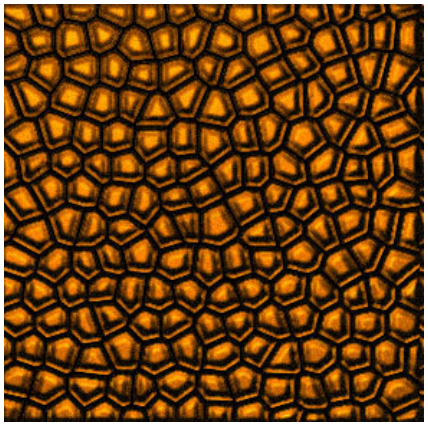
Description of region features: texture

Texture

- Texture is a very general notion that can be attributed to almost everything in nature.
- For a human, the texture relates mostly to:
  - A specific, spatially repetitive (micro)structure of surfaces formed by **repeating a particular element** or several elements in different relative spatial positions.
  - Generally, the repetition involves local variations of scale, orientation, or other geometric and optical features of the elements.



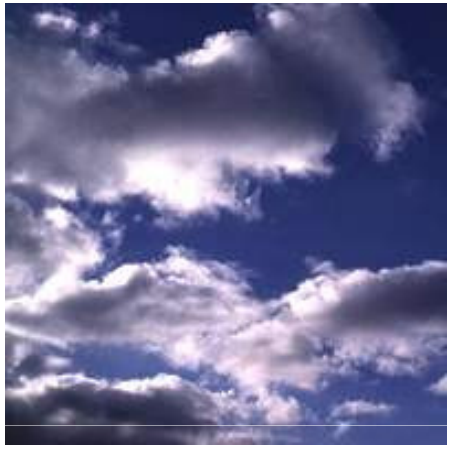
Texture

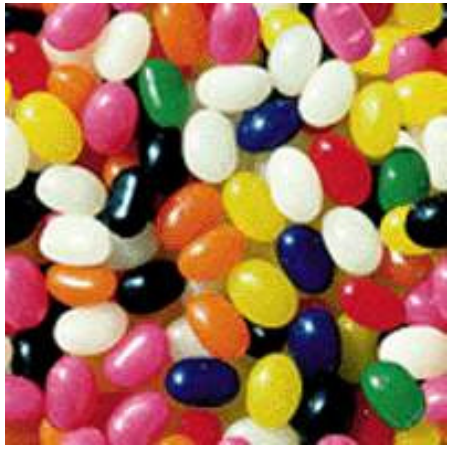
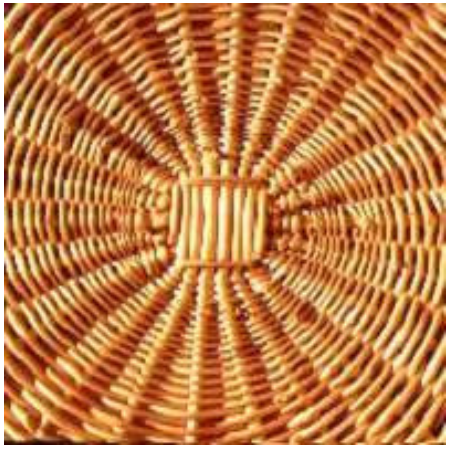
- Texture is a **property of regions**: the texture of a point is undefined
- Texture involves the spatial distribution of **gray levels or colors**
- Texture in an image can be perceived at **different scales** or levels of resolution
- A region is perceived to have texture when the number of primitive elements in the region is large. If only a few primitive elements are present, then a group of countable objects is perceived instead of a textured image

Texture

- It is almost impossible to provide a complete description of textures in words
- However, words capture some **informal qualitative features** that can help discriminate different textures

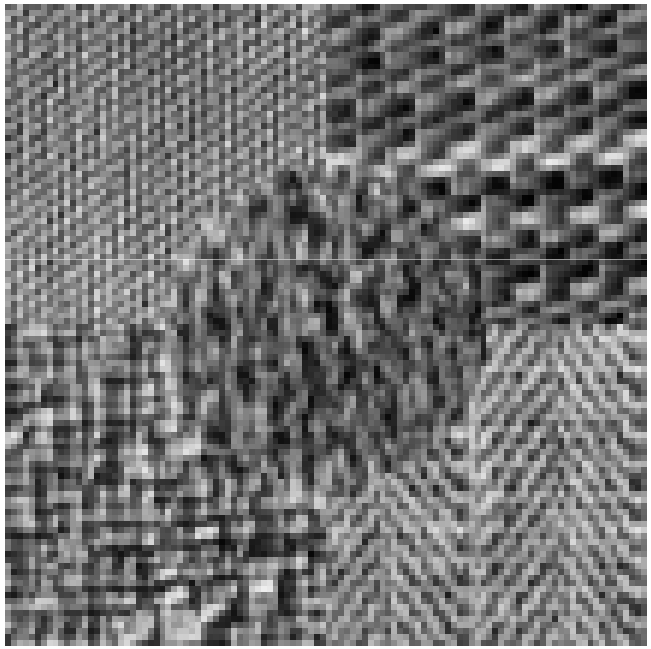
  - fineness - coarseness,
  - smoothness,
  - granularity,
  - lineation,
  - directionality,
  - regularity - randomness



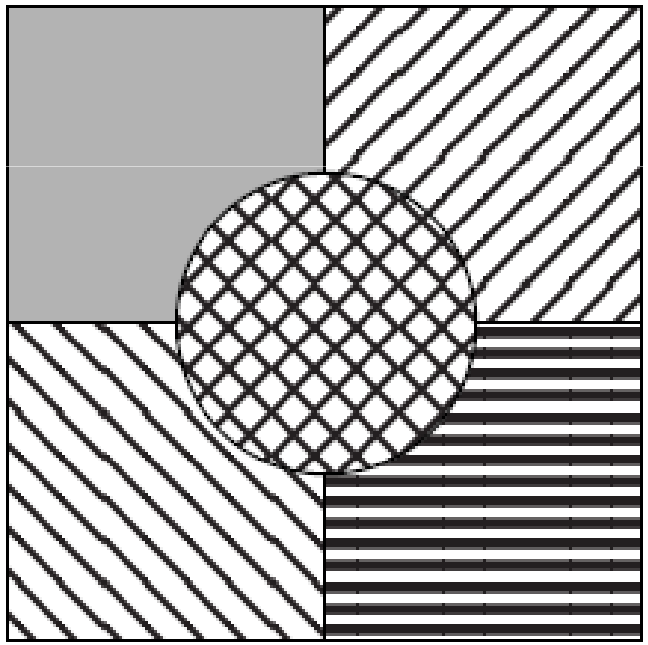
Texture

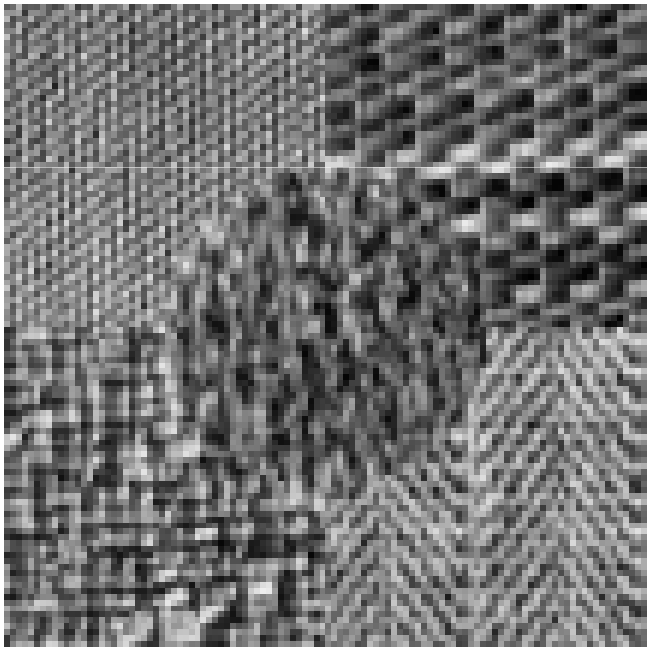
- Texture models can be of interest for three main purposes:
  - **Segmentation**
  - **Classification**
  - **Synthesis**



Texture

- Texture models can be of interest for three main purposes:
  - **Segmentation**: partition the input image into regions of uniform texture
  - **Classification**
  - **Synthesis**

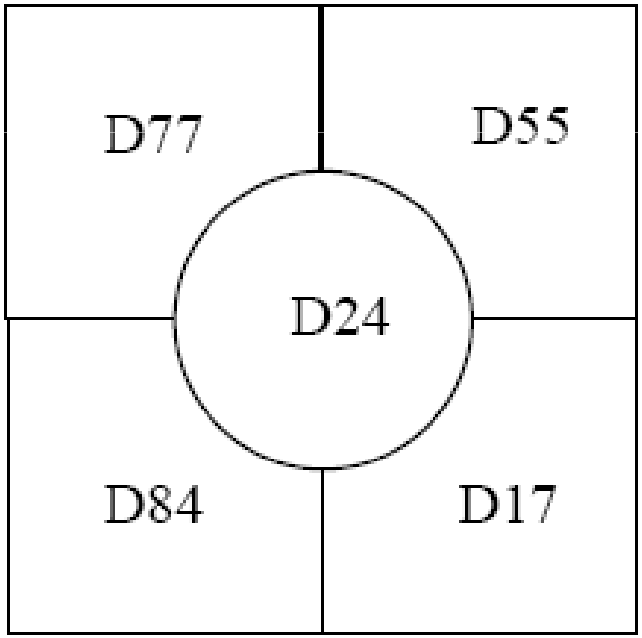


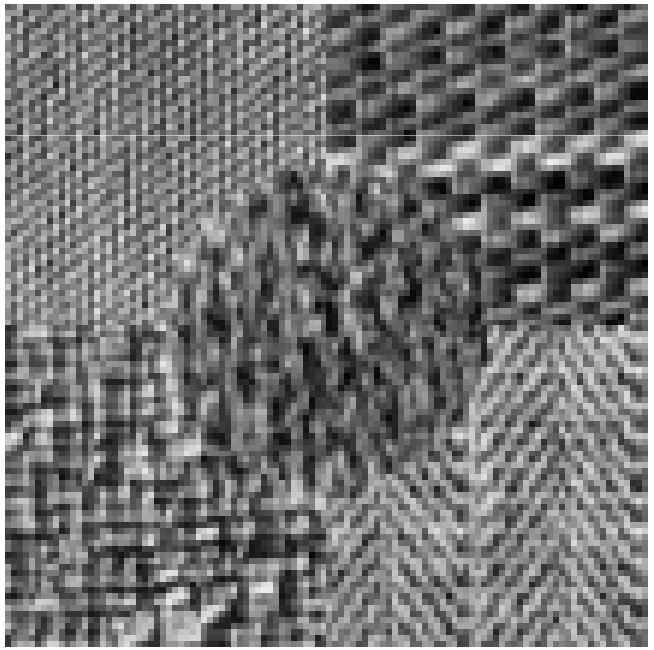


Texture

- Texture models can be of interest for three main purposes:
  - **Segmentation**
  - **Classification**: produce a classification map of the input image where each uniform textured region is identified with the texture class it belongs to
  - **Synthesis**

D77: cotton canvas
D55: straw matting
D84: raffia
D17: herringbone weave
D24: pressed calf leather





Texture

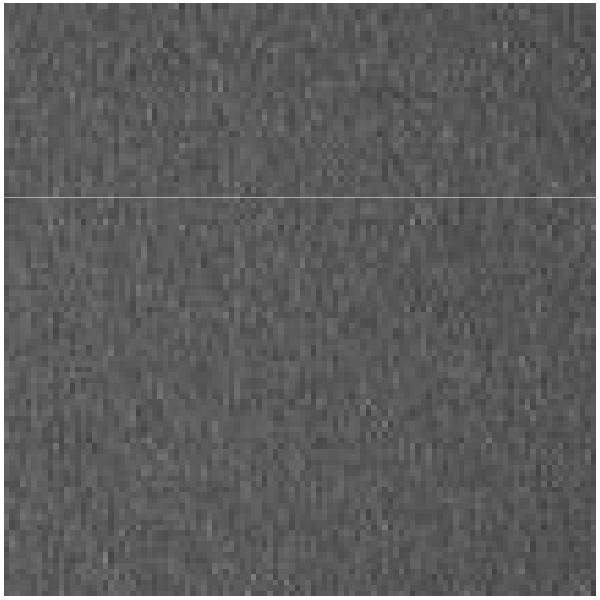
- Texture models can be of interest for three main purposes:
  - **Segmentation**
  - **Classification**
  - **Synthesis**: generate an image with a texture that looks like a requested texture class

Applications of Texture Analysis

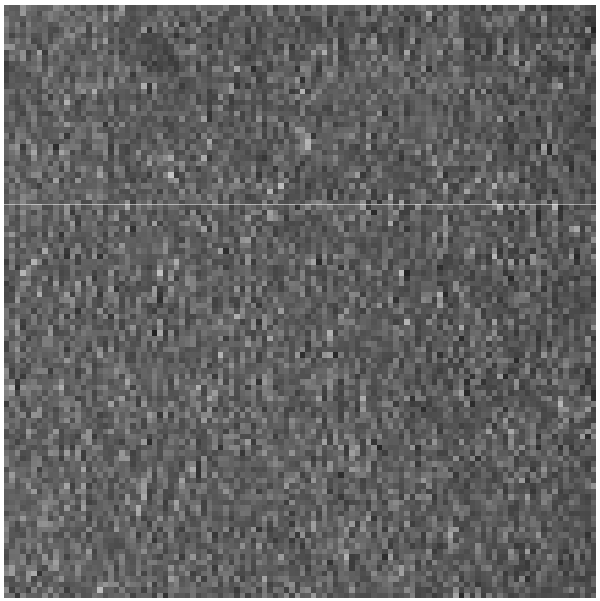
- Texture analysis methods have been used successfully in a variety of application domains:
  - Inspection
  - Medical image analysis
  - Document analysis
  - Remote sensing

Applications of Texture Analysis

- *Inspection*
  - defect detection in images of textiles: detect point defects and line defects in texture images
  - automated inspection of automobile paints: automatically classify the quality of painted metallic surfaces



uniform coating



blotchy coating

Applications of Texture Analysis

- *Medical image analysis*:
  - Detection of normal/abnormal tissue samples (normal/diseased lungs, leukemic malignancy in blood cells)

Applications of Texture Analysis

- *Document analysis*:
  - In many postal document processing applications (recognition of destination address and zip code), the first step is the ability to separate the regions in the image which contain useful information from the background
  - Texture segmentation methods can be used for locating text blocks in newspapers

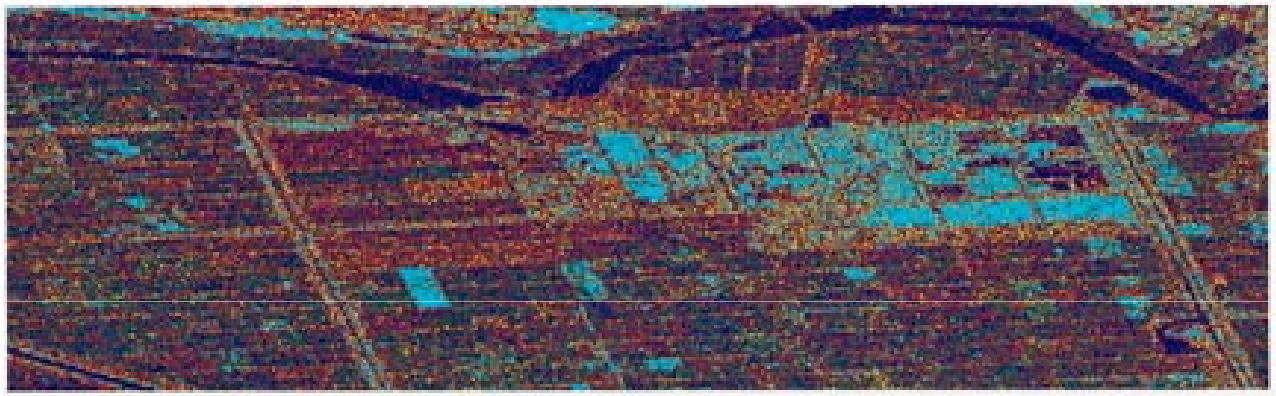
text

no text

transition

Applications of Texture Analysis

- *Remote sensing*:
  - Land use classification: identify homogeneous regions with different *types* of terrains (wheat, bodies of water, urban regions, etc.)



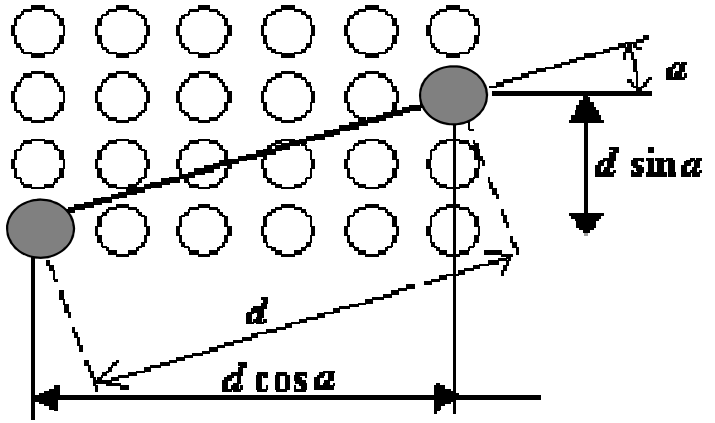
Original SAR image

N-th order statistics

- *First-order statistics*
  - measure the likelihood of observing a gray value at a randomly-chosen location in the image.
  - can be computed from the histogram of pixel intensities in the image.
  - depend only on individual pixel values and not on the interaction or co-occurrence of neighboring pixel values.
  - The average intensity in an image is an example of the first-order statistic.
- *Second-order statistics*
  - are defined as the likelihood of observing a pair of gray values occurring at the endpoints of a dipole (or needle) of random length placed in the image at a random location and orientation.
  - are properties of pairs of pixel values

Co-occurrence matrix

- Many statistical texture features are based on **co-occurrence matrices**
- These represent the distribution of grey levels in pairs of pixels in an image.
- A co-occurrence matrix shows how frequent is every particular pair of grey levels in pixel pairs separated by a certain distance *d* along a certain direction *a*.

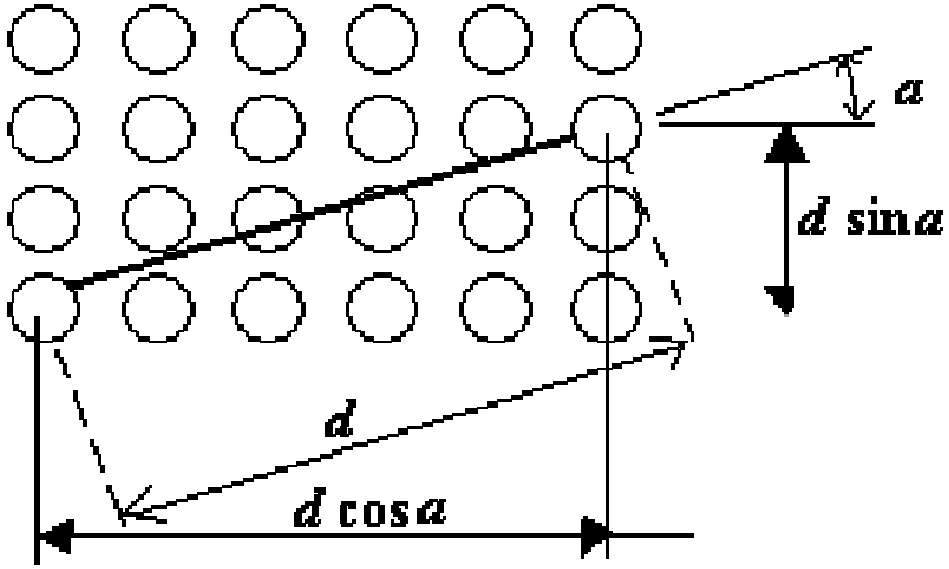


Co-occurrence matrix

- Let g(x,y) with x = 1, ..., M and y = 1, ..., N be a digital image and Q={0, ..., q_{max}} be the set of grey levels.
- The co-occurrence matrix is computed with respect to a predefined **displacement vector *d*** specified either in polar *(d,a)* or cartesian *(dx,dy)* coordinates:

$$COOC_{dx,dy}(g) = \alpha\{c_{ij}\}_{i,j=0}^{q_{\max}} \qquad c_{ij} = \Pr((g(x,y)=i, g(x+dx, y+dy)=j))$$

- where αc_{ij} is the cardinality of the set composed of pixel pairs $[(x,y), (x+dx, y+dy)]$ such that $g_{x,y}=i$ and $g_{x+dx,y+dy}=j$.



Co-occurrence matrix

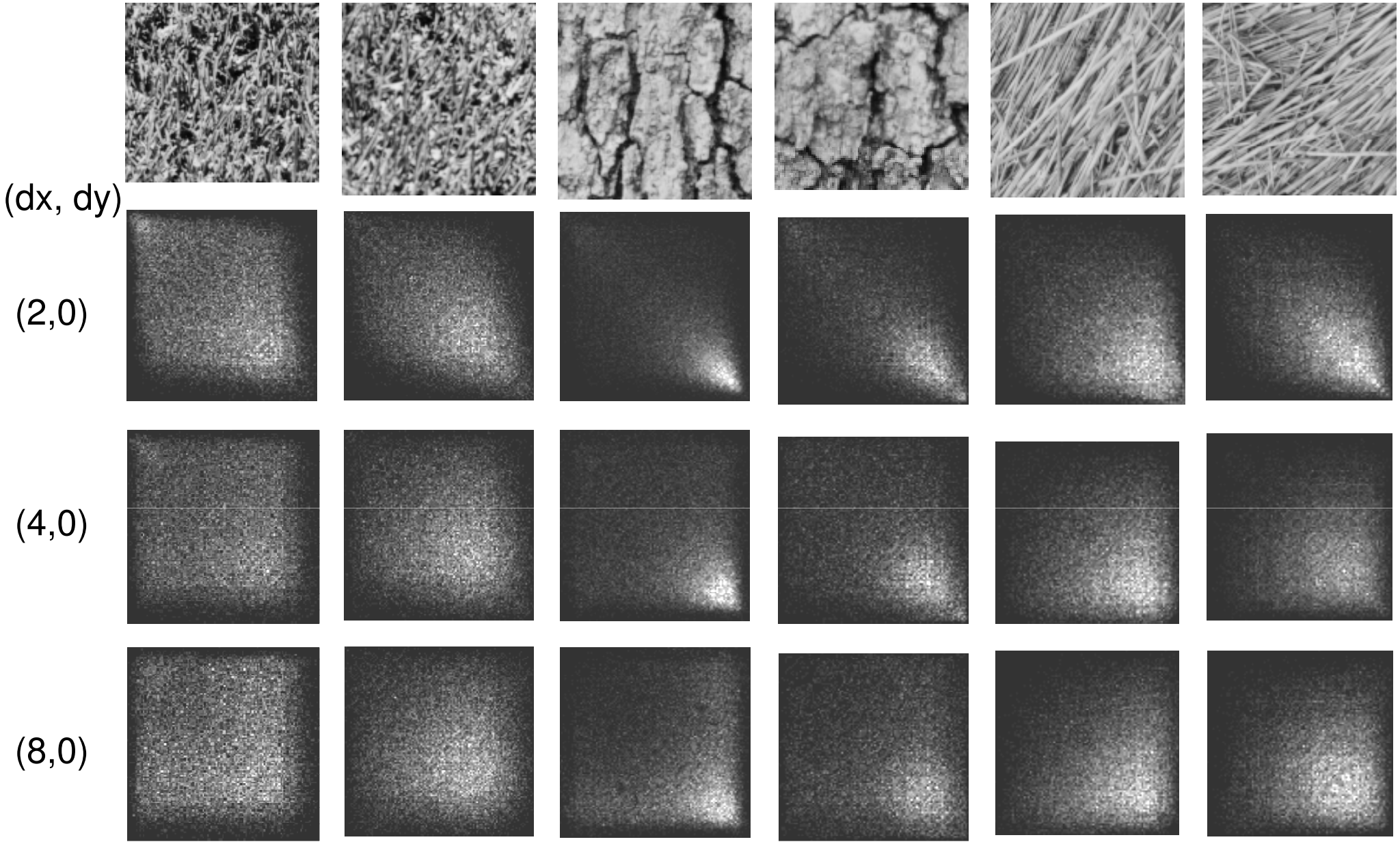
- How to compute the COOC matrix for the offset (di, dj):

```
1. Initialize to 0 all entries of the COOC
2. For each image pixel (i,j)
2.1. Let k1 be the graylevel of pixel (i,j) and
     k2 the graylevel of pixel (i+di,j+dj)
2.2. COOC(k1,k2) += 1
3. Normalize COOC entries by the number of pix-pairs
4. end
```

Co-occurrence matrix

- What is the effect of graylevel quantization (reduction of the graylevels) on the COOC matrix?
- What is the effect of the image size on the COOC matrix?
- The COOC matrix is not symmetric
  - A symmetric COOC matrix can be constructed by incrementing not only entry (k1,k2) but also (k2,k1)

Co-occurrence matrix



Co-occurrence matrix

- Use of the entire co-occurrence matrix as descriptor doesn't make it explicit the actual properties of the texture and makes it difficult to define a distance between textures

- For this purpose, various statistical and information theoretic properties of the co-occurrence matrices can be used:
  - Contrast
  - Energy
  - Entropy
  - Homogeneity
  - Variance

Co-occurrence matrix features

- Contrast $$F_{con} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} (i-j)^2 f(i,j)$$
- Energy $$F_{ene} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} f(i,j)^2$$
- Entropy $$F_{ent} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} f(i,j) \log(f(i,j))$$

f(i,j) is the **normalized** value of entry (i,j) of the co-occurrence matrix.

Entropy is computed using only non-zero entries.

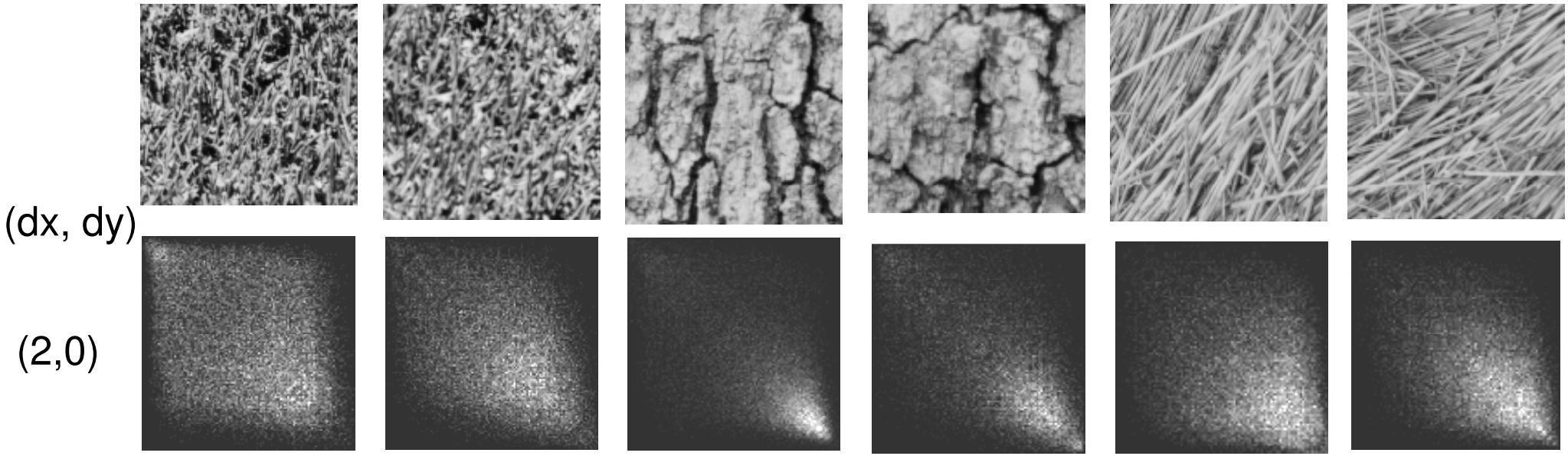
Co-occurrence matrix features

- Homogeneity $F_{\text{hom}} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} \frac{f(i,j)}{1+|i-j|}$

- Variance $F_{\text{var}} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} \frac{\left(f(i,j) - \overline{f(i,j)}\right)^2}{(q_{\max}+1)^2}$

with $\overline{f(i,j)} = \sum_{i=0}^{q_{\max}} \sum_{j=0}^{q_{\max}} \frac{f(i,j)}{(q_{\max}+1)^2}$

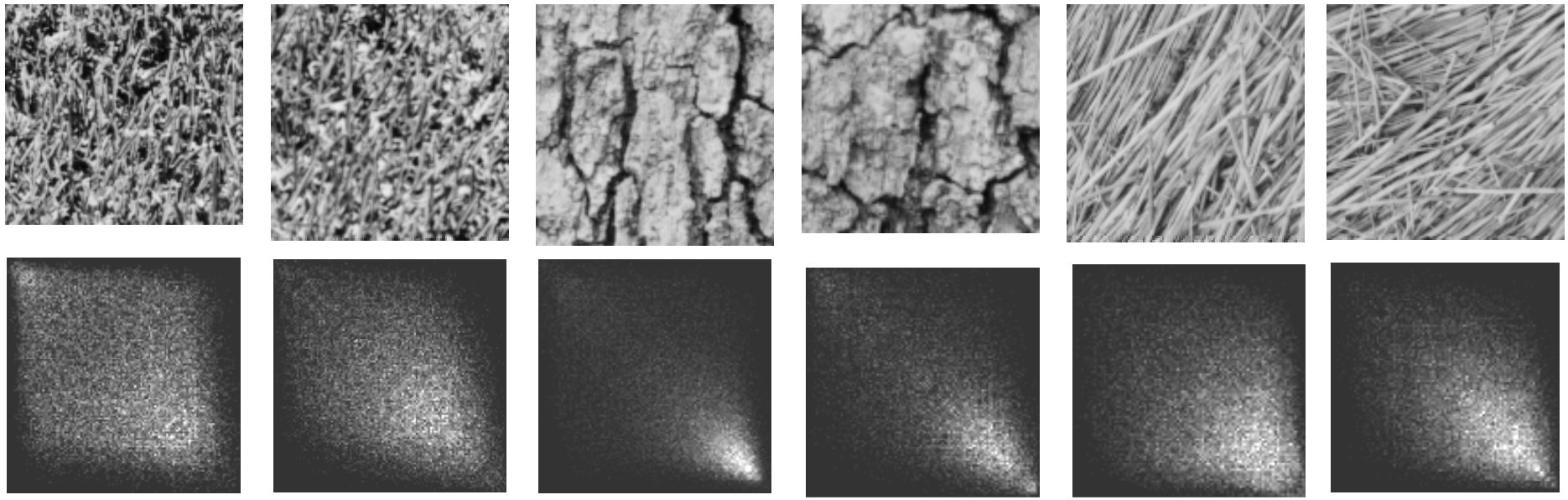
Co-occurrence matrix features



Con.	1.7979	0.5469	-0.7069	-0.6896	-0.3267	-0.6216
Ene.	1.5268	0.7139	-1.1372	-0.9115	-0.0050	-0.1870
Entr.	-1.5585	-0.9625	0.7619	0.6301	0.4424	0.6867
Hom.	1.4496	0.9155	-1.1964	-0.6922	-0.3734	-0.1031
Var.	0.5985	0.2113	-1.4471	-1.0413	0.9103	0.7682

Normalized to zero mean and unitary std

Co-occurrence matrix features

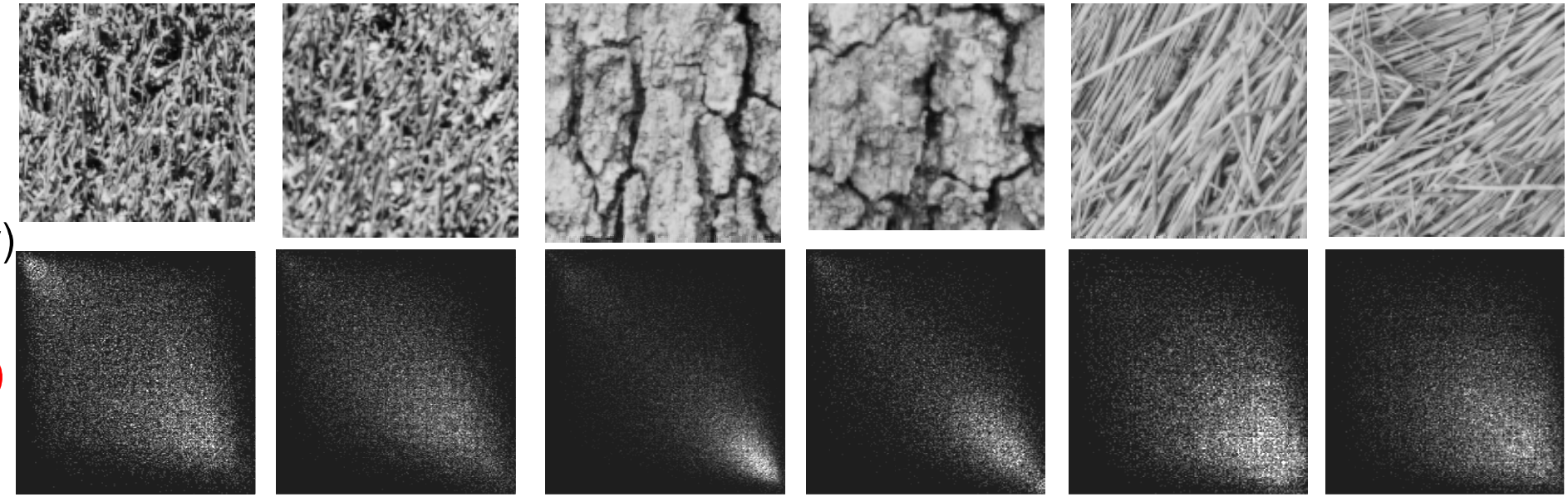


Con.	1.7979	0.5469	-0.7069	-0.6896	-0.3267	-0.6216
Ene.	1.5268	0.7139	-1.1372	-0.9115	-0.0050	-0.1870
Entr.	-1.5585	-0.9625	0.7619	0.6301	0.4424	0.6867
Hom.	1.4496	0.9155	-1.1964	-0.6922	-0.3734	-0.1031
Var.	0.5985	0.2113	-1.4471	-1.0413	0.9103	0.7682

Distance matrix

0.00	1.73	5.47	4.91	3.77	4.03
1.73	0.00	3.89	3.29	2.32	2.49
5.47	3.89	0.00	0.69	2.78	2.64
4.91	3.29	0.69	0.00	2.21	2.03
3.77	2.32	2.78	2.21	0.00	0.52
4.03	2.49	2.64	2.03	0.52	0.00

Co-occurrence matrix features



Con.	1.8524	0.1863	-0.8925	-0.8020	-0.0401	-0.3041
Ene.	1.6413	-0.1614	-1.0736	-0.7391	0.6792	-0.3464
Entr.	-1.6253	-0.6913	0.9588	0.8981	0.0735	0.3862
Hom.	1.8179	0.0908	-0.8869	-0.4951	0.2449	-0.7716
Var.	0.9322	-0.5511	-1.1555	-0.6740	1.4274	0.0210

Distance matrix

0.00	3.47	5.76	5.19	3.18	4.49
3.47	0.00	2.45	2.04	2.29	1.58
5.76	2.45	0.00	0.71	3.54	1.61
5.19	2.04	0.71	0.00	2.86	1.10
3.18	2.29	3.54	2.86	0.00	2.05
4.49	1.58	1.61	1.10	2.05	0.00

Region covariance matrix

- Following another approach, the **covariance** of several **image statistics** computed inside a region of interest, can be used as the descriptor of the texture within the region
  - Possible image statistics include graylevels, their first and second derivatives, output of filter bank, etc.
- Since the covariance matrices are not elements of the Euclidean space, a suitable distance metric defined on the space of positive definite symmetric matrices should be used for matching texture descriptors

Region covariance matrix

- Let I be an image and R $\subseteq$ I a region of the image composed of n pixels
- Let $\{\mathbf{z}_k\}$, k=1…n be the feature vectors extracted from the region pixels, with $\mathbf{z}_k \in R^d$
- Region features are captured through the dxd symmetric covariance matrix

$$\mathbf{C} = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{z}_k - \boldsymbol{\mu})(\mathbf{z}_k - \boldsymbol{\mu})^T$$

being $\boldsymbol{\mu}$ the mean of vectors $\mathbf{z}_k$

Region covariance matrix

- The covariance matrix represents a natural way of fusing multiple features which might be correlated
  - The **diagonal entries** of the covariance matrix represent the variance of each feature
  - The **nondiagonal entries** represent the correlations
- The noise corrupting individual samples are largely filtered out with an average filter during covariance computation

Region covariance matrix

- The covariance matrices do not lie on Euclidean space
  - For example, the space is not closed under multiplication with negative scalars
  - Accordingly, the distance between two covariance matrices cannot be computed using the Euclidean distance
- It can be demonstrated that a metric in the space of covariance matrices is the following one:

$$\rho(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{i=1}^{d} \ln^2 [\lambda_i(\mathbf{C}_1, \mathbf{C}_2)]}$$

being λ_i the **generalized eigenvalues** of C_1 and C_2

Region covariance matrix

- It can be demonstrated that a metric in the space of covariance matrices is the following one:

$$\rho(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{i=1}^{d} \ln^2 [\lambda_i(\mathbf{C}_1, \mathbf{C}_2)]}$$

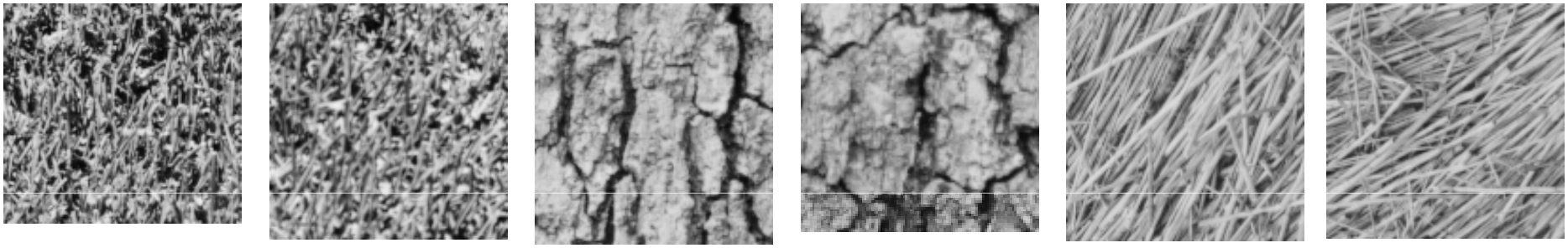
being λ_i the generalized eigenvalues of C_1 and C_2

- The generalized eigenvalues λ_i and eigenvectors $\mathbf{x}_i$ are the solution to the equation

$$\lambda_i \mathbf{C}_1 \mathbf{x}_i - \mathbf{C}_2 \mathbf{x}_i = 0 \quad i = 1, \ldots, d$$

In Matlab, if A and B are square matrices the function d = **eig**(A,B) returns a vector containing the generalized eigenvalues.

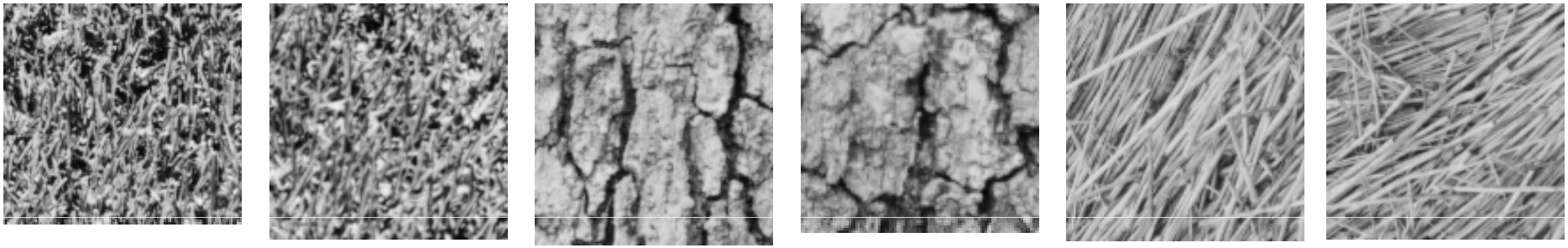
Region covariance matrix



- As an example, feature vectors can be extracted by combining the image intensities and norms of first and second order derivatives along *x* and *y* direction:

$$\mathbf{z}_i = \left[I(x,y), \left\| \frac{\partial I(x,y)}{\partial x} \right\|, \left\| \frac{\partial I(x,y)}{\partial y} \right\|, \left\| \frac{\partial^2 I(x,y)}{\partial x^2} \right\|, \left\| \frac{\partial^2 I(x,y)}{\partial y^2} \right\| \right]$$

Region covariance matrix



- As an example, feature vectors can be extracted by combining the image intensity and norm of the first and second order derivatives along *x* and *y* direction:

$$\mathbf{z}_i = \left[I(x,y), \left\| \frac{\partial I(x,y)}{\partial x} \right\|, \left\| \frac{\partial I(x,y)}{\partial y} \right\|, \left\| \frac{\partial^2 I(x,y)}{\partial x^2} \right\|, \left\| \frac{\partial^2 I(x,y)}{\partial y^2} \right\| \right]$$

Distance matrix

0.00	0.73	1.57	1.45	1.77	1.98
0.73	0.00	1.03	0.89	1.34	1.40
1.57	1.03	0.00	0.42	0.91	1.21
1.45	0.89	0.42	0.00	1.00	0.90
1.77	1.34	0.91	1.00	0.00	1.43
1.98	1.40	1.21	0.90	1.43	0.00

Tamura features

- An alternative, generally preferred approach is based on a set of six visual features, called **Tamura features**, selected on the basis of psychological experiments:

  - coarseness
  - contrast
  - directionality
  - linelikeness
  - regularity
  - roughness

Tamura features

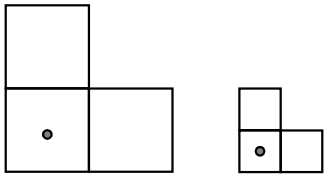
- **Coarseness** relates to distances between notable spatial variations of grey levels
  - that is, to the size of the primitive elements (textons) forming the texture.
- The proposed computational procedure accounts for differences between the average signals for non-overlapping windows of different size

Tamura features

- Algorithm:
  1. At each pixel (x,y), compute five averages for the windows of size $(2^k+1)\times(2^k+1)$, $k = 1,\ldots,5$ around the pixel (3x3 – 33x33)

$$A_k(x,y) = \frac{1}{2^{2k}} \sum_{i=x-2^{k-1}}^{x+2^{k-1}} \sum_{j=y-2^{k-1}}^{y+2^{k-1}} f(i,j)$$

  2. At each pixel, compute absolute differences $E_k(x,y)$ between the pairs of non-overlapping averages in the horizontal and vertical directions



$$E_k(x,y) = \max\left\{\left|A_k(x,y) - A_k\left(x+2^k, y\right)\right|, \left|A_k(x,y) - A_k\left(x, y+2^k\right)\right|\right\}$$

  3. At each pixel, find the value of k that maximizes the difference $E_k(x,y)$ in either directions and set the best size $S_{\text{best}}(x,y)=2^k$

Tamura features

- Algorithm:
  4. Compute the coarseness feature F_{crs} by averaging $S_{best}(x,y)$ over the entire image. Instead of the average of $S_{best}(x,y)$, an improved coarseness feature to deal with textures having multiple coarseness properties is a histogram characterizing the whole distribution of the best sizes over the image

Tamura features

- **Contrast** measures how grey levels vary in the image
- For this purpose, the second- and fourth-order central moments of the grey level histogram are used:

$$F_{con} = \frac{\sigma}{(\alpha_4)^n}$$

where:

- σ^2 is the variance (second central moment of gray level prob. distrib.)

$$\sigma^2 = \sum_{q=0}^{q_{max}} (q - m)^2 \Pr(q \mid g)$$

- $\alpha_4 = \mu_4/\sigma^4$ is the kurtosis (μ_4 is the fourth central moment of gray level p.d.)

$$\mu_4 = \sum_{q=0}^{q_{max}} (q - m)^4 \Pr(q \mid g)$$

- n is a positive number to be chosen experimentally. The value n=0.25 is suggested as the best

Tamura features

- The **directionality** measures the frequency distribution of gradient magnitude against the directional angle
- The magnitude $e(x,y)$ and the directional angle $a(x,y)$ are computed using the Prewitt operator by approximating the pixel-wise x- and y-derivatives of the image:

$$e(x,y) = 0.5(|d_x(x,y)| + |d_y(x,y)|)$$

$$a(x,y) = \tan\text{-}1(d_y(x,y) / d_x(x,y))$$

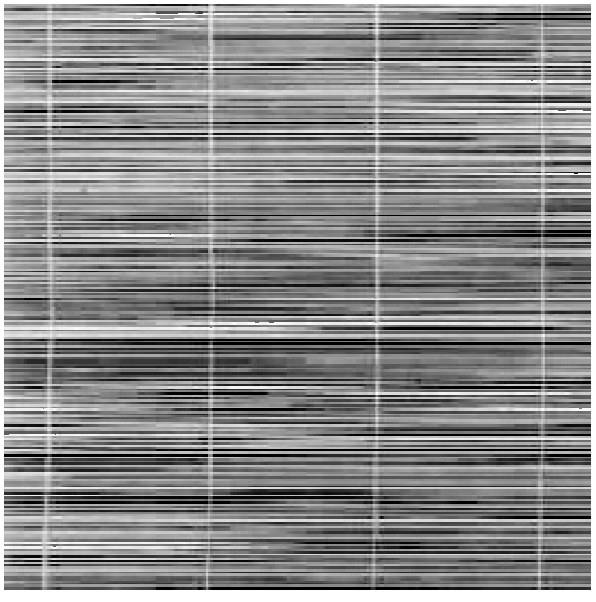
- where $d_x(x,y)$ and $d_y(x,y)$ are the horizontal and vertical grey level differences between the neighboring pixels, respectively.
- Differences are measured using Prewitt 3×3 moving windows

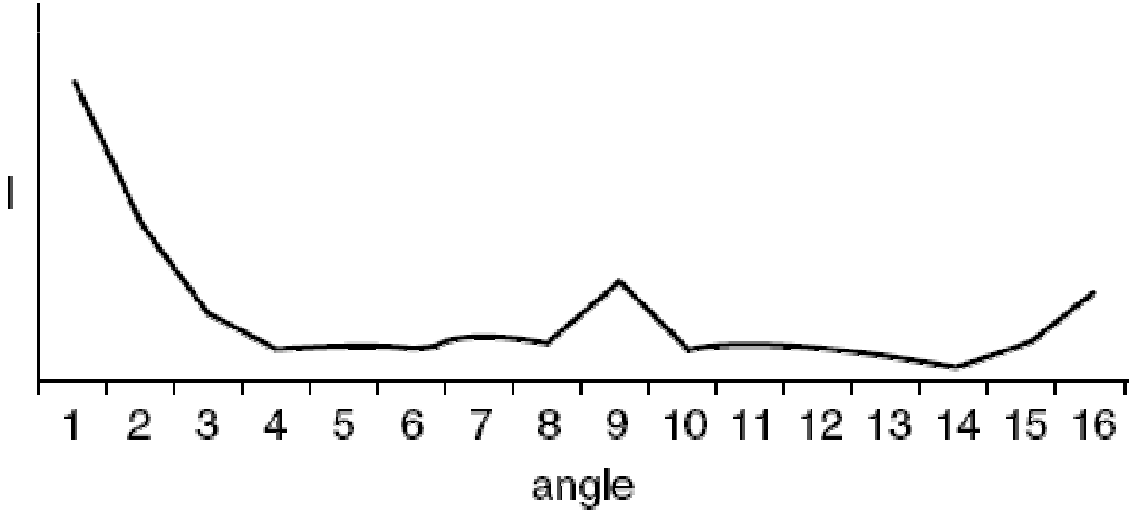
-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

Tamura features

- A histogram $H_{dir}(a)$ of quantized direction values a is constructed by counting numbers of the pixels with the corresponding directional angles and gradient magnitude greater than a predefined threshold
- The histogram is relatively uniform for images without strong orientation and exhibits peaks for highly directional images.



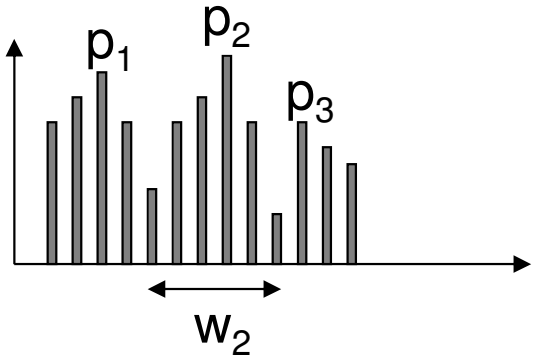


Tamura features

È una stima della varianza attorno al picco: se piccola, il picco è marcato

- The degree of directionality re[illegible]s to the sharpness of the peaks:

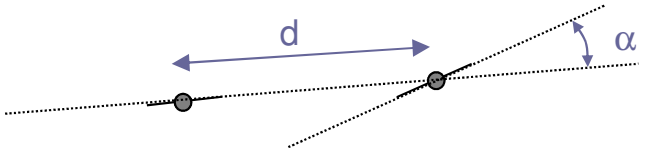
$$F_{dir} = 1 - r\, n_p \sum_{p=1}^{n_p} \sum_{a \in w_p} (a - a_p)^2 H_{dir}(a)$$



- where
  - n_p is the number of peaks,
  - a_p is the position of the p-th peak,
  - w_p is the range of the angles attributed to the p-th peak (that is, the range between valleys around the peak),
  - r denotes a normalizing factor related to quantizing levels of the angles a,
  - a is the quantized directional angle (modulo 180°).

Tamura features

- Three other features are highly correlated with the above three features and do not add much to the effectiveness of texture description.
- The **linelikeness** feature F_{lin} is defined as an average coincidence of the edge directions (more precisely, coded directional angles) that co-occurred in the pairs of pixels separated by a distance *d* along the edge direction in every pixel



- The edge strength is expected to be greater than a given threshold eliminating trivial "weak" edges.
- Coincidence is measured through the cosine of twice (to get values in [-1,1]) the difference between the angles:
  - co-occurrences in the same direction are measured by +1 and those in perpendicular directions by -1.

Tamura features

- The **regularity** feature is defined as

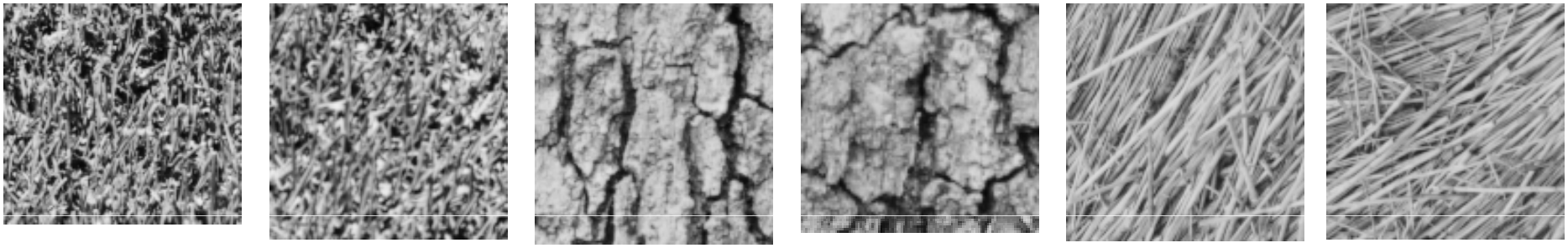
$$F_{reg}=1-r(s_{crs}+s_{con}+s_{dir}+s_{lin})$$

- where r is a normalizing factor and each s_i means the standard deviation of the corresponding feature F_i in each subimage the texture is partitioned into.

- The **roughness** feature is given by simply summing the coarseness and contrast measures: $F_{rgh}=F_{crs}+F_{con}$

- In most cases, only the first three Tamura's features are used.
  - These features capture the high-level perceptual attributes of a texture well.
  - However, they are not very effective for finer texture discrimination.

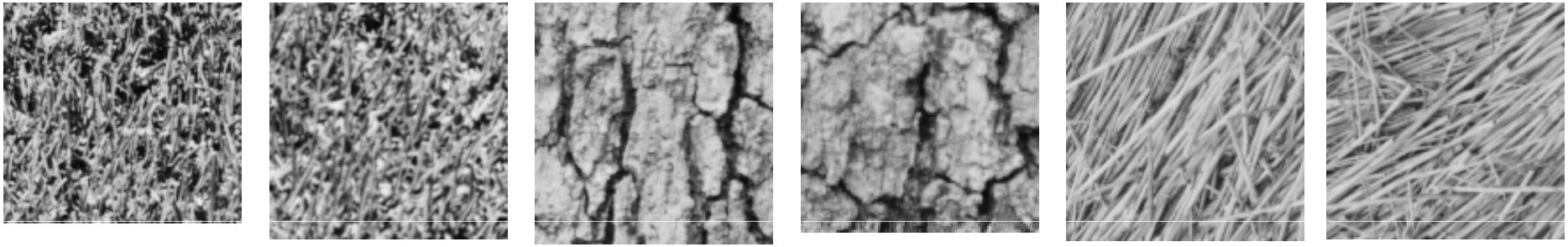
Tamura features



Coars.	30.41	31.35	32.14	26.51	29.20	28.98
Contr.	46.29	39.86	38.72	37.55	29.66	28.26
Direct.	0.0	0.0	0.0	0.0	0.0	0.0
Coars.	0.32	0.79	1.18	-1.62	-0.28	-0.39
Contr.	1.41	0.46	0.29	0.12	-1.04	-1.25
Direct.	0.0	0.0	0.0	0.0	0.0	0.0

Normalized to zero mean and unitary std

Tamura features



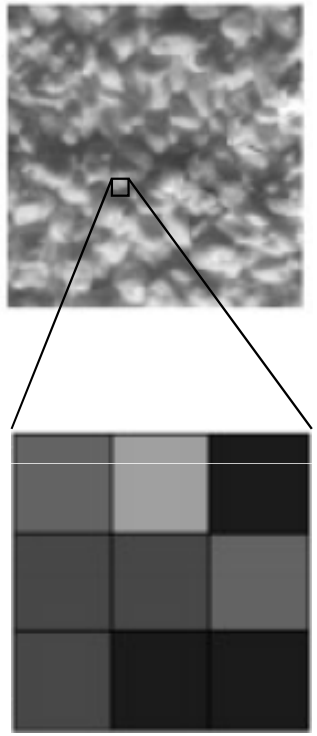
Coars.	30.41	31.35	32.14	26.51	29.20	28.98
Contr.	46.29	39.86	38.72	37.55	29.66	28.26
Direct.	0.0	0.0	0.0	0.0	0.0	0.0
Coars.	0.32	0.79	1.18	-1.62	-0.28	-0.39
Contr.	1.41	0.46	0.29	0.12	-1.04	-1.25
Direct.	0.0	0.0	0.0	0.0	0.0	0.0

Distance matrix

0.00	1.06	1.41	2.33	2.53	2.76
1.06	0.00	0.42	2.43	1.85	2.08
1.41	0.42	0.00	2.81	1.98	2.21
2.33	2.43	2.81	0.00	1.78	1.84
2.53	1.85	1.98	1.78	0.00	0.23
2.76	2.08	2.21	1.84	0.23	0.00

Local Binary Patterns - basic

- Describe the statistics of graylevel differences between a pivot pixel and the surrounding pixels
- Basic approach:
  - The surrounding pixels are the 8 neighbors of the pivot
- Subtract the graylevel of the pivot to the graylevel of the surrounding pixels
- Use a threshold τ to binarize the mask of graylevel differences
- Parse mask values counterclockwise and convert them into a binary number



85	99	21
54	54	86
57	12	13

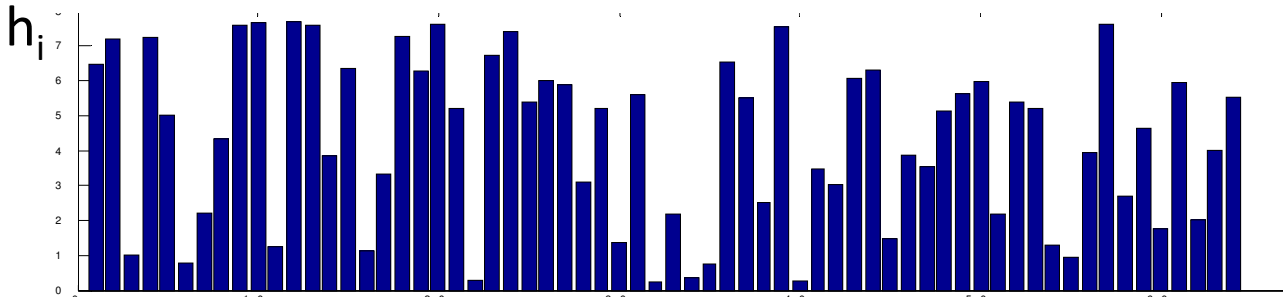
τ

1	1	0
1		1
1	0	0

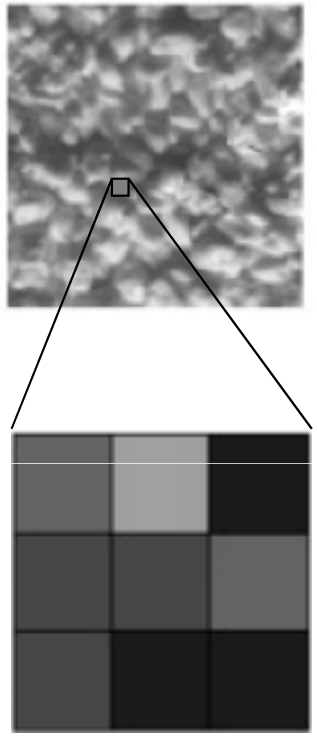
Binary code (8 bits): 11001011

Local Binary Patterns - basic

- In turn, every image pixel acts as pivot and is associated with one 8-bit binary code
  - Each binary code corresponds to one decimal code in {0,…,255}
- The LBP descriptor of the image is the normalized histogram h_i , i=0,…,255 storing, for each code i, the normalized number of pixels associated with code i



Binary code (8 bits): 11001011 = 203 $base_{10}$



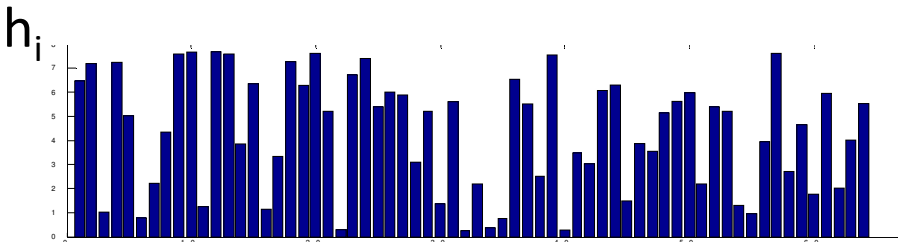
85	99	21
54	54	86
57	12	13

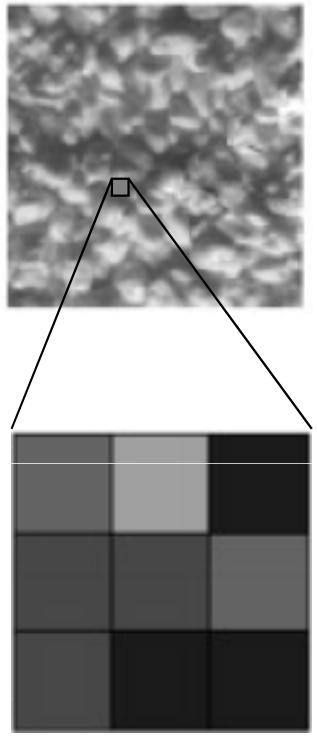
τ

1	1	0
1		1
1	0	0

Local Binary Patterns - basic

- Main traits:
  - **Invariance to uniform grayscale scaling**
  - **No scale control**: the statistics is extracted from 8-neighbors but this is not adequate to model relationships between points at a coarse scale
  - **Robustness to noise**: just a few elements of the histogram capture true texture features. Most of the elements are influenced by noise and do not bring relevant info for texture description
  - **No rotation invariant**: rotating the image yields a circular shift of the binary code and a change of the code in base$_{10}$.





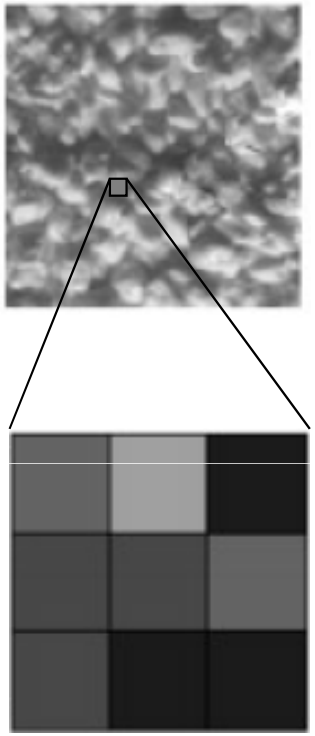
85	99	21
54	54	86
57	12	13

τ

1	1	0
1		1
1	0	0

Local Binary Patterns - adv

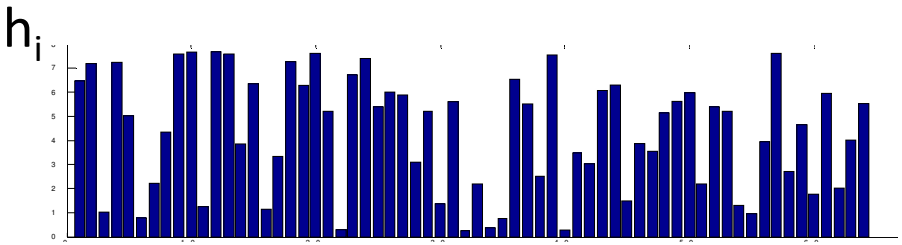
- Improvements:
  - **No scale control**: circular neighborhoods
  - **Robustness to noise**: uniform patterns
  - **No rotation invariant**: code equivalent classes



85	99	21
54	54	86
57	12	13

τ

1	1	0
1		1
1	0	0

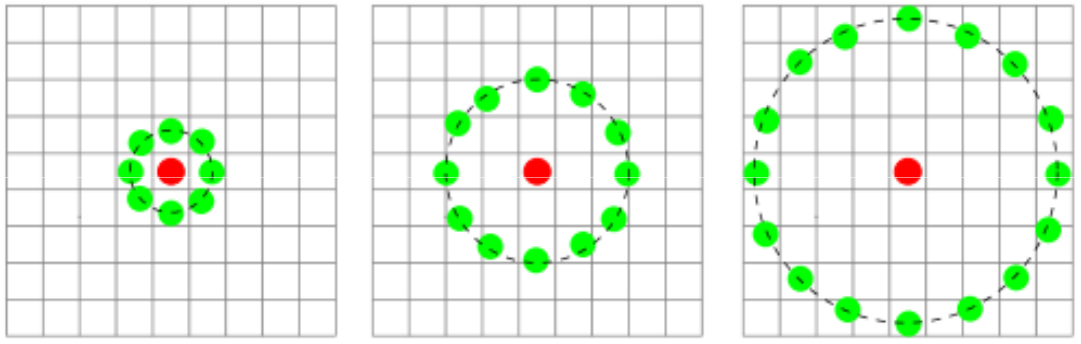


LBP – Circular neighborhoods

- The graylevel of the pivot pixel is compared to the graylevels of P equispaced pixels on a circle of radius R
- Coordinates of 'neighboring' pixels are expressed (wrt the pivot coordinates) as:

$$\left(-R\sin\left(\frac{2\pi\, p}{P}\right), R\cos\left(\frac{2\pi\, p}{P}\right)\right) \qquad p = 0,\ldots,P-1$$

- The pivot is associated with a P-bits code

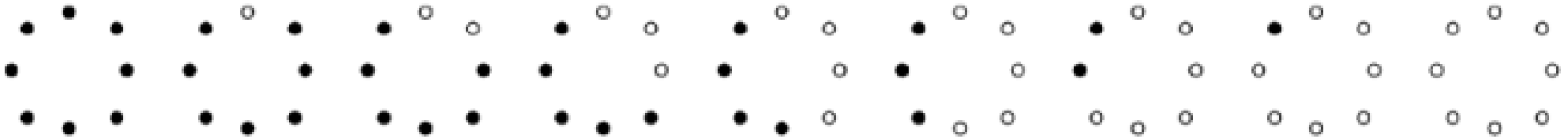


- Since neighboring pixel coordinates do not fit the integer coordinates of the image grid, inter-pixel interpolation is required

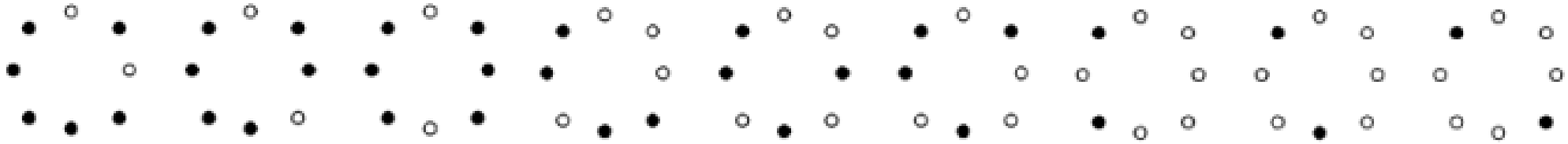
LBP – Uniform patters

- Instead of considering all the 2^P codes originating from all the P-bits combinations, the histogram is built by considering only **uniform codes**
  - A binary code is uniform if at most 2 bit transitions (from 1 to 0 and from 0 to 1) occurr between adjacent pixels

Uniform



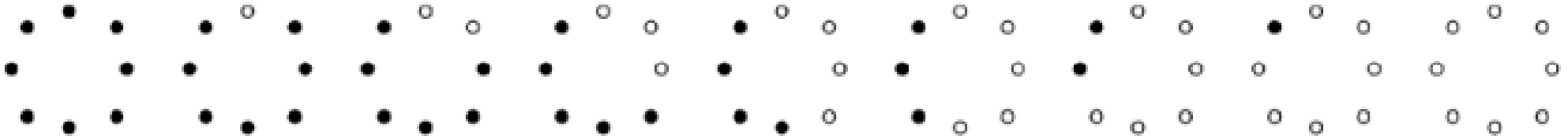
Not uniform



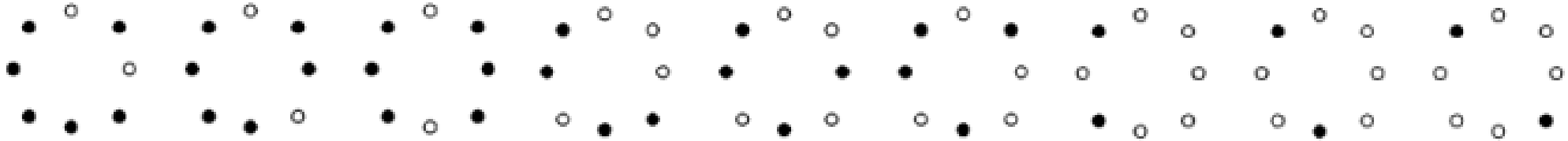
LBP – Uniform patters

- Uniform LBP patterns are demonstrated to depend on relevant texture features and to be robust to noise
- Furthermore, building the histogram using only uniform patterns yields much more compact histograms
  - P=8 yields 256 codes. Among these, only 58 are uniform

Uniform

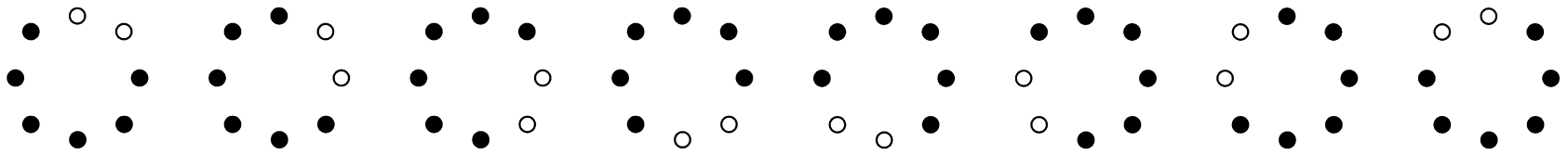


Not uniform



LBP – Rotation invariant codes

- To remove the effect of rotation, each LBP code must be rotated back to a reference position, effectively making all rotated versions of a binary code the same.



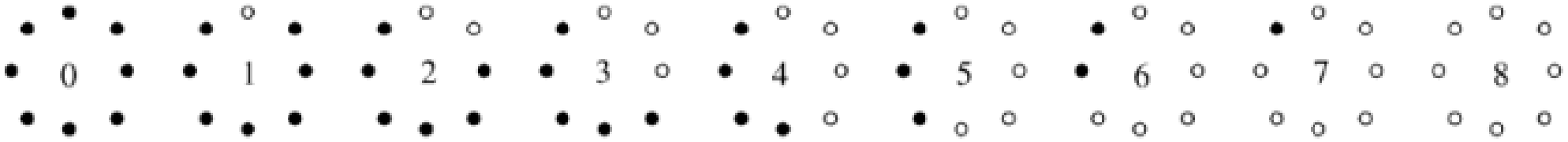
- Each LBP code is represented through its **rotation invariant equivalent** corresponding to the min value among all its circularly rotated versions

$$LBP^{ri}_{P,R} = \min_i \{ROR(LBP_{P,R}, i)\} \quad i = 0,\ldots,P-1$$

being *ROR(LBP,i)* the operator that applies a circular rotation of i bits to the binary LBP code

LBP – Rotation invariant codes

- In the case P=8, if only uniform and rotation invariant codes are considered, the histogram used to describe the texture has just 9 elements, representing the relative occurrence of the following codes



Distance measures

- Several possible dissimilarity measures have been proposed for comparison of a template histogram ***T*** to a model histogram ***M***

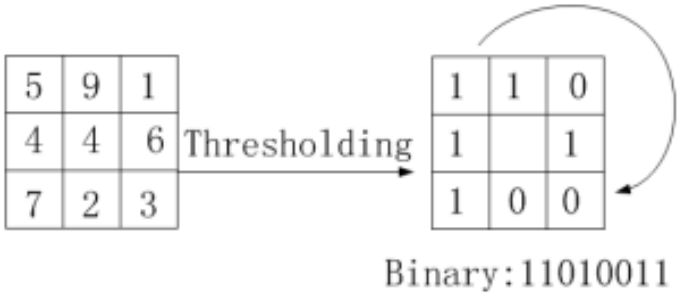
  - Histogram intersection $$D_{HI}(T,M)=\sum_i \min(T_i,M_i)$$

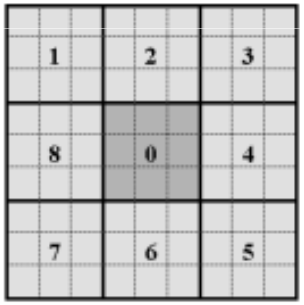
  - Log-likelihood statistic $$D_{LL}(T,M)=-\sum_i T_i \log(M_i)$$

  - Chi square statistic (χ^2) $$D_{\chi^2}(T,M)=\sum_i \frac{(T_i-M_i)^2}{T_i+M_i}$$

Multi-scale Block LBP

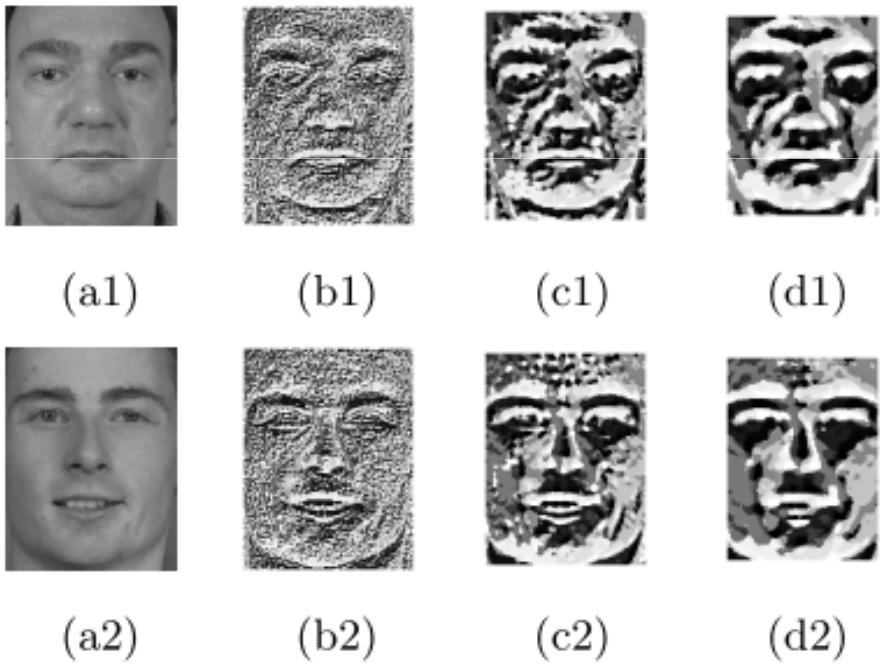
- To improve robustness of the descriptor, a multi-scale block-based approach has been proposed
  - Motivation: In the original LBP the bit-wise comparison between two pixel values is much affected by noise
- In MB-LBP, the comparison operator between single pixels is replaced with comparison between average gray-values of sub-regions
  - The whole filter is composed of 9 blocks





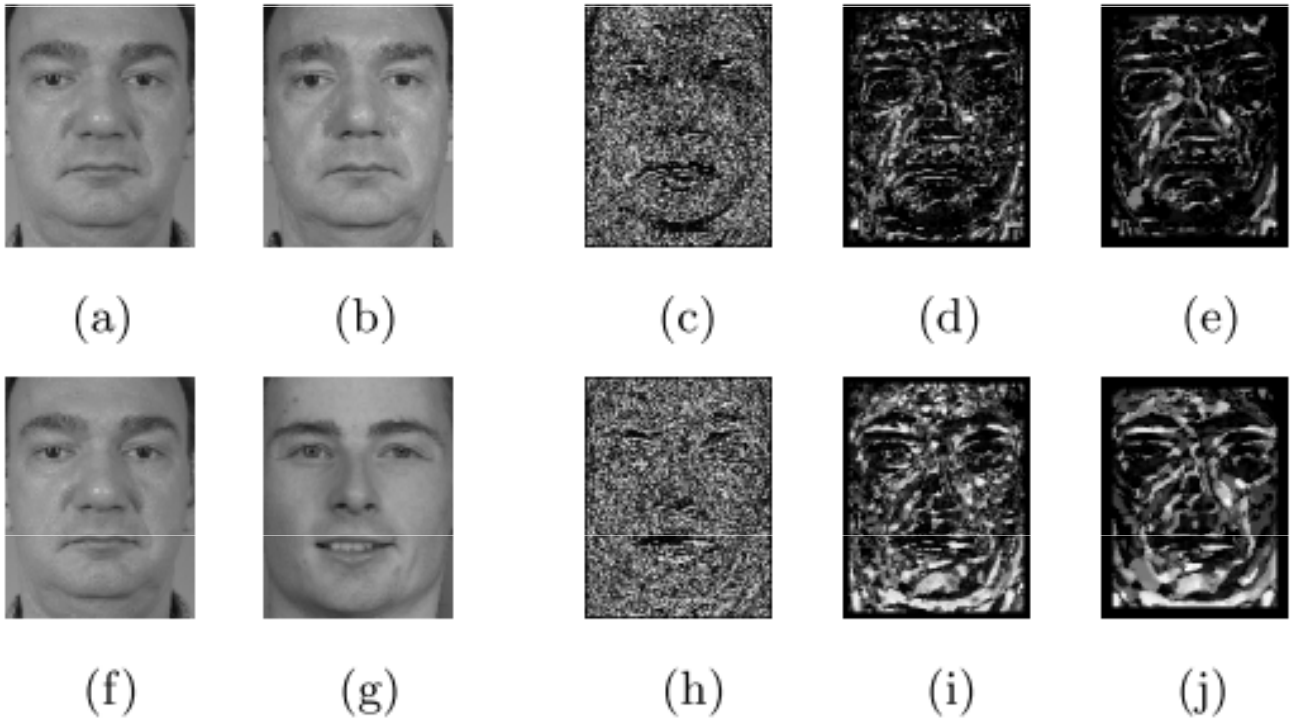
Multi-scale Block LBP

- MB-LBP filtered face images by 3 × 3, 9 × 9 and 15 × 15 blocks
  - For a small scale, local, micro patterns of a face structure is well represented, which may beneficial for discriminating local details of faces
  - The large scale filters reduce noise, and makes the representation more robust
  - Large scale information is complementary to small scale details: much discriminative information is also dropped!



Multi-scale Block LBP

- With larger scale MB-LBP, intra-personal differences are smaller than inter-personal differences

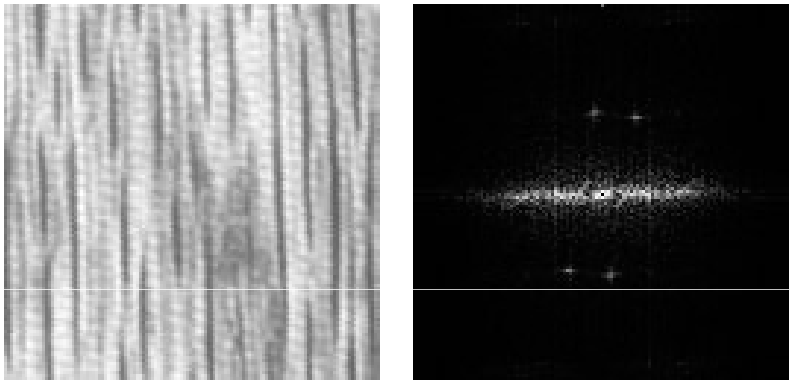


Fourier, cosine transform

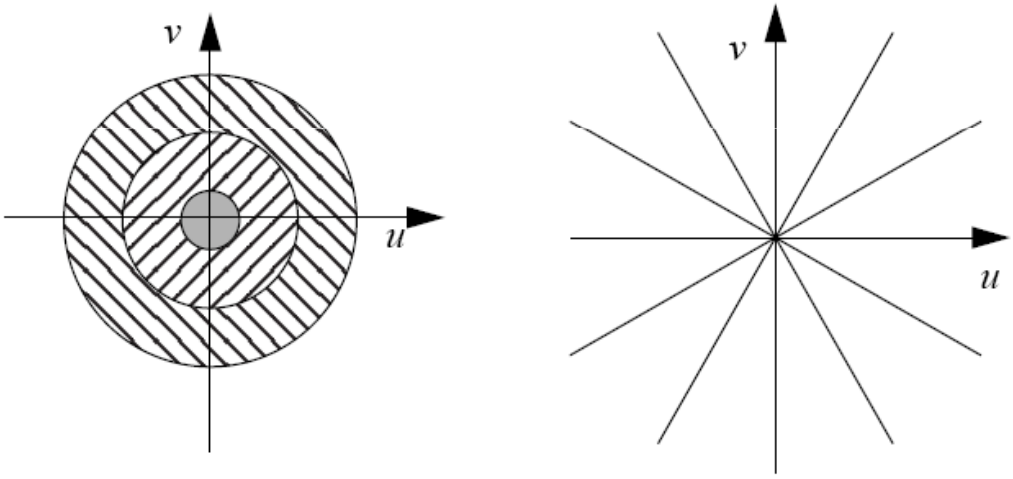
- An alternative to the spatial domain for computing the texture features is to use domains of specific transforms:
  - the discrete Fourier transform (DFT),
  - the discrete cosine transform (DCT),
  - the discrete wavelet transforms (DWT)
- For instance, given an image I(x,y) and its Fourier transform F(u,v), the power spectrum $|F(u,v)|^2$ has been used to extract texture features
  - Some texture properties are reflected in the distribution of energy in the power spectrum

Power spectrum

- Directional textures generate power spectra distributed along a few directions



- The frequency domain is divided into **rings** (for frequency content, i.e. coarse/fine) and **wedges** (for directional content)



- The **total energy in each region** is used to characterize the texture

Fourier, cosine transform

- However, global power spectra computed from the transformed domains are not always effective, compared to local features in small windows

- As an example, the magnitude spectra of the Fourier transform of images tends to be very similar
  - As a result, the magnitude spectra of an image is surprisingly uninformative

Structural approaches

- Image textures consist of organized patterns of quite regular sub-elements, usually called ***textons***.

- One way to represent a texture (**structural approach**) is to find the textons and then describe the way in which they are laid out
  - However, there is no canonical set of textons that is guaranteed to represent generic texture elements

- Alternatively, instead of looking for patterns at the level of textons it makes sense to look for very simple pattern elements (such a dots and bars) and describe their spatial layout

- The advantage of this approach being that simple elements can be detected by simple image filtering

Barness and blobness

- Convolution of an image with a linear filter yields a representation of the image on a different basis
- Representation using a different basis can be advantageous to make clearer the local structure of the image:
  - There is a strong response when the local image pattern looks similar to the filter kernel (and a weak response otherwise)
- This suggests representing image texture in terms of responses to a collection of filters

Barness and blobness

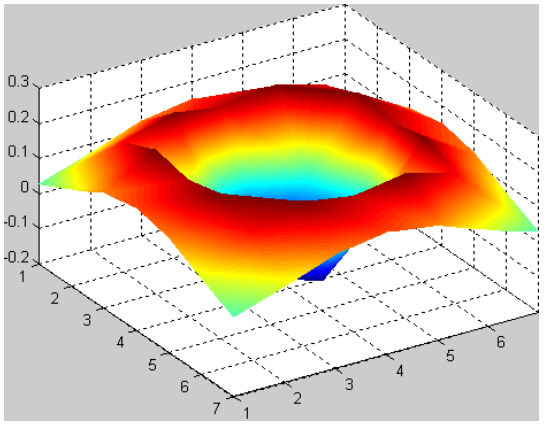
- The filter collection would consist of a set of simple patterns (typically circular and linear elements) at several scales
- Local image characteristics are represented in terms of “spottiness” or “barriness” at each scale.
  - Spot filters are more easy to manage than bar filters as they have a non oriented structure
- However, what should be the specific filter profile (mask) to use?
  - A variety of answers have been tried, however…

Barness and blobness

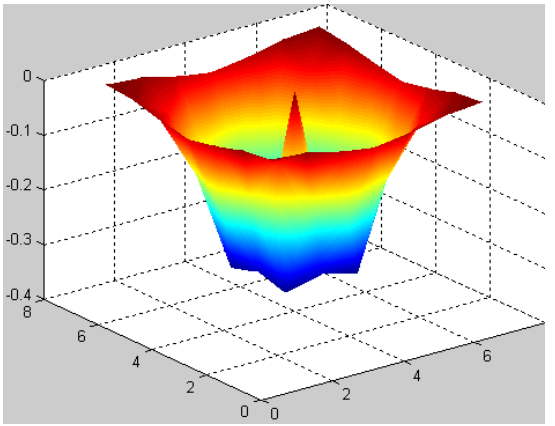
- Studies on the behavior of the human visual cortex suggest usage of at least one spot filter and a collection of oriented bar filters at different orientations and scales

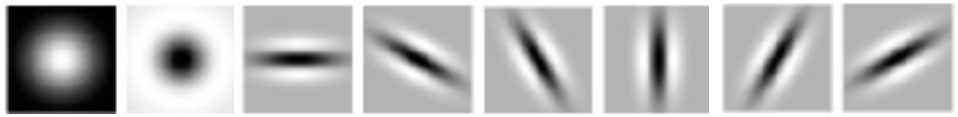
- One way to obtain these filters is to form a weighted difference of Gaussian filters at different scales

Barness and blobness

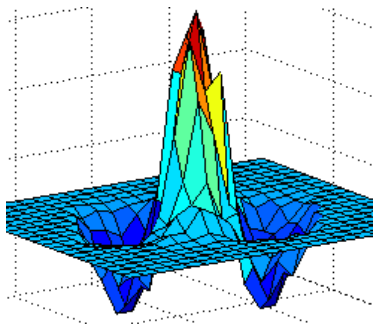


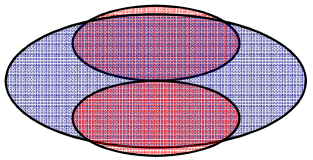
- For instance, a possible choice can be:
  - One spot given by the weighted sum of three concentric, symmetric Gaussians with weights 1, -2 and 1 and σ 0.62, 1 and 1.6
  - Another spot given by the weighted sum of two concentric, symmetric Gaussians with weights 1 and -1 and σ 0.71 and 1.14





  - Six oriented bars, obtained by rotating a horizontal bar.
    - The horizontal bar is given by the weighted sum of three Gaussians with weights 1, -2 and 1. All Gaussians have sigma 2 in the x direction and 1 in the y direction. Centers of the Gaussians are offset along the y axis, lying at (0,1), (0,0) and (0,-1)





Barness and blobness

- A set of filtered images is not a representation of texture
  - Some representation of the overall distribution of texture elements is necessary
- If the size of the image window in which we want to represent the texture is known, a set of statistics of filter outputs for that windows can be considered:
  - Mean value of filter outputs over the window
  - Mean and standard deviation values of filter output over the window
- Alternative representation of the filter output include
  - Textons: by filter output clustering
  - Histograms: by filter output quantization

Gabor filters

- As an alternative to simple bar and spot templates, representation of image features based on **Gabor filters** has also been investigated

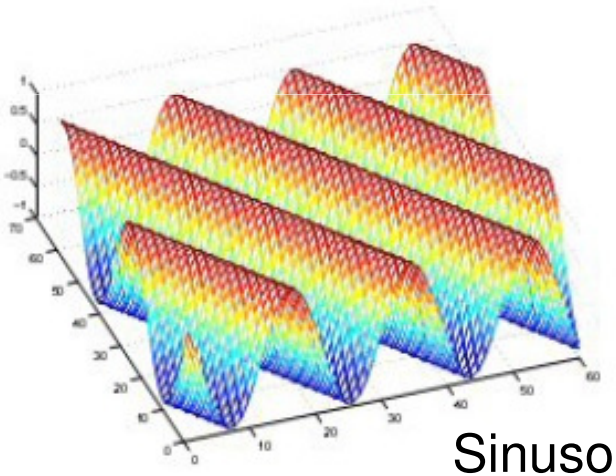
- Gabor filters perform a transform similar to Fourier with the advantage of preserving spatial information:
  - Gabor kernels look like Fourier basis elements (sinusoids) that are multiplied by Gaussians

- Gabor filters come in pairs (quadrature pairs)
  - One of the pair recovers symmetric components in a particular direction
  - The other one recovers antisymmetric components

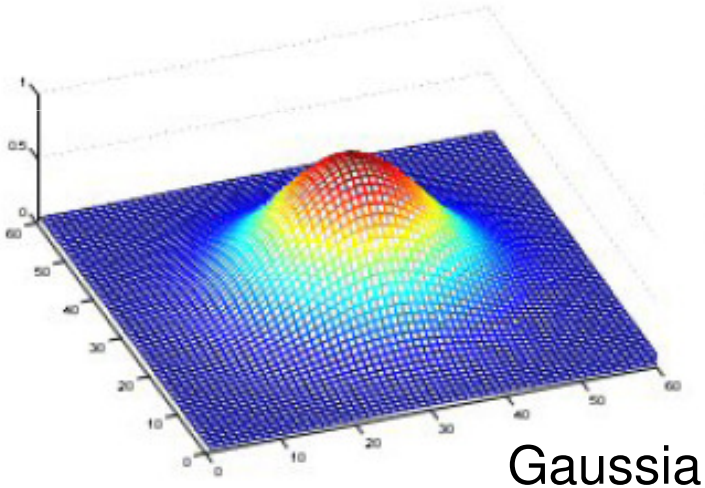
Gabor filters

$$G_S(x,y) = \cos(k_x x + k_y y)e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \qquad G_A(x,y) = \sin(k_x x + k_y y)e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

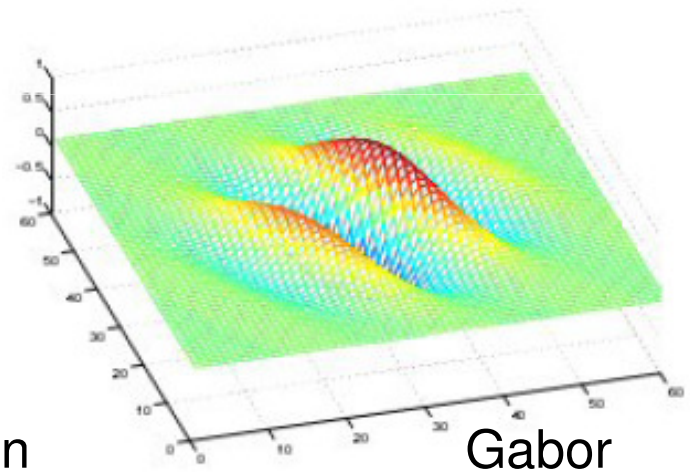
- k_x and k_y give the **spatial frequency** of the filter
- The ratio k_x/k_y determines the **orientation** of the filter
- σ is referred to as the **scale** of the filter



Sinusoid

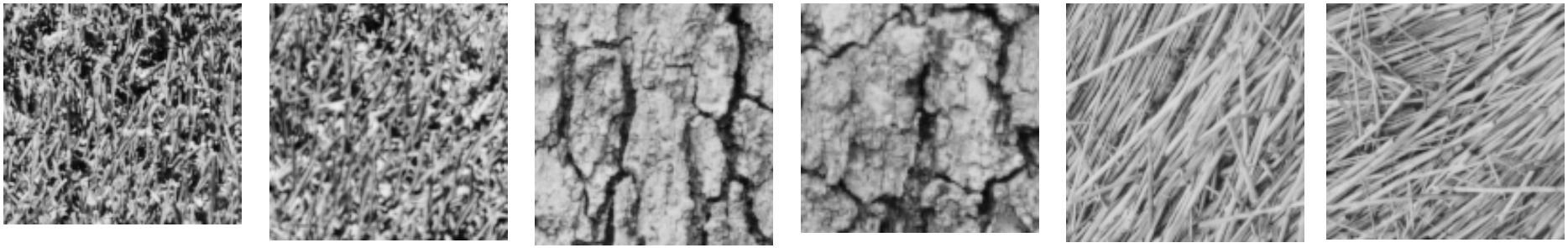


Gaussian

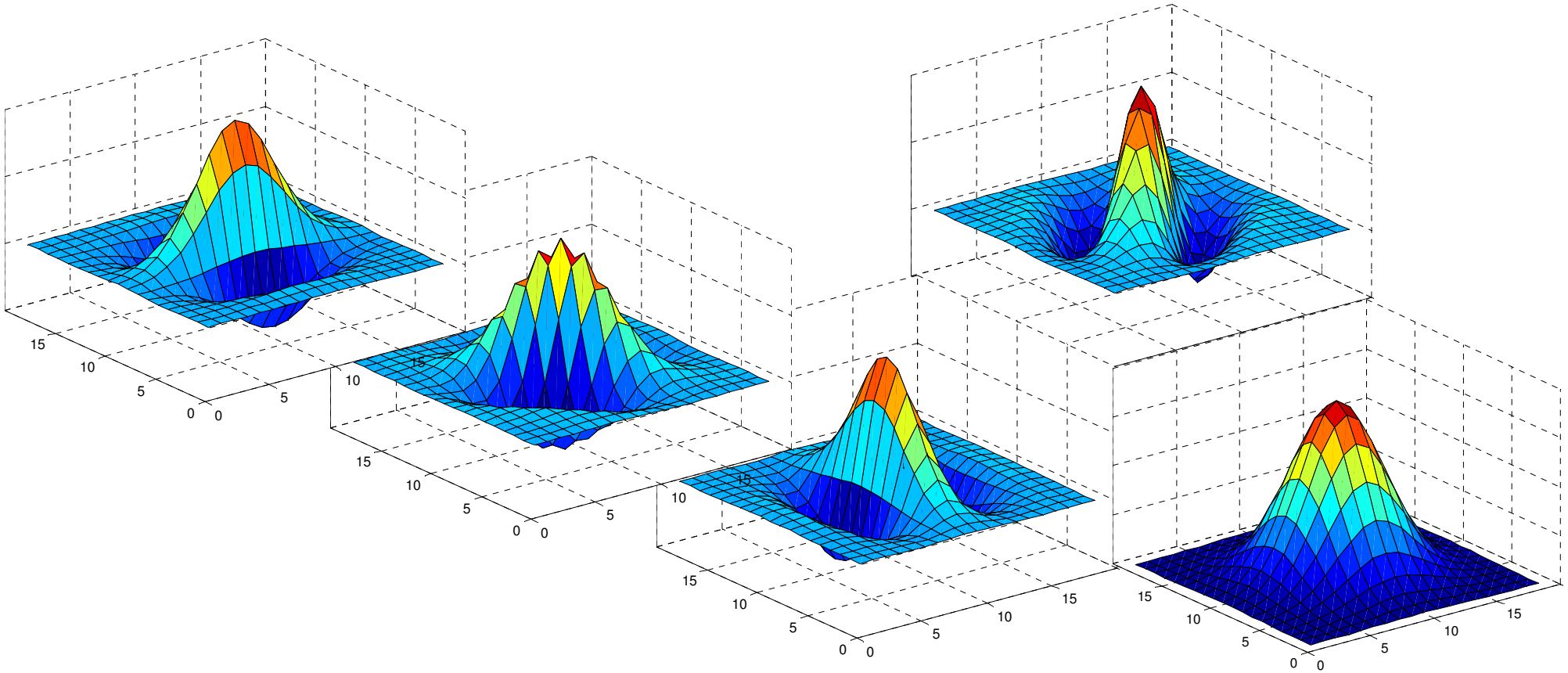


Gabor

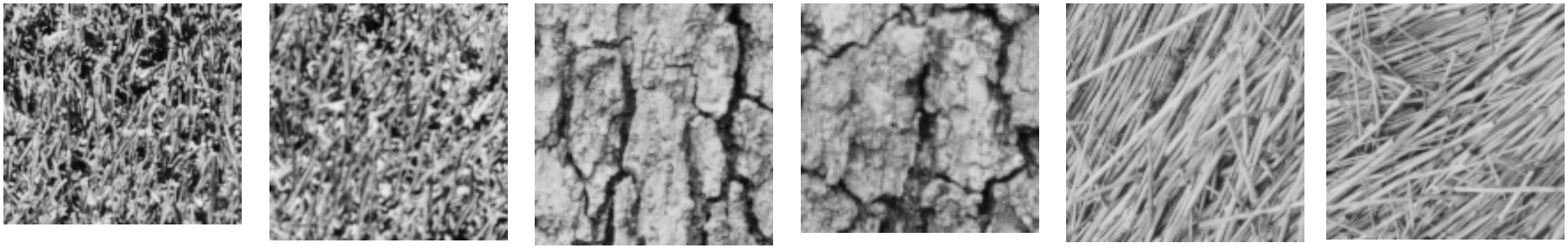
Covariance of Gabor filters



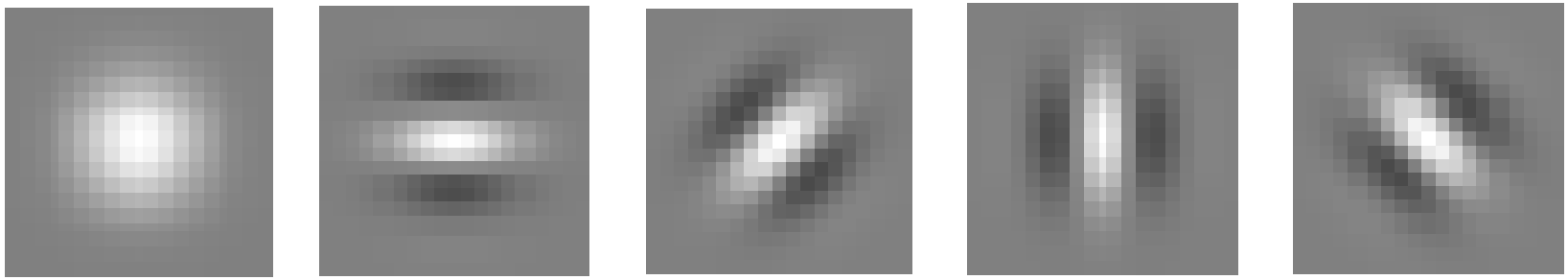
- Feature vectors are extracted by combining convolution with a Gaussian and four symmetric Gabor masks oriented at 0, 45, 90, 135 degree:



Covariance of Gabor filters



- Each region is described by a 5x5 covariance matrix:

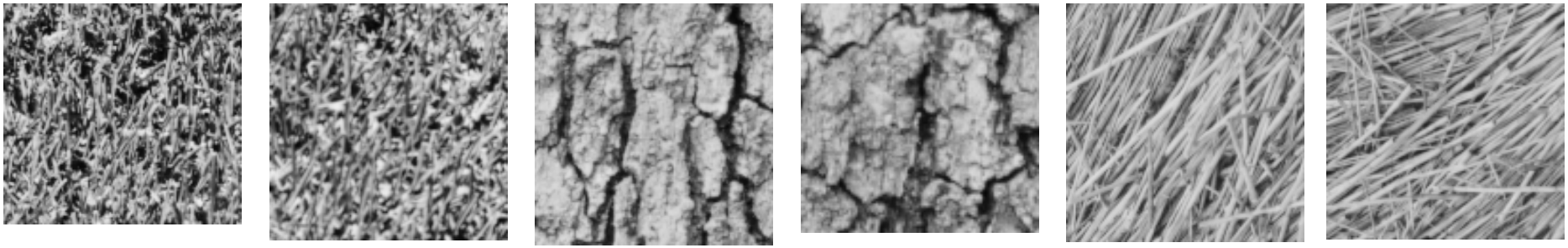


Distance matrix

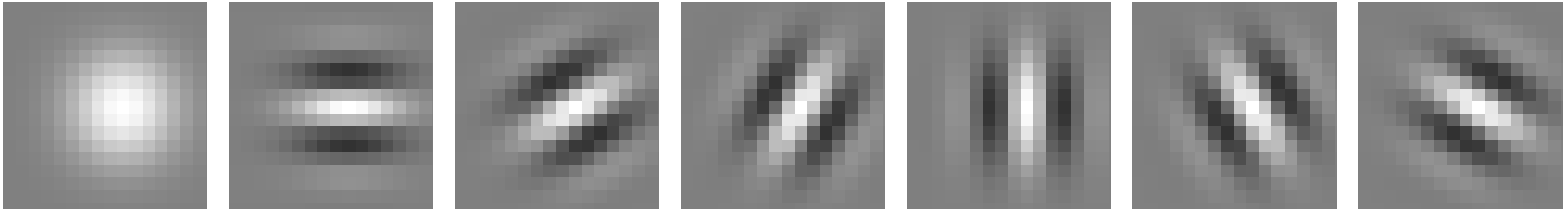
0.00	0.35	1.10	1.13	2.23	2.50
0.35	0.00	1.31	1.27	2.12	2.27
1.10	1.31	0.00	0.49	2.21	2.61
1.13	1.27	0.49	0.00	1.99	2.33
2.23	2.12	2.21	1.99	0.00	1.32
2.50	2.27	2.61	2.33	1.32	0.00

Use of a finer quantization of the orientation is expected to boost the accuracy or not?

Covariance of Gabor filters



- Each region is described by a 5x5 covariance matrix:



Distance matrix

0.00	0.35	1.48	1.42	2.56	2.48
0.33	0.00	1.42	1.28	2.40	2.25
1.37	1.28	0.00	0.38	1.81	1.75
1.35	1.17	0.48	0.00	1.83	1.72
1.77	1.64	1.49	1.51	0.00	1.05
1.86	1.68	1.92	1.57	1.28	0.00

Gray level Moments

- **Moments** can be used to represent any data distribution, including gray level values
- The general form of a moment of order (p+q) of a function f(i,j) over a region R is:

$$M_{pq} = \iint_R \psi_{pq}(i,j) f(i,j)\, di\, dj \approx \sum_{(i,j)\in R} \psi_{pq}(i,j) f(i,j)$$

being ψ_{pq} the **weighting kernel** or **basis function**

- Depending on values of the basis functions, moments may have useful properties such as invariance under rotation, scale and translation

Cartesian moments

- **Cartesian moments** are defined by considering the following values for the basis functions:

$$\psi_{pq}(i,j)=i^p j^q$$

- **Uniqueness theorem**:
  The sequence of Cartesian moments m_{pq} is uniquely defined by function f(i,j) and conversely, f(i,j) is uniquely defined by the sequence m_{pq}

- However, computation of the formula expressing f(i,j) based on m_{pq} values is not straightforward

Cartesian moments

- The zero order Cartesian moment m_{00} is defined as the **total mass** (or power) of the image f(i,j)

$$m_{00} = \sum_{(i,j)\in R} f(i,j)$$

- For a binary image m_{00} represents the number of non-zero pixel of f(i,j) on R
- The two first order Cartesian moments are used to find the **center of mass** of the image f(i,j)

$$\bar{x} = \frac{m_{01}}{m_{00}} = \frac{\sum_{(i,j)\in R} f(i,j)\, j}{m_{00}} \qquad \bar{y} = \frac{m_{10}}{m_{00}} = \frac{\sum_{(i,j)\in R} f(i,j)\, i}{m_{00}}$$

Centralized moments

- **Centralized moments** are defined by translating Cartesian moments wrt the center of mass:

$$\mu_{pq} = \sum_{(i,j)\in R} (i - \bar{y})^p (j - \bar{x})^q f(i,j)$$

- Centralized moments are invariant to translation

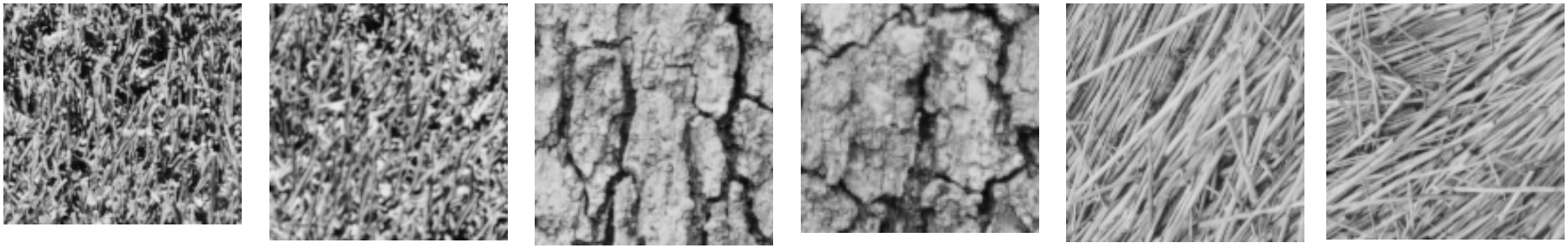
- To support invariance also to scale, **normalized moments** should be considered:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

where $\gamma = \frac{p+q}{2} + 1 \qquad \forall (p+q) \geq 2$

Since $\mu_{10} = \mu_{01} = 0$ then (p+q =1) $\eta_{01} = \eta_{10} = 0$

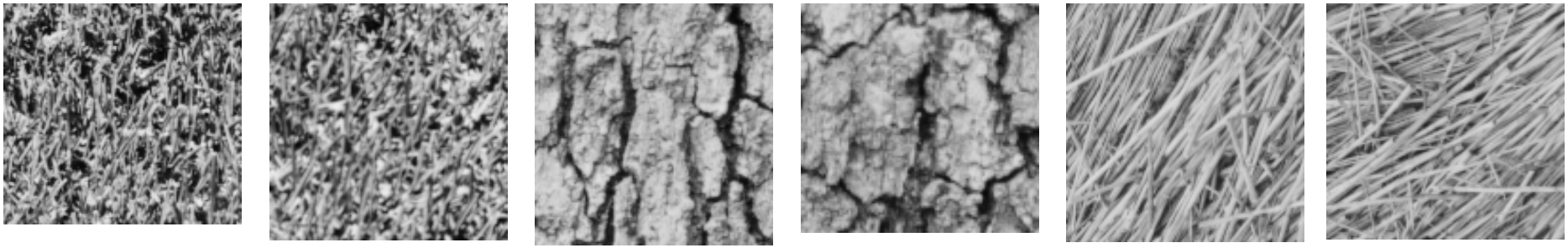
Normalized moments



η_{11}	0.3557	-1.5405	-0.1463	1.5245	0.1643	-0.3576
η_{20}	1.7159	0.2485	-0.8198	0.3538	-0.9414	-0.5571
η_{02}	1.4165	1.0248	-0.0895	-0.5830	-1.0090	-0.7599
η_{30}	1.0549	-1.5356	0.6265	0.7045	-0.0519	-0.7984
η_{03}	-1.5400	-0.4730	0.1841	1.5187	0.2447	0.0655
η_{21}	-0.5801	-0.6667	0.1751	1.8149	0.1844	-0.9276
η_{12}	-1.0746	-0.4859	-0.4579	-0.4265	1.5607	0.8842

Normalized to zero mean and unitary std

Normalized moments



η_{11}	0.3557	-1.5405	-0.1463	1.5245	0.1643	-0.3576
η_{20}	1.7159	0.2485	-0.8198	0.3538	-0.9414	-0.5571
η_{02}	1.4165	1.0248	-0.0895	-0.5830	-1.0090	-0.7599
η_{30}	1.0549	-1.5356	0.6265	0.7045	-0.0519	-0.7984
η_{03}	-1.5400	-0.4730	0.1841	1.5187	0.2447	0.0655
η_{21}	-0.5801	-0.6667	0.1751	1.8149	0.1844	-0.9276
η_{12}	-1.0746	-0.4859	-0.4579	-0.4265	1.5607	0.8842

Distance matrix

0.00	3.75	3.61	4.78	4.99	4.51
3.75	0.00	3.18	5.20	4.01	2.83
3.61	3.18	0.00	2.98	2.34	2.37
4.78	5.20	2.98	0.00	3.53	4.24
4.99	4.01	2.34	3.53	0.00	1.66
4.51	2.83	2.37	4.24	1.66	0.00

Hu invariant set

- Normalized moments are invariant to translation and scale

- By considering expressions of algebraic invariants applied to the moment generating function under a rotation transform, Hu defined a set of moment invariants that can be used for **scale, position and rotation invariant** pattern identification

- These invariants are computed from normalized moments up to order three

Hu invariant set

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} + \eta_{02})^2 + 4(n_{11})^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

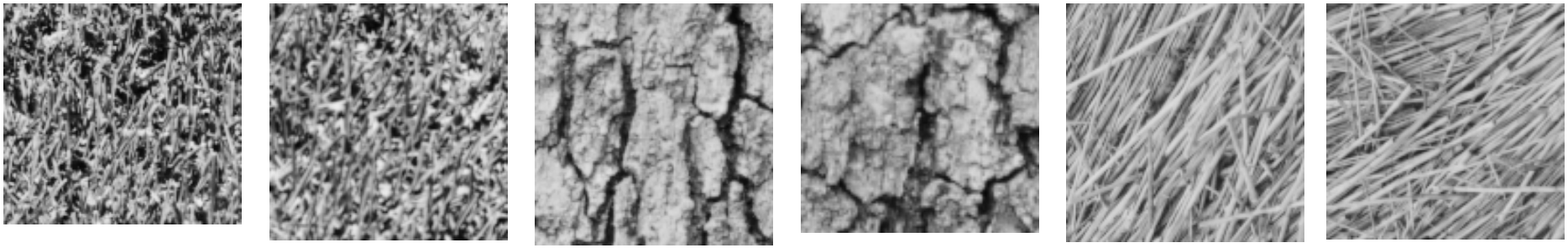
$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})]$$

and the last one, that is also invariant to skewing

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

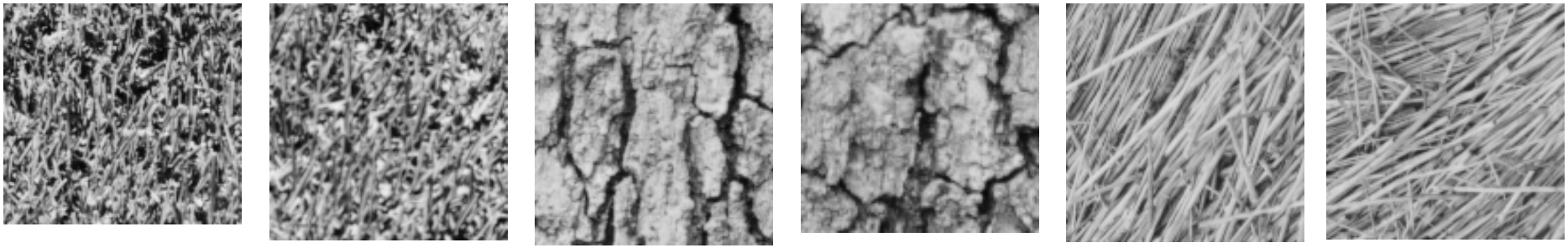
Hu invariant features



I_1	1.6997	0.5316	-0.6076	0.0458	-1.0132	-0.6563
I_2	1.6845	-0.2065	-0.6585	0.7095	-0.8479	-0.6811
I_3	1.8183	-0.7512	0.1632	0.2214	-0.7255	-0.7262
I_4	0.6887	0.2534	-0.8911	1.5429	-0.8882	-0.7057
I_5	0.0948	0.2578	0.5311	-2.0037	0.5473	0.5727
I_6	-1.1110	0.2789	0.7356	-1.4154	0.7434	0.7685
I_7	-1.5897	-0.0533	0.0160	1.5710	0.0223	0.0337

Normalized to zero mean and unitary std

Hu invariant features



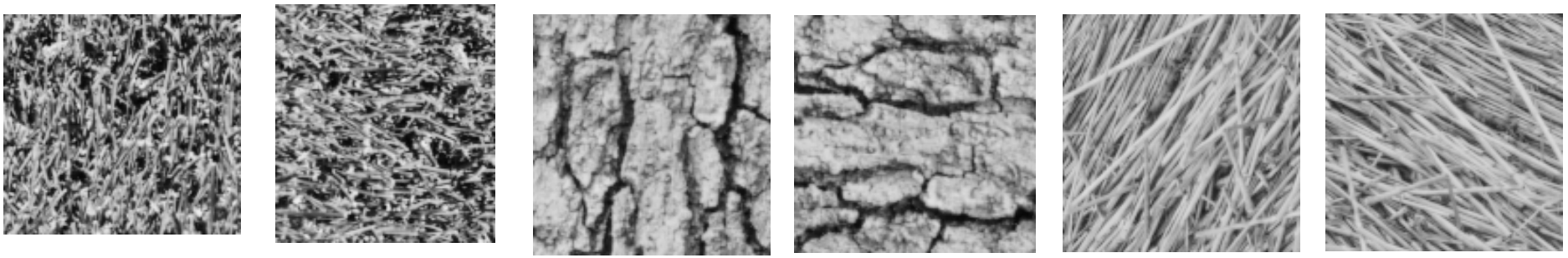
I_1	1.6997	0.5316	-0.6076	0.0458	-1.0132	-0.6563
I_2	1.6845	-0.2065	-0.6585	0.7095	-0.8479	-0.6811
I_3	1.8183	-0.7512	0.1632	0.2214	-0.7255	-0.7262
I_4	0.6887	0.2534	-0.8911	1.5429	-0.8882	-0.7057
I_5	0.0948	0.2578	0.5311	-2.0037	0.5473	0.5727
I_6	-1.1110	0.2789	0.7356	-1.4154	0.7434	0.7685
I_7	-1.5897	-0.0533	0.0160	1.5710	0.0223	0.0337

Distance matrix

0.00	4.00	4.71	4.63	5.38	5.09
4.00	0.00	1.98	3.78	2.09	1.70
4.71	1.98	0.00	4.65	0.99	0.91
4.63	3.78	4.65	0.00	4.89	4.70
5.38	2.09	0.99	4.89	0.00	0.43
5.09	1.70	0.91	4.70	0.43	0.00

Hu invariant features

- It should be noticed that the values of the invariants are exactly the same when computed on images that differ only by a rotation transform



I_1	1.2785	1.2785	-0.4843	-0.4843	-0.7942	-0.7942
I_2	1.2881	1.2881	-0.5690	-0.5690	-0.7191	-0.7191
I_3	1.2121	1.2121	-0.2213	-0.2213	-0.9908	-0.9908
I_4	1.2910	1.2910	-0.6472	-0.6472	-0.6437	-0.6437
I_5	-1.2904	-1.2904	0.6100	0.6100	0.6803	0.6803
I_6	-1.2910	-1.2910	0.6414	0.6414	0.6496	0.6496
I_7	-1.2910	-1.2910	0.6417	0.6417	0.6493	0.6493