Image Processing

Cosimo Distanté

Lecture: Texture
Today: Texture

What defines a texture?
Includes: more regular patterns
Includes: more random patterns
Scale: objects vs. texture

Often the same thing in the world can occur as texture or an object, depending on the scale we are considering.
Why analyze texture?

Importance to perception:

• Often indicative of a material’s properties
• Can be important appearance cue, especially if shape is similar across objects
• Aim to distinguish between shape, boundaries, and texture

Technically:

• Representation-wise, we want a feature one step above “building blocks” of filters, edges.
Texture-related tasks

• Shape from texture
  – Estimate surface orientation or shape from image texture
Shape from texture

• Use deformation of texture from point to point to estimate surface shape

Texture-related tasks

- **Shape from texture**
  - Estimate surface orientation or shape from image texture

- **Segmentation/classification from texture cues**
  - Analyze, represent texture
  - Group image regions with consistent texture

- **Synthesis**
  - Generate new texture patches/images given some examples
Recall: These looked very similar in terms of their color distributions (when our features were R-G-B)

But how would their *texture* distributions compare?
Psychophysics of texture

• Some textures distinguishable with preattentive perception—without scrutiny, eye movements [Julesz 1975]

Same or different?
• Textons: analyze the texture in terms of statistical relationships between fundamental texture elements, called “textons”.
• It generally required a human to look at the texture in order to decide what those fundamental units were...
Texture representation

• Textures are made up of repeated local patterns, so:
  – Find the patterns
    • Use filters that look like patterns (spots, bars, raw patches...)
    • Consider magnitude of response
  – Describe their statistics within each local window
    • Mean, standard deviation
    • Histogram
    • Histogram of “prototypical” feature occurrences
Texture representation: example

- Original image
- Derivative filter responses, squared
- Statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Win. #1</th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Texture representation: example

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th></th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows
Texture representation: example

Original image

Derivative filter responses, squared

Statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Window</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Texture representation: example

Original image

Derivative filter responses, squared

Statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Window</th>
<th>Mean $d/\text{dx}$</th>
<th>Mean $d/\text{dy}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Texture representation: example

<table>
<thead>
<tr>
<th></th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows
Texture representation: example

Windows with primarily horizontal edges

Windows with small gradient in both directions

Windows with primarily vertical edges

Both

Statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Windows</th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Dimension 1 (mean $d/dx$ value)
Dimension 2 (mean $d/dy$ value)
Texture representation: example

original image

derivative filter responses, squared

visualization of the assignment to texture “types”
Texture representation: example

- **Dimension 1 (mean d/dx value)**
- **Dimension 2 (mean d/dy value)**

**Far: dissimilar textures**

**Close: similar textures**

<table>
<thead>
<tr>
<th>Win. #</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>#2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Statistics to summarize patterns in small windows.
Texture representation: window scale

- We’re assuming we know the relevant window size for which we collect these statistics.

Possible to perform scale selection by looking for window scale where texture description not changing.
Filter banks

• Our previous example used two filters, and resulted in a 2-dimensional feature vector to describe texture in a window.
  – x and y derivatives revealed something about local structure.

• We can generalize to apply a collection of multiple (d) filters: a “filter bank”

• Then our feature vectors will be d-dimensional.
  – still can think of nearness, farness in feature space
$d$-dimensional features
Filter banks

- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

Matlab code available for these examples: http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html
Influential paper:

Early vision and texture perception

James R. Bergen* & Edward H. Adelson**

* SRI David Sarnoff Research Center, Princeton, New Jersey 08540, USA
** Media Lab and Department of Brain and Cognitive Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
Learn: use filters.

Bergen and Adelson, Nature 1988

Fig. 1 Top row, Textures consisting of Xs within a texture composed of Ls. The micropatterns are placed at random orientations on a randomly perturbed lattice. a. The bars of the Xs have the same length as the bars of the Ls. b. The bars of the Ls have been lengthened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is enhanced. c. The bars of the Ls have been shortened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is impaired. Bottom row: the responses of a size-tuned mechanism d, response to image a; e, response to image b; f, response to image c.
Learn: use lots of filters, multi-ori&scale.

Malik and Perona

Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms. J OPT SOC AM A 7: (5) 923-932 MAY 1990
If matching the averaged squared filter values is a good way to match a given texture, then maybe matching the entire marginal distribution (eg, the histogram) of a filter’s response would be even better.

Jim Bergen proposed this...
Pyramid-Based Texture Analysis/Synthesis

David J. Heeger
Stanford University

James R. Bergen†
SRI David Sarnoff Research Center

SIGGRAPH 1994
**LBP** (Local Binary Pattern)

- A powerful mean of texture description

- **LBP operator:**
  - Standard LBP:
    
    $$
    \text{LBP}_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_c)2^i, \quad S(x) = \begin{cases} 
    1 & \text{if } x \geq 0, \\
    0 & \text{otherwise.}
    \end{cases}
    $$

- **Illustration:**

  ![LBP Operator Illustration](image)
LBP (LOCAL BINARY PATTERN)

- **Example:**

<table>
<thead>
<tr>
<th>Example</th>
<th>Thresholded</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 5 2</td>
<td>1 0 0</td>
<td>1 2 4</td>
</tr>
<tr>
<td>7 6 1</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>9 8 7</td>
<td>1 1 1</td>
<td>64 32 16</td>
</tr>
</tbody>
</table>

Pattern = 1110001
LBP = 1 + 16 + 32 + 64 + 128 = 241

- **Parameters:**
  - P : Number of neighboring pixels
  - R : Radius

  ![Diagram](P=8, R=1, P=16, R=2)
LTP (Local Trinary Pattern)

- **LTP operator:**

  \[
  \text{LTP}_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_r)3^i, \quad S(x) = \begin{cases} 
  1 & \text{if } x \geq t, \\
  0 & \text{if } |x| < t, \\
  -1 & \text{if } x \leq -t,
\end{cases}
\]

- **t:** threshold

- **Illustration:**

  ![Illustration of LTP operator with threshold and ternary code](image)
1. Face tagging
Local binary patterns

Computing of face object histogram

Weight matrix:

```
  2 1 1 1 1 1 1 2
  2 4 4 3 4 4 2
  1 1 1 2 1 1 1
  0 1 1 2 1 1 0
  0 1 1 1 1 1 0
  0 1 1 2 1 1 0
  0 1 1 1 1 1 0
```
Fig. 1. Examples of the different demographics studied. (a-c) Age demographic. (d-e) Gender demographic. (f-h) Race/ethnicity demographic. Within each demographic, the following cohorts were isolated: (a) ages 18 to 30, (b) ages 30 to 50, (c) ages 50 to 70, (d) female gender, (e) male gender, (f) Black race, (g) White race, and (h) Hispanic ethnicity. The first row shows the “mean face” for each cohort. A “mean face” is the average pixel value computed from all the aligned face images in a cohort. The second and third rows show different sample images within the cohorts.
Face Recognition Performance: Role of Demographic Information
Brendan F. Klare, Mark J. Burge, Joshua C. Klontz, Richard W. Vorder Bruegge, and Anil K. Jain
HOG (Histogram of Oriented Gradients)

**HOG** *(Histogram of Oriented Gradients)*

- **Gradient Computation:**

<table>
<thead>
<tr>
<th>Mask Type</th>
<th>1-D centred</th>
<th>1-D uncentred</th>
<th>1-D cubic-corrected</th>
<th>2 × 2 diagonal</th>
<th>3 × 3 Sobel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator</td>
<td>[-1, 0, 1]</td>
<td>[-1, 1]</td>
<td>[1, -8, 0, 8, -1]</td>
<td>[0 1; -1 0], [ -1 0]</td>
<td>[ -1 0 1; -2 0 2]</td>
</tr>
<tr>
<td>Miss rate at 10⁻⁴ FPPW</td>
<td>11%</td>
<td>12.5%</td>
<td>12%</td>
<td>12.5%</td>
<td>14%</td>
</tr>
</tbody>
</table>

*Table 4.1. Different gradient masks and their effect on detection performance. All results are without Gaussian smoothing (σ = 0).*
HOG (HISTOGRAM OF ORIENTED GRADIENTS)

- Spatial/Orientation Binning:
  - Weighted votes
    - Function of magnitude
  - Avoid aliasing
    - Interpolation
  - Parameters:
    - Number of orientation bins
    - Cell size
    - Block size
HOG (Histogram of Oriented Gradients)

- Spatial/Orientation Binning:
  - Parameters:
    - Number of orientation bins: 9 bins/18 bins
    - Cell size: 8x8 pixels
    - Block size: 2x2 cells
HOG (HISTOGRAM OF ORIENTED GRADIENTS)
HOG (Histogram of Oriented Gradients)

HOG (HISTOGRAM OF ORIENTED GRADIENTS)

- **Normalization:**
  - Group cells to larger blocks and normalize each block separately \((illumination\ \text{invariant})\)

- **Normalization Schemes:**
  
  - \(L2\text{-norm}, \ v \leftarrow \frac{v}{\sqrt{||v||_2^2 + \epsilon^2}}\);
  - \(L2\text{-Hys}, \ L2\text{-norm} \text{ followed by clipping (limiting the maximum values of } v \text{ to 0.2) and renormalising, as in Lowe [2004];}\)
  - \(L1\text{-norm}, \ v \leftarrow \frac{v}{||v||_1 + \epsilon}\);
  - \(L1\text{-sqrt}, \ v \leftarrow \sqrt{v/(||v||_1 + \epsilon)}, \ i.e. L1\text{-norm followed by square root, amounts to treating the descriptor vectors as probability distributions and using the Bhattacharya distance between them.}\)
HOG (Histogram of Oriented Gradients)

- Normalization:
  - Normalization Schemes:
**Comparison (SIFT vs. HOG)**

Comparison:

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>HOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest Region Detection</td>
<td>Keypoints</td>
<td>Dense grids</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>Scale invariant</td>
<td>Not scale invariant</td>
</tr>
<tr>
<td></td>
<td>Rotation invariant</td>
<td>Not rotation invariant</td>
</tr>
<tr>
<td>Descriptor Construction</td>
<td>Quantize 4x4, 8 bins</td>
<td>Quantize 2x2, 9 bins,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 different normalizations</td>
</tr>
<tr>
<td>Descriptor Normalization</td>
<td>Illustration invariant</td>
<td>Illustration invariant</td>
</tr>
</tbody>
</table>
HOG Variation

- ‘Object Detection with Discriminatively Trained Part Based Models’

- **Pixel-Level Feature Maps:**
  - Use [-1, 0, 1] to calculate gradient
  - Contrast sensitive ($B_1$), Contrast insensitive ($B_2$)
    
    $B_1(x, y) = \text{round} \left( \frac{p\theta(x, y)}{2\pi} \right) \mod p$
    
    $B_2(x, y) = \text{round} \left( \frac{p\theta(x, y)}{\pi} \right) \mod p$

  - $(\rho = 9)$
    
    - Quantize into orientation bins

    $F(x, y)_b = \left\{ \begin{array}{ll}
    r(x, y) & \text{if } b = B(x, y) \\
    0 & \text{otherwise}
    \end{array} \right.$

    $r$: gradient magnitude
HOG Variation

- **Spatial Aggregation:**
  - Rectangular cell: 8×8 pixels
  - Cell-based feature \( C(i, j) \)
    - Reduce the size of feature map
  - Avoid aliasing:
    - Bilinear interpolation

- **Normalization:**

\[
N_{\delta, \gamma}(i, j) = (\| C(i, j) \|^2 + \| C(i + \delta, j) \|^2 + \\
\| C(i, j + \gamma) \|^2 + \| C(i + \delta, j + \gamma) \|^2)^{\frac{1}{2}}.
\]

\( \delta, \gamma \in \{-1, 1\} \).
**HOG Variation**

- **Truncation:**
  
  \[
  H(i, j) = \begin{pmatrix}
  T_\alpha(C(i, j)/N_{-1,-1}(i, j)) \\
  T_\alpha(C(i, j)/N_{-1,-1}(i, j)) \\
  T_\alpha(C(i, j)/N_{-1,-1}(i, j)) \\
  T_\alpha(C(i, j)/N_{-1,-1}(i, j))
  \end{pmatrix}
  \]

  maximum 0.2
  
  - No renormalization

- **Dimension:**

  9 bins \(\times\) 4 different normalization = 36 (contrast insensitive)
HOG Variation

- PCA analysis:

- Top 11 eigenvectors captures most of information of HOG
HOG Variation

- PCA analysis:
  - Top eigenvectors lie (approximately) in a linear subspace

\[
V = \{u_1, \ldots, u_9\} \cup \{v_1, \ldots, v_4\} \text{ with }
\]

\[
\begin{align*}
  u_k(i, j) &= \begin{cases} 
    1 & \text{if } j = k \\
    0 & \text{otherwise}
  \end{cases} \\
  v_k(i, j) &= \begin{cases} 
    1 & \text{if } i = k \\
    0 & \text{otherwise}
  \end{cases}
\end{align*}
\]

- 13-dimensional features:
  - Project 36-dimensional HOG feature into \( u_k \), \( v_k \)
  - Projection into \( u_k \): sum over 4 normalization over fixed orientation
  - Projection into \( v_k \): sum over 9 orientation over fixed normalization
HOG Variation

- For Contrast Insensitive ($B_2$):
  - 9 bins $\times$ 4 different normalization = 36 (contrast insensitive)

- For Contrast Sensitive ($B_1$):
  - 18 bins $\times$ 4 different normalization = 72 (contrast insensitive)

- Reduce to $(18 + 9) + 4 = 31$ dimension
REFERENCE

- “Description of Interest Regions With Local Binary Patterns”, Pattern Recognition ’09  Marko Heikkilä
  - [http://www.tele.ucl.ac.be/~devlees/ref_ELEC2885/projects/RoiDescriptionLBP-pr-accepted.pdf](http://www.tele.ucl.ac.be/~devlees/ref_ELEC2885/projects/RoiDescriptionLBP-pr-accepted.pdf)
- “Effective Pedestrian Detection Using Center-symmetric Local Binary/Trinary Patterns”, Youngbin Zheng
- “Scale-space Theory”  Tony Lindeberg
- “Histogram of Oriented Gradients for Human Detection”, CVPR ‘05  Navneet Dalal
- “Finding People in Images and Videos”, Navneet Dalal
- “Feature matching” Yung-Yu Chuang
- “Scale & Affine Invariant Interest Point Detectors”, IJCV ’04  Krystian Mikolajczyk
REFERENCE

- “Object Detection with Discriminatively Trained Part Based Models”
- “Distinctive Image Features from Scale-Invariant Keypoints”, IJCV ’04 David G. Lowe
Gray Level Co-occurrence

- The statistical measures described so far are easy to calculate, but do not provide any information about the repeating nature of texture.

- A gray level co-occurrence matrix (GLCM) contains information about the positions of pixels having similar gray level values.
A co-occurrence matrix is a two-dimensional array, $P$, in which both the rows and the columns represent a set of possible image values.

A GLCM $P_d[i,j]$ is defined by first specifying a displacement vector $d=(dx,dy)$ and counting all pairs of pixels separated by $d$ having gray levels $i$ and $j$. 
The GLCM is defined by:

\[ P_d[i, j] = n_{ij} \]

– where \( n_{ij} \) is the number of occurrences of the pixel values \((i,j)\) lying at distance \(d\) in the image.

– The co-occurrence matrix \( P_d \) has dimension \( n \times n \), where \( n \) is the number of gray levels in the image.
GLCM

For example, if \( d=(1,1) \)

there are 16 pairs of pixels in the image which satisfy this spatial separation. Since there are only three gray levels, \( P[i,j] \) is a \( 3 \times 3 \) matrix.
GLCM

Algorithm:

• Count all pairs of pixels in which the first pixel has a value $i$, and its matching pair displaced from the first pixel by $d$ has a value of $j$.

• This count is entered in the $i^{th}$ row and $j^{th}$ column of the matrix $P_d[i,j]$.

• Note that $P_d[i,j]$ is not symmetric, since the number of pairs of pixels having gray levels $[i,j]$ does not necessarily equal the number of pixel pairs having gray levels $[j,i]$. 
Normalized GLCM

- The elements of $P_d[i,j]$ can be normalized by dividing each entry by the total number of pixel pairs.

Normalized GLCM $N[i,j]$, defined by:

$$N[i, j] = \frac{P[i, j]}{\sum_i \sum_j P[i, j]}$$

which normalizes the co-occurrence values to lie between 0 and 1, and allows them to be thought of as probabilities.
Numeric Features of GLCM

• Gray level co-occurrence matrices capture properties of a texture but they are not directly useful for further analysis, such as the comparison of two textures.

• Numeric features are computed from the co-occurrence matrix that can be used to represent the texture more compactly.
Quantitative Texture Measures

- Numeric quantities or statistics that describe a texture can be calculated from the intensities (or colors) themselves.
- One problem with deriving texture measures from co-occurrence matrices is how to choose the displacement vector $d$.
- The choice of the displacement vector is an important parameter in the definition of the GLCM.
- Occasionally the GLCM is computed from several values of $d$ and the one which maximizes a statistical measure computed from $P[i,j]$ is used.
- Zucker and Terzopoulos used a $\chi^2$ measure to select the values of $d$ that have the most structure; i.e. to maximize the value:

$$\chi^2(d) = \sum_i \sum_j \frac{P_d[i,j]^2}{P_d[i]P_d[j]} - 1$$
Windowing

- Algorithms for texture analysis are applied to an image in a series of windows of size $w$, each centered on a pixel $(i,j)$.
- The value of the resulting statistical measure are assigned to the position $(i,j)$ in the new pixel.
Maximum Probability

• This is simply the largest entry in the matrix, and corresponds to the strongest response. This could be the maximum in any of the matrices or the maximum overall.

\[ C_m = \max_{i,j} P_d[i, j] \]

• Maximum probability with \( w=21 \), and \( d=(2,2) \).
Range

One of the simplest of the *texture operators* is the *range* or *difference between maximum and minimum intensity values in a neighborhood*.

The range operator converts the original image to one in which brightness represents texture.
Moments

• The order $k$ element difference moment can be defined as:

$$Mom_k = \sum_{i} \sum_{j} (i - j)^k P_d[i, j]$$

• This descriptor has small values in cases where the largest elements in $P$ are along the principal diagonal. The opposite effect can be achieved using the inverse moment.

$$Mom_k = \sum_{i} \sum_{j} \frac{P_d[i, j]}{(i - j)^k}, \quad i \neq j$$
Moments

• Moments with $w=21$, and $d=(2,2)$
Contrast

- **Contrast** is a measure of the local variations present in an image.

\[
C(k, n) = \sum_i \sum_j (i - j)^k P_d[i, j]^n
\]

- If there is a large amount of variation in an image the P[i,j]’s will be concentrated away from the main diagonal and contrast will be high (*typically* \(k=2, n=1\)).
Contrast

- Contrast with $w=21$, and $d=(2,2)$
Homogeneity

- A homogeneous image will result in a **co-occurrence matrix** with a combination of high and low $P[i,j]$’s.

\[ C_h = \sum_i \sum_j \frac{P_d[i,j]}{1+|i-j|} \]

- Where the **range of gray levels** is small the $P[i,j]$ will tend to be clustered around the main diagonal.
- A heterogeneous image will result in an even spread of $P[i,j]$’s.
Homogeneity

- Homogeneity with $w=21$, and $d=(2,2)$
Entropy

- Entropy is a measure of information content. It measures the randomness of intensity distribution.

\[ C_e = - \sum_i \sum_j P_d[i, j] \ln P_d[i, j] \]

- Such a matrix corresponds to an image in which there are no preferred gray level pairs for the distance vector \( \mathbf{d} \).
- Entropy is highest when all entries in \( \mathbf{P}[i,j] \) are of similar magnitude, and small when the entries in \( \mathbf{P}[i,j] \) are unequal.
Entropy

- Entropy with $w=21$, and $d=(2,2)$
Correlation

- Correlation is a measure of image linearity

\[
C_c = \frac{\sum_i \sum_j [ijP_d[i, j]] - \mu_i \mu_j}{\sigma_i \sigma_j}
\]

\[
\mu_i = \sum i P_d[i, j], \quad \sigma_i^2 = \sum i^2 P_d[i, j] - \mu_i^2
\]

- Correlation will be high if an image contains a considerable amount of linear structure.
Energy

• One approach to generating texture features is to use local kernels to detect various types of texture.

• After the convolution with the specified kernel, the **texture energy measure (TEM)** is computed by summing the absolute values in a local neighborhood:

\[ L_e = \sum_{i=1}^{m} \sum_{j=1}^{n} |C(i, j)| \]

• If \( n \) kernels are applied, the result is an \( n \)-dimensional feature vector at each pixel of the image being analyzed.
Gray-level Difference Statistics

- Gray-level differences are based on absolute differences between pairs of gray-levels.

- The gray-level differences are contained in a 256-element vector, and are computed by taking the absolute differences of all possible pairs of gray levels distance \( \mathbf{d} \) apart at angle \( \Theta \), and counting the number of times the difference is 0, 1, ..., 255.

- Let \( \mathbf{d} = (dx, dy) \) be the displacement vector between two image pixels, and \( g(d) \) the gray-level difference at distance \( \mathbf{d} \).

\[
g(d) = \left| f(i, j) - f(i + dx, j + dy) \right|
\]

- \( p_g(g, \mathbf{d}) \) is the histogram of the gray-level differences at the specific distance, \( \mathbf{d} \). One distinct histogram exists for each distance \( \mathbf{d} \).
Gray-level Difference Statistics (2)

- The difference statistics are then normalized by dividing each element of the vector by the number of possible pixel pairs.
- Several texture measures can be extracted from the histogram of gray-level differences:
  - Mean:
    $$\mu_d = \sum_{k=1}^{N} g_k p_g(g_k, d)$$
    - Small mean values \( \mu_d \) indicate coarse texture having a grain size equal to or larger than the magnitude of the displacement vector.
  - Entropy:
    $$H_d = -\sum_{k=1}^{N} p_g(g_k, d) \ln p_g(g_k, d)$$
    - This is a measure of the homogeneity of the histogram. It is maximized for uniform histograms.
Gray-level Difference Statistics

(3)

• Variance:

\[ \sigma_d^2 = \sum_{k=1}^{N} (g_k - \mu_d)^2 \cdot p_g(g_k, d) \]

- The variance is a measure of the dispersion of the gray-level differences at a certain distance, \(d\).

• Contrast:

\[ C_d = \sum_{k=1}^{N} g_k^2 \cdot p_g(g_k, d) \]
Gray-level Difference Statistics

Some experimental results:

- Mean
- Standard Deviation
- Entropy
- Contrast
Edges and Texture

It should be possible to locate the edges that result from the intensity transitions along the boundary of the texture.

Since a texture will have large numbers of texels, there should be a property of the edge pixels that can be used to characterise the texture.

- a set of common directions
- a measure of the local density of the edge pixels

Compute the co-occurrence matrix of an edge-enhanced image.
Edges and Texture

Original          Sobel-Enhanced          Contrast
Biometry – Iris recognition through co-occurrence matrix

1) **Energia:** \[ f_1 = \sum_{i} \sum_{j} p(i, j)^2 \]

2) **Contrasto:** \[ f_2 = \sum_{i} \sum_{j} (i-j)^2 p(i, j) \]

3) **Correlazione:** \[ f_3 = \frac{\sum_{i} \sum_{j} (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \]

4) **Omogeneità:** \[ f_4 = \sum_{i} \sum_{j} \frac{1}{1 + (i-j)^2} p(i, j) \]

5) **Autocorrelazione:** \[ f_5 = \sum_{i} \sum_{j} (ij) p(i, j) \]

6) **Dissimilarità:** \[ f_6 = \sum_{i} \sum_{j} |i-j| p(i, j) \]

7) **Massima probabilità:** \[ f_7 = \max_{(i,j)} p(i, j) \]
Biometry – Iris recognition through co-occurrence matrix

```
[glcm] = graycomatrix(pattern,'NumLevels',255,'GrayLimits',[0 255],'Offset',[0 1; -1 0; -1 -1; 0 3; -3 3; -3 0; -3 -3; 0 5; -5 5; -5 0; -5 -5; 0 10; -10 10; -10 0; -10 -10]);
```

The resulting feature vector is of 112 elements (16 matrix for 7 statistics) for each eye image.
Biometria - Riconoscimento Iride mediante descrittori estratti dalle matrici di cooccorrenza
Content based Image Retrieval

Figure 1: Rock texture classes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>$\sum_i \sum_j p_d^2(i,j)$</td>
</tr>
<tr>
<td>Entropy</td>
<td>$-\sum_i \sum_j p_d(i,j) \log p_d(i,j)$</td>
</tr>
<tr>
<td>Contrast</td>
<td>$\sum_i \sum_j (i-j)^2 p_d(i,j)$</td>
</tr>
<tr>
<td>Inverse Difference Moment</td>
<td>$\sum_i \sum_j \frac{p_d(i,j)}{</td>
</tr>
</tbody>
</table>

Table 1: Features extracted from gray level co-occurrence matrix
Content based Image Retrieval

Image Database

Query Image

Query Blobs

Building Index

Retrieved Images

Feature Space
Texture characterization by means of a filter bank

\[ \psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left[ -\frac{1}{2} \left( \frac{\ddot{x}^2}{\sigma_x^2} + \frac{\ddot{y}^2}{\sigma_y^2} \right) \right] \exp \left[ j (\omega_0^x x + \omega_0^y y) \right] \]

\[ \begin{align*}
\ddot{x} &= x \cos \theta + y \sin \theta \\
\ddot{y} &= -x \sin \theta + y \cos \theta
\end{align*} \]

Fig. 5. The 2D Gabor wavelet (real part) shows capabilities for tuning information at different scales (reading by rows) and orientations (reading by columns), allowing to detect vessels having any spatial configuration.
Color-texture Segmentation

Ground truth
Color-texture Segmentation (cont.)