Texture analysis.

Textures

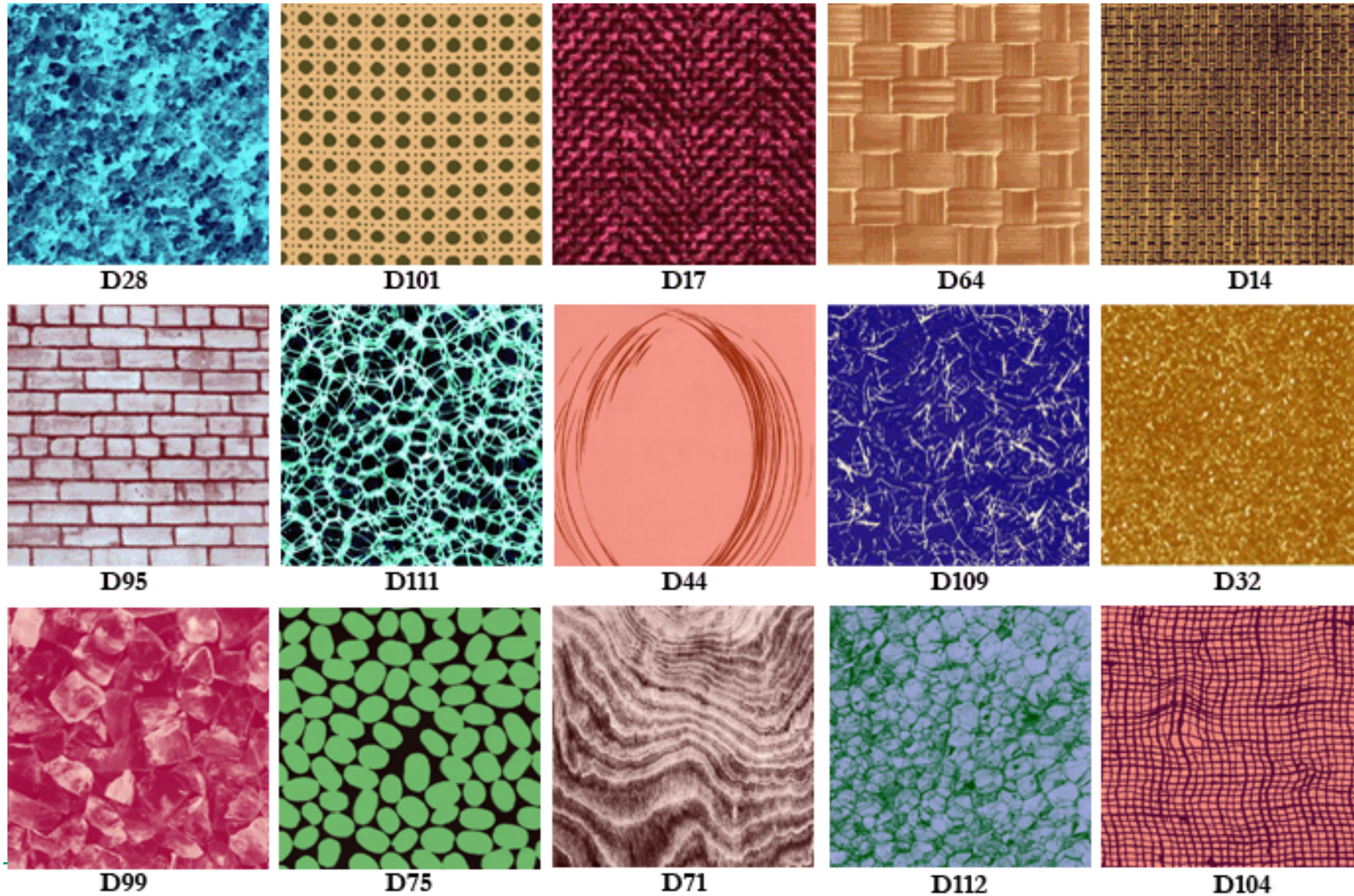

D28
D101
D17
D64
D14
D95
D111
D44
D109
D32
D99
D75
D71
D112
D104

Image Texture

- Texture analysis refers to the characterization of regions in an image by their texture content
- Texture analysis attempts to quantify intuitive qualities described by terms such as rough, smooth, silky, or bumpy as a function of the spatial variation in pixel intensities
- In this sense, the roughness or bumpiness refers to variations in the intensity values, or gray levels

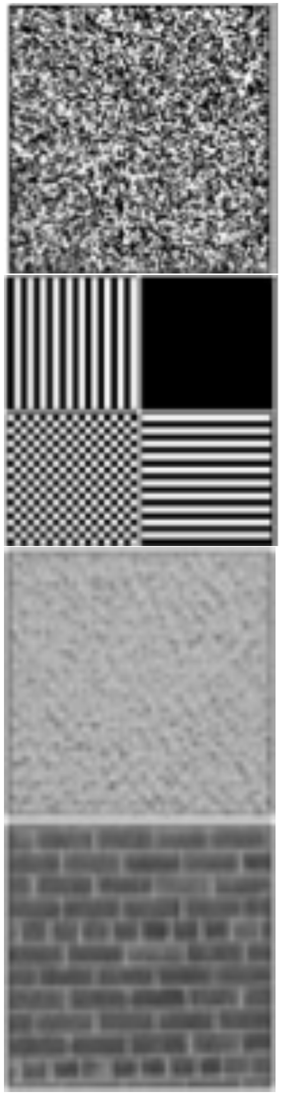
Image Texture

- Texture analysis is used in a variety of applications, including remote sensing, automated inspection, and medical image processing
- Texture analysis can be used to find the texture boundaries, called texture segmentation
- Texture analysis can be helpful when objects in an image are more characterized by their texture than by intensity, and traditional thresholding techniques cannot be used effectively

Features

- Image features can be found from:
  - Edges
    - Gives the borders between image regions (sometimes incomplete)
  - Homogenous regions
    - Mean and variance are useful for describing them
  - Texture of a local sliding window

Visual Texture

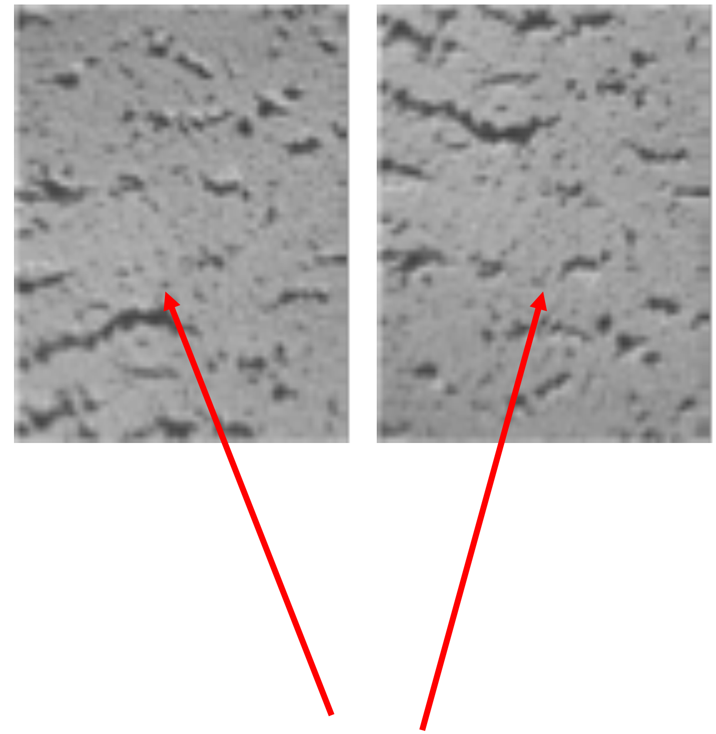


Stochastic
(isotropic)

Deterministic
(anisotropic)

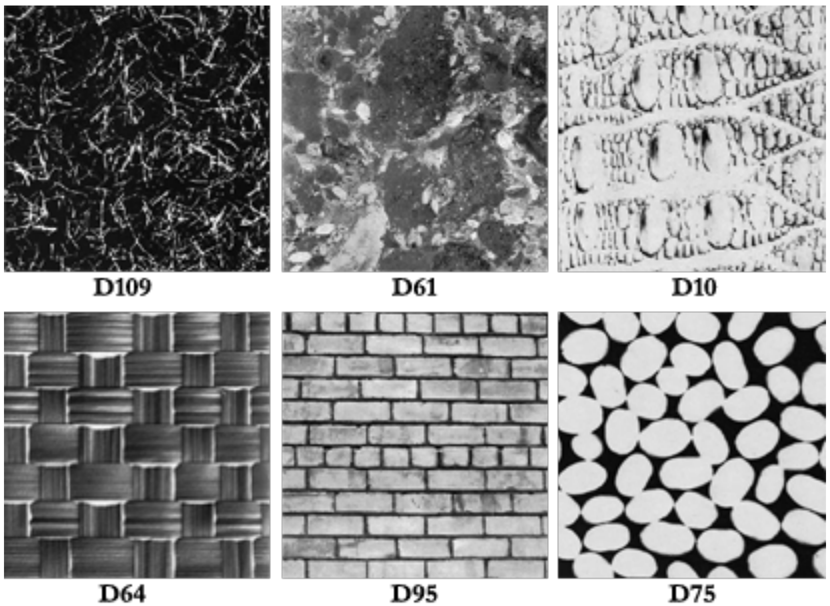
Semi Stochastic

Semi Deterministic



Visual Texture
Perception

Brodatz Textures



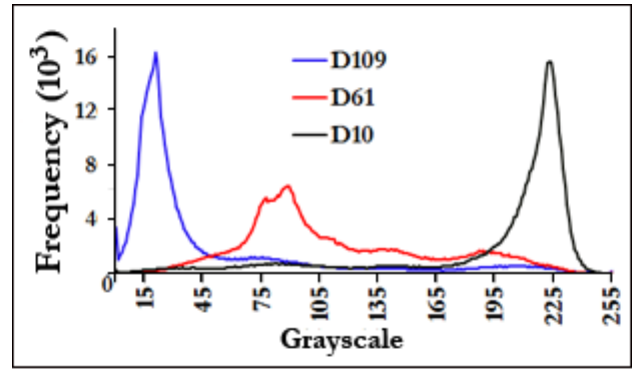


Figure 1

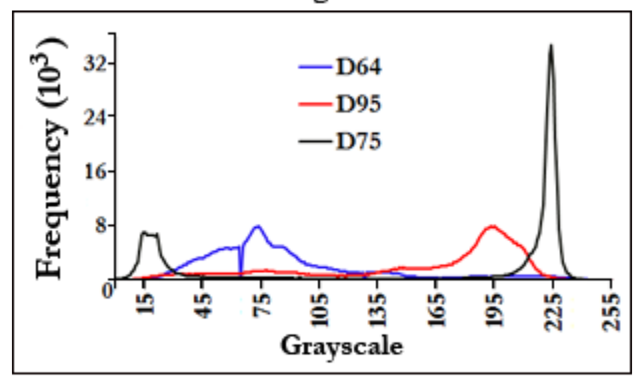
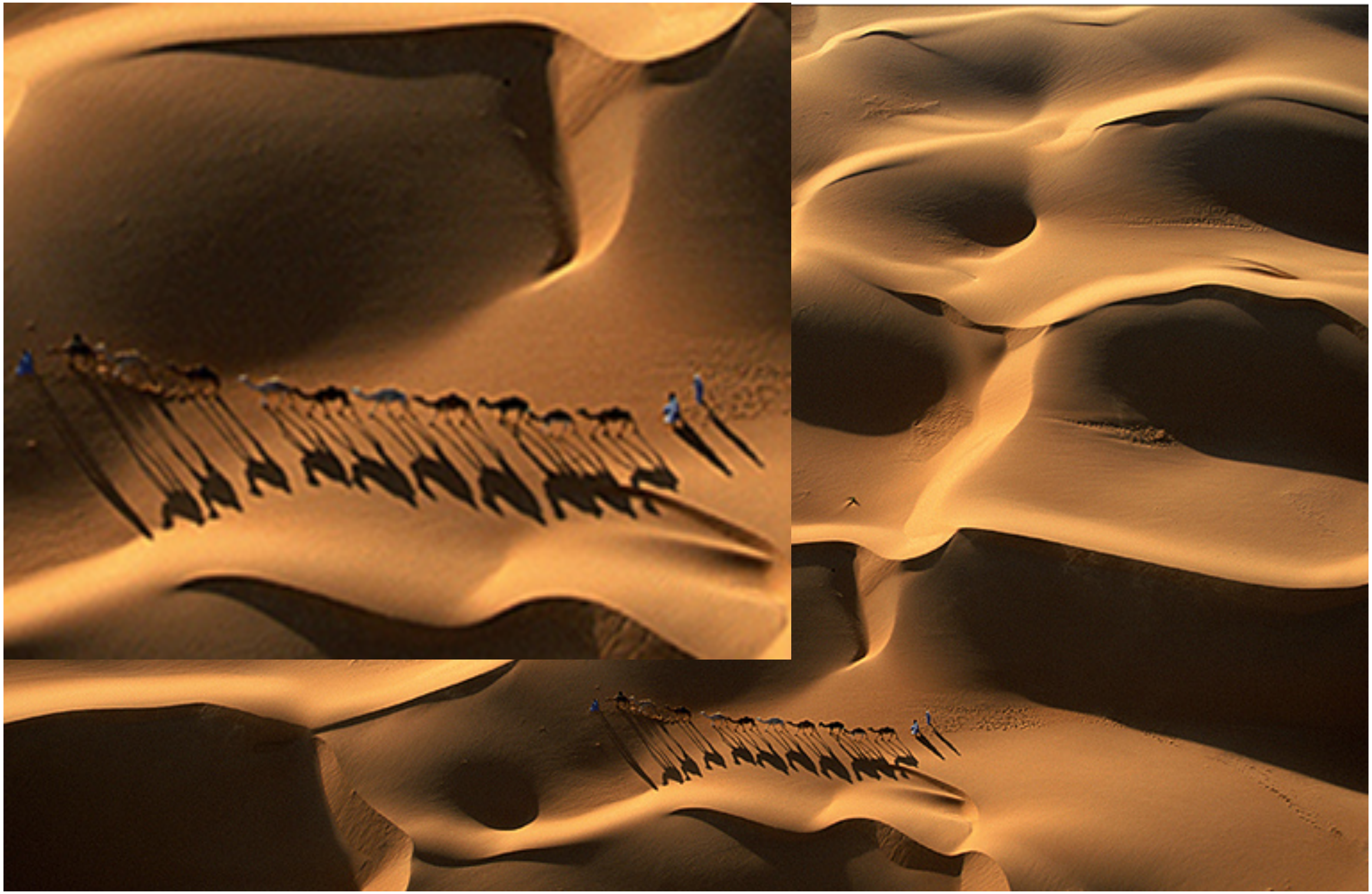


Figure 2

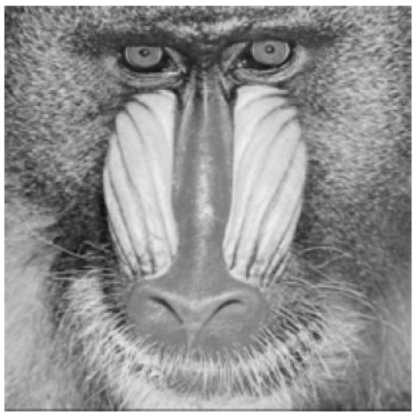
Brodatz album contains 112 texture images

http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html

Texture analysis: A matter of scale



Example of scale dependence



Original image

Variance feature computed in window of size 3x3

Variance feature computed in window of size 15x15

Texture feature extraction

Histograms

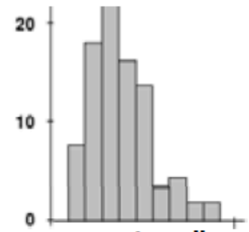
GLCM – Grey Level Co-occurance matrix

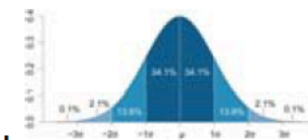
Histograms

First order statistics from histograms

- Mean (hardly a useful feature)

$$\mu = \frac{\sum_{i=0}^{G-1} ip(i)}{\sum_{i=0}^{G-1} p(i)} = \frac{\sum_{i=0}^{G-1} ip(i)}{n} = \sum_{i=0}^{G-1} iP(i)$$





- Variance (a more credible feature, measures region "roughness")

$$\sigma^2 = \sum_{i=0}^{G-1} (i-\mu)^2 P(i)$$

- Skewness (are the texel intensities usually darker/lighter than average?)

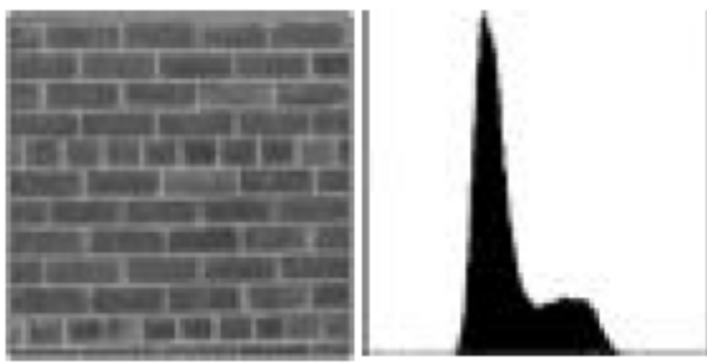
$$\gamma_3 = \frac{1}{\sigma^3} \sum_{i=0}^{G-1} (i-\mu)^3 P(i) = \frac{m_3}{\sigma^3}$$

- Kurtosis (how "peaked" is the graylevel distribution?)

$$\gamma_4 = \frac{1}{\sigma^4} \sum_{i=0}^{G-1} (i-\mu)^4 P(i) - 3 = \frac{m_4}{\sigma^4} - 3$$

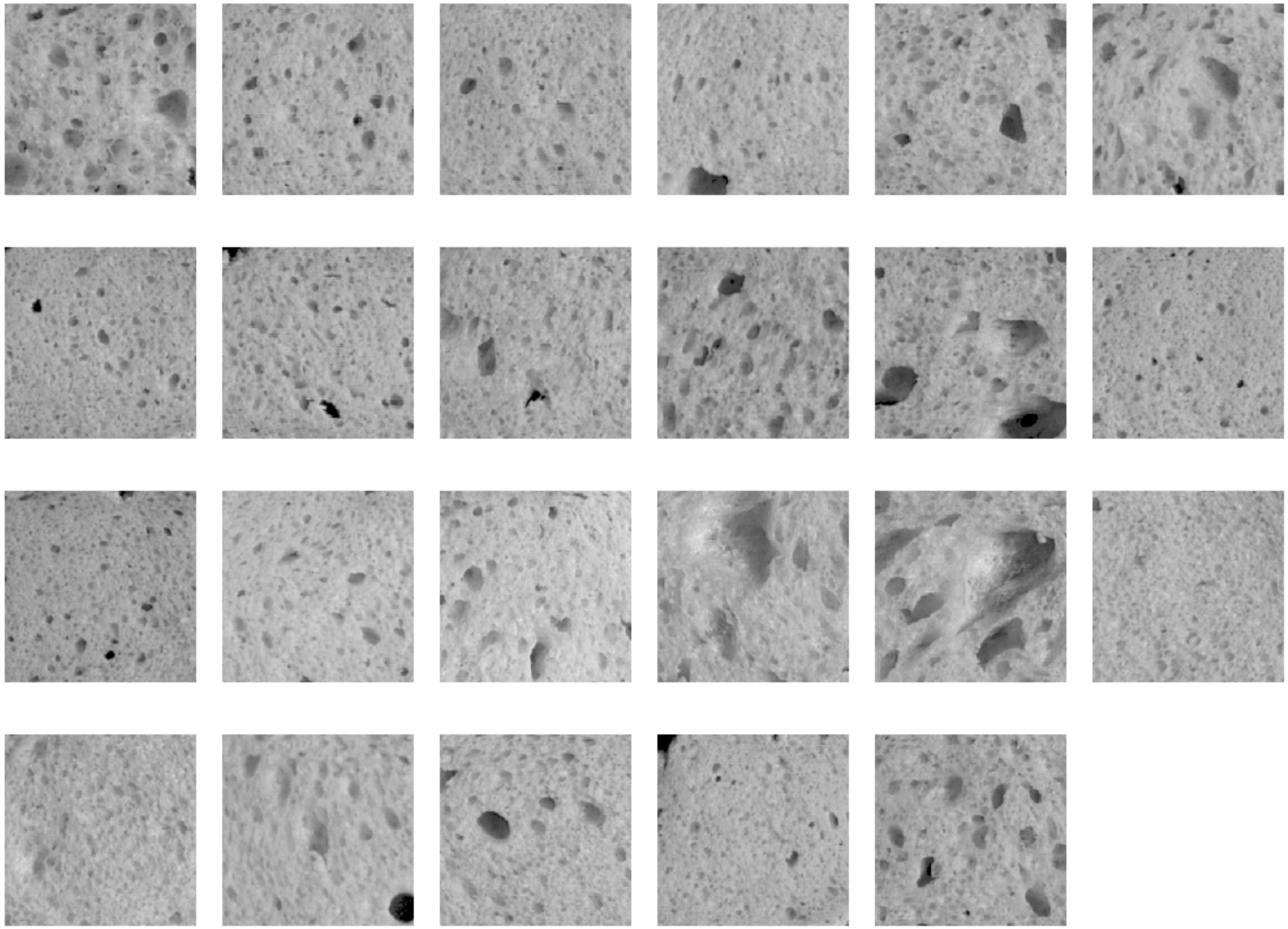
Example of first order statistics

Mean	167.9
Variance	669.0
CV	0.15
Skewness	-0.82
Curtosis	0.01

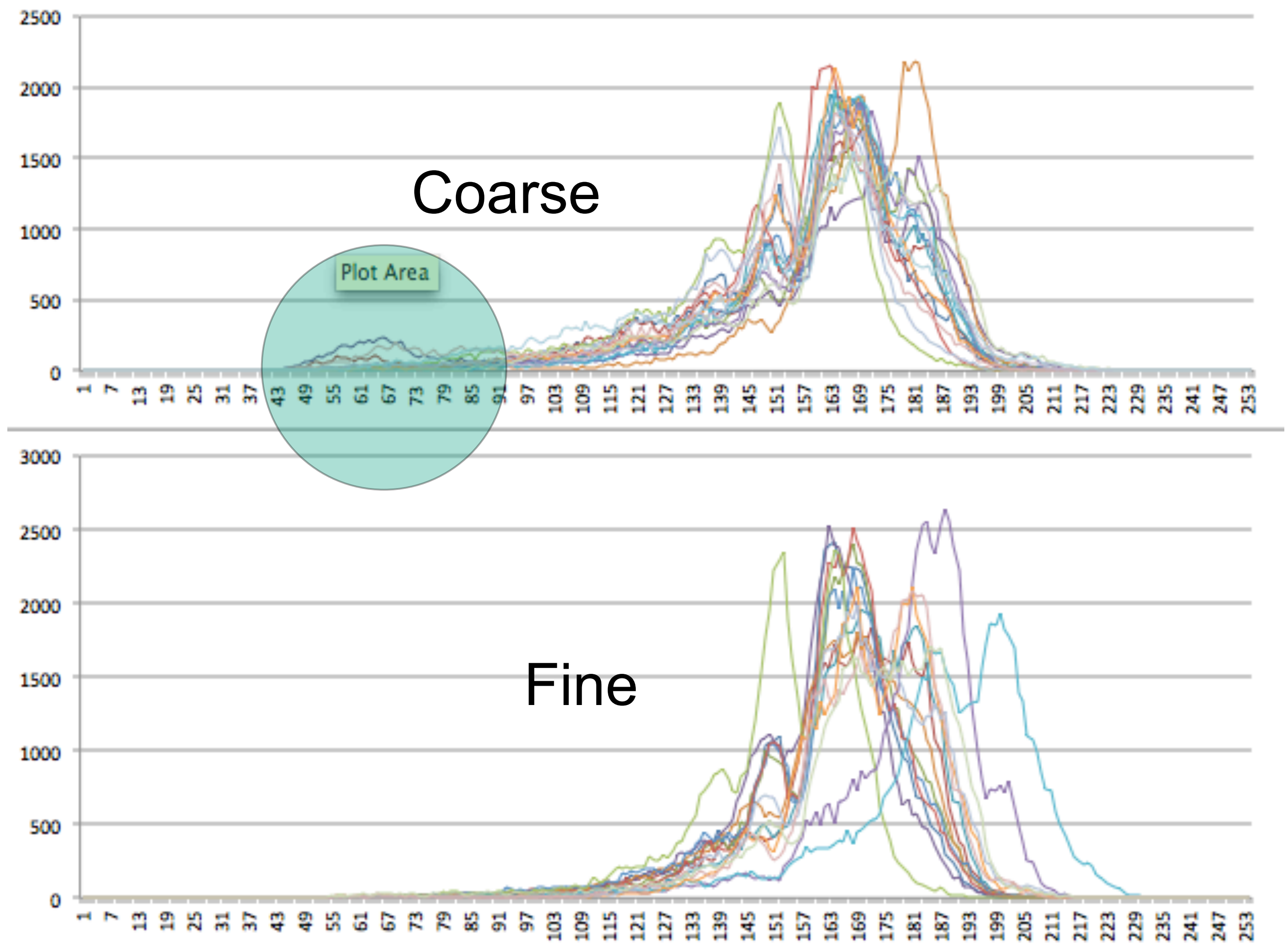


Mean	105.1
Variance	720
CV	0.26
Skewness	1.15
Curtosis	0.30

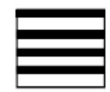
Baguette textures



Baguette textures and histograms



Limitation with first order statistics

- Edges around objects are exaggerated
- 1. order statistics can not describe geometry or context
  - Can not desicriminate between  
  - Solution:
    - Calculate 1. order statistics with different resolutions, obtain information about higher order statistics
    - Use 2. or higher order statistics

Grey Level Co-occurance Matrix (GLCM)

The gray level co-occurrence matrix (GLCM)

- Statistical calculations on the second-order histograms of the gray scale images.
- Calculate how often two pixels, in the matrix element A_δ (i, j), with intensity values i and j at a particular displacement distance δ from along a given direction θ (horizontally, vertically, or diagonally) occurs in the image

δ =1

θ =0

A_δ

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

GLCM

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

The GLCM matrix P

1	2	3	2	4
5	3	2	1	1
3	2	1	2	1
1	1	5	2	1
2	5	4	1	3

	1	2	3	4	5
1	2	2	1	0	1
2	4	0	1	1	1
3	0	3	0	0	0
4	1	0	0	0	0
5	0	1	1	1	0

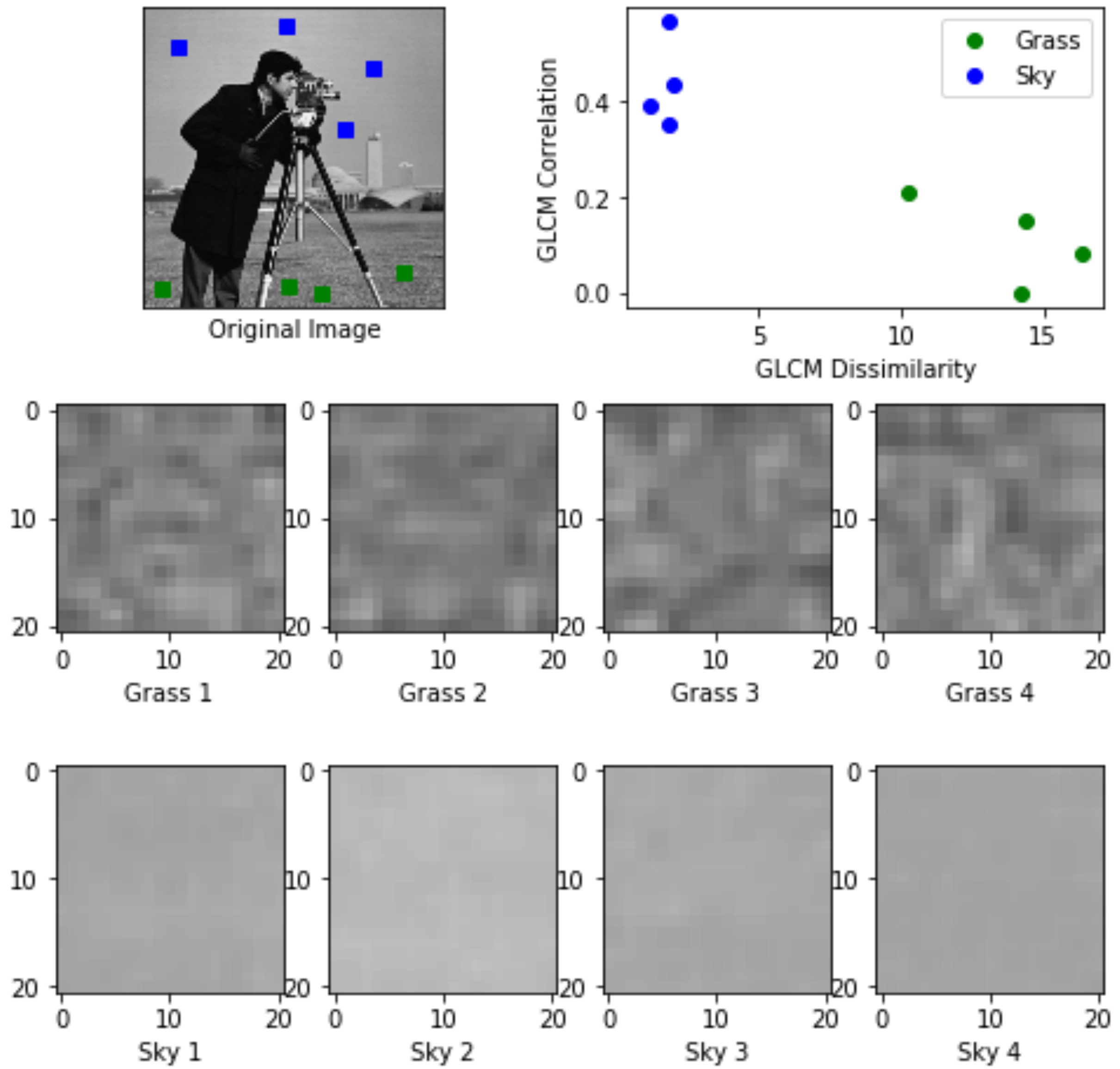
GLCM (0°)

1	2	3	2	4
5	3	2	1	1
3	2	1	2	1
1	1	5	2	1
2	5	4	1	3

	1	2	3	4	5
1	3	1	0	1	0
2	3	1	0	0	0
3	0	0	2	0	0
4	0	1	0	0	0
5	0	2	0	0	1

GLCM (45°)

Grey level co-occurrence matrix features



http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_glcm.html

Computing texture images

- Select a window size and select feature
- For each pixel compute

 the texture feature

similar to filtering

- For each pixel,

compute a local

homogeneity measure

 in a local window

GLCM Variance

Image thresholded with GLCM Variance