Texture

Texture is a description of the spatial arrangement of color or intensities in an image or a selected region of an image.

Structural approach: a set of texels in some regular or repeated pattern

Problem with Structural Approach

How do you decide what is a texel?

Ideas?
Natural Textures from VisTex

What/Where are the texels?

The Case for Statistical Texture

- Segmenting out texels is difficult or impossible in real images.
- Numeric quantities or statistics that describe a texture can be computed from the gray tones (or colors) alone.
- This approach is less intuitive, but is computationally efficient.
- It can be used for both classification and segmentation.

Some Simple Statistical Texture Measures

1. Edge Density and Direction

   - Use an edge detector as the first step in texture analysis.
   - The number of edge pixels in a fixed-size region tells us how busy that region is.
   - The directions of the edges also help characterize the texture

Two Edge-based Texture Measures

1. edgeness per unit area

   \[
   F_{\text{edgeness}} = \frac{\left| \{ p \mid \text{gradient magnitude}(p) \geq \text{threshold} \} \right|}{N}
   \]

   where \( N \) is the size of the unit area

2. edge magnitude and direction histograms

   \[
   F_{\text{magdir}} = (H_{\text{magnitude}}, H_{\text{direction}})
   \]

   where these are the normalized histograms of gradient magnitudes and gradient directions, respectively.

How would you compare two histograms?
Local Binary Partition Measure

- For each pixel p, create an 8-bit number \( b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8 \), where \( b_i = 0 \) if neighbor \( i \) has value less than or equal to p’s value and 1 otherwise.

- Represent the texture in the image (or a region) by the histogram of these numbers.

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
100 & 101 & 103 & \text{1111100}
\end{array}
\]

Fids (Flexible Image Database System) is retrieving images similar to the query image using LBP texture as the texture measure and comparing their LBP histograms.

Low-level measures don’t always find semantically similar images.
Co-occurrence Matrix Features

A co-occurrence matrix is a 2D array \( C \) in which

- Both the rows and columns represent a set of possible image values.
- \( C_{d}(i,j) \) indicates how many times value \( i \) co-occurs with value \( j \) in a particular spatial relationship \( d \).
- The spatial relationship is specified by a vector \( d = (dr, dc) \).

\[
\begin{array}{cccc}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 2 & 2 \\
0 & 0 & 2 & 2 \\
0 & 0 & 2 & 2 \\
0 & 0 & 2 & 2
\end{array}
\]

(Co-occurrence matrix)

From \( C_{d} \) we can compute \( N_{d} \), the normalized co-occurrence matrix, where each value is divided by the sum of all the values.

Co-occurrence Features

What do these measure?

- **Energy** measures uniformity of the normalized matrix.

\[
\text{Energy} = \sum_{i} \sum_{j} N_{d}^{2}(i,j)
\]

(7.7)

- **Entropy** measures the amount of information in the normalized matrix.

\[
\text{Entropy} = -\sum_{i} \sum_{j} N_{d}(i,j) \log_{2} N_{d}(i,j)
\]

(7.8)

- **Contrast** measures the local variation of the normalized matrix.

\[
\text{Contrast} = \sum_{i} \sum_{j} (i-j)^{2} N_{d}(i,j)
\]

(7.9)

- **Homogeneity** measures the local uniformity of the normalized matrix.

\[
\text{Homogeneity} = \sum_{i} \sum_{j} \frac{N_{d}(i,j)}{1+|i-j|}
\]

(7.10)

- **Correlation** measures the linear dependence of the normalized matrix.

\[
\text{Correlation} = \sum_{i} \sum_{j} \frac{(i-\mu_{i})(j-\mu_{j})N_{d}(i,j)}{\sigma_{i}\sigma_{j}}
\]

(7.11)

where \( \mu_{i}, \sigma_{i} \) are the means and \( \sigma_{i}, \sigma_{j} \) are the standard deviations of the row and column sums.

But how do you choose \( d \)?

- This is actually a critical question with all the statistical texture methods.

- Are the “texels” tiny, medium, large, all three …?

- Not really a solved problem.

Zucker and Terzopoulos suggested using a \( \chi^{2} \) statistical test to select the value(s) of \( d \) that have the most structure for a given class of images. See transparencies.
Laws’ Texture Energy Features

- Signal-processing-based algorithms use texture filters applied to the image to create filtered images from which texture features are computed.

- The Laws Algorithm
  - **Filter** the input image using texture filters.
  - **Compute texture energy** by summing the absolute value of filtering results in local neighborhoods around each pixel.
  - **Combine features** to achieve rotational invariance.

Law’s texture masks (1)

<table>
<thead>
<tr>
<th>Mask</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>L5</td>
<td>Level</td>
</tr>
<tr>
<td>E5</td>
<td>Edge</td>
</tr>
<tr>
<td>S5</td>
<td>Spot</td>
</tr>
<tr>
<td>R5</td>
<td>Ripple</td>
</tr>
</tbody>
</table>

- (L5) (Gaussian) gives a center-weighted local average
- (E5) (gradient) responds to row or col step edges
- (S5) (LOG) detects spots
- (R5) (Gabor) detects ripples

Law’s texture masks (2)

Creation of 2D Masks

- 1D Masks are “multiplied” to construct 2D masks:
  - mask E5L5 is the “product” of E5 and L5
  
  \[
  \begin{bmatrix}
  -1 & -4 & -6 & -4 & -1 \\
  -2 & -8 & -12 & -8 & -1 \\
  0 & 0 & 0 & 0 & 0 \\
  2 & 8 & 12 & 8 & 2 \\
  1 & 4 & 6 & 4 & 1 \\
  \end{bmatrix}
  \]

9D feature vector for pixel

- Subtract mean neighborhood intensity from pixel
- Dot product 16 5x5 masks with neighborhood
- 9 features defined as follows:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>L5E5/E5L5</td>
<td>L5S5/S5L5</td>
</tr>
<tr>
<td>L5R5/R5L5</td>
<td>E5E5</td>
</tr>
<tr>
<td>E5S5/S5E5</td>
<td>E5R5/R5E5</td>
</tr>
<tr>
<td>S5S5</td>
<td>S5R5/R5S5</td>
</tr>
<tr>
<td>R5R5</td>
<td></td>
</tr>
</tbody>
</table>
Table 7.2: Laws texture energy measures for major regions of the images of Figure 7.8.

<table>
<thead>
<tr>
<th>Region</th>
<th>E006</th>
<th>E005</th>
<th>R005</th>
<th>E015</th>
<th>R015</th>
<th>R010</th>
<th>R005</th>
<th>E010</th>
<th>R010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiger</td>
<td>168.1</td>
<td>84.4</td>
<td>207.7</td>
<td>155.7</td>
<td>134.4</td>
<td>101.6</td>
<td>116.3</td>
<td>339.2</td>
<td>217.4</td>
</tr>
<tr>
<td>Water</td>
<td>69.5</td>
<td>93.8</td>
<td>186.5</td>
<td>53.7</td>
<td>143.3</td>
<td>59.4</td>
<td>49.6</td>
<td>139.1</td>
<td>171.3</td>
</tr>
<tr>
<td>Flag</td>
<td>258.1</td>
<td>113.6</td>
<td>297.7</td>
<td>107.6</td>
<td>362.2</td>
<td>203.5</td>
<td>132.4</td>
<td>311.5</td>
<td>356.8</td>
</tr>
<tr>
<td>Fence</td>
<td>199.5</td>
<td>80.7</td>
<td>624.3</td>
<td>761.7</td>
<td>177.5</td>
<td>801.1</td>
<td>126.8</td>
<td>287.5</td>
<td>215.6</td>
</tr>
<tr>
<td>Grass</td>
<td>245.5</td>
<td>103.4</td>
<td>605.7</td>
<td>626.3</td>
<td>136.3</td>
<td>111.6</td>
<td>148.6</td>
<td>427.5</td>
<td>325.6</td>
</tr>
<tr>
<td>Small flowers</td>
<td>116.9</td>
<td>48.6</td>
<td>285.1</td>
<td>465.6</td>
<td>94.3</td>
<td>48.3</td>
<td>73.6</td>
<td>162.2</td>
<td>168.3</td>
</tr>
<tr>
<td>Big flowers</td>
<td>17.9</td>
<td>29.4</td>
<td>177.1</td>
<td>361.5</td>
<td>36.4</td>
<td>210.9</td>
<td>45.8</td>
<td>89.7</td>
<td>62.9</td>
</tr>
<tr>
<td>Border</td>
<td>11.3</td>
<td>6.4</td>
<td>64.4</td>
<td>92.8</td>
<td>36.3</td>
<td>74.8</td>
<td>9.7</td>
<td>26.1</td>
<td>19.5</td>
</tr>
</tbody>
</table>

**Interpreting autocorrelation**

- Coarse texture → function drops off slowly
- Fine texture → function drops off rapidly
- Can drop differently for r and c
- Regular textures → function will have peaks and valleys; peaks can repeat far away from [0, 0]
- Random textures → only peak at [0, 0]; breadth of peak gives the size of the texture
Fourier power spectrum

- High frequency power → fine texture
- Concentrated power → regularity
- Directionality → directional texture

What else?

- Gabor filters
- Wold decomposition
- Global Signatures (CANDID)
- Second Moment Matrix (Belongie paper)
- DOOG filter
  
  etc.