Image Processing – lesson 8

**Image Representation**

Quad Trees
Gaussian pyramids
Laplacian Pyramids
Wavelet Pyramids
Applications
Quad Trees

Quad tree image representation = a tree representation which represents recursive subdivisions of an image.

Example:
Quad tree representation of an image
Quad Tree representation

Image

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

```
1 2 3 4
```

Quad Tree:

1. Let's assume the image is divided into four quadrants.
2. Each quadrant is represented by a node in the tree.
3. If all the pixels in a quadrant are the same color, the quadrant is represented by a leaf node.
4. If the quadrant contains pixels of different colors, it is split into four smaller quadrants, each represented by a child node.
5. This process continues recursively until all quadrants are leaf nodes.
6. The root of the tree represents the entire image.

The image and its corresponding quad tree representation are shown in the diagram.
Quad Tree Applications:

- Compression
- Segmentation (Split & Merge)
- Smoothing
- Binary Image Operations ("And" "Or" "Not")
Quad Tree Representation - Example
Quad Tree Representation - Example
Quad Tree Representation

Original

Thresh = 0.20

Thresh = 0.40

Thresh = 0.55
Binary Operations Using Quad Trees

Circle = 1  

Circle = 0

“Not”

“And”

“And”

“And”

“And”
Image Pyramids

Image features at different scales require filters at different scales.

Edges (derivatives):

\[ f(x) \]

\[ f'(x) \]

Objects (correlation):
Image Pyramid = Hierarchical representation of an image

Image Pyramid = A collection of images at different resolutions.

Low Resolution

No details in image - (blurred image)
low frequencies

High Resolution

Details in image - low+high frequencies
Image Pyramid

Low resolution

High resolution
Image Blurring = low pass filtering
Gaussian Pyramid

- Level M
  1 x 1

- Level 1
  $2^{n-1} \times 2^{n-1}$

- Level 0
  $2^n \times 2^n$

$w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4$

$w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4$
Gaussian Pyramid

Burt & Adelson (1981)

Normalized: $\Sigma w_i = 1$

Symmetry: $w_i = w_{-i}$

Unimodal: $w_i \geq w_j$ for $0 < i < j$

Equal Contribution: for all $j$ $\Sigma w_{j+2i} = $ constant

\[
\begin{align*}
  a + 2b + 2c &= 1 \\
  a + 2c &= 2b \\
  a > 0.25 \\
  b &= 0.25 \\
  c &= 0.25 - a/2
\end{align*}
\]
For $a = 0.4$ most similar to a Gaussian filter

$$g = [0.05 \ 0.25 \ 0.4 \ 0.25 \ 0.05]$$

$$\text{low_pass_filter} = g \ast g' =$$

$$\begin{bmatrix}
0.0025 & 0.0125 & 0.0200 & 0.0125 & 0.0025 \\
0.0125 & 0.0625 & 0.1000 & 0.0625 & 0.0125 \\
0.0200 & 0.1000 & 0.1600 & 0.1000 & 0.0200 \\
0.0125 & 0.0625 & 0.1000 & 0.0625 & 0.0125 \\
0.0025 & 0.0125 & 0.0200 & 0.0125 & 0.0025 \\
\end{bmatrix}$$
<table>
<thead>
<tr>
<th>image</th>
<th>pattern</th>
<th>correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="image" /></td>
<td><img src="pattern1.png" alt="pattern" /></td>
<td><img src="correlation1.png" alt="correlation" /></td>
</tr>
<tr>
<td><img src="image2.png" alt="image" /></td>
<td><img src="pattern2.png" alt="pattern" /></td>
<td><img src="correlation2.png" alt="correlation" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="image" /></td>
<td><img src="pattern3.png" alt="pattern" /></td>
<td><img src="correlation3.png" alt="correlation" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="image" /></td>
<td><img src="pattern4.png" alt="pattern" /></td>
<td><img src="correlation4.png" alt="correlation" /></td>
</tr>
</tbody>
</table>
Gaussian Pyramid - Computational Aspects

Memory:

\[ 2^N \times 2^N \times \left(1 + \frac{1}{4} + \frac{1}{16} + \ldots\right) = 2^N \times 2^N \times \frac{4}{3} \]

Computation:

Level \( i \) can be computed with a single convolution with filter: \( h_i = g \ast g \ast g \ast \ldots \)

Example:

\[ h_2 = g \ast g = \]

\[ g \ast g \ast g \ast \ldots \]

\[ i \text{ times} \]
Laplacian Pyramid

Compression - compression rates are higher for predictable values. e.g. values around 0.

\[ G_0, G_1, \ldots = \text{the levels of a Gaussian Pyramid.} \]

Predict level \( G_l \) from level \( G_{l+1} \) by Expanding \( G_{l+1} \) to obtain \( G'_l \)

\[
\text{Denote by } L_l \text{ the error in prediction:}
\]

\[
L_l = G_l - G'_l
\]

\( L_0, L_1, \ldots = \text{the levels of a Laplacian Pyramid.} \)
Gaussian Pyramid

Laplacian Pyramid

expand

expand

expand

= 

= 

= 

=
Laplacian Pyramid - Example
Laplacian Pyramid - No scaling
Reconstruction of the original image from the Laplacian Pyramid

Laplacian Pyramid

Expand

Expand

Expand

Original Image
Laplacian Pyramid - Computational Aspects

Memory:

\[ 2^{N \times 2^N} (1 + 1/4 + 1/16 + \ldots) = 2^{N \times 2^N} \times 4/3 \]

However, coefficients are highly compressable.

Computation:

\[ L_i \] can be computed from \( G_0 \) with a single convolution with filter: \( k_i = h_{i-1} - h_i \)
When splining two images, transition from one image to the other should behave:

- **High Frequencies**
- **Middle Frequencies**
- **Low Frequencies**
Wavelet Decomposition

Fourier Space
Wavelet Transform - Example
Wavelet Transform - Example
# Image Pyramids - Comparison

<table>
<thead>
<tr>
<th>Transform</th>
<th>Basis</th>
<th>Frequency</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier</td>
<td>Sines+Cosines</td>
<td></td>
<td>Not localized in space&lt;br&gt;Localized in Frequency</td>
</tr>
<tr>
<td>Gaussian Pyramid</td>
<td>Gaussian Filters</td>
<td></td>
<td>Localized in space&lt;br&gt;Not localized in Frequency</td>
</tr>
<tr>
<td>Laplacian Pyramid</td>
<td>Laplacian Filters</td>
<td></td>
<td>Localized in space&lt;br&gt;Not localized in Frequency</td>
</tr>
<tr>
<td>Wavelet Pyramid</td>
<td>Wavelet Filters</td>
<td></td>
<td>Localized in space&lt;br&gt;Localized in Frequency</td>
</tr>
</tbody>
</table>
Image Pyramids - Comparison

Image pyramid levels = Filter then sample.

Filters:

- Gaussian Pyramid
- Laplacian Pyramid
- Wavelet Pyramid