (1D)

\[ g_l(i) = \sum_{m=-2}^{2} \hat{w}(m)g_{l-1}(2i+m) \]

\[ g_l(2) = \hat{w}(-2)g_{l-1}(4-2) + \hat{w}(-1)g_{l-1}(4-1) + \hat{w}(0)g_{l-1}(4) + \hat{w}(1)g_{l-1}(4+1) + \hat{w}(2)g_{l-1}(4+2) \]

\[ g_l(2) = \hat{w}(-2)g_{l-1}(2) + \hat{w}(-1)g_{l-1}(3) + \hat{w}(0)g_{l-1}(4) + \hat{w}(1)g_{l-1}(5) + \hat{w}(2)g_{l-1}(6) \]
Reduce

Gaussian Pyramids

\[ g_{l,n}(i, j) = \sum_{p=-2}^{2} \sum_{q=-2}^{2} w(p, q) g_{l,n-1}\left(\frac{i-p}{2}, \frac{j-q}{2}\right) \]

\[ g_{l,n} = EXPAND[g_{l,n-1}] \]
Expand (1D)

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

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

\[ g_{l,n}(4) = \hat{w}(-2)g_{l,n-1}(1) + \hat{w}(0)g_{l,n-1}(2) + \hat{w}(2)g_{l,n-1}(3) \]
Expand

Convolution Mask

\[ [w(-2), w(-1), w(0), w(1), w(2)] \]
Convolution Mask

• Separable

\[ w(m, n) = \hat{w}(m) \hat{w}(n) \]

• Symmetric

\[ \hat{w}(i) = \hat{w}(-i) \]

[\(c, b, a, b, c\)]

Convolution Mask

• The sum of mask should be 1.

\[ a + 2b + 2c = 1 \]

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

\[ a + 2c = 2b \]
Convolution Mask

\[
\hat{w}(0) = a \\
\hat{w}(-1) = \hat{w}(1) = \frac{1}{4} \\
\hat{w}(-2) = \hat{w}(2) = \frac{1}{4} - \frac{a}{2}
\]

a=.4 GAUSSIAN, a=.5 TRINGULAR
Triangular

Approximate Gaussian
Gaussian

\[ g(x) = e^{-\frac{x^2}{2\sigma^2}} \]
Separability
Algorithm

• Apply 1-D mask to alternate pixels along each row of image.
• Apply 1-D mask to each pixel along alternate columns of resultant image from previous step.

Gaussian Pyramid
Laplacian Pyramids

- Similar to edge detected images.
- Most pixels are zero.
- Can be used for image compression.

\[ L_1 = g_1 - \text{EXPAND}[g_2] \]
\[ L_2 = g_2 - \text{EXPAND}[g_3] \]
\[ L_3 = g_3 - \text{EXPAND}[g_4] \]
Coding using Laplacian Pyramid

• Compute Gaussian pyramid

\[ g_1, g_2, g_3, g_4 \]

• Compute Laplacian pyramid

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

• Code Laplacian pyramid

Decoding using Laplacian pyramid

• Decode Laplacian pyramid.
• Compute Gaussian pyramid from Laplacian pyramid.

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

• is reconstructed image.
Gaussian

- Most natural phenomenon can be modeled by Gaussian.
- Take a bunch of random variables of any distribution, find the mean, the mean will approach to Gaussian distribution.
- Gaussian is very smooth function, it has infinite no of derivatives.

Gaussian

- Fourier Transform of Gaussian is Gaussian.
- If you convolve Gaussian with itself, it is again Gaussian.
- There are cells in human brain which perform Gaussian filtering.
  - Laplacian of Gaussian edge detector
Carl F. Gauss

- Born to a peasant family in a small town in Germany.
- Learned counting before he could talk.
- Contributed to Physics, Mathematics, Astronomy,…
- Discovered most methods in modern mathematics, when he was a teenager.

Carl F. Gauss

- Some contributions
  - Gaussian elimination for solving linear systems
  - Gauss-Seidel method for solving sparse systems
  - Gaussian curvature
  - Gaussian quadrature
Laplacian Pyramid

Image Compression (Entropy)

7.6

4.4

5.0

5.6

6.2

1.9

3.3

4.2
Huffman Coding (Example-1)

Huffman Coding

Entropy \[ H = -\sum_{i=0}^{255} p(i) \log_2 p(i) \]

\[ H = -0.5 \log_0.5 - 0.25 \log_0.25 - 0.125 \log_0.125 - 0.125 \log_0.125 = 1.75 \]
Image Compression

1.58

1

.73

(c)

(d)

Combining Apple & Orange
Combining Apple & Orange

Algorithm

- Generate Laplacian pyramid Lo of orange image.
- Generate Laplacian pyramid La of apple image.
- Generate Laplacian pyramid Lc by copying left half of nodes at each level from apple and right half of nodes from orange pyramids.
- Reconstruct combined image from Lc.
Lucas Kanade with Pyramids

- Compute ‘simple’ LK at highest level
- At level $i$
  - Take flow $u_{i-1}$, $v_{i-1}$ from level $i-1$
  - bilinear interpolate it to create $u_i^*$, $v_i^*$ matrices of twice resolution for level $i$
  - multiply $u_i^*$, $v_i^*$ by 2
  - compute $f_i$ from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections $u_i'$, $v_i'$, i.e. $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$. 

- [http://www-bcs.mit.edu/people/adelson/papers.html](http://www-bcs.mit.edu/people/adelson/papers.html)
Pyramids

\[ u_i = u_i^* + u^*, v_i = v_i^* + v^* \]

\( f_1 \) pyramid \quad \( f_2 \) pyramid

Interpolation

\[
\begin{array}{ccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & o & o & o & o & o & o & o \\
u = 1 & o & o & o & o & o & o & o \\
2 & o & o & o & o & o & o & o \\
3 & o & o & o & o & o & o & o \\
\end{array}
\]

\[
\begin{array}{ccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & o & o & o & o & o & o & o \\
v = 1 & o & o & o & o & o & o & o \\
2 & o & o & o & o & o & o & o \\
v' = 3 & o & o & o & o & o & o & o \\
\end{array}
\]
1-D Interpolation

\[ y = mx + c \]
\[ f(x) = mx + c \]

\[ f(1) \quad f(2) \]

2-D Interpolation

\[ f(x,y) = a_1 + a_2 x + a_3 y + a_4 xy \quad \text{Bilinear} \]

\[
\begin{array}{cc}
X & X \\
O & X \\
\end{array}
\]
Bi-linear Interpolation

Four nearest points of \((x,y)\) are:

\[(x, y), (x, y), (x, y), (x, y)\]

\[(3,5), (4,5), (3,6), (4,6)\]

\[\bar{x} = \text{int}(x) \quad 3 \quad (3.2, 5.6)\]

\[\bar{y} = \text{int}(y) \quad 5 \quad X (3,6) \quad X (4,6)\]

\[\bar{x} = x + 1 \quad 4 \quad X(3,5) \quad X (4,5)\]

\[\bar{y} = y + 1 \quad 6\]

\[
f''(x, y) = \bar{\varepsilon}_x \bar{\varepsilon}_y f(x, y) + \bar{\varepsilon}_x \bar{\varepsilon}_y f(x, y) + \bar{\varepsilon}_x \bar{\varepsilon}_y f(x, y) + \bar{\varepsilon}_x \bar{\varepsilon}_y f(x, y)
\]

\[
\bar{\varepsilon}_x = x - \bar{x} \quad \bar{\varepsilon}_y = y - \bar{y} \quad \bar{\varepsilon}_x = x - \bar{x} = 4 - 3.2 = 0.8 \\
\bar{\varepsilon}_y = y - \bar{y} \quad \bar{\varepsilon}_y = y - \bar{y} = 6 - 5.6 = 0.4 \\
\bar{\varepsilon}_x = x - \bar{x} \quad \bar{\varepsilon}_x = x - \bar{x} = 3.2 - 2 = 0.2 \\
\bar{\varepsilon}_y = y - \bar{y} \quad \bar{\varepsilon}_y = y - \bar{y} = 5.6 - 5 = 0.6
\]
Lucas-Kanade without pyramids

Fails in areas of large motion

Lucas-Kanade with Pyramids