

→ textures, shapes → region of int'l.

---

# Mid Level Image Features

## : Textures

Basic Level : pixel val.

→ Color, Edge.

---

Eun Yi Kim

---



Artificial Intelligence  
& Computer Vision  
Laboratory

---

# I N D E X

---

Texture

Structural Approaches

Statistical Approaches

- Simple features
- LBPs and GLCMs
- Edge based features
- Raw' energy

Interpreting Textures

---



Artificial Intelligence  
& Computer Vision  
Laboratory

# Texture



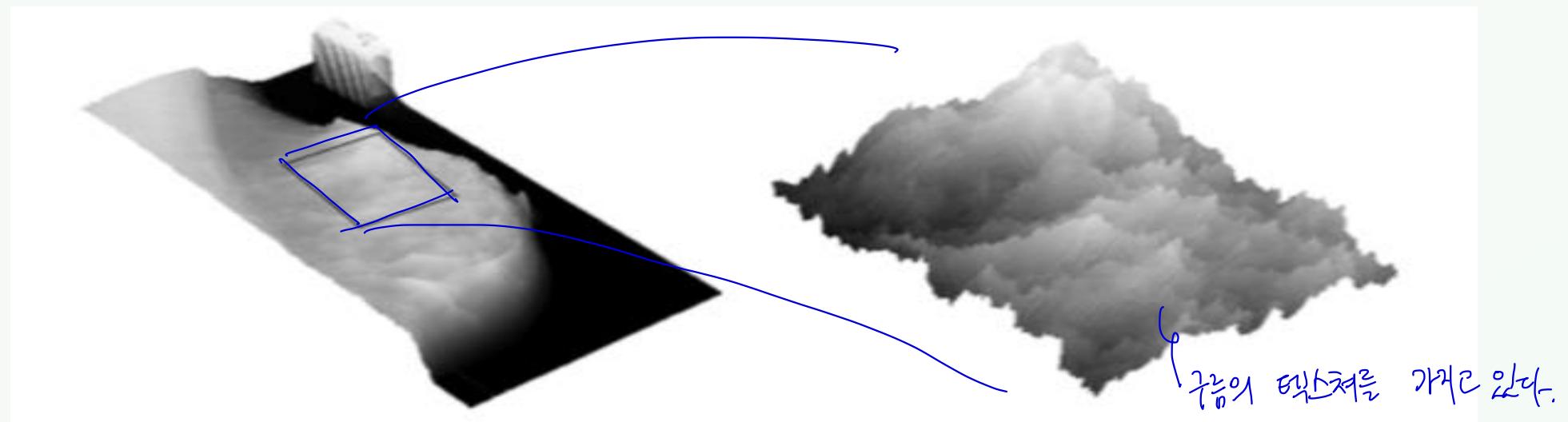
- Texture is a feature used to partition images into regions of interest and /to classify those regions
- Texture provides information in the "spatial arrangement of colors" or intensities in an image.
- Texture is characterized by the spatial distribution of intensity levels in a neighborhood.

# Texture



Artificial Intelligence  
& Computer Vision  
Laboratory

- Texture is a "repeating pattern" of local variations in image intensity
  - Texture cannot be defined for a point



# Texture



- For example, an image has a 50% black and 50% white distribution of pixels

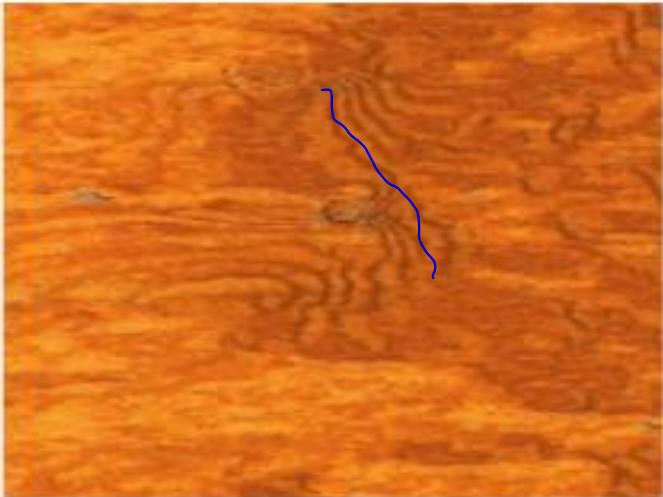


- Three different images with the same intensity distribution, but with different textures

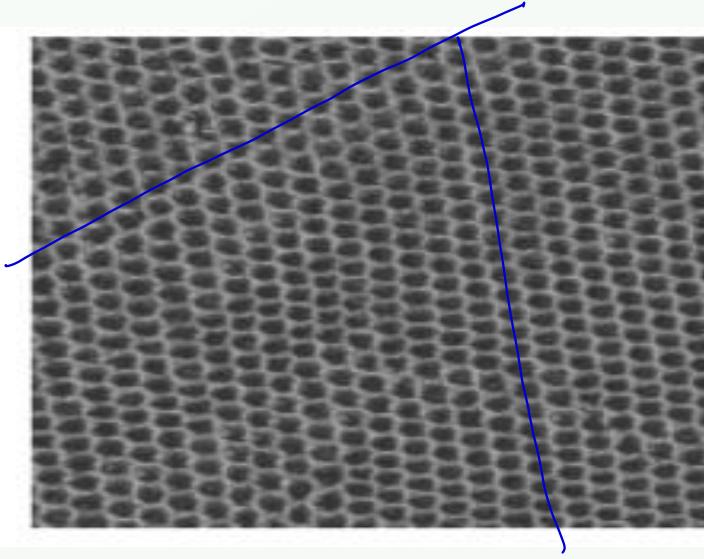
# Texture



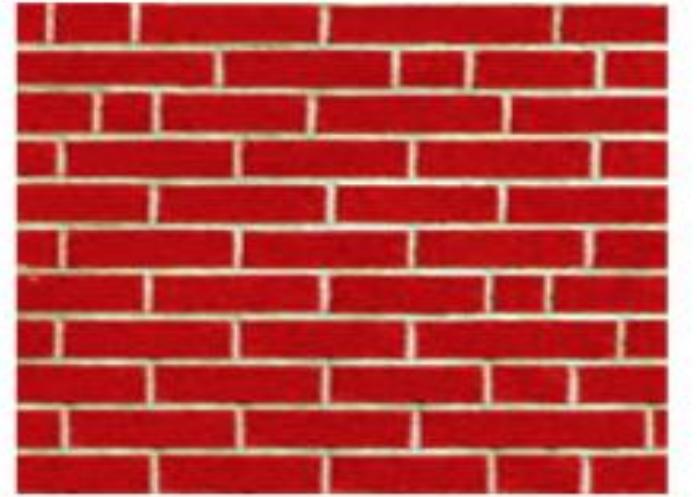
Artificial Intelligence  
& Computer Vision  
Laboratory



{  
natural texture.  
object class.



{  
grid texture.  
object class.



# Understanding Texture



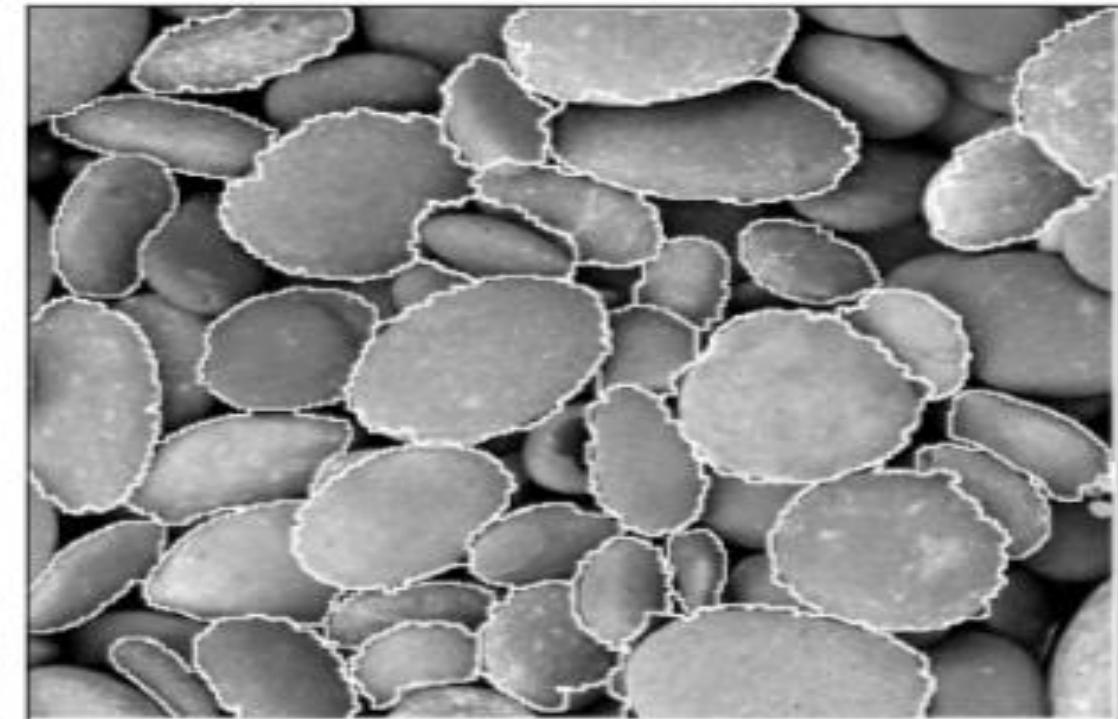
- There are three approaches to defining exactly what texture is:
  1. **Structural** : texture is a set of primitive **texels** in some regular or repeated relationship.
  2. **Statistical** : texture is a quantitative measure of the arrangement of intensities in a regions. This set of measurements is called a **feature vector**.
  3. **Modeling** : texture modeling techniques involve constructing models to specify textures

???:  
\* \* \* \* \*

# Structural approach to describing texture



Artificial Intelligence  
& Computer Vision  
Laboratory

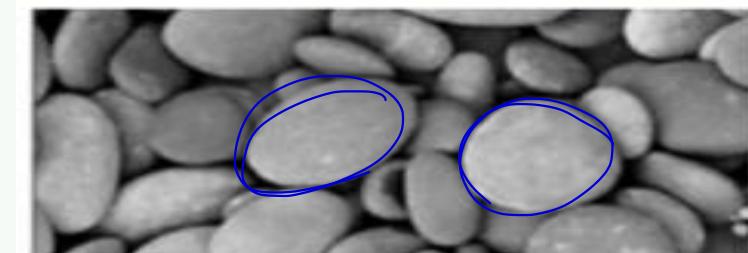
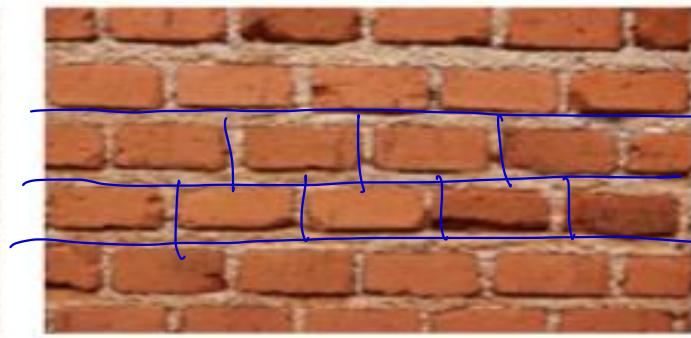


<http://vision.ai.uiuc.edu/~sintod/>



# Aspects of texture

- Size/Granularity
  - Sand versus pebbles versus boulders
- Directionality/Orientation
- Random or regular
  - Stucco versus bricks



Artificial Intelligence  
& Computer Vision  
Laboratory

# Problem with Structural Approach



Artificial Intelligence  
& Computer Vision  
Laboratory



**What/Where are the texels?**

texel은 투영하기 어렵다.

Extracting texels in real images may be difficult or impossible



# Statistical Approach to Texture



Artificial Intelligence  
& Computer Vision  
Laboratory

- Characterize texture using statistical measures computed from gray-scale intensities (or colors) alone
- Less intuitive, but applicable to all images and computationally efficient
- Can be used for both classification of a given input texture and segmentation of an image into different texture regions

# Texture Analysis



- There are two primary issues in texture analysis:
  - Texture classification → 영상 내의 다양한 textures를 정식으로 구분해
  - Texture segmentation → 이미지 내의 영상 단위를 잘 구분하고 이를 boundary를 찾는다
- **Texture segmentation** is concerned with automatically determining the boundaries between various texture regions in an image
- **Texture classification** is concerned with identifying a given textured regions from a given set of **texture classes**
  - Each of these regions has unique texture characteristics
  - Statistical methods are extensively used



# Simple Statistical Texture Measure

- ① Range
- ② Variance

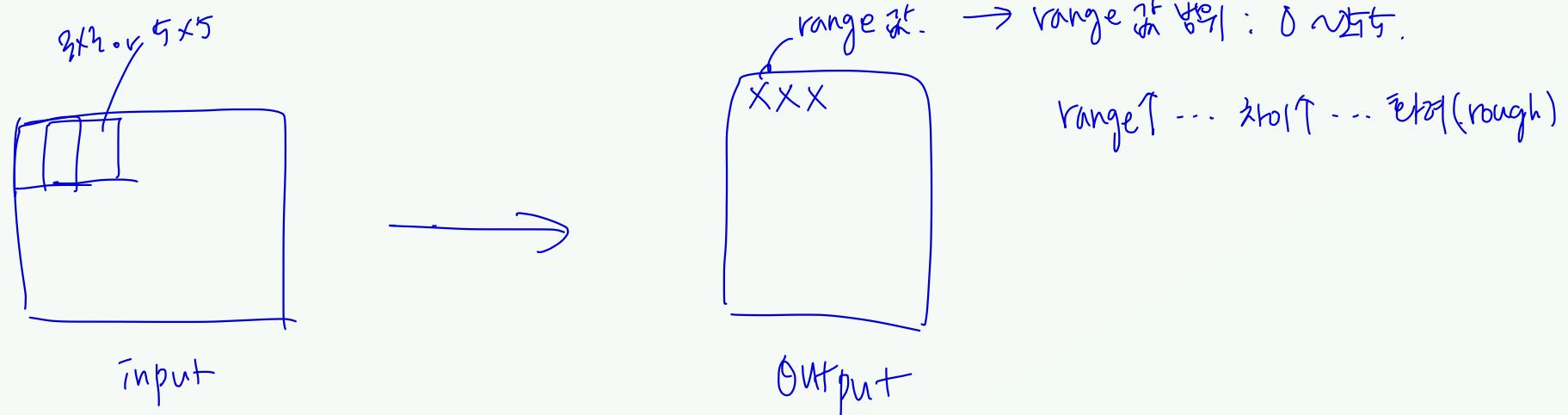
# Range

→ 3x3 or 5x5 size neighbor window.  $\Rightarrow$  range =  
 pixel's intensity  $\max - \min = \text{range}$ ,  
 $\Rightarrow$  higher range  $\Rightarrow$  output value.



One of the simplest of the texture operator is the **range** or difference between maximum and minimum intensity values in a neighbor

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



2

# Variance

~~(range, variance, ...)~~  
 : range, variance, ...

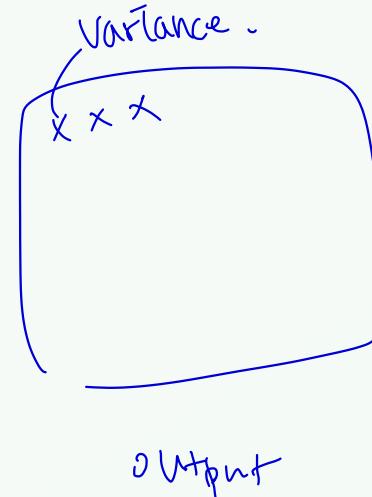
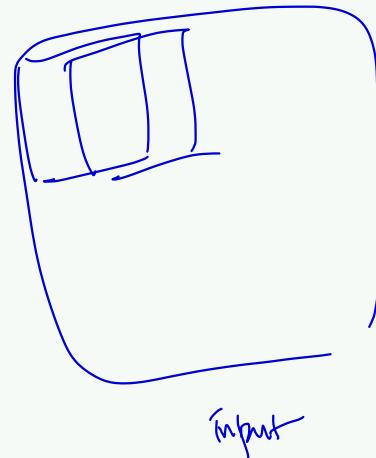
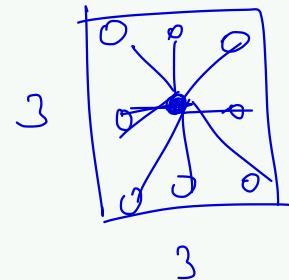


Artificial Intelligence  
 & Computer Vision  
 Laboratory

Another estimator of texture is the **variance** in neighborhood regions

- This is the sum of the squares of the differences between the intensity of the central pixel and its neighbors

$\sum_{i=1}^n (x_i - \bar{x})^2$



# Examples: Range and Variance



Artificial Intelligence  
& Computer Vision  
L a b o r a t o r y



# Quantitative Texture Measures

↳ \* Colorning theory



- Numeric quantities or statistics that describe a texture can be calculated from the intensities (or colors) themselves
1. Local Binary Pattern
  1. Grey Level Co-occurrence

# Local Binary Pattern 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

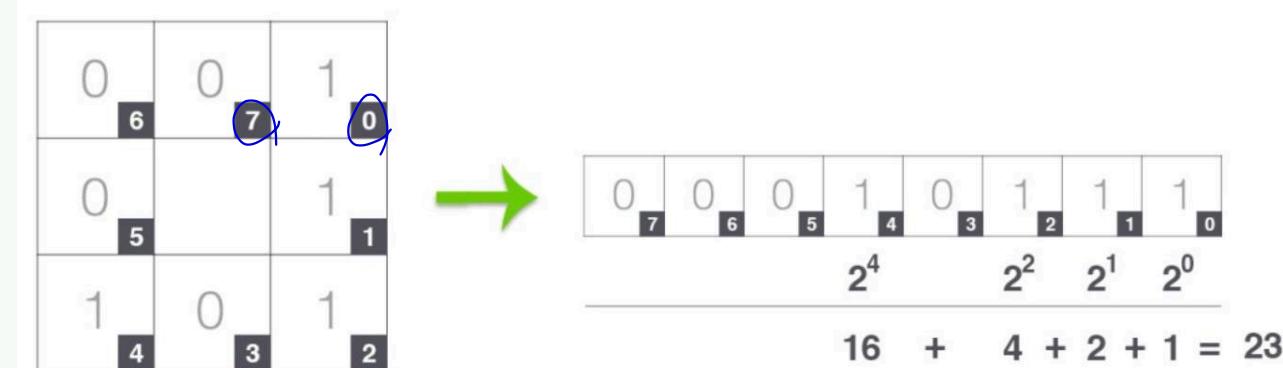
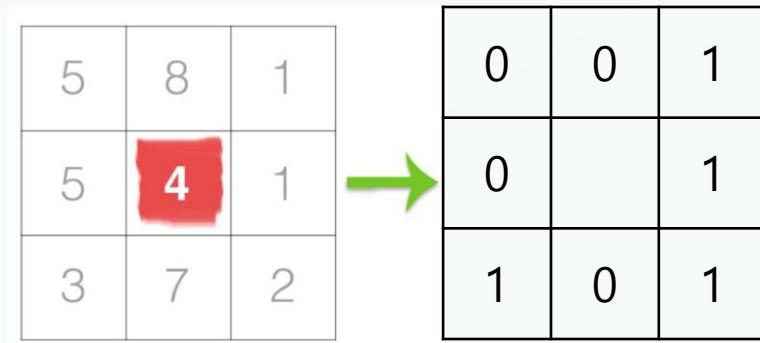
	1	2	3	
8	100	101	103	
	40	50	80	
	50	60	90	

$b_1 \text{ int } 3 \text{rd } 1.$   
 $\longrightarrow$   
 Center int 3rd 0.  
1 1 1 1 1 1 0 0  
 → 8bit uint 11111100,  $255 - 3 = 252$ .



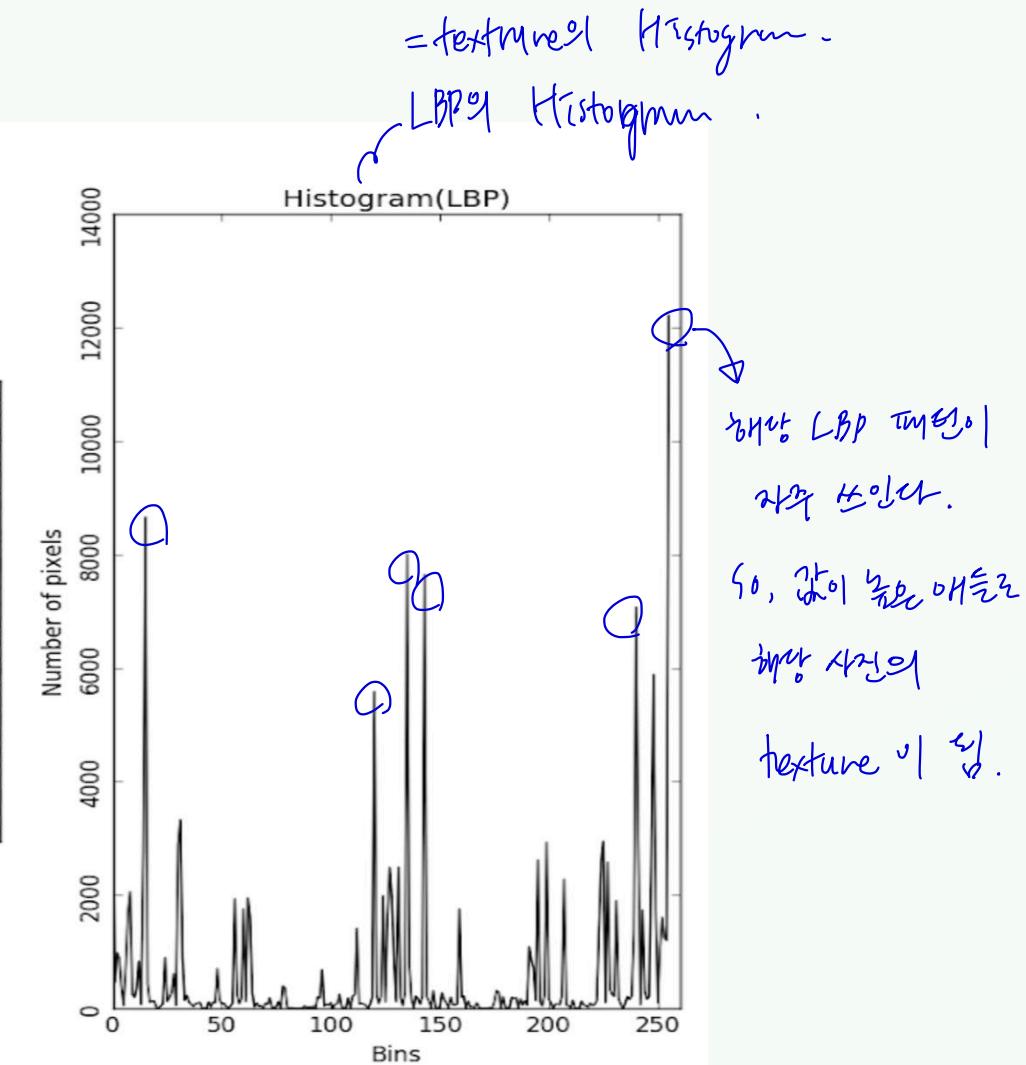
# Examples: LBPs

- For each pixel  $p$ , create an 8-bit number



# Examples: LBPs

- Image description with local binary pattern





# Example: LBPs

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**

The screenshot shows a user interface for the Fids demo. At the top, it says "Fids demo". Below that is a grid of six images. The first image in the top-left corner is highlighted with a red border. Below the grid, there are navigation buttons: a left arrow, "Random", "Go", "ZoomIn", and a right arrow. To the right of these buttons, it says "Found 191 matches. Displaying 1 - 6". Underneath the images, there are two rows of search parameters. The first row includes "distance measures" and "loose ... strict" buttons. The second row lists several measures with checkboxes: "ColorHistL1 4x4x4" (unchecked), "ColorHist8x8x8" (checked), "SobelEdgeHist" (unchecked), "LBPHist" (checked), "fleshiness" (unchecked), and "Wavelets" (unchecked). To the right of these checkboxes are five horizontal sliders, each with a value of 5. On the far right, there is a legend with three options: "And" (selected), "Or", and "Sum". At the bottom, it says "Server Connected".

# Example: LBPs

Low-level measures don't always find semantically similar images.

**Fids demo**

The screenshot shows a search interface for image similarity. At the top, the title "Fids demo" is displayed in red. Below it is a grid of six images. The first image in the top row is highlighted with a red border, indicating it is the query image. The other five images are search results. To the right of the grid are two buttons: "Put In Cart" and "Check Out". Below the grid, there are navigation buttons: a left arrow, a "Random" button, a "Go" button (which is currently selected), a "ZoomIn" button, and a right arrow. The text "Found 119 matches. Displaying 1 - 6" is shown. Underneath these controls is a section for "distance measures" with sliders for "loose" and "strict" settings. The measures listed are: ColorHistL14x4x4, ColorHist8x8x8, SobelEdgeHist, LBPHist (which is checked), fleshiness, and Wavelets. To the right of the sliders is a legend for "And", "Or", and "Sum" operations. A callout box on the right side provides instructions: "A double click on an image means:" followed by options "Set query / Go" (selected) and "Zoom in". At the bottom of the interface, the status "Server Connected" is shown.

Put In Cart

Check Out

Random Go ZoomIn Found 119 matches. Displaying 1 - 6

distance measures loose ... strict

- ColorHistL14x4x4
- ColorHist8x8x8
- SobelEdgeHist
- LBPHist
- fleshiness
- Wavelets

And

Or

Sum

A double click on an image means:

- Set query / Go
- Zoom in

Server Connected

# Extension and Applications

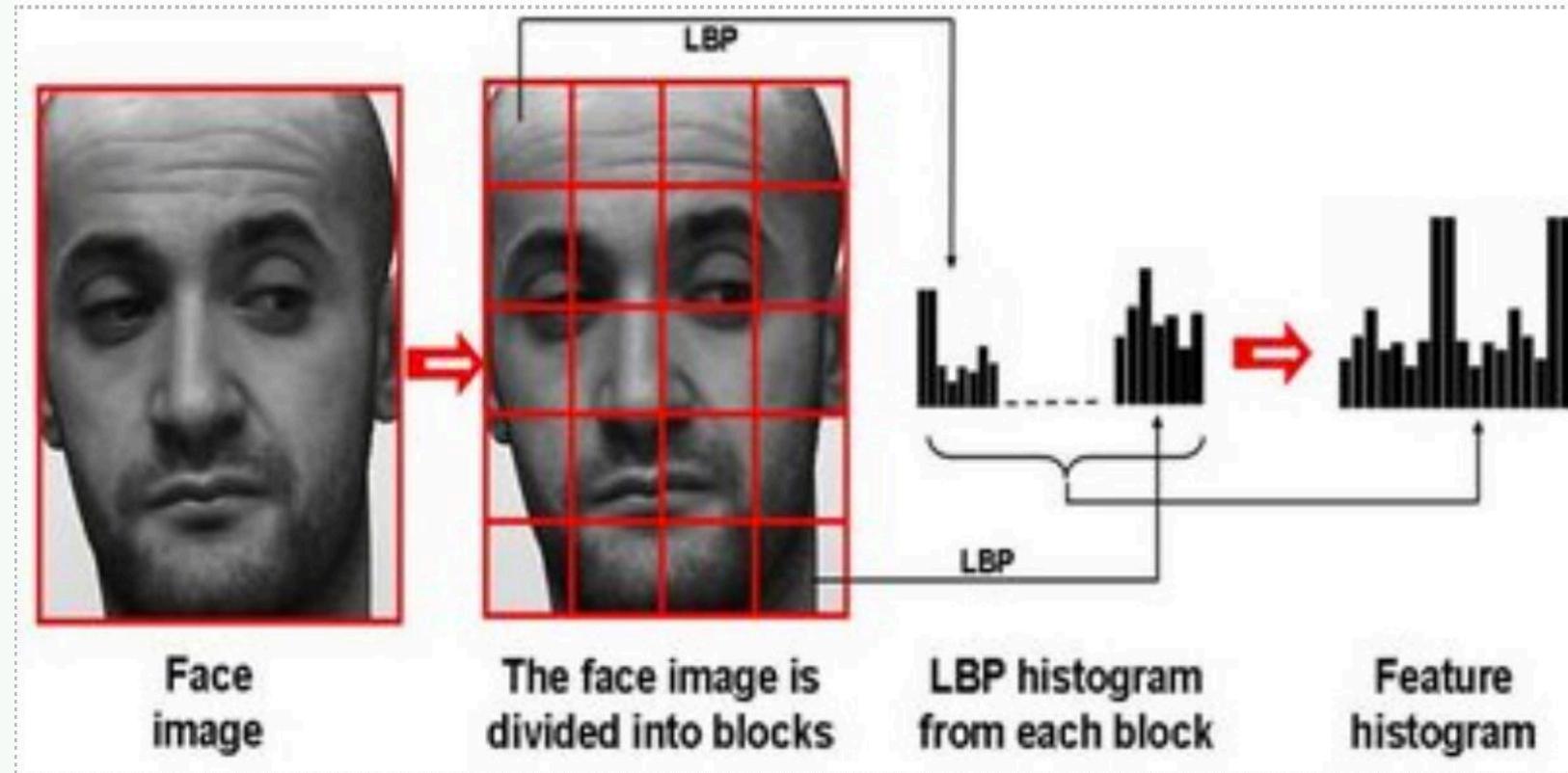


- Due to its discriminative power and computational simplicity, it is widely used all over the world both in research and application
- To increase the applicability of LBP, various extensions and modifications have been proposed
  - → Paper reading

# Applications: LBPs



- Face description with local binary patterns \*

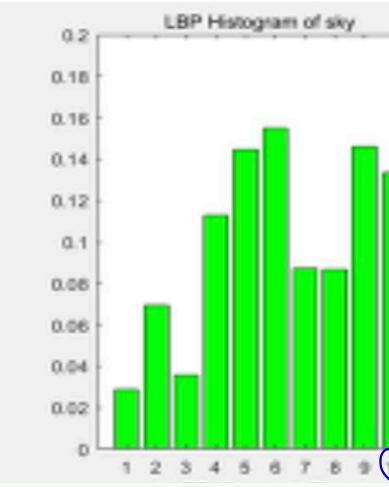
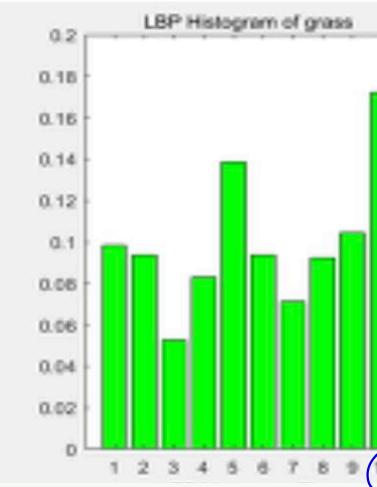
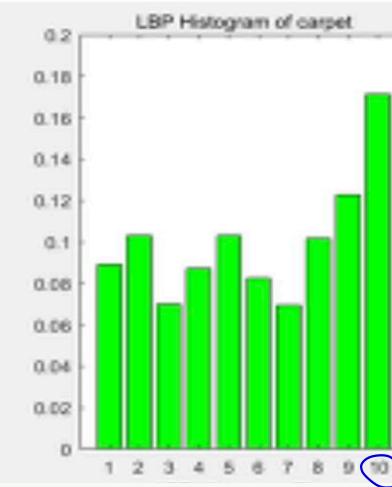
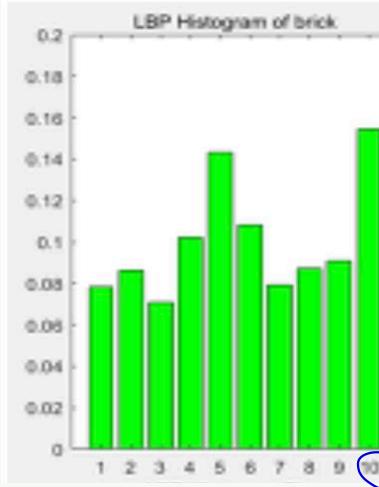


\* T. Ahonen, A. Hadid, and M. Pietikinen, "Face description with local binary patterns: Application to face recognition," PAMI 2006

# Applications: LBPs



- Texture classification with uniform & rotation-invariant LBP\*



\* Ojala T, Pietikäinen M, Harwood D. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns [M]// Computer Vision - ECCV 2000. Springer Berlin Heidelberg, 2000:404-420.

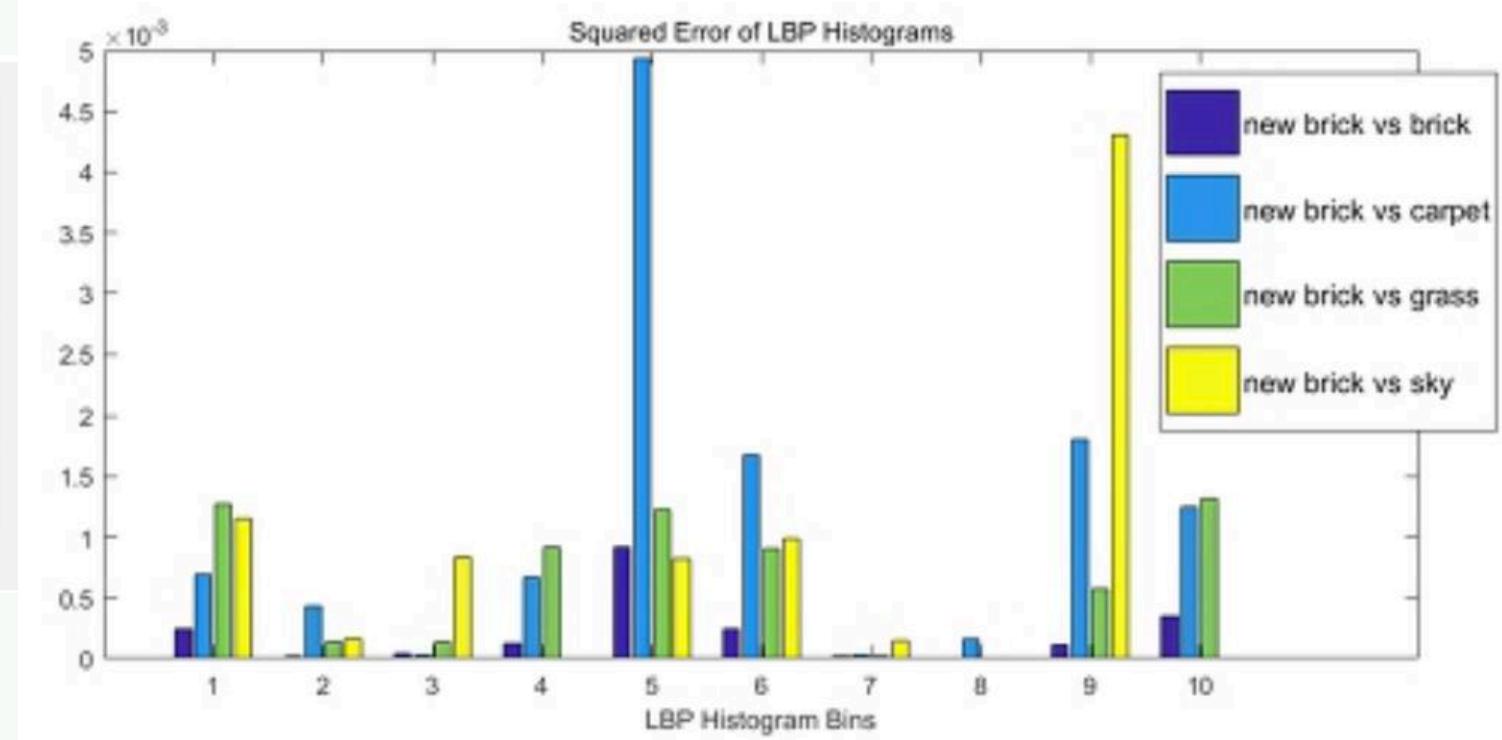
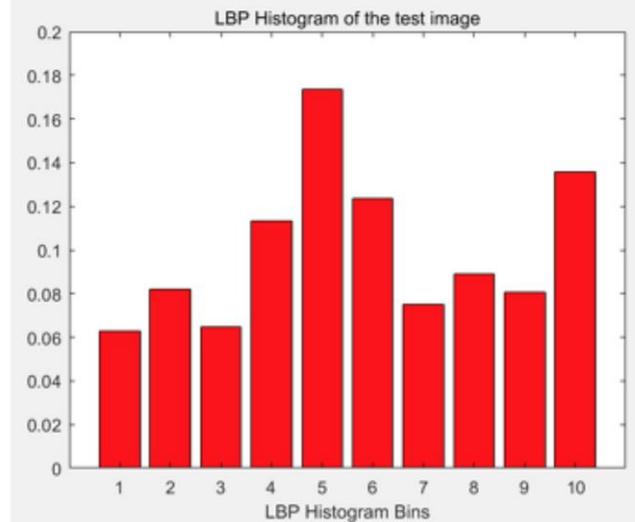
# Applications: LBPs



Artificial Intelligence  
& Computer Vision  
Laboratory



Test image



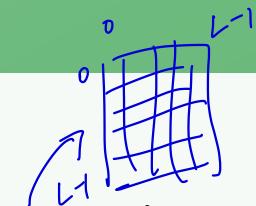
# Gray Level Co-occurrence



- 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.

gray level of texture + 월(월)정리

: 모양의 월(월)정리 틀에 gray level들이 모여 월(월)정리의 수를 matrix 형태로 정리.



- A **co-occurrence matrix** is a two-dimensional array,  $\mathbf{P}$ , 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:  $P_d[i,j] = n_{ij}$ 
    - $n_{ij}$  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

# Example: GLCM



- For example, if  $d = (1,1)$  예를 들어  $X$ .

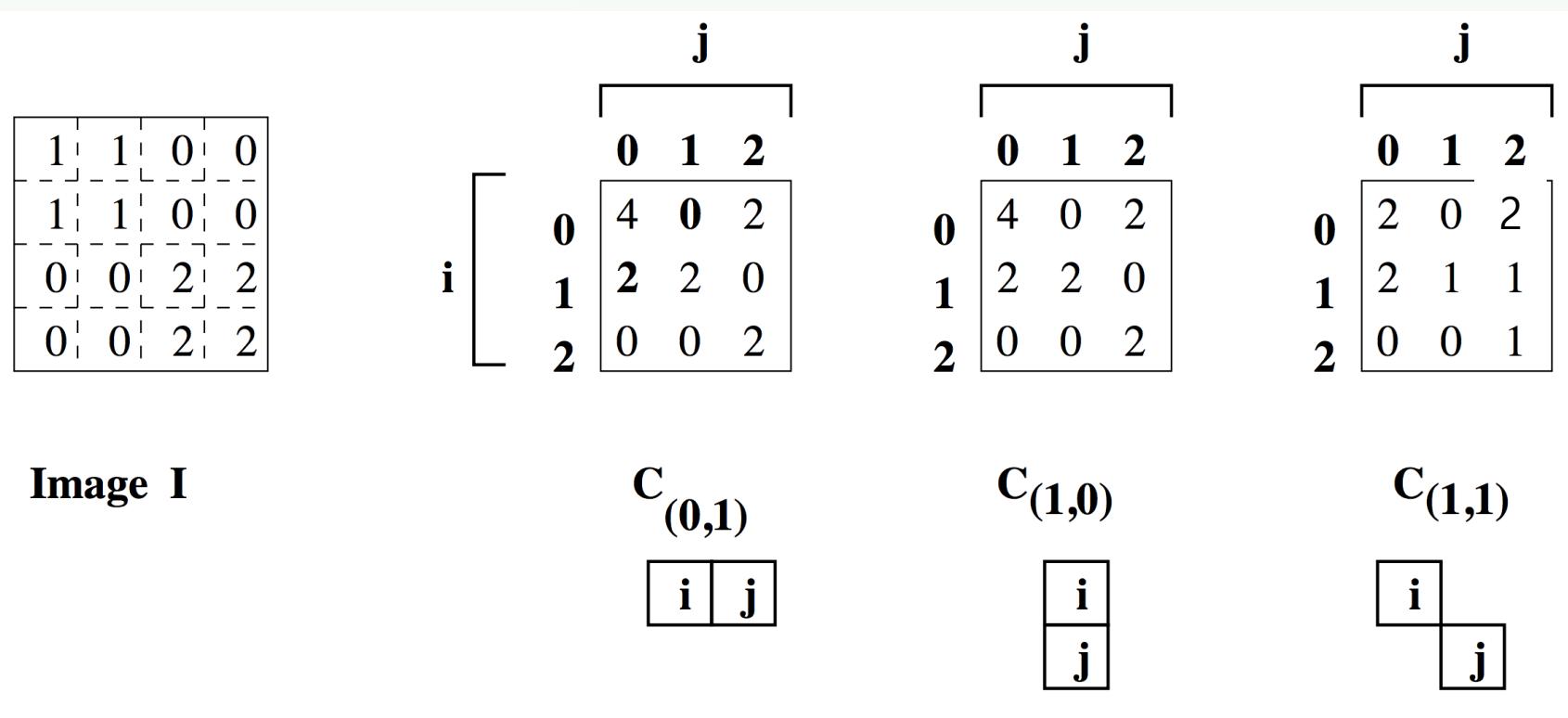
2	1	2	0	1
0	2	1	1	2
0	1	2	2	0
1	2	2	0	1
2	0	1	0	1

## input

$$\begin{array}{c} i \\ \hline j \end{array}$$

$$P_d = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 3 & 2 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \end{matrix}$$

- There are 16 pairs of pixels in the image which satisfy this spatial separation
  - Since there are only three gray levels,  $P_d[i,j]$  is  $3 \times 3$  matrix





### 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  $i^{\text{th}}$  row and  $j^{\text{th}}$  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

가설의 증명 방법.



Artificial Intelligence  
& Computer Vision  
Laboratory

- The elements of  $P_d[i,j]$  can be normalized by dividing each entry by the total number of pixel pairs
- Normalized GLCM,  $N[i,j]$  is defined by:

$$N[i,j] = \frac{P[i,j]}{\sum_i \sum_j P[i,j]} \Rightarrow \text{2x2x3 probability.}$$

- It normalizes the co-occurrence values to lie between 0 and 1, and allows them to be thought of as probabilities



# Numeric Features of GLCM

↳ open image window 보기

각각의 feature는 GLCM의 대각선.

- 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

$L \times L$ 은 정복 일정이 아니므로 쉽게 침습.  
기준화하기

- Numeric features are computed from the occurrence matrix that can be used to represent the texture more compactly

- Maximum probability
- Moments
- Contrast
- Homogeneity
- Entropy
- Correlation

# Maximum Probability

↳ 1번 뒤에 2번 째 번째 원소의 element, 3 번째 원소 .



- 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

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

자기에게 가장  
가까운 것.  
이제 확률적인 전략.

$C_m$  ... 확률적인 행동을 가하는 행동이다.

# Moments

$\Delta \beta_{24}$



- The **order k element difference moment** can be defined as:

$$MOM_k = \sum_i \sum_j (i - j)^k P_d[i, j] \quad \text{if } \dots \text{ short.}$$

- 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**

$$MOM_k = \sum_i \sum_j \frac{P_d[i, j]}{(i - j)^k}, \quad i \neq j$$

Moment
$0^{\text{th}}(M_0)$ : size
$1^{\text{st}}(M_1)$ : $E\bar{P}$
$2^{\text{nd}}(M_2)$ : variance
= contrast

# ③ Contrast

$\hookrightarrow k=2 \text{ or } M_2$



- 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,  $k=2$  and  $n=1$

$$\begin{aligned} \overline{i} \gg \bar{j} \text{ or } \bar{i} \ll \bar{j} &\Rightarrow C(k,n) \text{ is high} \\ \bar{i} = \bar{j} \text{ or } \bar{i} \approx \bar{j} &\Rightarrow C(k,n) \text{ is low.} \end{aligned}$$

# Homogeneity

유사도  
유사도



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

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

$C_h$  T ... 유사한 이미지 T.

- 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

여기까지 듣기 49분.

# Entropy

$= \text{info}$   $= \text{randomness}$ .



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

$$C_e = - \sum_{i} \sum_{j} P_d[i, j] \ln P_d[i, j]$$

- Entropy is highest when all entries in  $P[i, j]$  are of similar magnitude, and small when the entries in  $P[i, j]$  are unequal

자주 사용하는 용어 Entropy T.

# Correlation

- Correlation is a measure of image linearity

$$C_e = \frac{\sum_{i} \sum_{j} ij P_d[i, j] - \mu_i \mu_j}{\sigma_i \sigma_j}$$

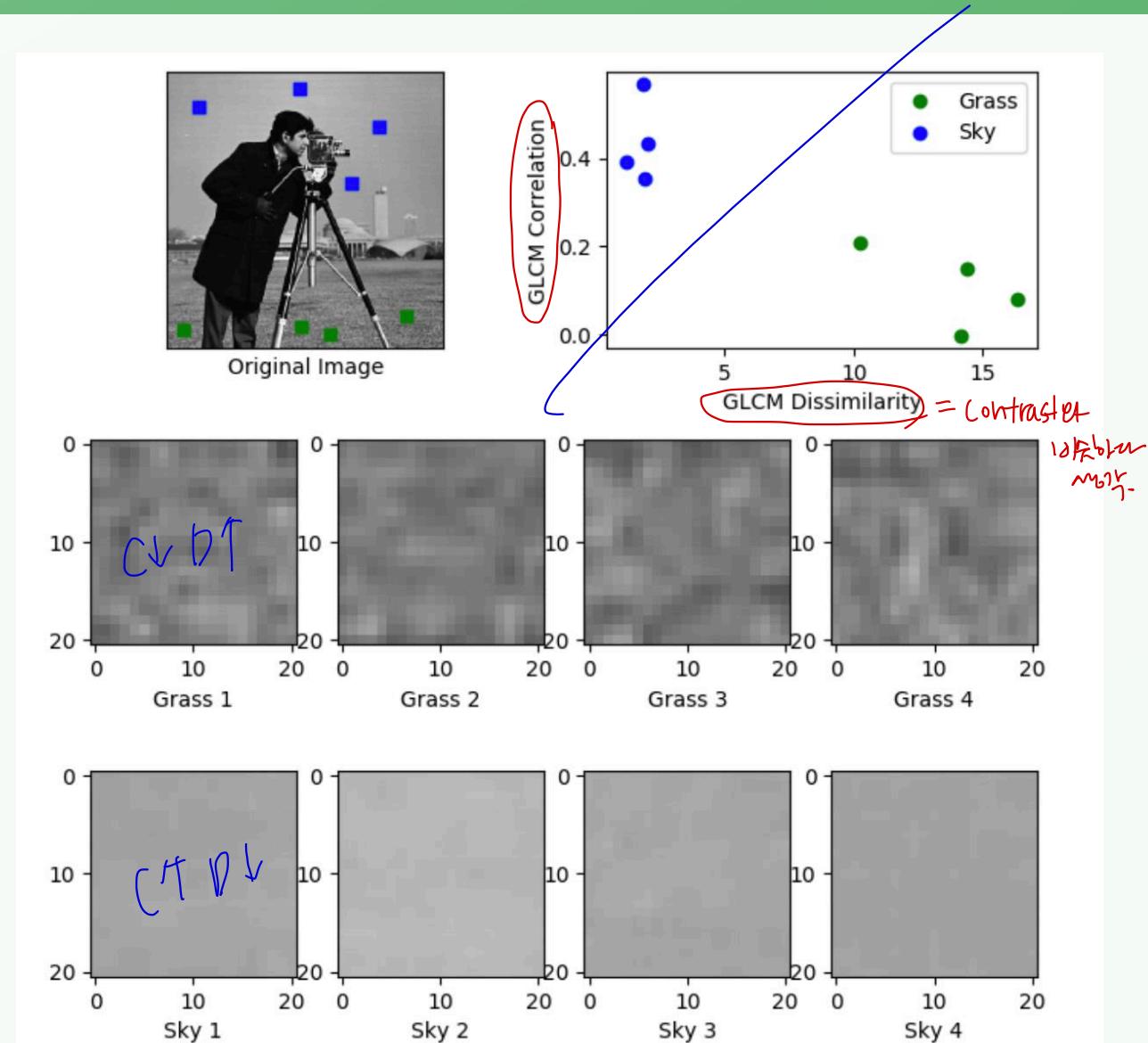
$$\mu_i = \sum i P_d[i, j], \quad \sigma_i^2 = \sum i^2 P_d[i, j] - \mu_i^2$$

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



# Examples: GLCM

- Texture classification with GLCM numeric feature





# Problems with GLCM

- One problem with deriving texture measures from co-occurrence matrices is how to choose the displacement vector  $\mathbf{d}$  → 이제 각 상관계 구조를
- 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  $\mathbf{d}$  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  $\mathbf{d}$  that have the most structure, i.e., to maximize the value

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

(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

region or pixel  
matrix size  $m \times m$ .



# Edges and Texture

# Edges and Texture

દ્વારા નિરૂપિત



Artificial Intelligence  
& Computer Vision  
Laboratory

- 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 characterize the texture
- Compute the co-occurrence matrix of an edge-enhanced image



# Edges and Texture

- **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 for a region R

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

- N is the size of the unit area  
 $= \text{no. pixels in unit}$

1011 1111 Fedgeness.  
0 " "  
" "

## 2. Histograms of edge magnitude and direction for a region R

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

- These are the normalized histograms of gradient magnitudes and gradient directions, respectively

▷ Mag. Histogram of gradient magnitudes



# Example

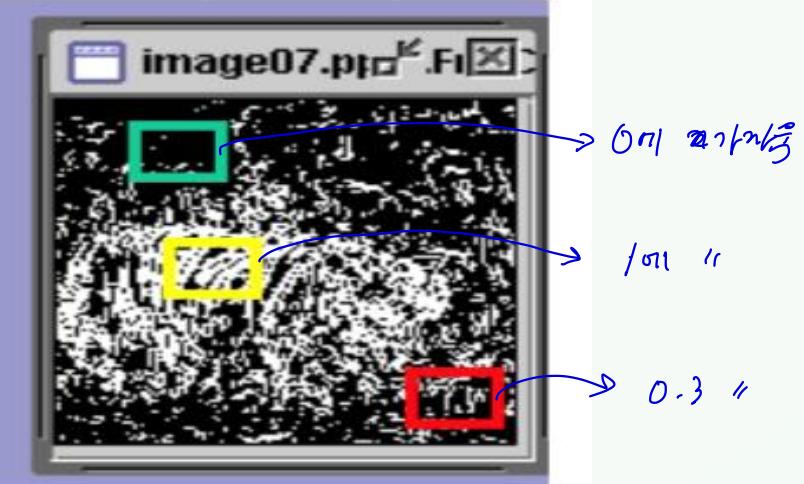
Original Image



Edge Image

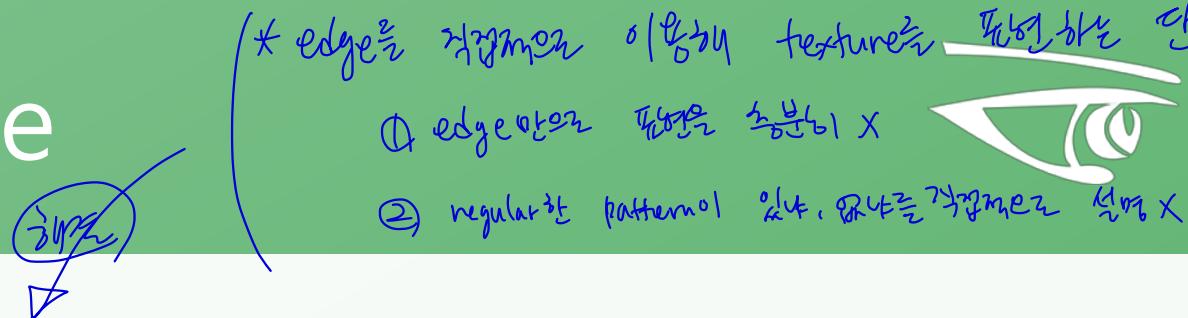


Thresholded  
Edge Image



Different  $F_{\text{edgeness}}$  for different regions

# Energy and Texture



- One approach to generate texture features is to use local kernels to detect various types of texture
- Laws<sup>1)</sup> developed a texture-energy approach that measures the amount of variation within a fixed size window
- 1) Laws, K. I. "Rapid texture identification". in SPIE Image Processing for Missile Guidance, pp. 370-380, 1980.



# Law's Texture Energy

texture w/ detecting 8x8 kernel  
local (kernel)

- 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 (부수적)

부수적 = 각각의 각각의 8x8 .

(      or      ) .

# Law's Texture Mask



- A set of convolution mask are used to compute texture energy
- The mask are computed from the following basic mask

kernel.

→ gaussian smoothing filter .

- L5 (Gaussian) gives a center-weighted local average
- E5 (gradient) responds to row or col step edges
- S5 (LoG) detectss spots
- R5 (Gabor) detects ripples
- W5(wave) detects waves

δ<sub>y</sub>, δ<sub>x</sub> Highfreq. κ(x).

1 D kernel.

$$L5 = [1 \ 4 \ 6 \ 4 \ 1]$$

$$E5 = [-1 \ -2 \ 0 \ 2 \ 1] \rightarrow \delta f = 0 .$$

$$S5 = [-1 \ 0 \ 2 \ 0 \ -1] \rightarrow \delta f = 0 .$$

$$R5 = [1 \ -4 \ 6 \ -4 \ 1]$$

$$W5 = [-1 \ 2 \ 0 \ -2 \ +1]$$



# Law's Texture Mask (2D)

- The 2D convolution mask are obtained by computing the outer product of a pair of vectors  
*the outer product of two vectors.*
- For example,  $E5L5$  is computed as the product of  $E5$  and  $L5$  as follows

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

*5 × 5*

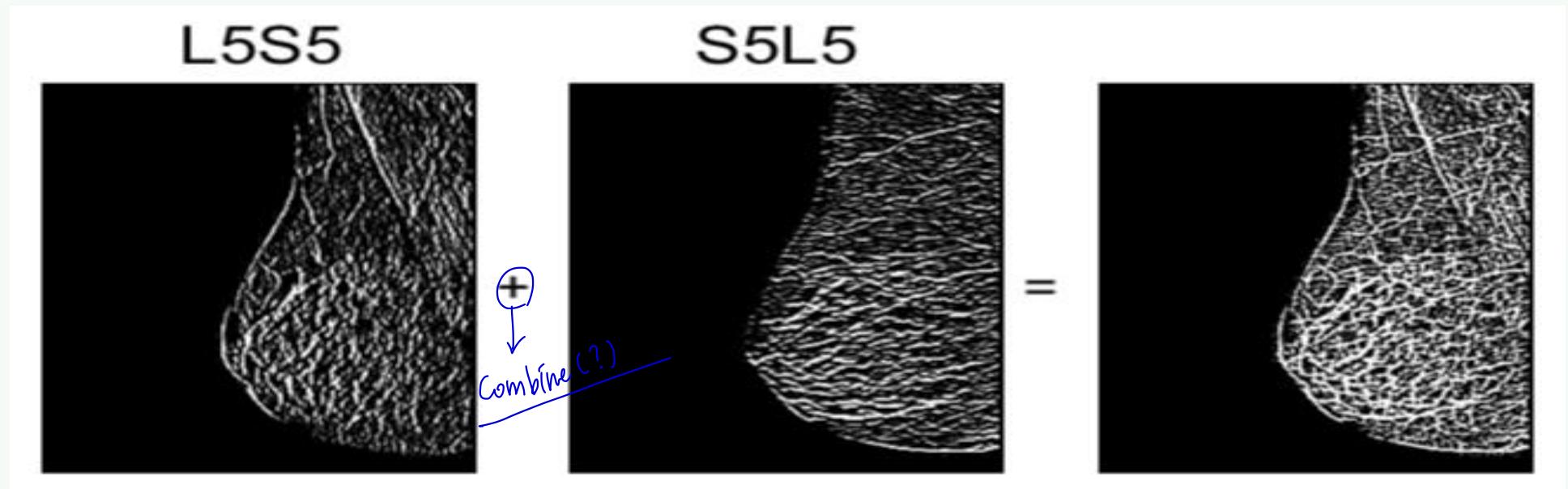
- This results in 25  $5 \times 5$  masks, 24 of the masks are zero-sum, the  $L5L5$  is not

*$L5L5$  whose sum is not 0.*

# Law's Texture Mask (2D)



- Bias from the "directionality" of textures can be removed by combining symmetric pairs of features, making them rotationally invariant
  - For example,  $S5L5 \text{ (H)} + L5S5 \text{ (V)} = L5S5R$



# Law's Texture Energy

Computing of  
Texture Energy



Artificial Intelligence  
& Computer Vision  
Laboratory

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

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

$\uparrow$  각 filter의 합계

- If  $k$  masks are applied, the result is an  $k$ -dimensional feature vector at each pixel of the image being analyzed

# Law's Texture Energy



Artificial Intelligence  
& Computer Vision  
Laboratory

- **Algorithm:**

1. Apply convolution masks
2. Calculate the texture energy measure (TEM) at each pixel. This is achieved by summing the absolute values in a local neighborhood
3. Normalize features – use L5L5 to normalize the TEM image

# Using Law's energy for segmentation



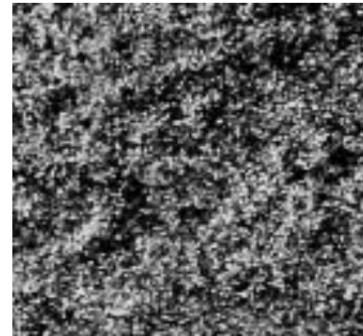
42:20

- (mm) • Subtract mean neighborhood intensity from pixel (to reduce illumination effects)
- Filter the neighborhood with 16 masks
- Compute energy at each pixel by summing absolute value of filter output across neighborhood around pixel
- Define 9 features as follows (replace each pair with average)
- |               |               |
|---------------|---------------|
| — L5E5 / E5L5 | — L5S5 / S5L5 |
| — L5R5 / R5L5 | — E5R5 / R5E5 |
| — E5S5 / S5E5 | — S5R5 / S5R5 |
| — S5S5        | — E5E5        |
| — R5R5        |               |

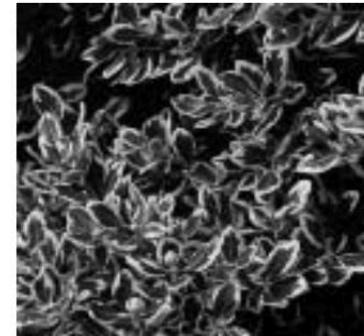


# Example: Law' Energy

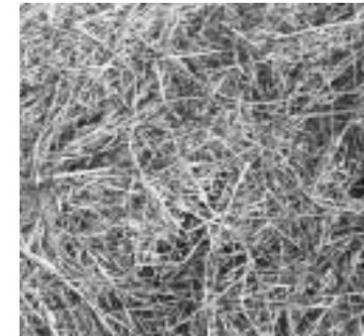
- Natural textures (from MIT Media Lab VisTex Database)



leaves



leaves



grass



brick



brick



stone



# Example: Law' Energy

- Natural textures (from MIT Media Lab VisTex Database)

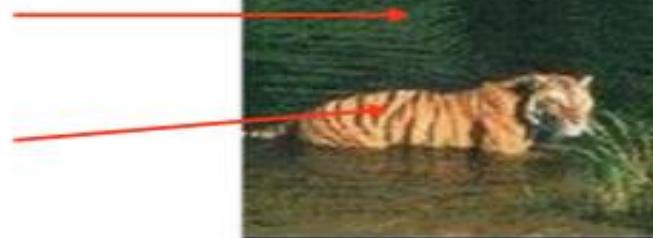
<b>Image</b>	<b>E5E5</b>	<b>S5S5</b>	<b>R5R5</b>	<b>E5L5</b>	<b>S5L5</b>	<b>R5L5</b>	<b>S5E5</b>	<b>R5E5</b>	<b>R5S5</b>
Leaves1	250.9	140.0	1309.2	703.6	512.2	1516.2	187.5	568.8	430.0
Leaves2	257.7	121.4	988.7	820.6	510.1	1186.4	172.9	439.6	328.0
Grass	197.8	107.2	1076.9	586.9	410.5	1208.5	144.0	444.8	338.1
Brick1	128.1	60.2	512.7	442.1	273.8	724.8	86.6	248.1	176.3
Brick2	72.4	28.6	214.2	263.6	130.9	271.5	43.2	93.3	68.5
Stone	224.6	103.2	766.8	812.8	506.4	1311.0	150.4	413.5	281.1

# Using Law's energy for segmentation



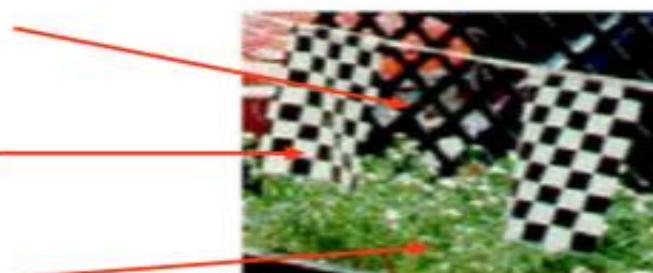
Artificial Intelligence  
& Computer Vision  
Laboratory

water



tiger

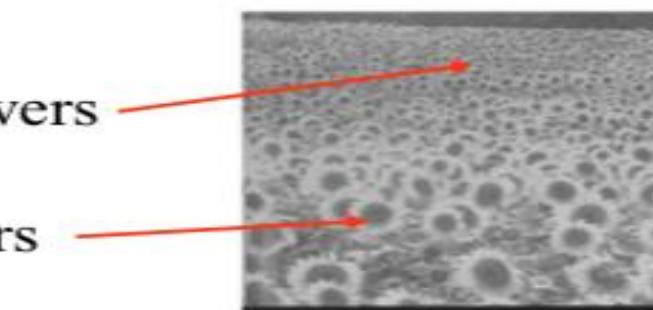
fence



flag

grass

small flowers



big flowers

# Using Law's energy for segmentation



Artificial Intelligence  
& Computer Vision  
Laboratory

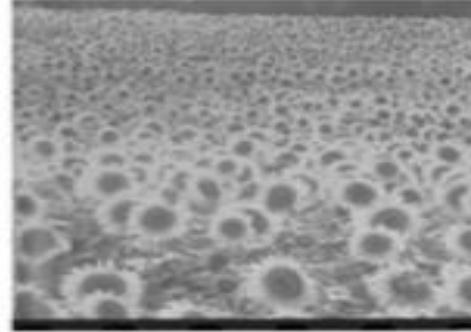


Table 7.2: Laws texture energy measures for major regions of the images of Figure 7.8.

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

# Using Law's energy for segmentation



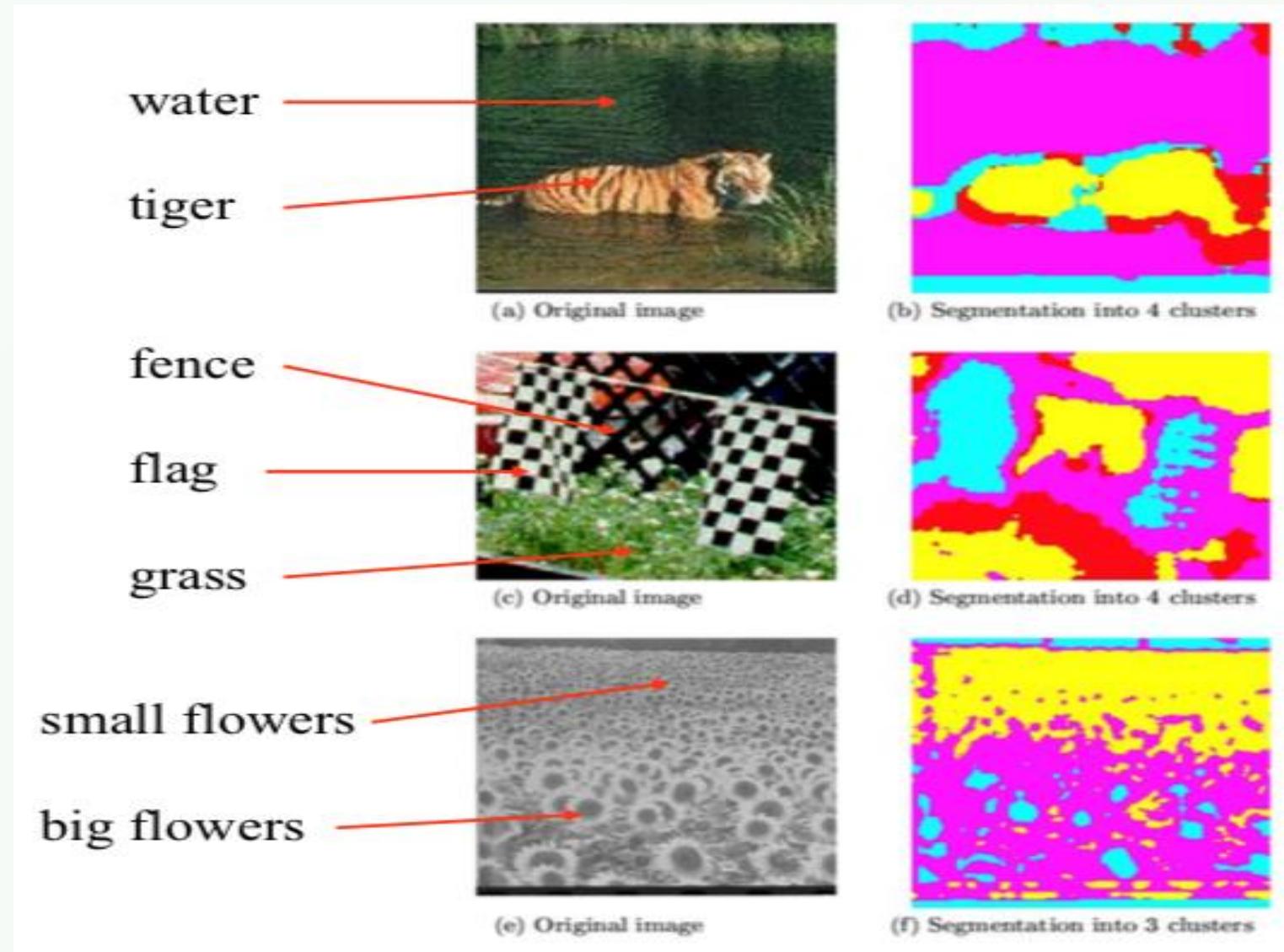
Table 7.3: Laws texture energy measures for tiger regions of several different images.

Image	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger1	171.2	96.8	1156.8	599.4	378.9	1162.6	124.5	423.8	332.3
Tiger2a	146.3	79.4	801.1	441.8	302.8	996.9	106.5	345.6	256.7
Tiger2b	177.8	96.8	1177.8	531.6	358.1	1080.3	128.2	421.3	334.2
Tiger3	168.8	92.2	966.3	527.2	354.1	1072.3	124.0	389.0	289.8
Tiger4	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Tiger5	146.9	80.7	868.7	474.8	326.2	1011.3	108.2	355.5	266.7
Tiger6	170.1	86.8	913.4	551.1	351.3	1180.0	119.5	412.5	295.2
Tiger7	156.3	84.8	954.0	461.8	323.8	1017.7	114.0	372.3	278.6

# Using Law's energy for segmentation



Artificial Intelligence  
& Computer Vision  
Laboratory



# Autocorrelation for texture

↳ ማሃን ገልፅ ዘዴትን በዚመሪያ አያዝዎ.

- Autocorrelation function computes the dot product (energy) of original image with shifted image for different shifts

$$\rho(dr, dc) = \frac{\sum_i \sum_j I[i, j] I[\underbrace{i + dr}_{\text{original}}, \underbrace{j + dc}_{\text{shift}}]}{\sum_i \sum_j I^2[i, j]} = \frac{I[i, j] \circ I_d[i, j]}{I[i, j] \circ I[i, j]}$$

- It can detect repetitive patterns of texels
- Also it can captures fineness/coarseness of the texture

# Interpreting Autocorrelation



Artificial Intelligence  
& Computer Vision  
Laboratory

- Regular textures : function will have peaks and valleys
- Random textures: only peak at [0,0] and breadth of peak gives the size of the texture
- Coarse texture: function drops off slowly
- Fine texture : function drops off rapidly
- Can drop differently for row and column

# Relationship to Fourier Analysis

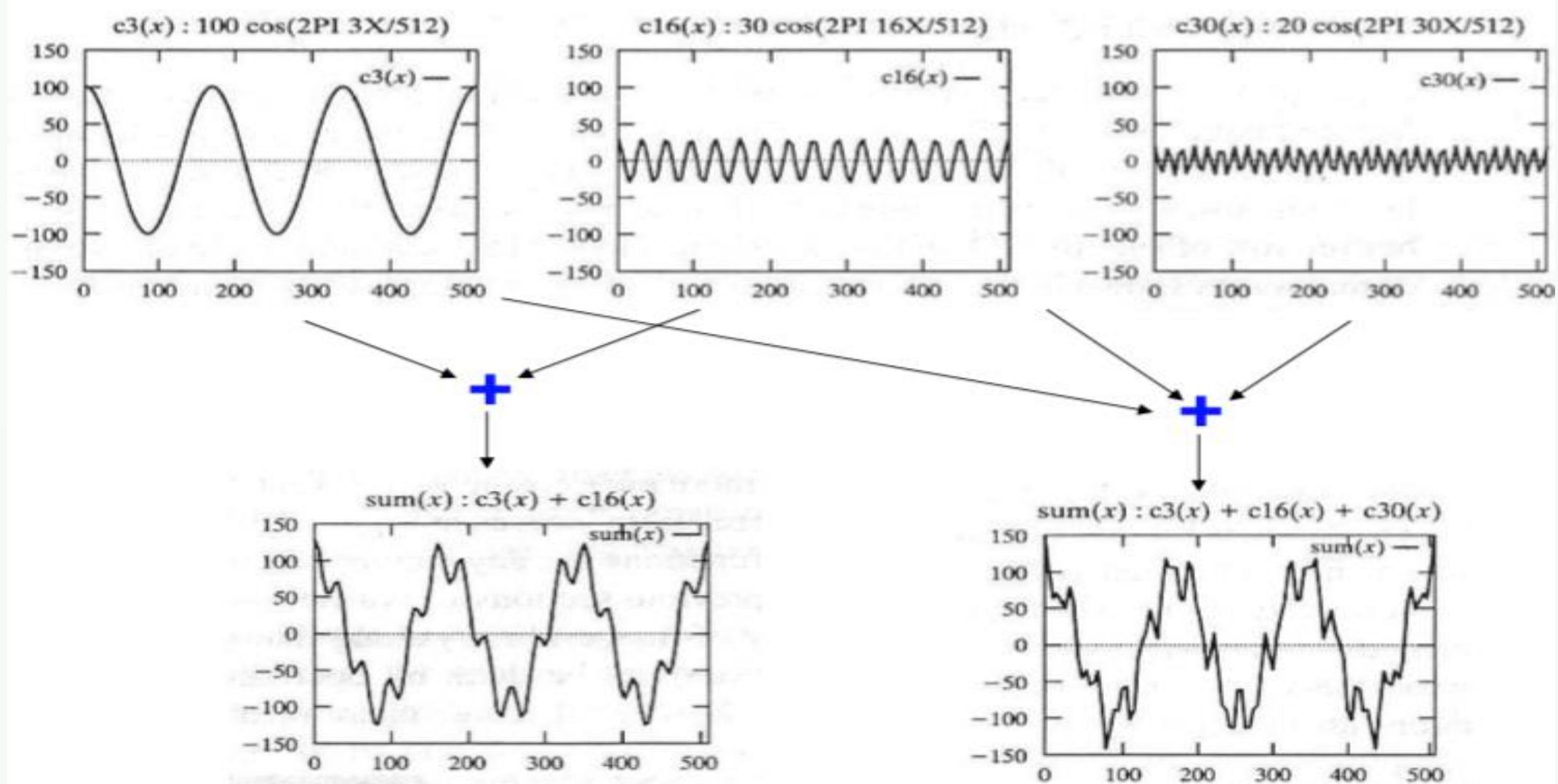


Artificial Intelligence  
& Computer Vision  
Laboratory

- The power spectrum of a signal is the Fourier transform of the autocorrelation function
- What is the Fourier transform?
  - Representing signals with sine/cosine waves



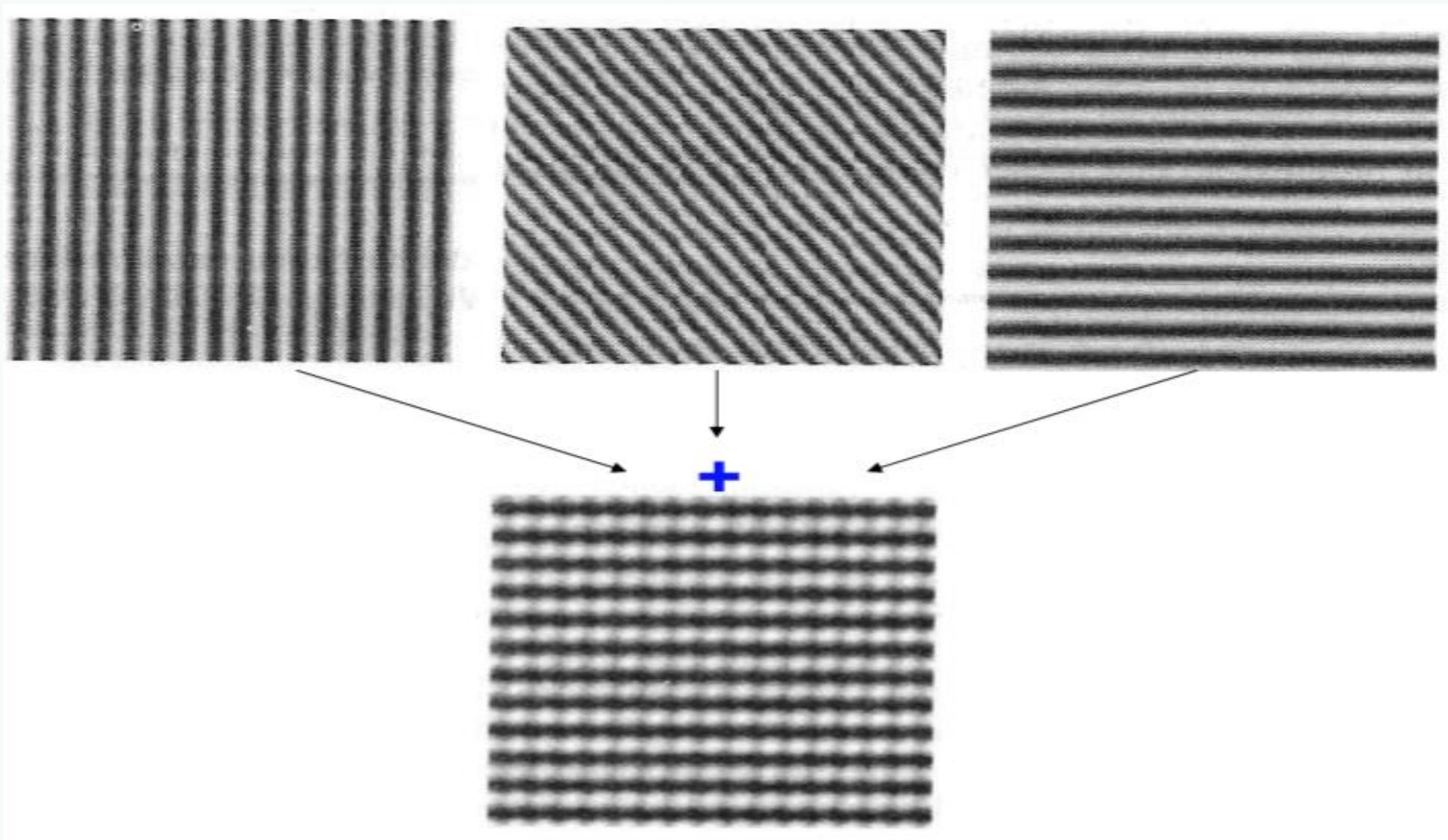
# 1D Example



# 2D Example



Artificial Intelligence  
& Computer Vision  
Laboratory

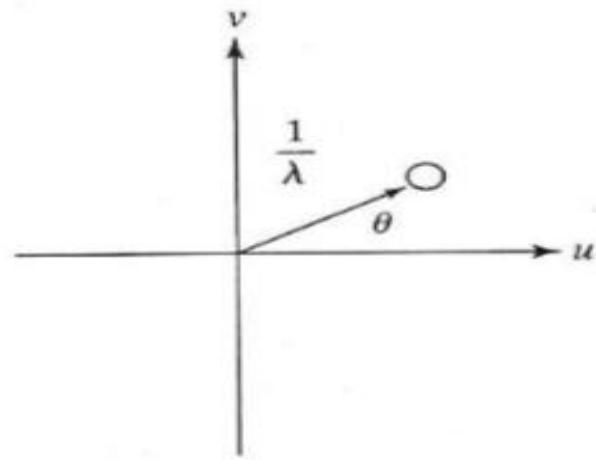
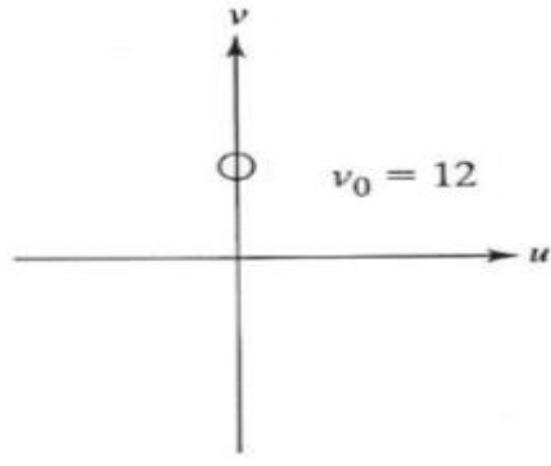
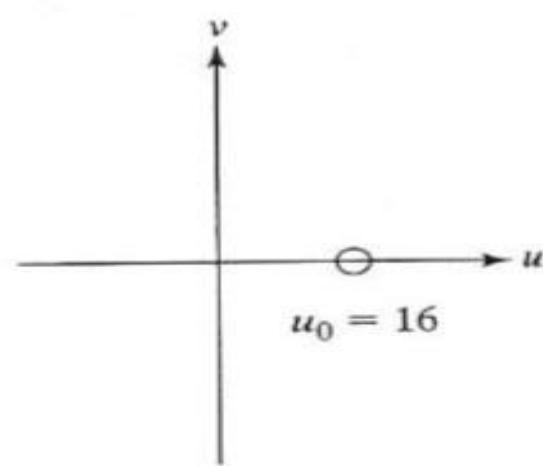
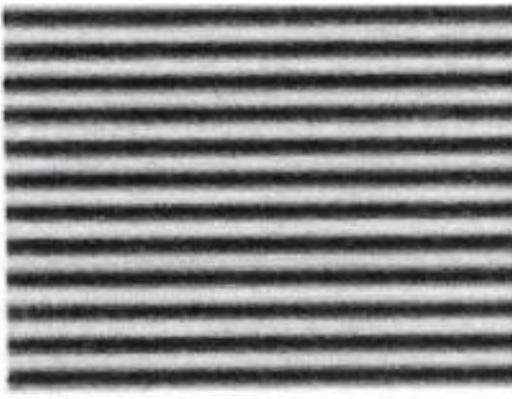
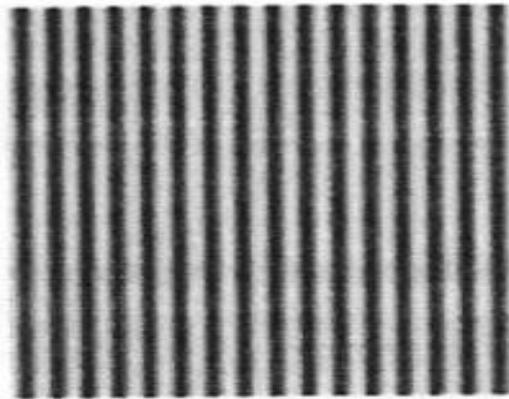


# Power Spectrum



Artificial Intelligence  
& Computer Vision  
Laboratory

$$P(u, v) \equiv (\text{Real}(F(u, v))^2 + \text{Imaginary}(F(u, v))^2)$$

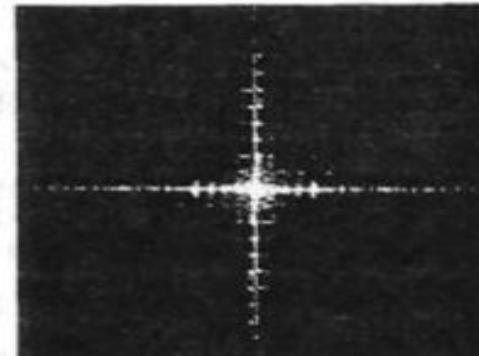
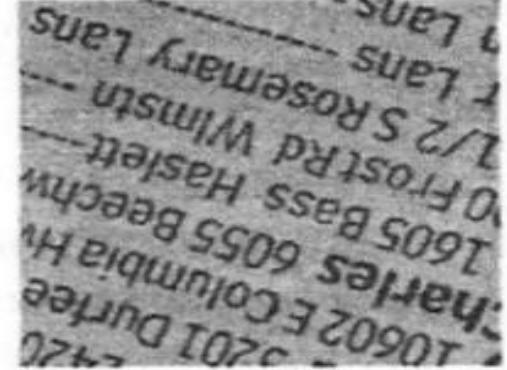
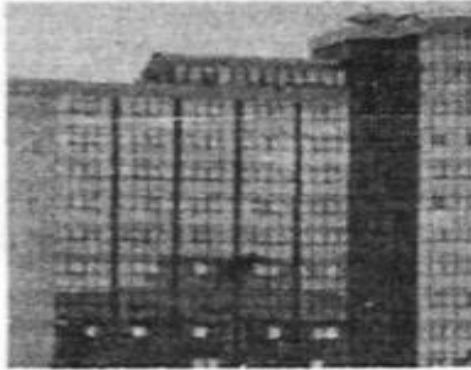


# Power Spectrum and Texture



Artificial Intelligence  
& Computer Vision  
Laboratory

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



# Summary



- Structural Approaches
- Statistical Approaches
  - - Simple features
  - - LBPs and GLCMs
  - - Edge based features
  - - Raw' energy
- Interpreting Textures



# Recommendations

- Paper reading for texture feature analysis
  - using gray level co-occurrence matrix
  - using Law's texture energy
  - using edge-based features
- For example,
  - Texture features analysis using GLCM for abnormality detection in chest CT images (2018)
  - Image retrieval system, Fingerprint recognition and range image classification using GLCM