INF 4300 – Digital Image Analysis

TEXTURE

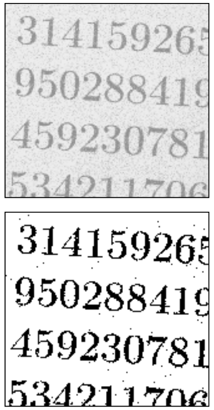
Fritz Albregtsen 02.09.2015

Plan for today

- Why texture, and what is it?
- Statistical descriptors
  - First order
    - Mean, variance, …
  - Second order
    - Gray level co-occurrence matrices
  - Higher order
    - Gray level runlength matrices
    - Laws’ texture energy measures
    - Fourier analysis

What is segmentation?

- A process that splits the image into meaningful regions.
- One of the most important elements of a complete image analysis system.
- Segmentation gives us regions and objects that we may later describe and recognize.
- The simplest case: two classes:
  - Foreground
  - Background



”Simple” example: find symbols for OCR

Segmentation problems

- Usually several objects in an image.
- Objects are seldom alike, even if they are of same class.
- Often several classes.
- Lighting may vary over image.
- Reflection, color etc. may vary.

- We perceive and utilize
  - intensity,
  - color,
  - and **texture**.

A more complex example:
What and where is the object in this image?

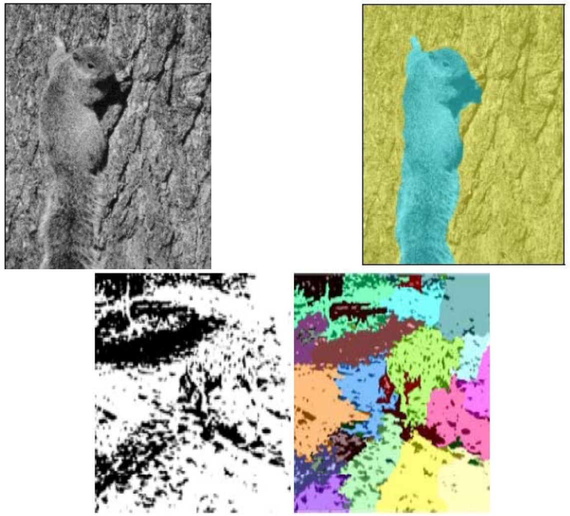
What is texture?

- Intuitively obvious, but no precise definition exists
  - ”fine, coarse, grained, smooth” etc

> “Spatially extended patterns of more or less accurate repetitions of some basic texture element, called texels.”

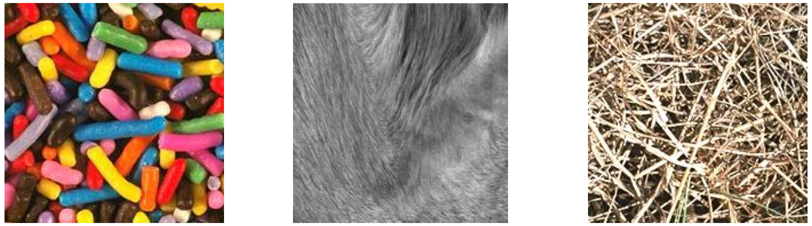
- Texture consists of texture primitives, texels,
  - a contiguous set of pixels with some tonal and/or regional property
- Texture can be characterized by
  - intensity (color) properties of texels
  - structure, spatial relationships of texels
- A texel is the characteristic object that the texture consists of (the ”brick in the wall”)
- Textures are highly scale dependent.

How do we segment these images?



What is a texel?

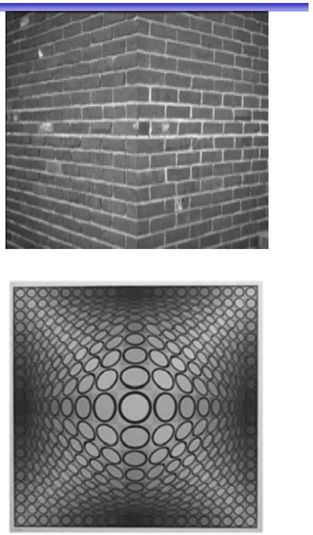
Texel = texture element, the fundamental unit of texture space.
Can be defined in a strict geometrical sense, or statistically.



Note that you can define texels in any image scale,
and that the best image scale for analysis is problem dependent.

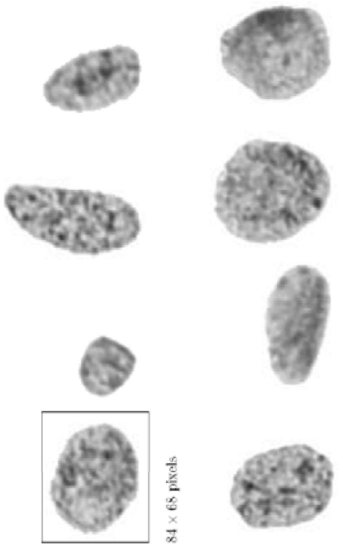
Uses for texture analysis

- Segment an image into regions with the same texture, i.e. as a complement to graylevel or color
- Recognize or classify objects in images based on their texture
- Find edges in an image, i.e. where the texture changes
- ”shape from texture”
- object detection, compression, synthesis
- Industrial inspection:
  - find defects in materials



Uses for texture analysis - II

- Reliable cancer prognostics.
- Microscopy images of monolayer cell nuclei from an early stage of ovarian cancer.
- Four monolayer cell nuclei from a good prognosis sample (left) and four nuclei from a bad prognosis sample (right).
- Aim: small set of differentiating textural features.

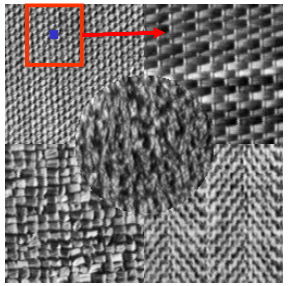


Features

- Image features can be found from:
  - Edges
    - Gives the (sometimes incomplete) borders between image regions
  - Homogeneous regions
    - Mean and variance are useful for describing the contents of homogeneous regions
  - The texture of the local (sliding) window
    - A feature that describes how the gray levels in a window varies, e.g. roughness, regularity, smoothness, contrast etc.

A simple approach to texture

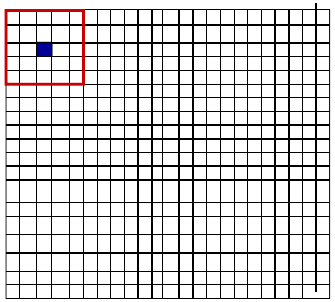
- To be able to find changes in the texture of an image, a simple strategy is to perform texture measurements in a sliding window
- Most texture features can be summed up as scalars, so we can assign features to each of the image pixels corresponding to window centers
- For each pixel, we now have a description of the ”texture” in its neighborhood
- Beware of image boundaries, artifacts will occur!



Compute a local texture feature in a local window.
Slide the window around in the image.
Each computed texture feature gives a new texture feature image!

Computing texture images

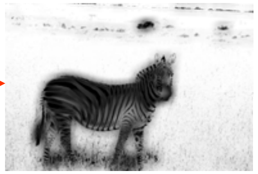
- Select a window size and select a texture feature
- For each pixel (i,j) in the image:
  - Center the window at pixel (i,j)
  - Compute the texture feature
    - One value is computed based on the gray-level variations of pixels inside the image
  - Assign the computed value to the center pixel (i,j) in a new output image of the same size
- This is similar to filtering
- Pixels close to the image border can be handled in the same manner as for filtering/convolution



Texture feature image example

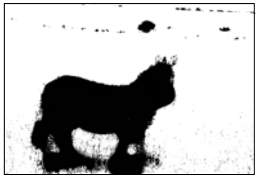
Input image.

For each pixel, compute a local homogeneity measure in a local sliding window.



New homogeneity image.

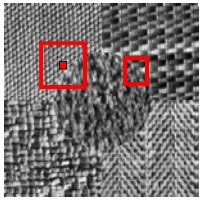
Try to get an image where pixels belonging to the same texture type get similar values.



Segmented feature image.

"Texture" – description of regions

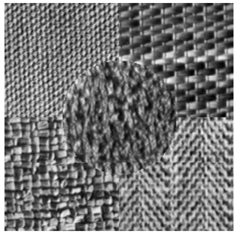
- Remember: we estimate local properties (features) to be able to isolate regions which are similar in an image (segmentation), usually with the goal of object description
  and possibly later identify these regions (classification),
- One can describe the "texture" of a region by:
  - smoothness, roughness, regularity, orientation...
- Problem: we want the local properties to be as "local" as possible
- Large region or window
  - Precise estimate of features
  - Imprecise estimate of location
- Small window
  - Precise estimate of location
  - Imprecise estimate of feature values



Uncertainty relation

- Large region or window
  - Precise feature value, but imprecise boundaries between regions
- Small window
  - Precise estimate of region boundaries, but imprecise feature value
- Related to Heisenberg's uncertainty relation in physics:

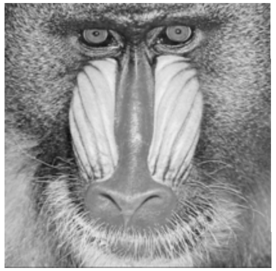
$$\Delta x \Delta p \approx h$$



Texture description is scale dependent

- What is our goal for texture description in the image?
- Scale impacts the choice of texels, and vice versa
- The curtain can be described as
  - a repetition of single threads,
  - a configuration of meshes
  - a repetition of folds.

Example of scale dependence



Original image

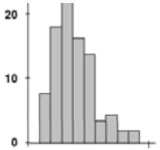
Variance feature computed in window of size 3x3

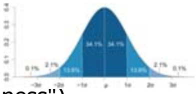
Variance feature computed in window of size 15x15

Statistical texture description

- Describe texture in a region by a vector of statistics (feature vector)
  - First order statistics from graylevel intensity histogram ***p(i)***
    - Mean, variance, 3. and 4. order moment
  - Second order statistics,
    describing relation between pixel pairs
    - How does the gray levels of pixels ***i*** and ***j*** at a distance ***d*** depend on each other. Are they similar or different ?
  - Higher order statistics,
    - describe region by ***runs*** of similar pixels

First order statistics from histogram - I





- 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?)

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

First order statistics from histogram - II

- Entropy
  (how uniform is the graylevel distribution?)

$$H = -\sum_{i=0}^{G-1} P(i) log_2 P(i)$$

- Energy
  (how non-uniform is the graylevel distribution?)

$$E = \sum_{i=0}^{G-1} [P(i)]^2$$

Using variance estimates

- Variance, σ^2, is directly a measure of ”roughness”
  - An unbounded measure ($\sigma^2 \geq 0$)
- A measure of ”smoothness” is

$$R = 1 - \frac{1}{1+\sigma^2}$$

  - A bounded measure ($0 \leq R \leq 1$)
    - R is close to 0 for homogenous areas
    - R tends to 1 as σ^2, ”roughness”, increases

Using variance estimates

- “Coefficient of variation”

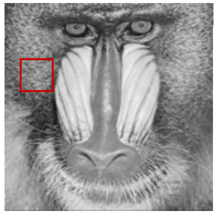
$$cv = \frac{\sigma_w(x,y)}{\mu_w(x,y)}$$

  - where σ_w and μ_w are computed within window w x w
  - *CV* is intensity scale invariant: $i' = Ai$
  - but not intensity shift invariant: $i' = i + B$
  - Alternatives:
    - use median instead of mean
    - interpercentile-distance instead of standard deviation
    - Also note “variance-to-mean” and “signal-to-noise”

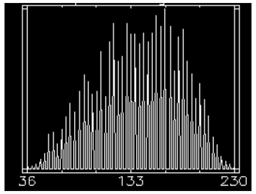
Skewness

- Higher order moments can also be used for texture description
- Skewness $\gamma_3 = \frac{1}{\sigma^3}\sum_{i=0}^{G-1}(i-\mu)^3 P(i) = \frac{m_3}{\sigma^3}$
  - **Skewness** is a measure of the asymmetry of the probability distribution
  - Measures if there is a "wider" range of either darker or lighter pixels
  - **Negative skew**: The left tail is longer; the mass of the distribution is concentrated on the right of the figure.
  - **Positive skew**: The right tail is longer; the *mass* of the distribution is concentrated on the left of the figure (more darker pixels than average).

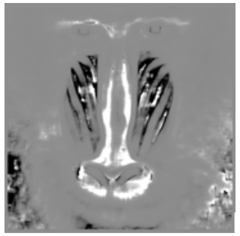
Skewness example



Region 1



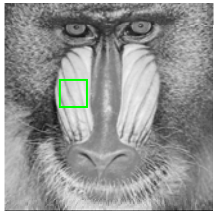
Histogram - Region 1



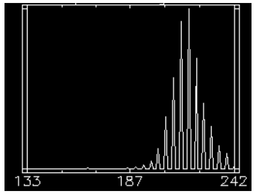
Skewness feature
Computed in 15x15 window

Region1: all gray levels occur
Histogram is fairly symmetric
Skewness is gray (average)

Region2: bright pixels more frequent. Histogram asymmetric.
Negative skew: Skewness is dark.



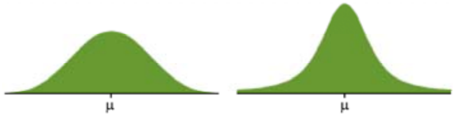
Region 2



Histogram - Region 2

Kurtosis

- Kurtosis



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

  - Measure of "peakedness" of the probability distribution.
  - Low kurtosis distribution has a more rounded peak with wider "shoulders"
  - A high kurtosis distribution has a sharper "peak" and flatter "tails"

First order statistics - Entropy

- Entropy (how uniform is the graylevel distribution?)

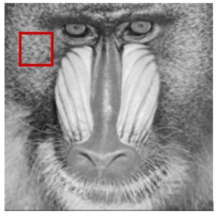
$$H = -\sum_{i=0}^{G-1} P(i) log_2 P(i)$$

- If all pixel values are the same, H = 0.
- If all pixel values are equally probable:
  - There are G = 2^b gray levels, each having a probability p(i) = 1/G = $1/2^b$, so:

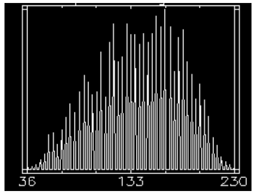
$$H = -\sum_{i=0}^{2^b-1} \frac{1}{2^b} \log_2(\frac{1}{2^b}) = -\log_2(\frac{1}{2^b}) = b$$

- We see that 0 ≤ H ≤ b (b = number of bits per pixel)
- Here, we use entropy as a texture feature, computed in a local window.

Entropy example



Region 1

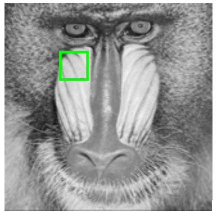


Histogram - Region 1

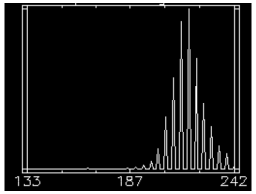
Entropy feature
Computed in 15x15
Window

Region1: high entropy

Region2: low entropy



Region 2



Histogram - Region 2

First order statistics - Energy

- Energy (how non-uniform?)

$$E = \sum_{i=0}^{G-1} [P(i)]^2$$

- A measure of homogeneity
- If all P(i) are equal (histogram is uniform), E=1/G
- If the image contains only one gray level:
  E=(G-1)×0+1×1=1
- **Thus, 1/G ≤ E ≤ 1**

1. order statistics discussion

- 1. order statistics can separate two regions even if $\mu_1 = \mu_2$, as long as $\sigma^2_1 \neq \sigma^2_2$, or skewness/kurtosis differ
- The statistics of a pixel *(x, y)* is found in a local window
- Problems:
  - Edges around objects are exaggerated
    - Solution: use adaptive windows
  - 1. order statistics does not describe geometry or context
    - Cannot discriminate between

    - Solution:
      - Calculate 1. order statistics with different resolutions, and obtain indirect information about 2. and higher order statistics.
      - Simply use 2. or higher order statistics.

Second order statistics

- Gray-level Co-Occurrence matrices
  - Intensity-change-"histograms" as a function of distance and direction
  - By far the most popular texture description method due to its simplicity
  - The co-occurrence matrix is an estimate of the second order joint probability,
    - which is the probability of
      - going from gray level *i* to gray level *j*,
      - given the distance *d* between two pixels
      - along a given direction θ.

Gray Level Coocurrence Matrices (GLCM)

- Matrix element *P(i,j)* in a GLCM is 2. order probability of changing from graylevel *i* to *j* when moving distance *d* in the direction θ of the image, or equivalent, $(\Delta x, \Delta y)$
- From a $M \times N$ image with *G* graylevels, and *f(m,n)* is the intensity. Then $P(i,j \mid \Delta x, \Delta y) = WQ(i,j \mid \Delta x, \Delta y)$, where

$$W = \frac{1}{(M-\Delta x)(N-\Delta y)} \quad , \quad Q(i,j \mid \Delta x, \Delta y) = \sum_{n=1}^{N-\Delta y} \sum_{m=1}^{M-\Delta x} A$$

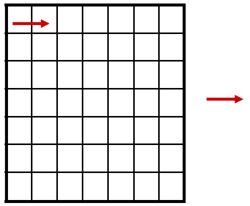
and

$$A = \begin{cases} 1 & \text{if } f(m,n) = i \text{ og } f(m+\Delta x, n+\Delta y) = j \\ 0 & \text{else} \end{cases}$$

- Alternative notation, dependent on distance and direction, $P(i,j \mid d, \theta)$

GLCM

- From one window of size $w \times w$ we get one GLCM matrix
- The dimension of the co-occurrence matrix is *GxG* if we have *G* gray-levels in the image.
- Choose a distance *d* and a direction θ



In this example, $d=1$ and $\theta=0$

- Check all pixel pairs with distance *d* and direction θ inside the window. $Q(i,j|d,\theta)$ is the number of pixel pairs where pixel *1* in the pair has pixel value *i* and pixel *2* has pixel value *j*.

GLCM

Image

0 → 1	1	1	2	3
0	0	2	3	3
0	1	2	2	3
1	2	3	2	2
2	2	3	3	2

d=1, θ =0 correspond to *dy=0, dx=1*

This row has no neighbors to the right. The number of pixel pairs that we can compute is N×(M-1) = 5 × (5-1) = 20

$Q(i,j|d,\theta)$

$$
\begin{array}{c|cccc}
 & \multicolumn{4}{l}{\text{gray level } j \rightarrow} \\
\hline
\text{gray level } i \downarrow & 1 & 2 & 1 & 0 \\
 & 0 & 1 & 3 & 0 \\
 & 0 & 0 & 3 & 5 \\
 & 0 & 0 & 2 & 2
\end{array}
$$

Cooccurrence Matrix

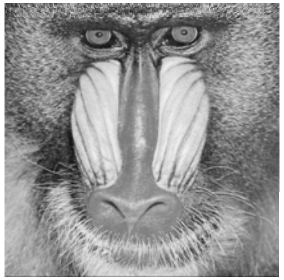
	j=0	1	2	3
i= 0	1/20	2/20	1/20	0
1	0	1/20	3/20	0
2	0	0	3/20	5/20
3	0	0	2/20	2/20

P(i,j|d,θ) is normalized by *W*, the number of pixel pairs inside the window.

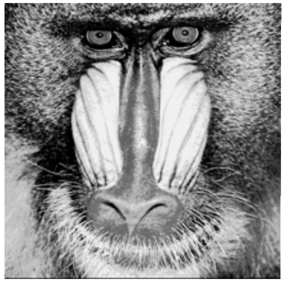
GLCM – practical issues

- The matrix must have a sufficient "average occupancy level" :
  - Reduce number of graylevels
    (Less precise description if the texture has low contrast)
    - Select **L** =number of gray levels
    - Rescale the image if necessary to use these levels using (histogram transform)
    - Requantize the scaled image from **G** to **L** gray levels before GLCM computation
  - Increase window size (Errors due to changes in texture)
- Heuristics:
  - *16* graylevels is usually sufficient
  - window should be *30 x 30* – *50* x *50* pixels.

Preprocessing examples

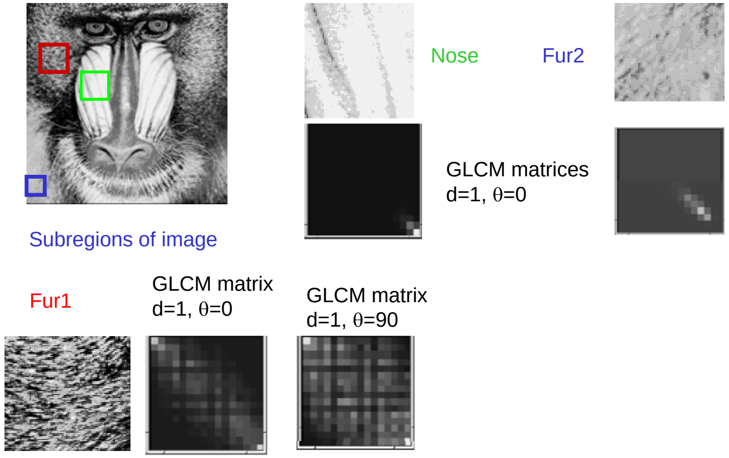


Original image



Histogram-equalized and requantized to 16 gray levels

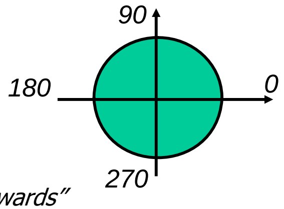
GLCM matrices for subregions



GLCM

- Usually a good idea to reduce the number of *(d,θ)* variations evaluated

- Simple pairwise relations:
  - $P(d,0^0) = P^t(d,180^0)$
  - $P(d,45^0) = P^t(d,225^0)$
  - $P(d,90^0) = P^t(d,270^0)$
  - $P(d,135^0) = P^t(d,315^0)$
  - *Symmetric GLCM:*
    - *Count "forwards" + "backwards"*
    - *Add matrix to its transpose*

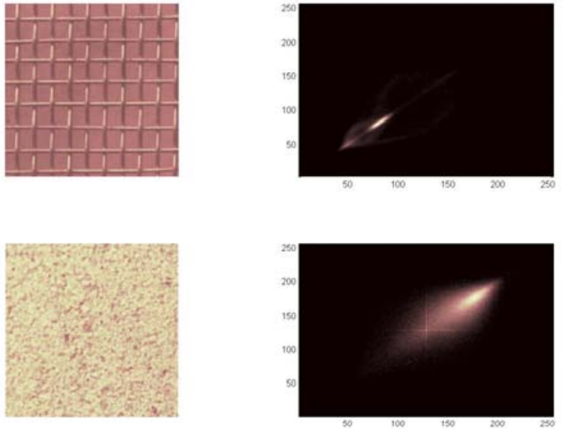


- Isotropic cooccurrence matrix by averaging
  $P(\theta)$, $\theta \in \{0^o, 45^o, 90^o, 135^o\}$
  - *Beware of differences in effective window size!*

- An isotropic texture is equal in all directions

- If the texture has a clear orientation, we select θ according to this.

Isotropic GLCM example



How to use the GLCM

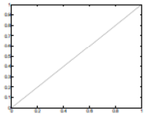
- Usually, used by extracting secondary features from GLCM
  - Haralick et al. and Conners et al.
  - Features are usually strongly correlated, using more than 4-5 simultaneously is not advisable
  - Need to evaluate several distances *d*
    - *Would you perform anti-aliasing filtering for d>1 ?*
  - Optimal set of features is problem dependent

- It is also advisable to preprocess by histogram transform to remove effect of absolute gray level.

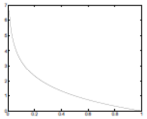
- Usually, we want to make the features "rotation" invariant by using the isotropic GLCM (remember different weights).

GLCM Features

- Angular Second Moment, $ASM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{P(i,j)\}^2$
  - ASM is a measure of homogeneity of an image.
  - Homogeneous scene will contain a few gray levels, giving a GLCM with few but high values of *P(i,j)*.
  - Thus, the sum of squares will be high.



- Entropy, $E = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) \times log(P(i,j))$
  - Inhomogeneous scenes have high entropy, while a homogeneous scene has a low entropy.
  - *Maximum Entropy is reached when all 2. order probabilities are equal.*



GLCM Features

- Correlation,
  - Correlation is a measure of gray level linear dependence between the pixels at the specified positions relative to each other.

$$COR = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j}$$

$$\mu_j = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} jP(i,j) \qquad \sigma_j^{\ 2} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(j-\mu_j)^2 P(i,j)$$

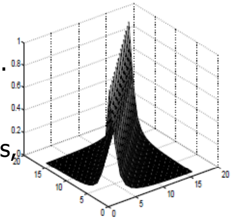
$$\mu_i = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} iP(i,j) \qquad \sigma_i^{\ 2} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-\mu_i)^2 P(i,j)$$

GLCM Features

- Contrast,
  - This measure of contrast or local intensity variation will favor contributions from *P(i,j)* away from the diagonal, i.e. $i \neq j$

$$CTR = \sum_{n=0}^{G-1} n^2 \{\sum_{i=1}^{G}\sum_{j=1}^{G} P(i,j)\}, \qquad | i-j |= n$$

- Inverse Difference Moment (also called homogeneity)
  - IDM is also influenced by the homogeneity of the image.
  - Because of the weighting factor $(1+(i-j)^2)^{-1}$ IDM will get small contributions from inhomogeneous areas ($i \neq j$).
  - The result is a low IDM value for inhomogeneous images, and a relatively higher value for homogeneous images.



$$IDM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i,j)$$

GLCM Features

- Sum of Squares, Variance, $VAR = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (i-\mu)^2 P(i,j)$
  - This feature puts relatively high weights on the elements that differ from the average value of *P(i,j)*
  - GLCM variance uses the GLCM, therefore it deals specifically with the dispersion around the mean of combinations of reference and neighbor pixels, i.e., encoding contextual (2. order) information
  - Variance calculated using i or j gives the same result, since the GLCM is symmetrical
  - Note that the contextuality is an integral part of this measure; one can measure the variance in pixels in one direction.
    Thus it is *not* the same as the 1. order statistic variance.

GLCM Features

- Sum Average, $AVE = \sum_{i=0}^{2G-2} iP_{x+y}(i)$
- Sum Entropy, $SEN = -\sum_{i=0}^{2G-2} P_{x+y}(i) \log \left(P_{x+y}(i)\right)$
- Difference Entropy, $DEN = -\sum_{i=0}^{G-1} P_{x+y}(i) \log(P_{x+y}(i))$
- Inertia, $INR = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \{i-j\}^2 \times P(i,j)$
- Cluster Shade, $SHD = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \{i+j-\mu_x-\mu_y\}^3 \times P(i,j)$
- Cluster Prominence, $PRM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \{i+j-\mu_x-\mu_y\}^4 \times P(i,j)$

GLCM feature image examples, w= 15

GLCM contrast

GLCM entropy

GLCM variance

GLCM contrast is
- negative correlated with IDM
- positively correlated with variance

GLCM entropy is negatively correlated with ASM

GLCM feature image examples, w= 15

GLCM IDM

GLCM ASM

GLCM correlation

Sum and difference histograms

- The sum histogram *S* is simply the histogram of the sums of all pixels *dx* and *dy* apart
- For example, the gray level at *I(x,y)* is added to the gray level at *I(x+dx,y+dy)* and the histogram bin corresponding to that sum is incremented
- The difference histogram D is simply the histogram of the difference of all pixels *dx* and *dy* apart

$$s_{\Delta x,\Delta y}(m,n) = f(m,n) + f(m+\Delta x, n+\Delta y)$$

$$d_{\Delta x,\Delta y}(m,n) = f(m,n) - f(m+\Delta x, n+\Delta y)$$

- The number of possible values of sum and difference histogram is *2G-1.*

Sum and difference histograms

- GLCM features can be derived from P_s and P_d
- Example:
  - Contrast from P_d

$$CON = \sum_{j=0}^{2G-2} j^2 P_d(j \mid \Delta x, \Delta y)$$

  - Contrast from GLCM

$$CON = \sum_{n=0}^{G-1} n^2 \{\sum_{i=1}^{G} \sum_{j=1}^{G} P(i,j)\}, \quad \mid i-j \mid = n$$

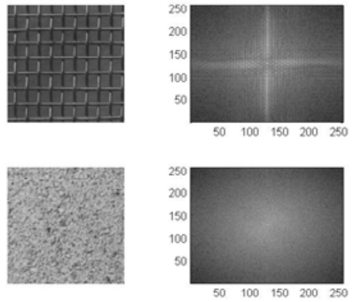
- Some of the features mentioned earlier is derived from the histograms
  - Sum Average, $AVE = \sum_{i=0}^{2G-2} P_s$
  - Sum Entropy, $SEN = -\sum_{i=0}^{2G-2} P_s(i) \log(P_s(i))$
  - Difference Entropy, $DEN = -\sum_{i=0}^{G-1} P_d(i) \log(P_d(i))$
  - Inverse Difference moment, $IDM = \sum_{i=0}^{G-1} \frac{1}{1+i^2} P_d(i)$

Fourier analysis

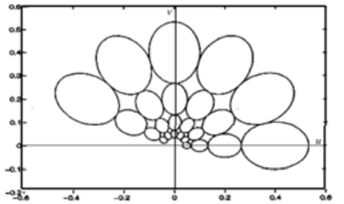
- The Fourier spectra give direction and frequency for periodic or near periodic 2D patterns
- Local FFT in windows
- Texture with a dominating direction will have peaks in the spectra along a line orthogonal to the texture orientation
- High frequency = fine texture = peaks in the spectra far from the origin
- Thus it is possible to separate fine and coarse spectra
- The spread in image frequencies = width of the peak in Fourier
- Isotropic textures with a defined frequency can be seen as rings in the spectra
- Scalar features can be extracted by integration over rings, wedges or from results of Gabor filtering

Fourier analysis example

- Transform to Fourier domain, integrate over rings or wedges



Gabor filters combine estimate of orientation and frequency.



Higher order statistics

- Higher order methods include
  - Gray level runlength matrices - ”histograms” of graylevel run lengths in different directions (INF 5300)
  - Laws’ texture masks - masks resulting from combinations of 0,1,2 derivatives

Laws' texture energy measures

- Based on convolution (INF 2310)
- Uses 3×3 or 5×5 separable masks that are symmetric or anti-symmetric
- This results in a new texture image for every convolution mask.
- From the results of convolution find the standard deviation or average of absolute values over a larger window (e.g. 15×15)
- This is a measure of texture energy in some direction depending on the mask chosen.

3×3 Law's texture estimator

- A set of masks is convolved with the image:

L3L3
$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

L3E3
$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

L3S3
$$\begin{matrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{matrix}$$

E3L3
$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

E3E3
$$\begin{matrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{matrix}$$

E3S3
$$\begin{matrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{matrix}$$

S3L3
$$\begin{matrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{matrix}$$

S3E3
$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

S3S3
$$\begin{matrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{matrix}$$

- The masks are separable.
- 3×3 masks made by convolving

L3 = [1 2 1]
E3 = [-1 0 1]
S3 = [-1 2 -1]

- The texture measure is found by computing the standard deviation over a larger window for every image convolved with the Law's masks

---

5×5 Law's texture masks

- **E5L5**

$$\begin{vmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{vmatrix}$$

- **E5S5**

$$\begin{vmatrix} -1 & 0 & 2 & 0 & 1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{vmatrix}$$

- **R5R5**

$$\begin{vmatrix} -1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{vmatrix}$$

- **L5S5**

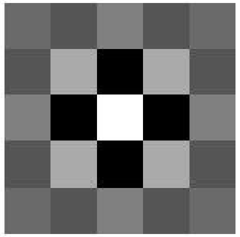
$$\begin{vmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{vmatrix}$$

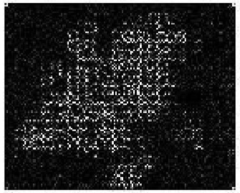
5×5 masks made by convolving vertical & horizontal 5-vector :

L5 = [1 4 6 4 1]   E5 = [-1 -2 0 2 1]
S5 = [-1 0 2 0 -1]   R5 = [1 -4 6 -4 1]

L5 = L3*L3   E5 = L3*E3   S5 = L3*S3   R5 = S3*S3

"Finding rippling texture" - R5R5 example





Multiscale texture analysis

- We have ascertained that texture description is scale dependent
- By representing the image in a pyramid structure
  we can describe image texture on different scales
- A discrete wavelet-transform will give four new images with reduced resolution, called s_{ll}, s_{lh}, s_{hl}, and s_{hh}

$$s_{ll}[n;m] = \left[\frac{\left[\frac{a+b}{2}\right] + \left[\frac{c+d}{2}\right]}{2}\right] \quad , \quad s_{lh}[n;m] = \left[\frac{(a-b+c-d)}{2}\right]$$

$$s_{hl}[n;m] = \left[\frac{a+b}{2}\right] - \left[\frac{c+d}{2}\right] \quad , \quad s_{hh}[n;m] = (a-b-c+d)$$

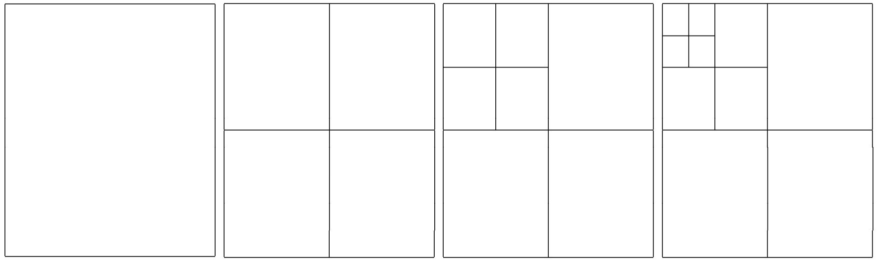
- where *a,b,c,d* are four neighboring pixels in the original image

$$a = f[2n;2m] \quad , \quad b = f[2n+1;2m]$$

$$c = f[2n;2m+1] \quad , \quad d = f[2n+1;2m+1]$$

Multiscale texture analysis

- By repeating this decomposition on s_{ll}, the result is a hierarchical pyramid structure for different resolutions, both in the lowpass image and the other subbands.
- The image can be represented in multiple resolutions, and multi-scale GLCM (for example) can be calculated
- Furthermore, each of the three other subbands can be viewed as edge information for that scale.



Learning goals - texture

- Understand what texture is, and the difference between first order and second order measures
- Understand the GLCM matrix, and be able to describe algorithm
- Understand how we go from an image to a GLCM feature image
  - Preprocessing, choosing d and θ, selecting some features that are not too correlated
- Understand Law's texture measures and how they are built based on basic filtering operations
- **There is no optimal texture features, it depends on the problem**
- **A good tutorial on texture: http://www.fp.ucalgary.ca/mhallbey/tutorial.htm**