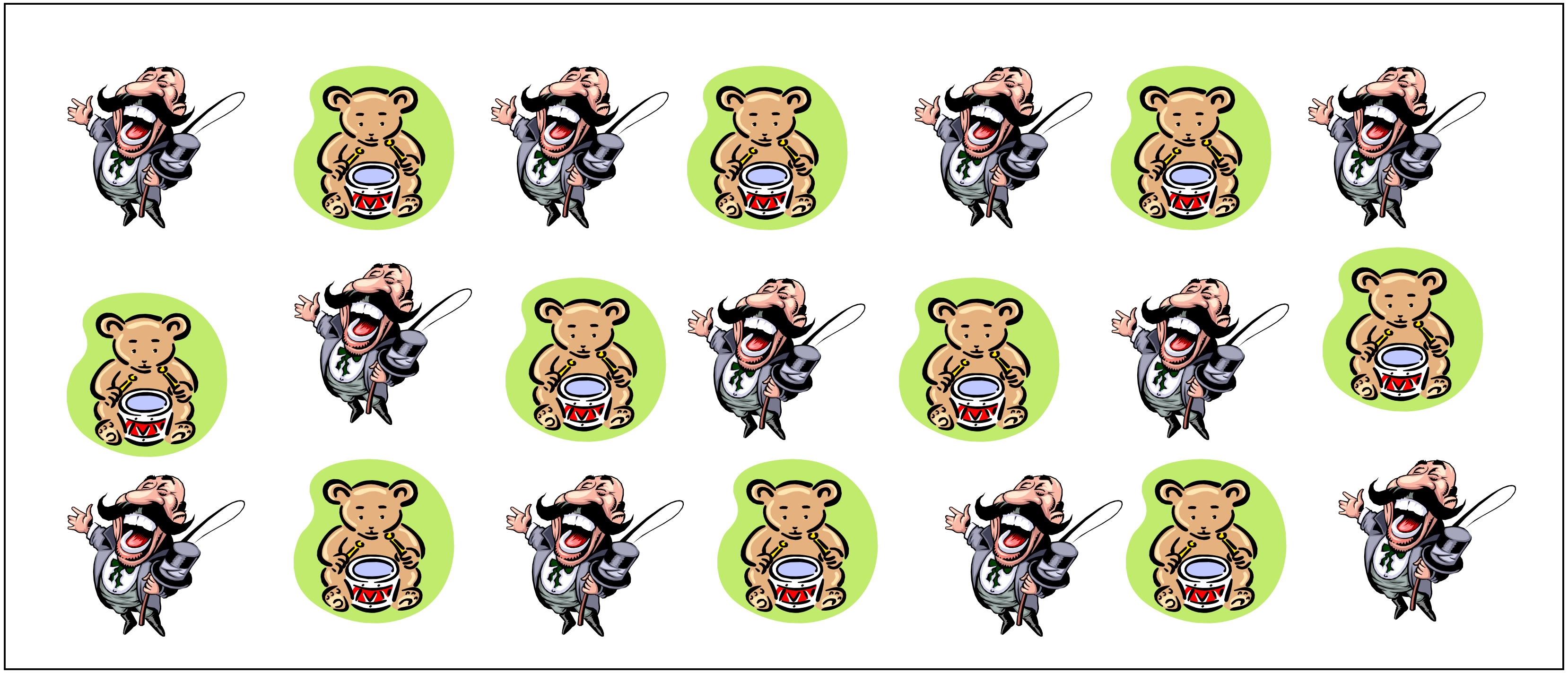
Texture

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



Why Texture Analysis?

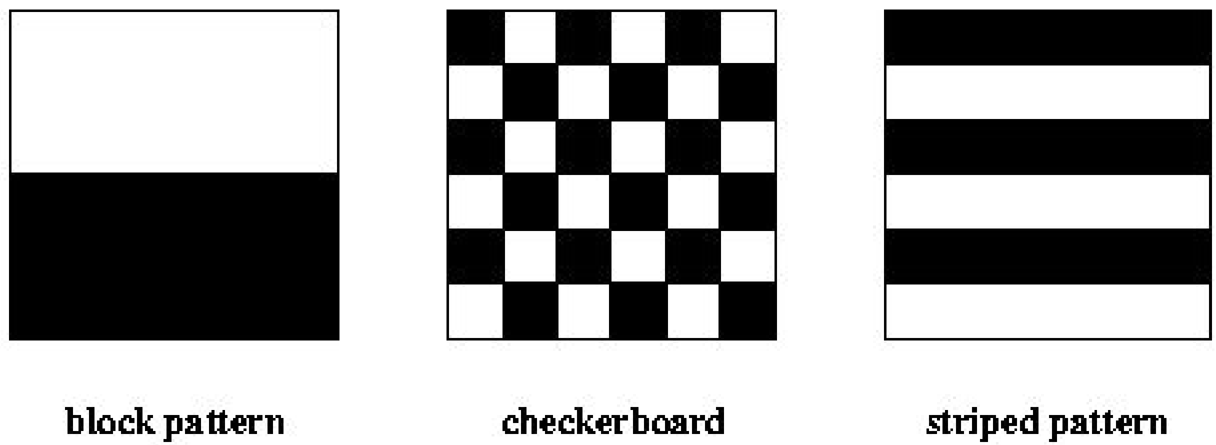


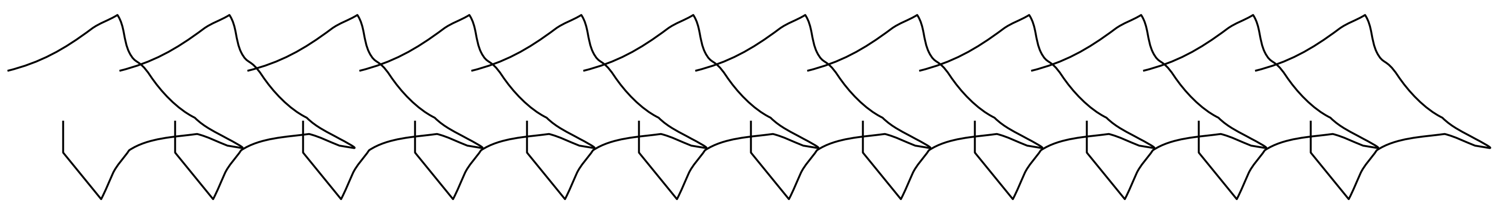
Figure 7.2: Three different textures with the same distribution of black and white.

Aspects of texture

- Size or granularity (sand versus pebbles versus boulders)
- Directionality (stripes versus sand)
- Random or regular (sawdust versus woodgrain; stucko versus bricks)
- Concept of texture elements (texel) and spatial arrangement of texels

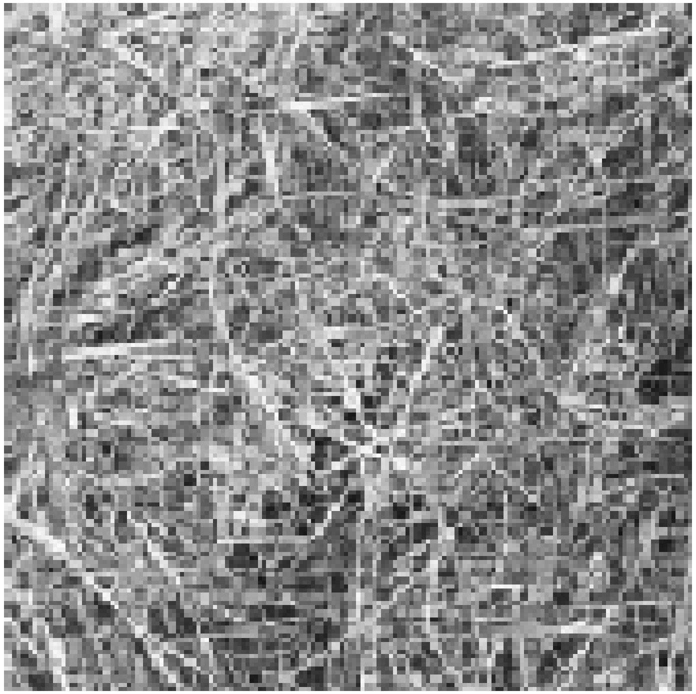
Problem with Structural Approach

How do you decide what is a texel?

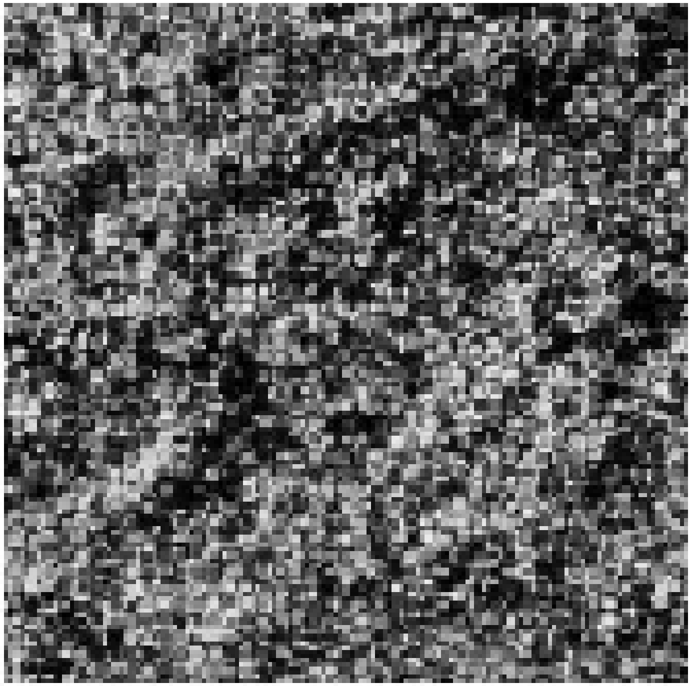


Ideas?

Natural Textures



grass



leaves

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

$$\boxed{F_{edgeness} = |\{ p \mid \text{gradient\_magnitude}(p) \geq \text{threshold}\}| / N}$$

where N is the size of the unit area

2. edge magnitude and direction histograms

$$\boxed{F_{magdir} = (H_{magnitude}, H_{direction})}$$

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

How would you compare two histograms?

Examples

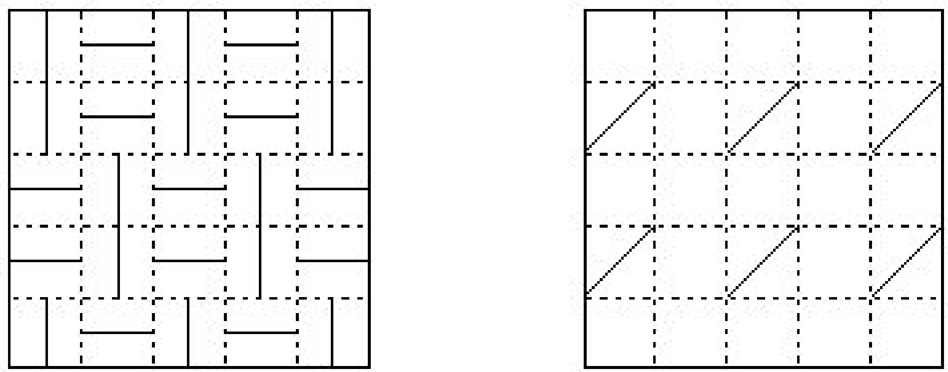


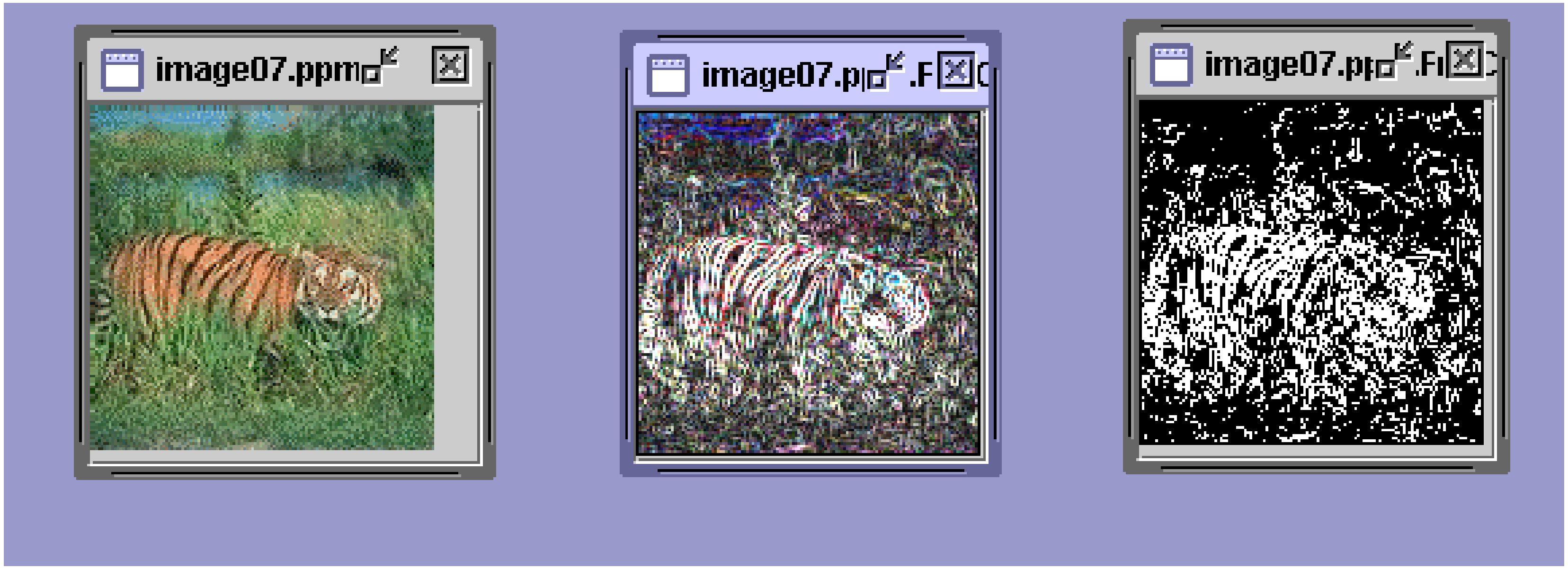
Figure 7.5: Two images with different edgeness and edge-direction statistics.

$F_e = 25/25$	$F_e = 6/25$
$H_m = (6,19)/25$	$H_m = (0,6)/25$
$H_d = (12,13,0)/25$	$H_d = (0,0,6)/25$

Original Image

Frei-Chen Edge Image

Thresholded Edge Image



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.

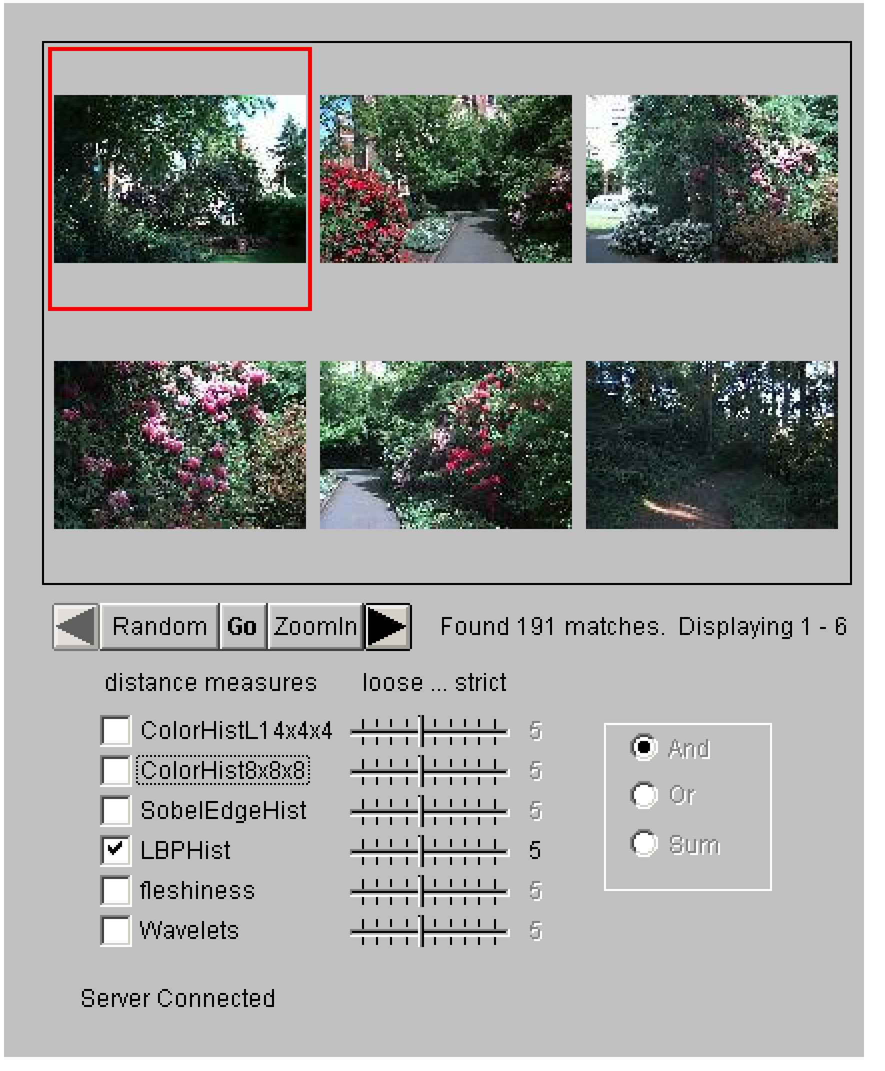
- Represent the texture in the image (or a region) by the histogram of these numbers.

	1	2	3	
	100	101	103	
8	40	50	80	4
	50	60	90	5
	7	6		

$\longrightarrow$ $\boxed{1\ 1\ 1\ 1\ 1\ 1\ 0\ 0}$

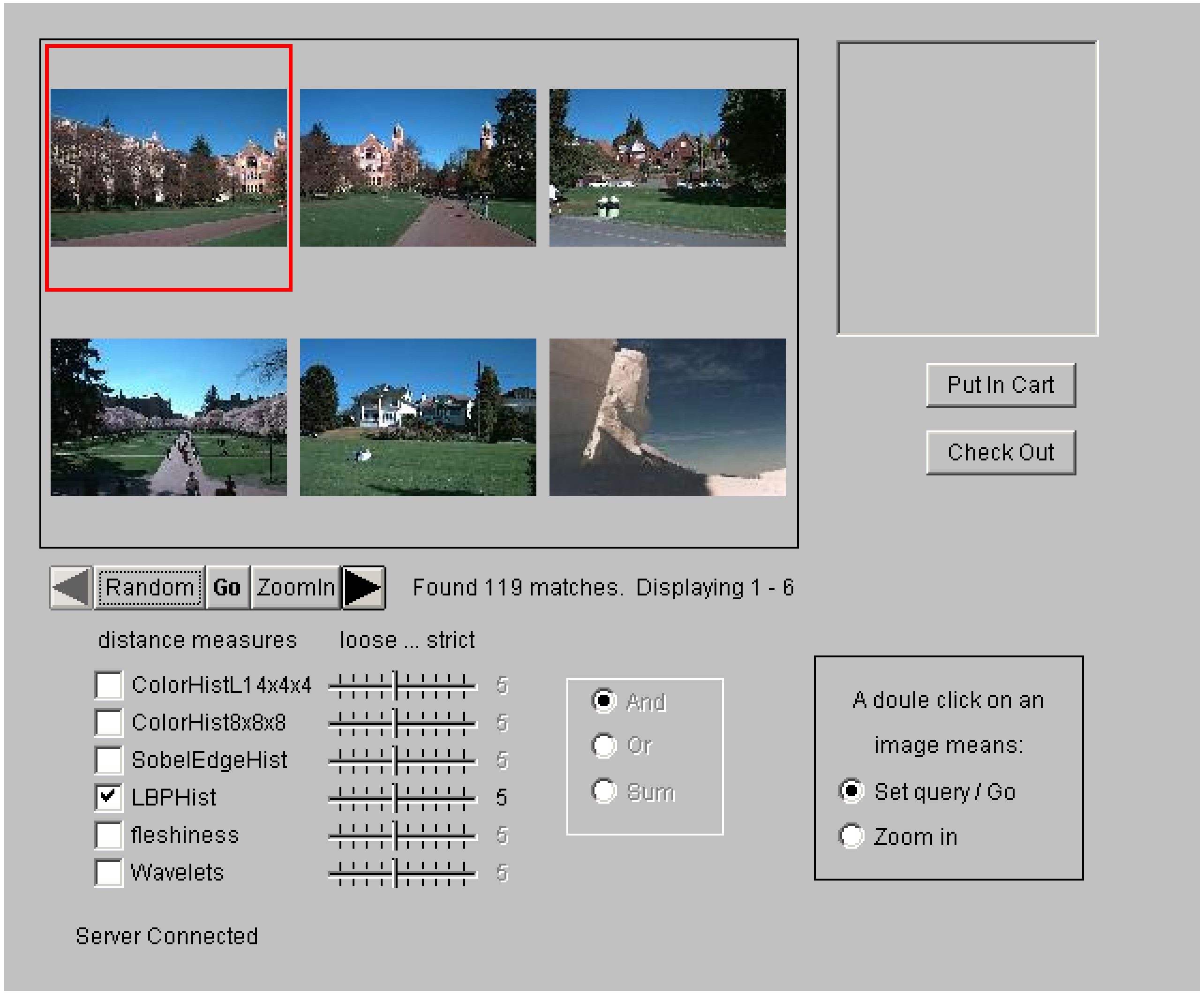
Fids demo

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



Fids demo

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

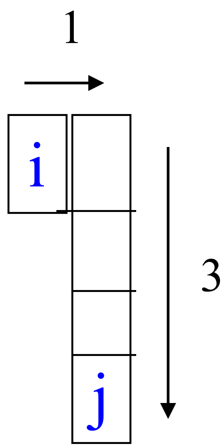
- Both the rows and columns represent a set of possible image values

- $C_d(i,j)$ 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).

$$\boxed{\begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{matrix}}$$

gray-tone image



$$\boxed{d = (3,1)}$$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 0 & 3 \\ 1 & 2 & 0 & 2 \\ 2 & 0 & 0 & 1 \end{array} \qquad C_d$$

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?

$$Energy = \sum_i \sum_j N_d^2(i,j) \tag{7.7}$$

$$Entropy = -\sum_i \sum_j N_d(i,j) log_2 N_d(i,j) \tag{7.8}$$

$$Contrast = \sum_i \sum_j (i-j)^2 N_d(i,j) \tag{7.9}$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i,j)}{1+ \mid i-j \mid} \tag{7.10}$$

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d(i,j)}{\sigma_i \sigma_j} \tag{7.11}$$

where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums.

Energy measures uniformity of the normalized matrix.

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 χ^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)

$$
\begin{array}{lll}
\text{L5} & \text{(Level)} & = [\ 1\ \ 4\ \ 6\ \ 4\ \ 1\ ] \\
\text{E5} & \text{(Edge)} & = [\ -1\ \ -2\ \ 0\ \ 2\ \ 1\ ] \\
\text{S5} & \text{(Spot)} & = [\ -1\ \ 0\ \ 2\ \ 0\ \ -1\ ] \\
\text{R5} & \text{(Ripple)} & = [\ 1\ \ -4\ \ 6\ \ -4\ \ 1\ ]
\end{array}
$$

- (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 –

E5

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

L5

E5L5

9D feature vector for pixel

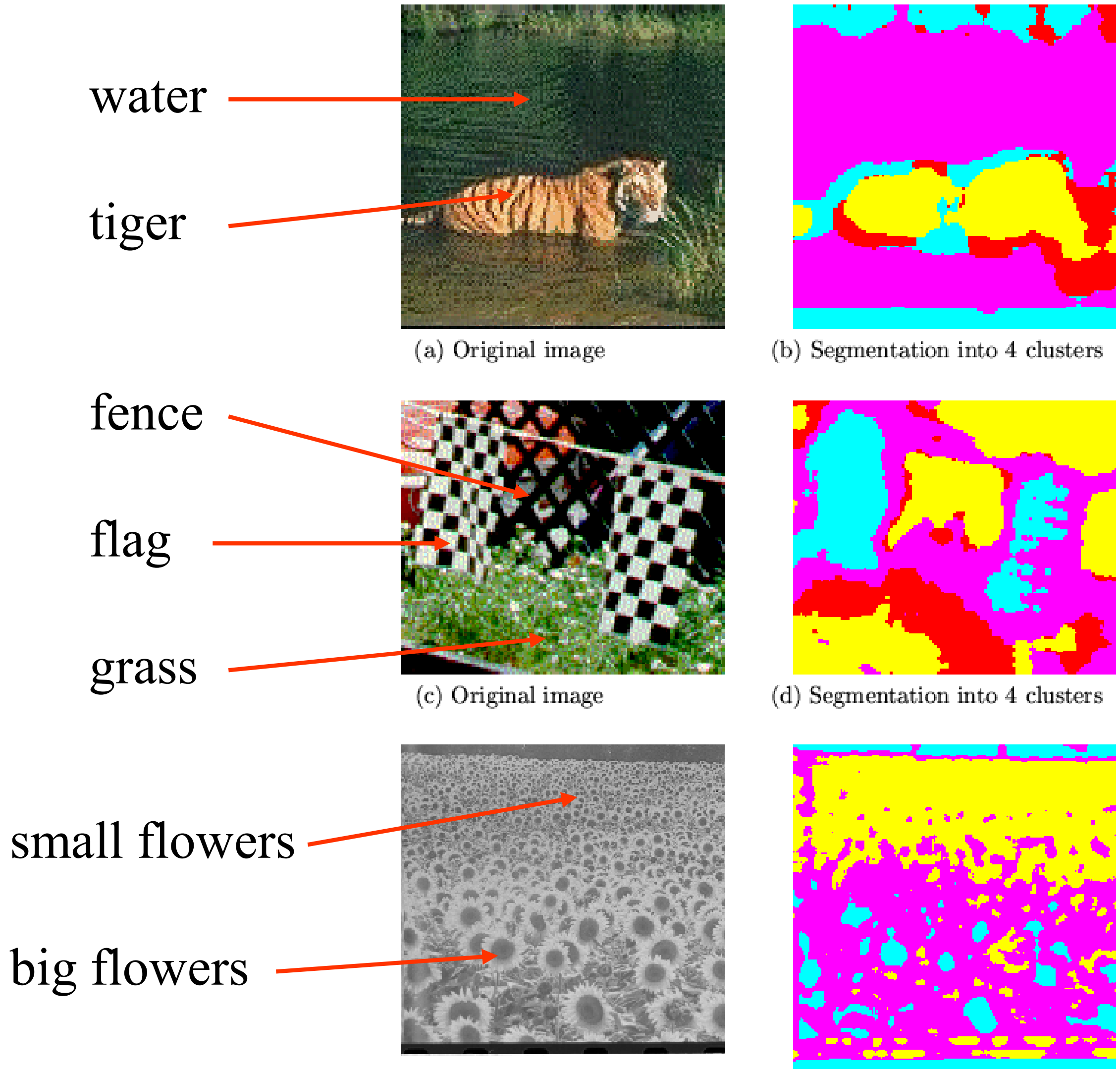
- Subtract mean neighborhood intensity from pixel
- Dot product 16 5x5 masks with neighborhood
- 9 features defined as follows:

L5E5/E5L5	L5S5/S5L5
L5R5/R5L5	E5E5
E5S5/S5E5	E5R5/R5E5
S5S5	S5R5/R5S5
R5R5	

Features from sample images

Table 7.2: Laws texture energy measures for major regions of the images of Figure 7.8.

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5



(a) Original image (b) Segmentation into 4 clusters

(c) Original image (d) Segmentation into 4 clusters

(e) Original image (f) Segmentation into 3 clusters

Is there a neighborhood size problem with Laws?

Autocorrelation function

- Autocorrelation function can detect repetitive patterns of texels
- Also defines fineness/coarseness of the texture
- Compare the dot product (energy) of non shifted image with a shifted image

$$\rho(dr, dc) = \frac{\sum_{r=0}^{N} \sum_{c=0}^{N} I[r,c]I(r+dr,c+dc]}{\sum_{r=0}^{N} \sum_{c=0}^{N} I^2[r,c]} = \frac{I[r,c] \circ I_d[r,c]}{I[r,c] \circ I[r,c]}$$

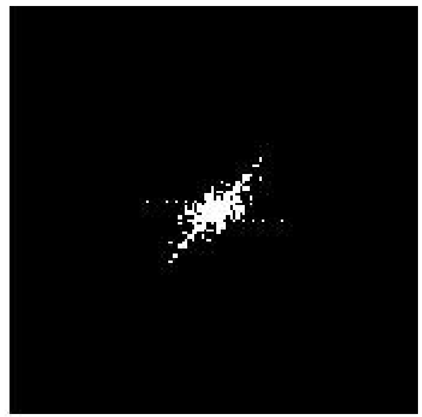
Interpreting autocorrelation

- 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

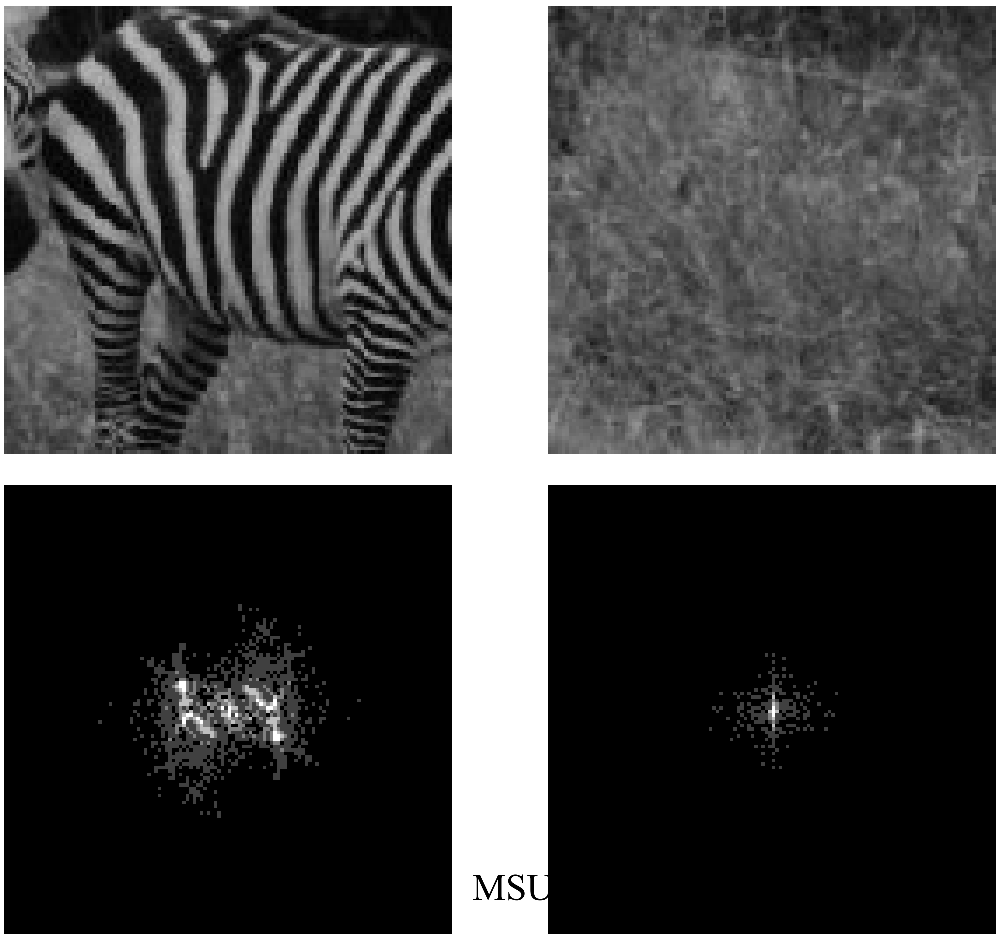
Fourier example

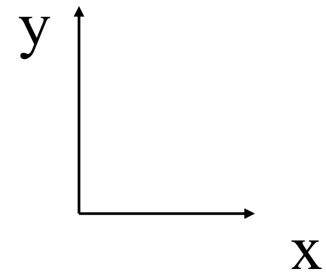


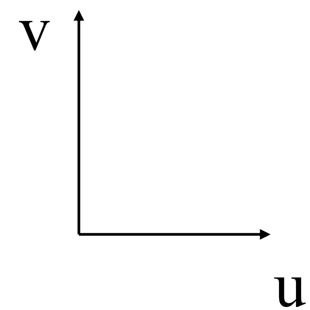
Notes on Texture by FFT

The power spectrum computed from the Fourier Transform reveals which waves represent the image energy.

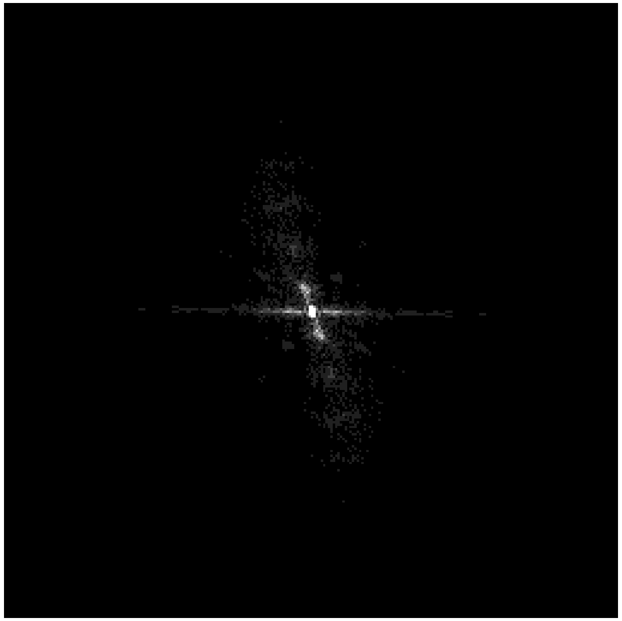
Stripes of the zebra create high energy waves generally along the u-axis; grass pattern is fairly random causing scattered low frequency energy





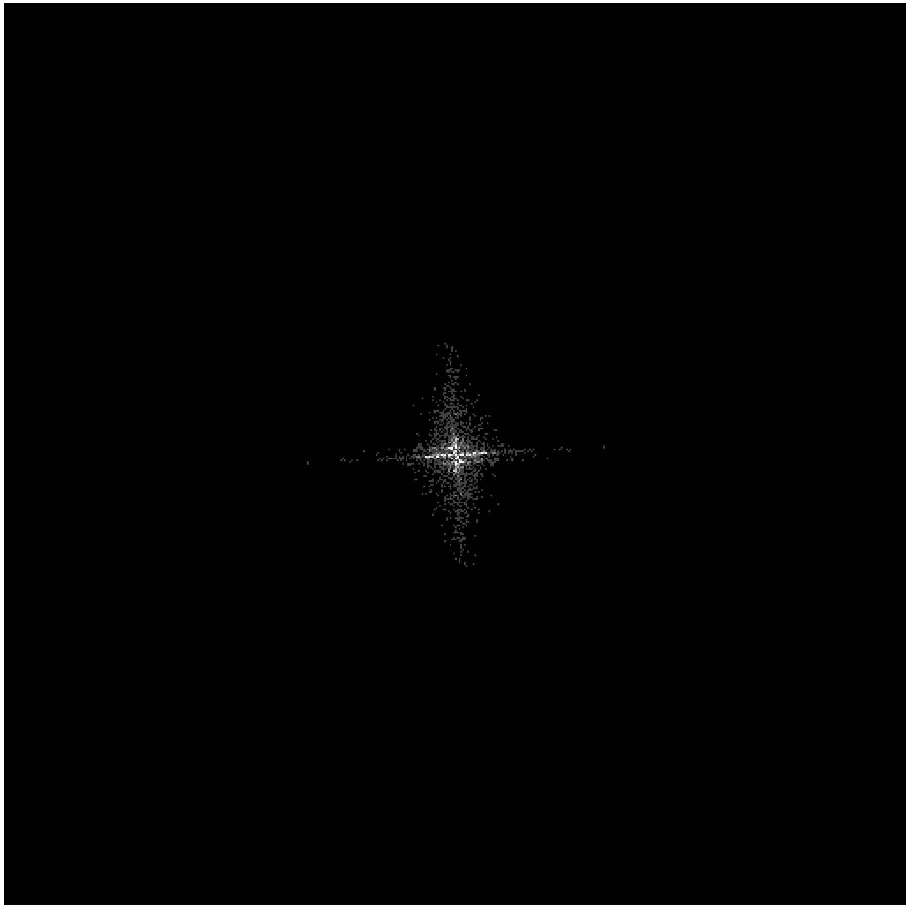


More stripes

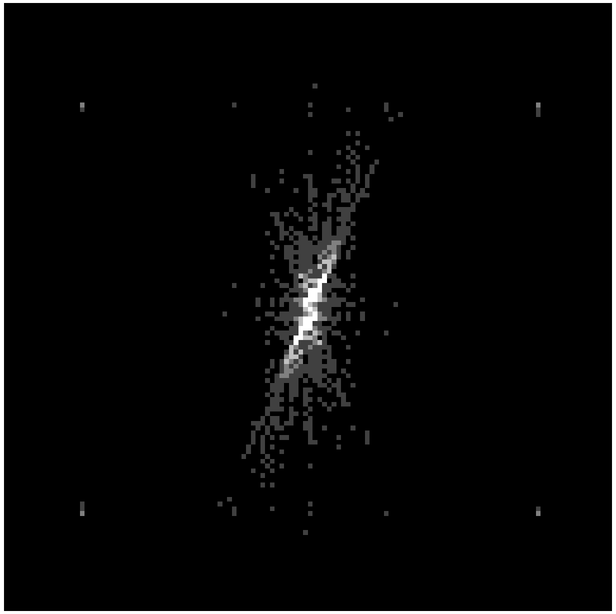


Power spectrum x 64

Spectrum shows broad energy along u axis and less along the v-axis: the roads give more structure vertically and so does the regularity of the houses



Spartan stadium: the pattern of the seats is evident in the power spectrum – lots of energy in (u,v) along the direction of the seats.



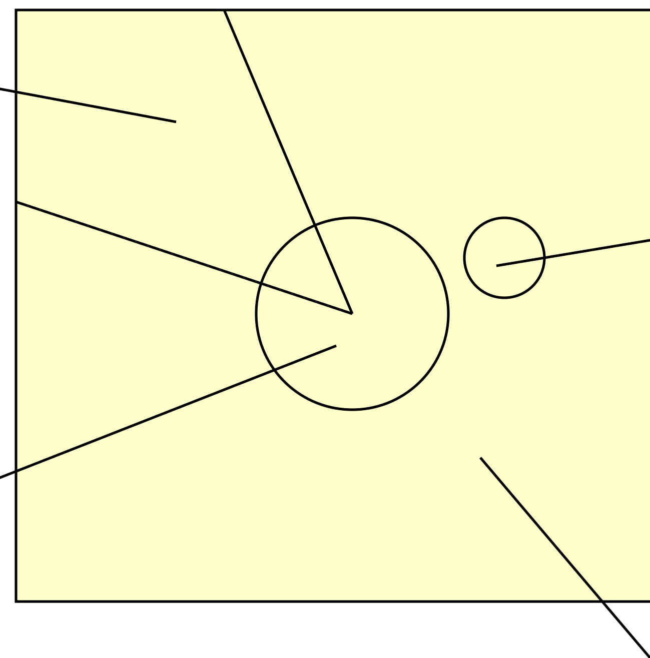
Getting features from the spectrum

- FT can be applied to square image regions to extract texture features
- set conditions on u-v transform image to compute features: f1 = sum of all pixels where R1 < || (u,v)|| < R2 (bandpass)
- f2 = sum of pixels (u,v) where u1 < u <u2
- f3 = sum of pixels

  where ||(u,v)-(u0,v0)|| < R

Filtering or feature extraction using special regions of u-v

F4 is all energy in directional wedge



F1 is all energy in small circle

F2 is all energy near origin (low pass)

F3 is all energy outside circle (high pass)