Biomedical Informatics 260

Computational Feature Extraction:

Texture Features

Lecture 6

David Paik, PhD

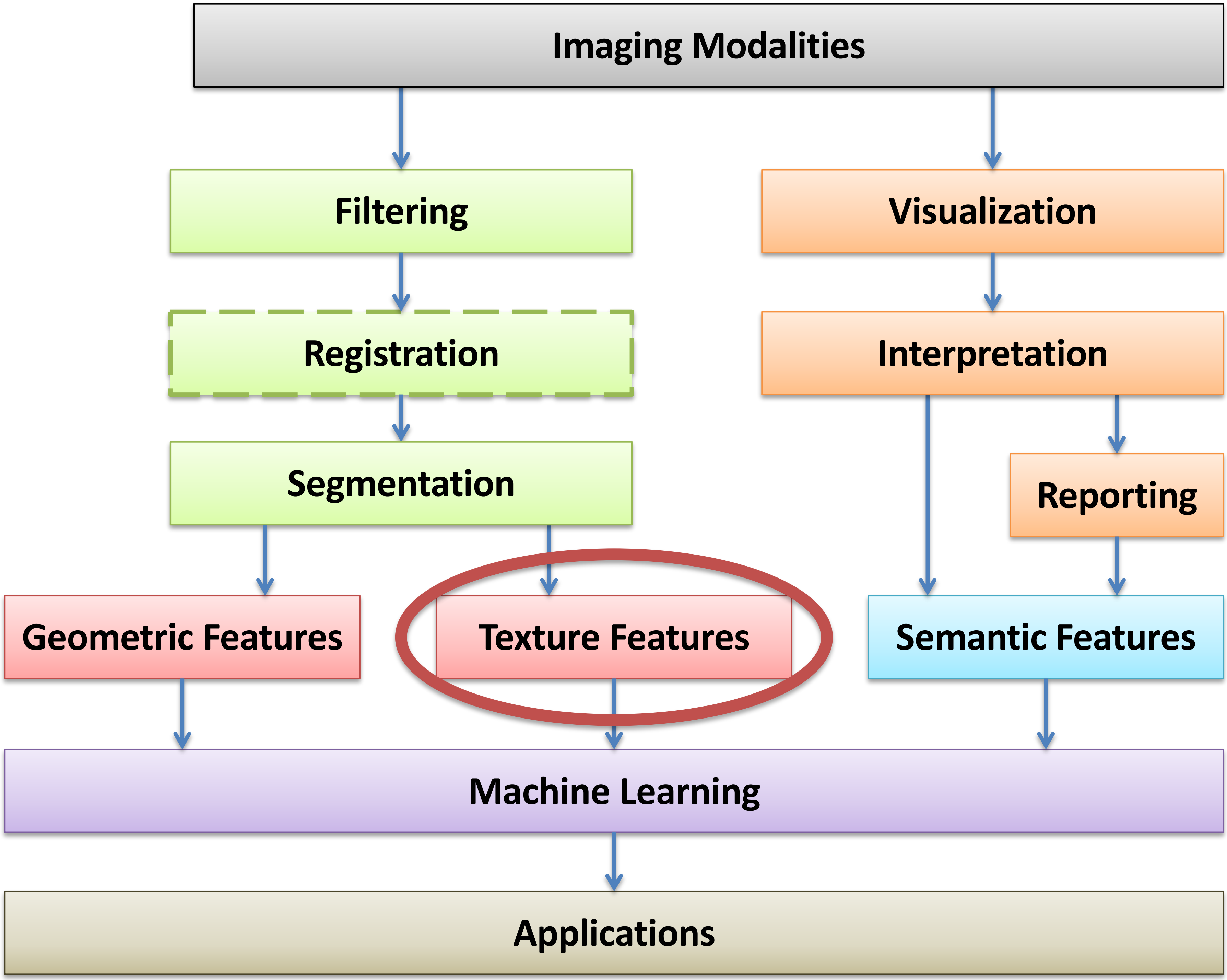
Spring 2019

Last Lecture: Computational Feature Extraction: Geometric Features

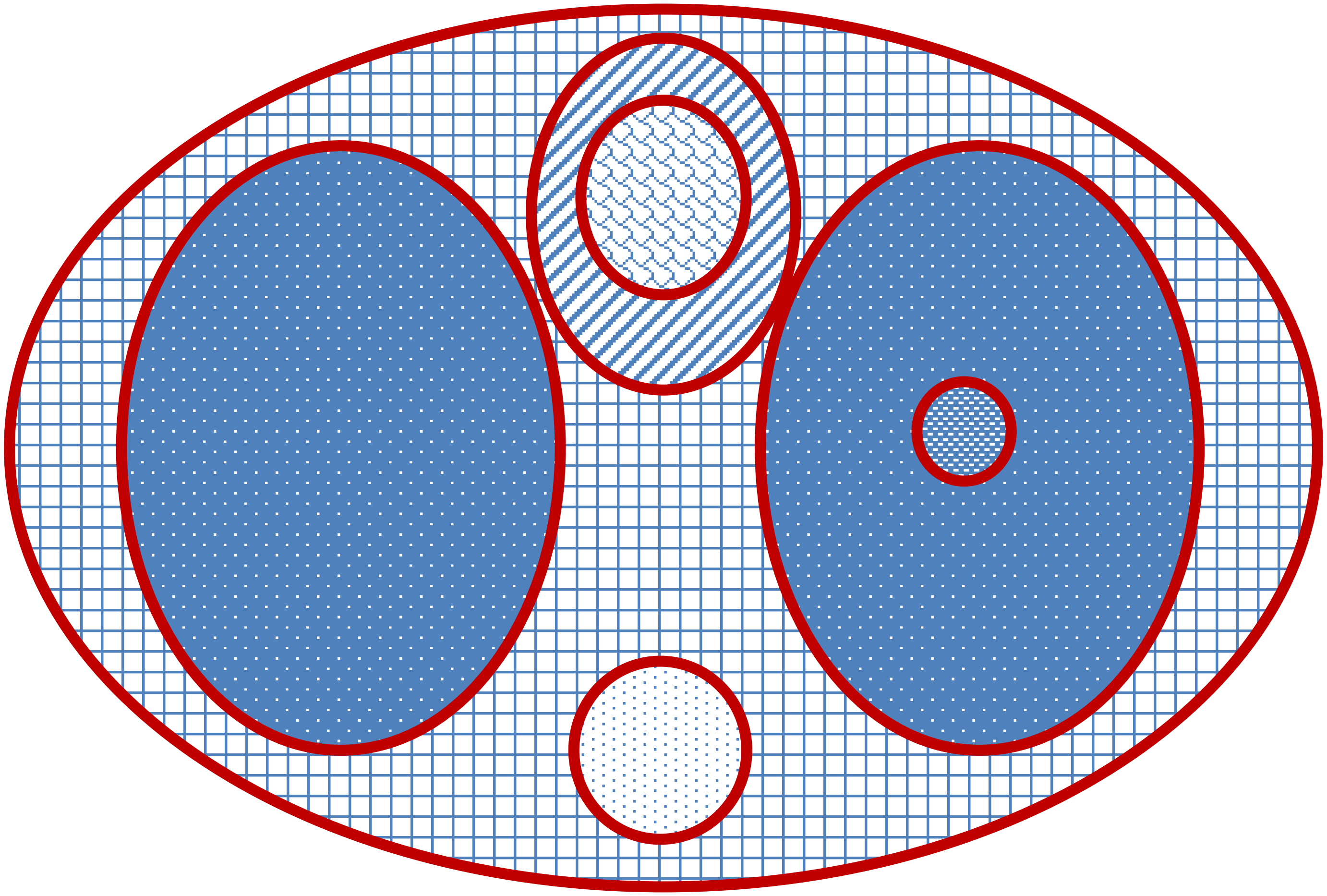
1. Methods:
    1. Local Pointwise Features
    2. Morphological Analysis
    3. Shape Features
    4. Shape Parameterization
2. Many features to describe shape and geometry
3. Considering natural parameterization of anatomy can be very useful

Today:
Computational Feature Extraction: Texture Features

- Texture is the "grain" that falls somewhere in between shapes and individual pixel values
- In many clinical imaging applications, the overall shape is less important the detailed features inside
- Topics:
  - Defining Texture
  - 1^{st} and 2^{nd} Order Statistical Features
  - Transforms
  - Fractal Analysis
  - Applications


Imaging Modalities
Filtering
Visualization
Registration
Interpretation
Segmentation
Reporting
Geometric Features
Texture Features
Semantic Features
Machine Learning
Applications

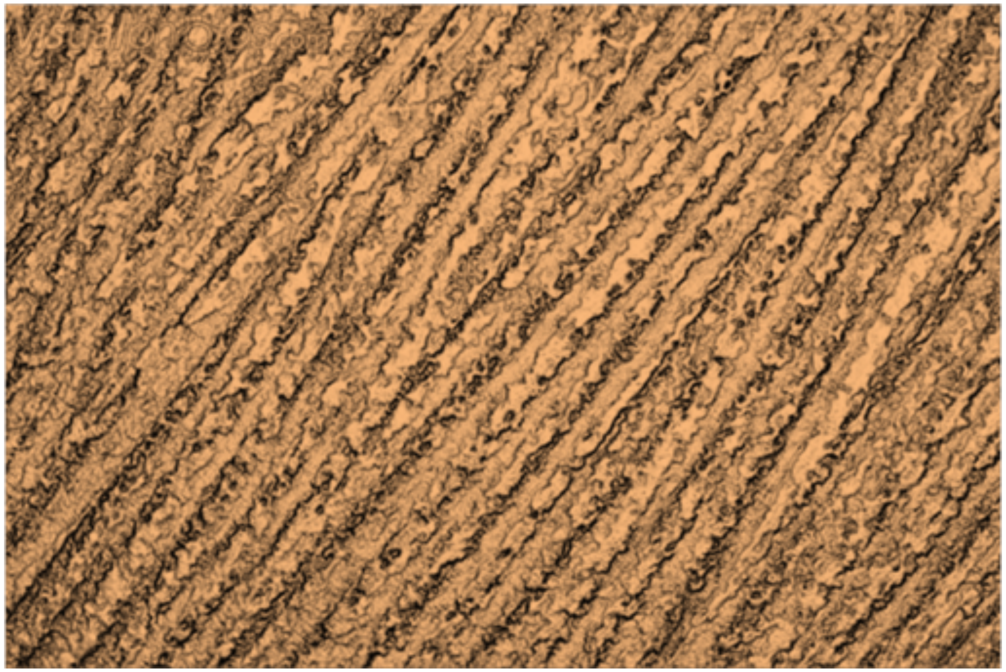
The Shape v Texture View of the World

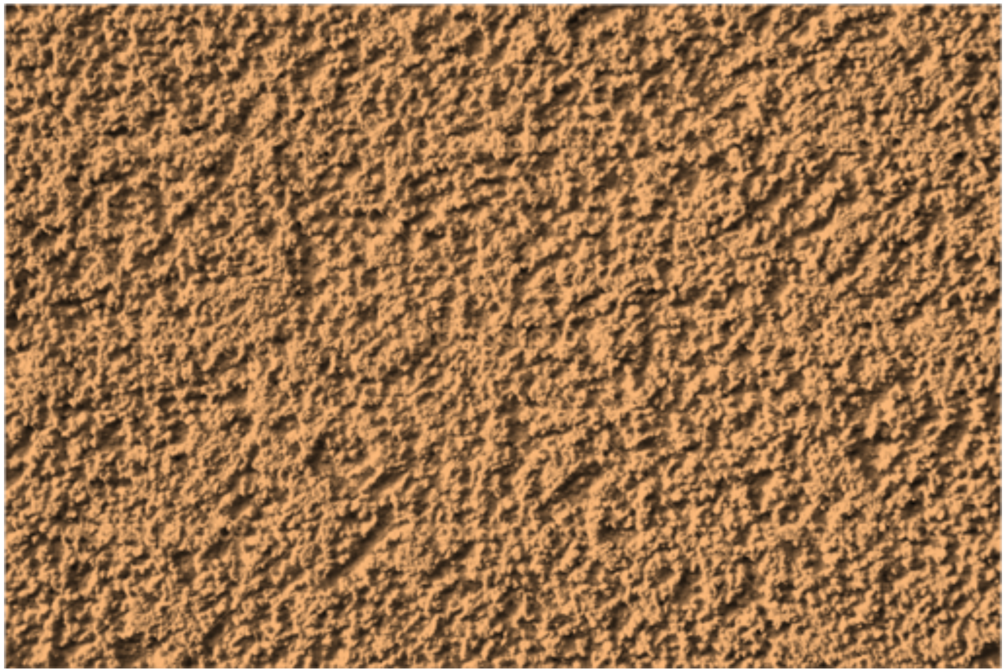


What is texture?

What different textures can you identify?

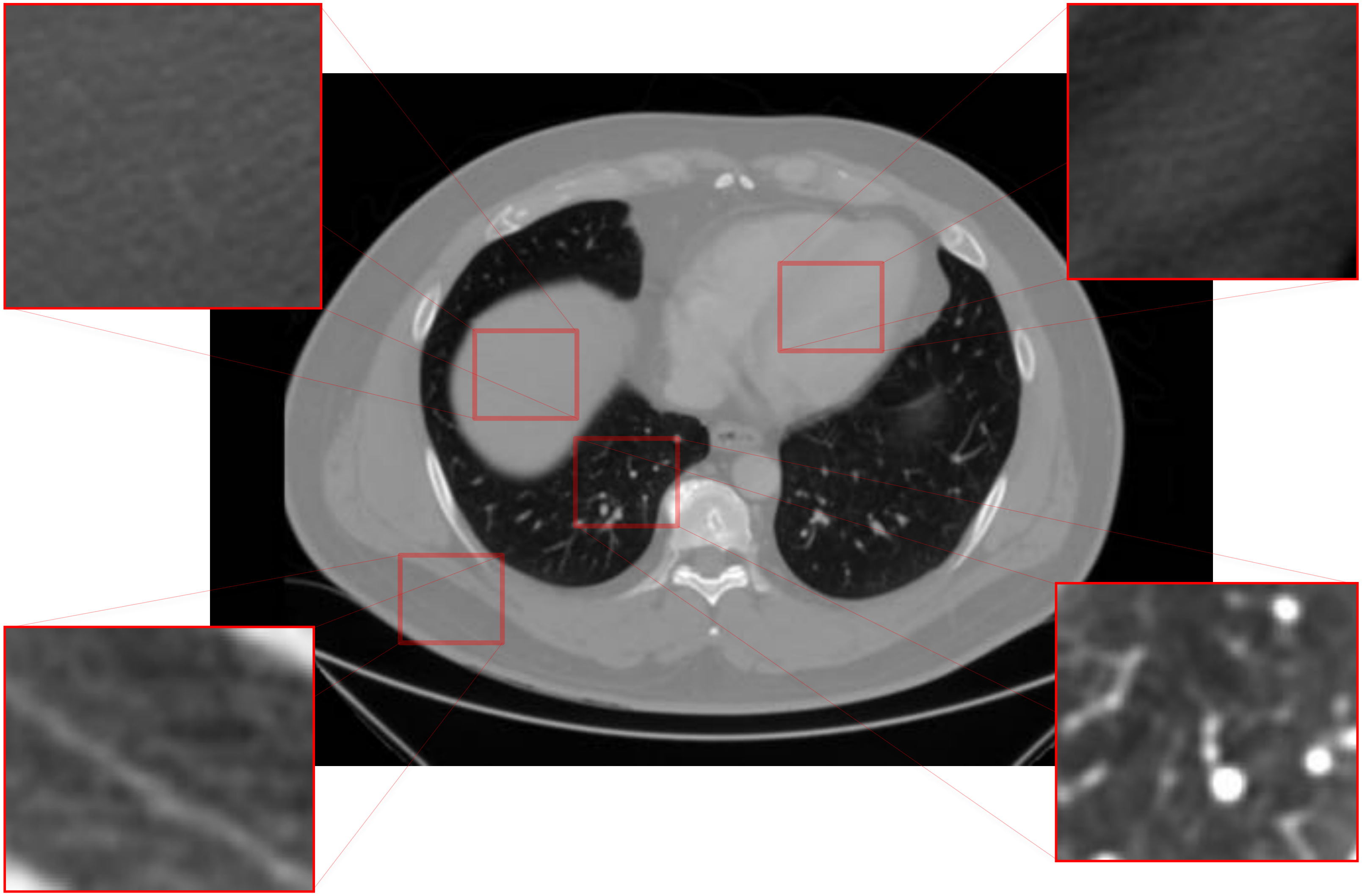
Could you tell the difference between these two rough surfaces with your finger tips?





What are the ways in which you could quantify these textures?

Texture is Challenging to Define



How do you mathematically describe the difference between these sub-images?
Can we classify different tissues based on their fine detail appearances?

First Order Statistical Texture

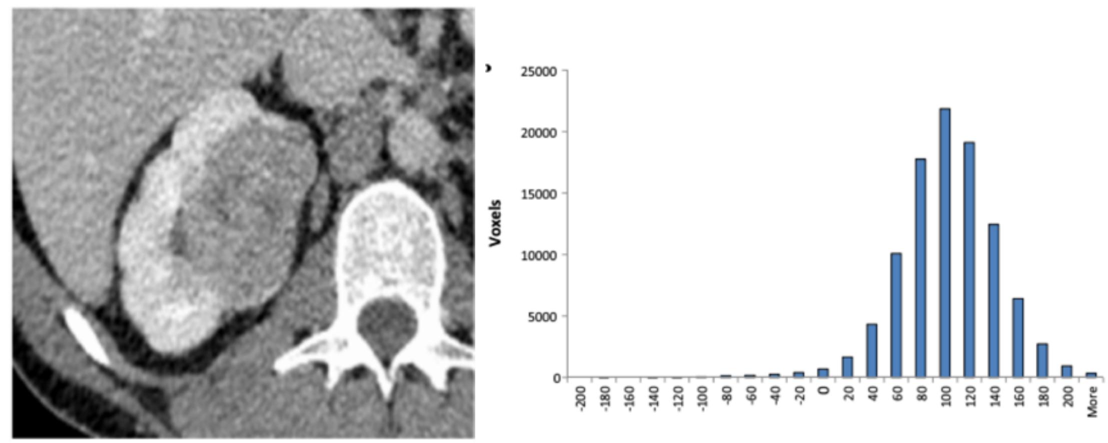
Histogram Analysis

Histogram Analysis

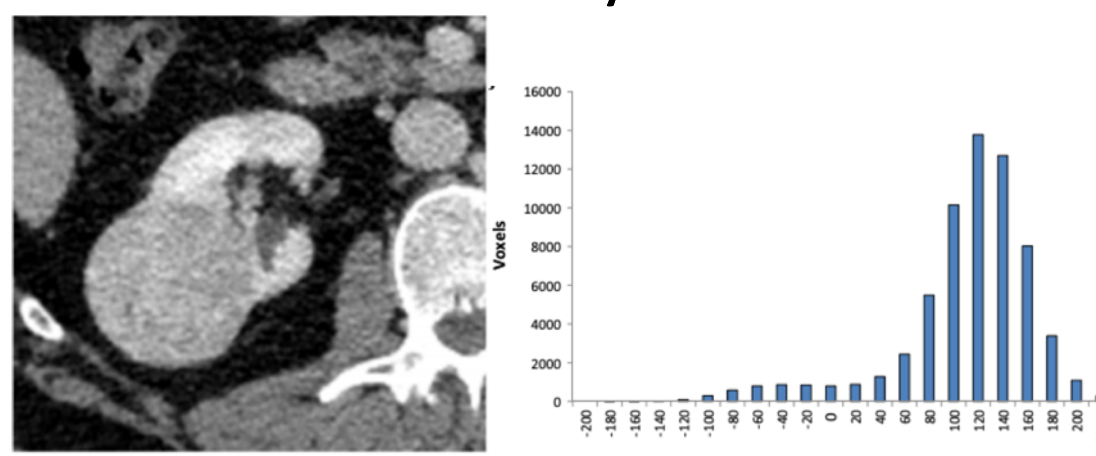
- Mean
- Standard Deviation
- Skewness
- Kurtosis
- Entropy
- Quartiles
- Min/Max

Often useful to mask regions of interest first

Clear cell renal cell carcinoma



Renal oncocytoma



Chen et al 2017

Second Order Statistical Texture

Haralick Texture Features / GLCM

Gray Level Co-occurrence Matrix (GLCM)

Joint Probability Distribution of Pixels with a Specific Spatial Relationship

$$GLCM_{\Delta i,\Delta j}(g_1,g_2)=\sum_{i=x-w}^{x+w}\sum_{j=y-w}^{y+w}\begin{cases}1 & if\ I(i,j)=g_1\ and\ I(i+\Delta i,j+\Delta j)=g_2\\ 0 & otherwise\end{cases}$$

g_1,g_2 *are grayscale values*

Δi=1

Δj=1

j ↑

2	**0**	**3**	**2**	**3**
2	**1**	**3**	$\boxed{g_2\ \mathbf{2}}$	**1**
1	**0**	$\boxed{g_1\ \mathbf{1}}$	**3**	**2**
2	**3**	**2**	**1**	**0**
0	**1**	**2**	**0**	**2**

i →

Moving kernel sub-image

Δi=1, Δj=1, $g\in[0\ldots3]$

g_2 \ g_1	0	1	2	3
0	**1**	**0**	**2**	**0**
1	**0**	**1**	**1**	**2**
2	**0**	$\boxed{\mathbf{3}}$	**0**	**1**
3	**2**	**1**	**2**	**0**

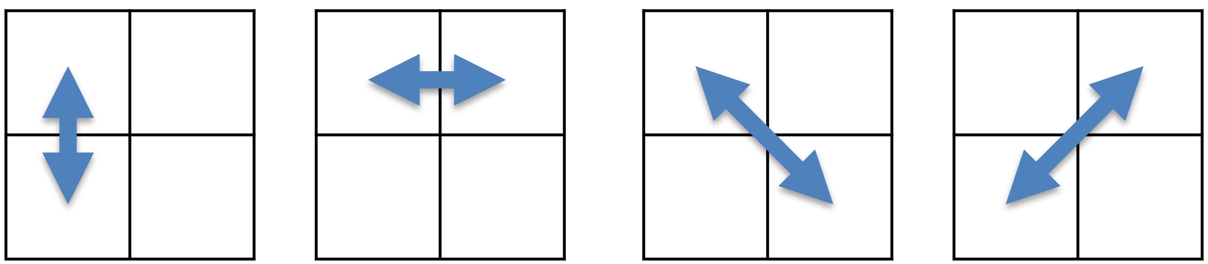
$GLCM_{1,1}(g_1,g_2)$

Directions of Adjacency

$$GLCM_{\Delta i,\Delta j}(g_1,g_2) = \sum_{i=x-w}^{x+w}\sum_{j=y-w}^{y+w}\begin{cases} 1 & \text{if } I(i,j)=\underline{g_1} \text{ and } I(i+\Delta i, j+\Delta j)=\underline{g_2} \\ 0 & \text{otherwise} \end{cases}$$

$$+\sum_{i=x-w}^{x+w}\sum_{j=y-w}^{y+w}\begin{cases} 1 & \text{if } I(i,j)=\underline{g_2} \text{ and } I(i+\Delta i, j+\Delta j)=\underline{g_1} \\ 0 & \text{otherwise} \end{cases}$$

We don't care about the ordering of the two pixels being considered
GLCM can be made symmetric by summing with transpose



(in 2D)

2 or 4 canonical directions in 2D (corresponding to 4- or 8-neighbors)
3, 9, or 13 canonical directions in 3D (corresponding to 6-, 18-, 26-neighbors)

(but not limited to just these directions)

Some Useful Shorthand Notation

$$P(g_1,g_2) = \frac{GLCM_{\Delta i,\Delta j}(g_1,g_2)}{\sum_{G_1=0}^{g_{\max}} \sum_{G_2=0}^{g_{\max}} GLCM_{\Delta i,\Delta j}(G_1,G_2)}$$

Joint Probability Distribution
(convert counts to probabilities)

$$\left.\begin{aligned} P_x(g) &= \sum_{g_2=0}^{g_{\max}} P(g,g_2) \\ P_y(g) &= \sum_{g_1=0}^{g_{\max}} P(g_1,g) \end{aligned}\right\} \text{Marginal Probabilities}$$

$$P_{x+y}(g) = \sum_{\substack{g_1=0 \\ g_1+g_2=g}}^{g_{\max}} \sum_{g_2=0}^{g_{\max}} P(g_1,g_2) \quad g \in [0,1,..,2g_{\max}]$$

Probability Distribution of the
Sum of Two Gray Levels

$$P_{x-y}(g) = \sum_{\substack{g_1=0 \\ |g_1-g_2|=g}}^{g_{\max}} \sum_{g_2=0}^{g_{\max}} P(g_1,g_2) \quad g \in [0,1,..,g_{\max}]$$

Probability Distribution of the
Difference of Two Gray Levels

Note: x and y represent gray levels, not spatial coordinates

Haralick Texture Features

Measures of Variation

Angular Second Moment :

$$f_1 = \sum_{g_1} \sum_{g_2} P(g_1, g_2)^2$$

Image Homogeneity
(noisy image has many small entries)

Contrast :

$$f_2 = \sum_{g=0}^{g_{\max}} g^2 P_{x-y}(g)$$

Variation between neighboring pixels
(larger differences get square law weights)

Correlation :

$$f_3 = \frac{\sum_{g_1} \sum_{g_2} g_1 g_2 P(g_1, g_2) - \mu_{P_x} \mu_{P_y}}{\sigma_{P_x} \sigma_{P_y}}$$

How correlated are pairs of pixel values?
Pearson Correlation Coefficient

$$\rho_{XY} = \frac{E[XY] - E[X]E[Y]}{\sigma_X \sigma_Y}$$

Haralick, IEEE Trans Sys Man and Cybernetics 1973

Haralick Texture Features

Difference Moments

$$\mathit{Sum\ of\ Squares\ (Variance)}:$$

$$f_4 = \sum_{g_1} \sum_{g_2} \left(g_1 - \mu_{P_{xy}}\right)^2 P(g_1, g_2)$$

Increasing weight given to greater gray value differences

$$\mathit{Inverse\ Difference\ Moment}:$$

$$f_5 = \sum_{g_1} \sum_{g_2} \frac{1}{1 + \left(g_1 - g_2\right)^2} P(g_1, g_2)$$

"Homogeneity"
Maximized when neighboring pixels have the same value

Haralick, IEEE Trans Sys Man and Cybernetics 1973

Haralick Texture Features

Sum and Difference of Neighboring Pixels

$$Sum\ Average:$$

$$f_6 = \sum_{g=0}^{2g_{\max}} gP_{x+y}(g)$$

Average sum of gray levels

$$Sum\ Variance:$$

$$f_7 = \sum_{g=0}^{2g_{\max}} \left(g - f_6\right)^2 P_{x+y}(g)$$

Variance of sum of gray levels
(typo in original paper)

$$Difference\ Variance:$$

$$f_{10} = variance\left\{P_{x-y}(g)\right\}$$

Variance of difference of gray levels

Haralick, IEEE Trans Sys Man and Cybernetics 1973

Haralick Texture Features

Entropy (Uncertainty) Measures

$$Sum\ Entropy:$$

$$f_8 = -\sum_{g=0}^{2g_{\max}} P_{x+y}(g)\log\{P_{x+y}(g)\}$$

Uniform (flat) distribution of sum of gray levels has maximum entropy

$$Difference\ Entropy:$$

$$f_{11} = -\sum_{g=0}^{2g_{\max}} P_{x-y}(g)\log\{P_{x-y}(g)\}$$

Uniform (flat) distribution of difference of gray levels has maximum entropy

$$Entropy:$$

$$f_9 = -\sum_{g_1}\sum_{g_2} P(g_1,g_2)\log\{P(g_1,g_2)\}$$

Uniform (flat) joint distribution of gray levels has maximum joint entropy

Haralick, IEEE Trans Sys Man and Cybernetics 1973

Haralick Texture Features

Information Theoretic Measures

$$HXY = -\sum_{g_1}\sum_{g_2} P(g_1,g_2)\log\left\{P(g_1,g_2)\right\}$$

$$HXY1 = -\sum_{g_1}\sum_{g_2} P(g_1,g_2)\log\left\{P_x(g_1)P_y(g_2)\right\}$$

$$HXY2 = -\sum_{g_1}\sum_{g_2} P_x(g_1)P_y(g_2)\log\left\{P_x(g_1)P_y(g_2)\right\}$$

Information Measure of Correlation 1:

$$f_{12} = \frac{HXY - HXY1}{\max\left\{HX, HY\right\}}$$

Normalized mutual information

Information Measure of Correlation 2:

$$f_{13} = \sqrt{1 - e^{-2(HXY2 - HXY)}}$$

Difference between joint entropy and joint entropy assuming independence

Haralick, IEEE Trans Sys Man and Cybernetics 1973

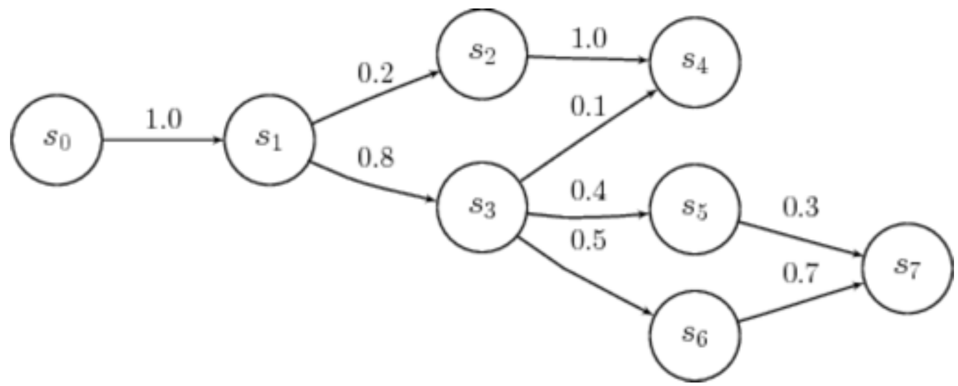
Haralick Texture Features

Markov Chain

$$Q(g_1, g_2) = \sum_g \frac{P(g_1, g)}{P_x(g_1)} \cdot \frac{P(g_2, g)}{P_y(g)}$$

$$= \sum_g P(g | g_1) \cdot P(g_2 | g)$$

Q is a transition matrix for a Markov chain of neighboring pixel gray levels



$$Maximal\ Correlation\ Coefficient:$$

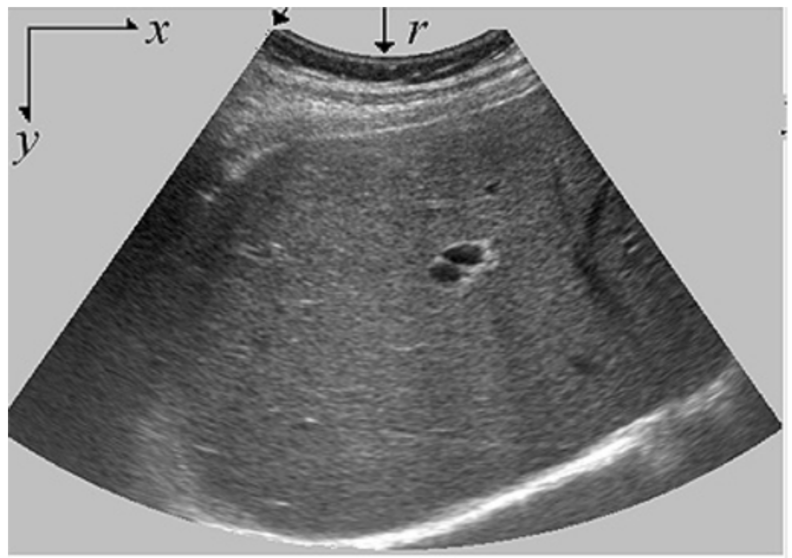
$$f_{14} = \sqrt{2^{nd}\ largest\ eignevalue\ of\ Q}$$

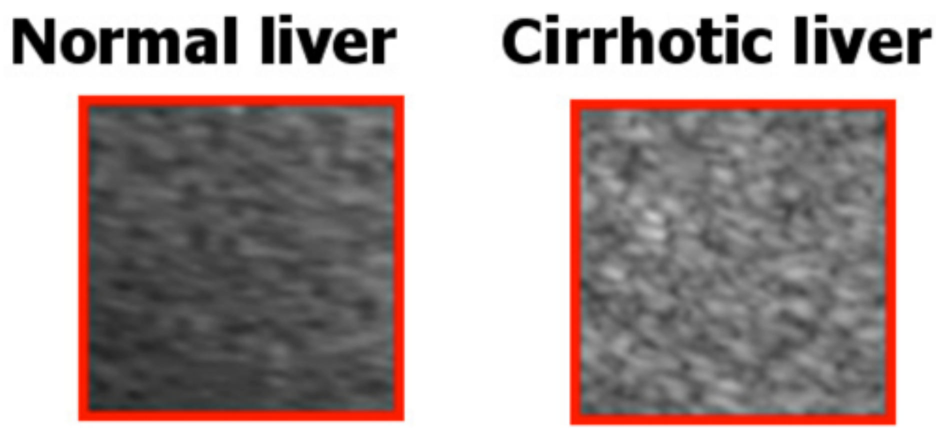
Relates to how fast the Markov chain converges

What are the pros and cons of these 14 texture features?

Haralick, IEEE Trans Sys Man and Cybernetics 1973

Texture Classification





63 patients w/chronic hepatitis B/C → adaptive filtering of speckle, nonlinear attenuation → cirrhosis stage correlated with texture entropy; earliest stages hardest to detect

Pao et al, SIIM 2011

Transform Analysis

Gabor, Wavelets, etc.

Global vs. Local Image Transforms

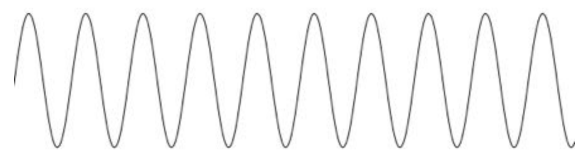
- Transforms in general:

$$F(u,v) = \sum_{x=0}^{N}\sum_{y=0}^{N} f(x,y)g(x,y,u,v) \qquad F(u,v) = \int\int f(x,y)g(x,y,u,v)\,dx\,dy$$

$f(\ )$ is the function of interest $\quad$ $g(\ )$ is the kernel specific to the transform

- Global Transform:

$f(\ )$ can be decomposed as a sum of waves of infinite extent



- Local Transform

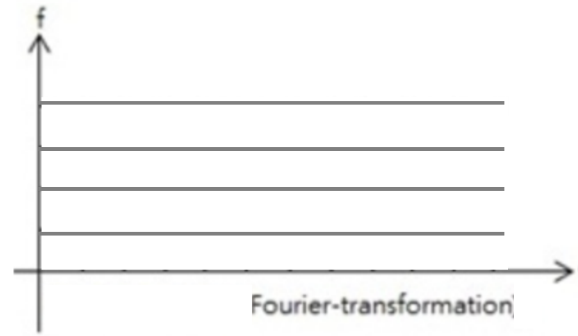
$f(\ )$ can be decomposed as a sum of waves of finite extent

- Heisenberg-Gabor Limit: $\sigma_t \cdot \sigma_f \geq \dfrac{1}{4\pi}$

Global vs. Local Image Transforms

Fourier Transform

$$\mathrm{F}(\omega) = \int_{-\infty}^{\infty} f(x) \cdot e^{-i\omega x} dx \qquad \omega = 2\pi u$$

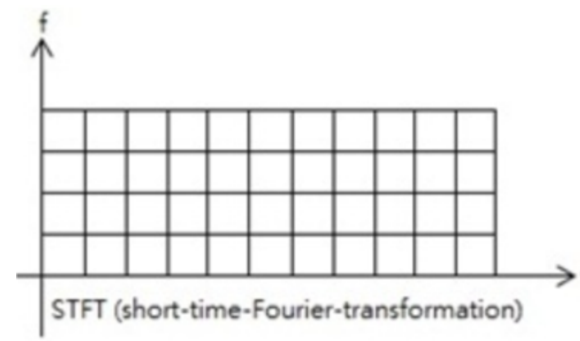


Short Time Fourier Transform

$$F_{STFT}(\omega, \tau) = \int_{-\infty}^{\infty} f(x) \cdot w(x - \tau) \cdot e^{-i\omega x} dx$$

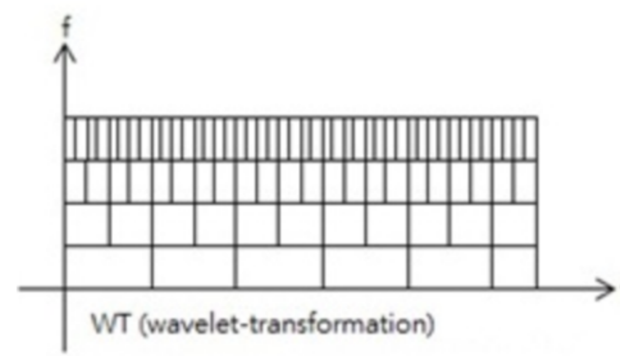
Gabor Transform

$$G(\omega, \tau) = \int_{-\infty}^{\infty} f(x) \cdot e^{-\pi\alpha(x-\tau)^2} \cdot e^{-i\omega x} dx$$



Wavelet Transform

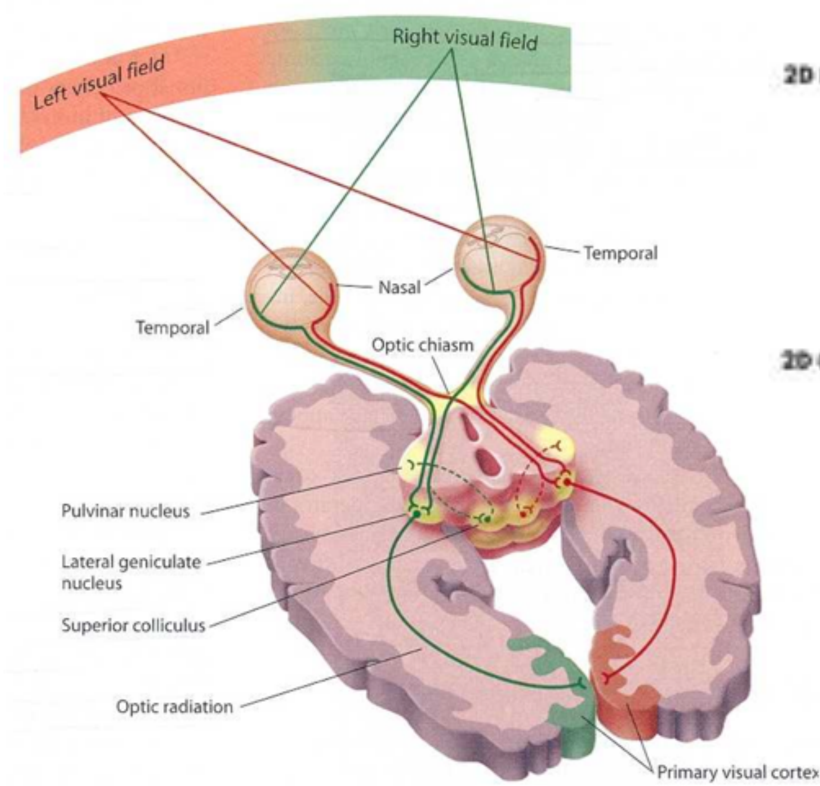
$$F_w(a, b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} f(x) \cdot \bar{\psi}\left(\frac{x - b}{a}\right) dx$$

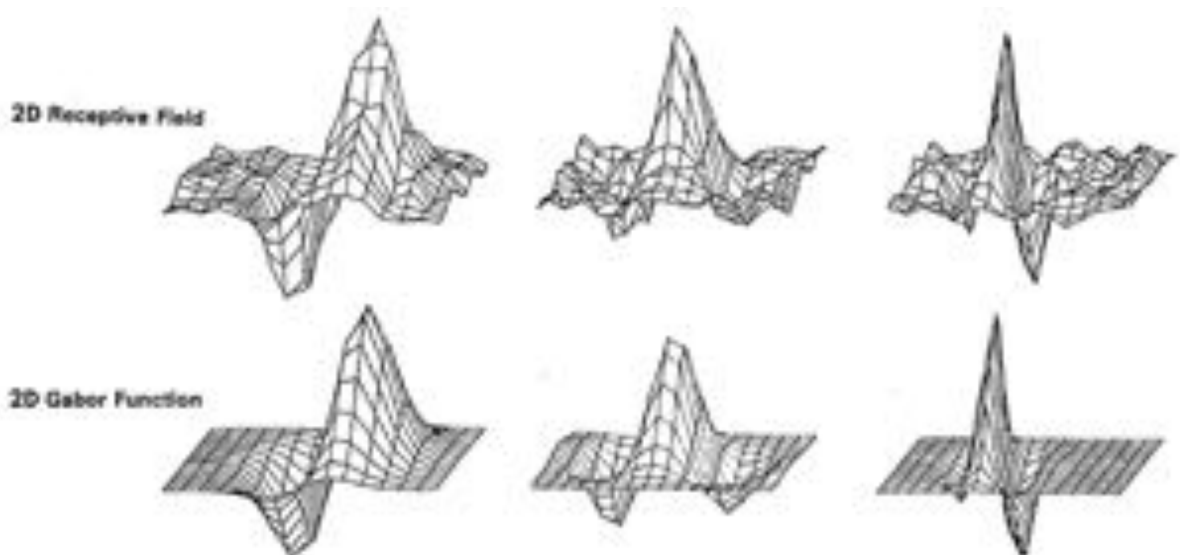


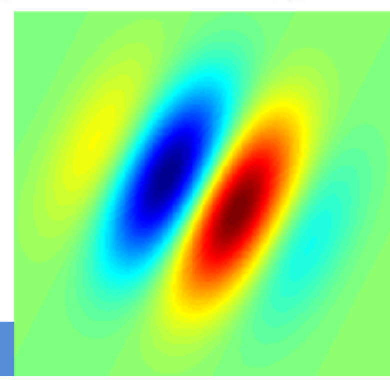
Gabor Filters

as an Approximation to Biological Vision

- Gabor kernel closely matches receptive field profiles in cat striate cortex
    - Stimulus alternates excitatory/inhibitory effect







Daugman, J Opt Soc Am 1985

Gabor Kernel

$$g(x,y) = e^{i\left(\frac{2\pi x'}{\lambda}+\psi\right)} e^{-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)}$$

$$= \cos\left(\frac{2\pi x'}{\lambda}+\psi\right) e^{-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)} + i\sin\left(\frac{2\pi x'}{\lambda}+\psi\right) e^{-\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)}$$

$$where \quad x' = x\cos\theta - y\sin\theta \quad and \quad y' = x\sin\theta + y\cos\theta$$

x and y are spatial coordinates
x' and y' are rotated spatial coordinates

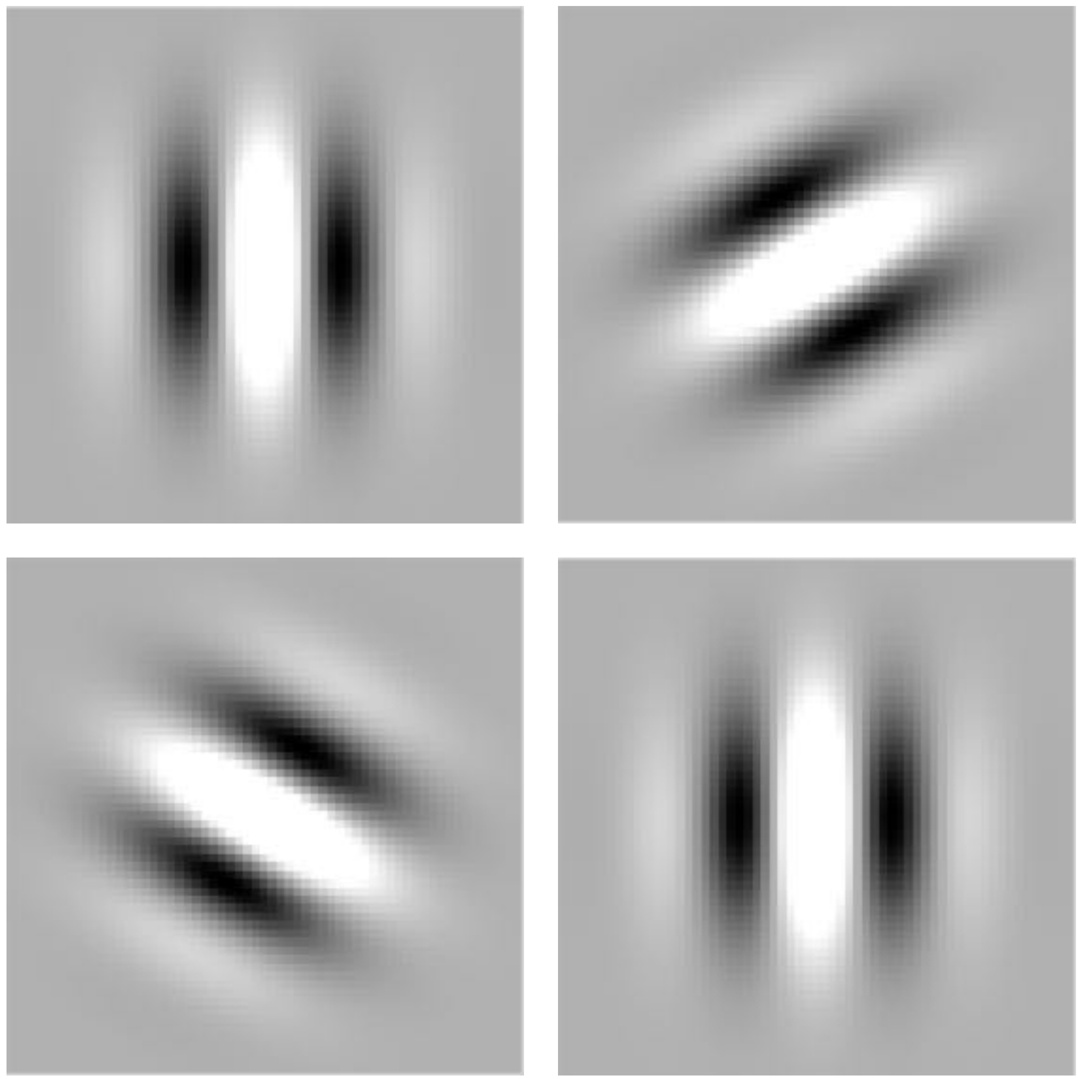
λ is wavelength
ψ is phase

$\sigma_{x,}$, σ_y are sizes of Gaussian envelope

$$e^{ix} = \cos x + i\sin x$$

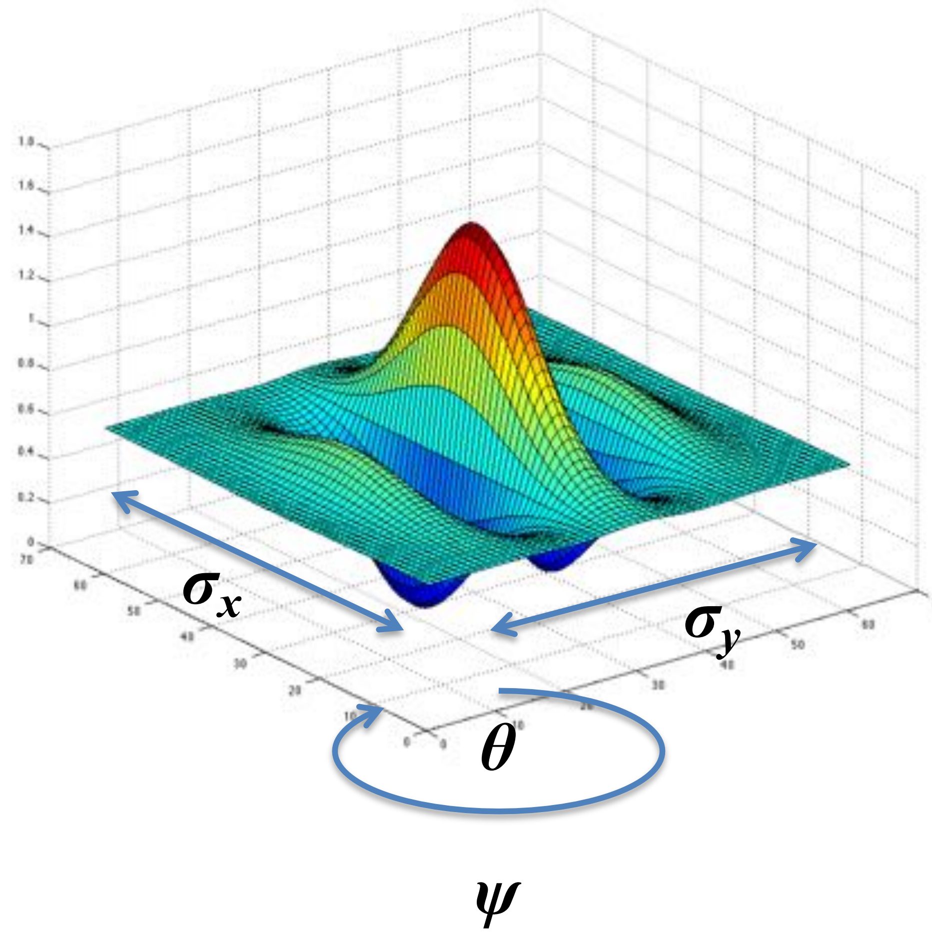
Different formula than 1st and 2nd derivatives of Gaussian but it can have a similar shape

Gabor kernels are localized "chirps" of frequency at various angles



Variable Filter Parameters

- Scale (σ_x,σ_y)
- Orientation (θ)
- Phase: (ψ)
  - even vs. odd symmetry
- Aspect Ratio (σ_x/σ_y)



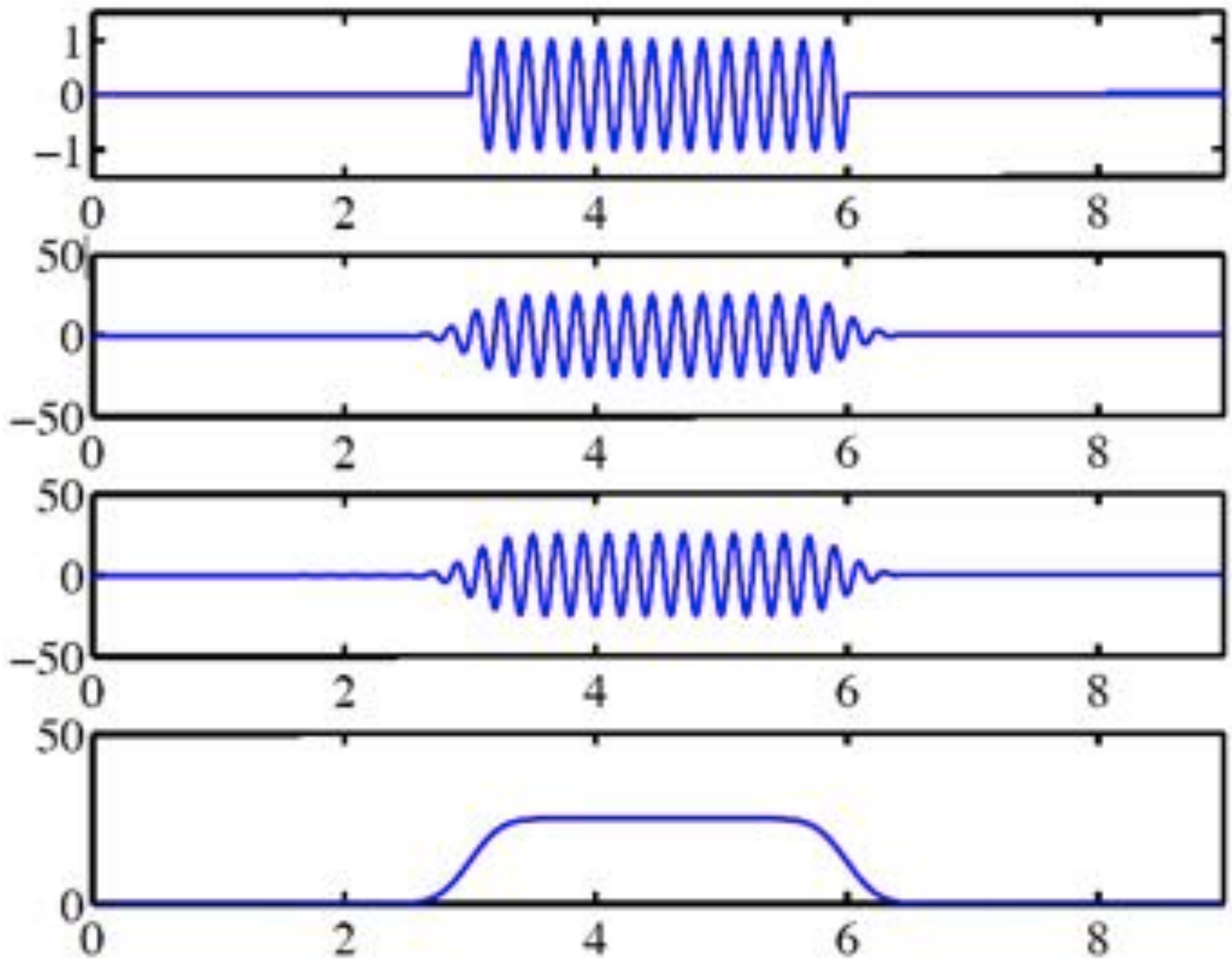
Quadrature Phase

Input

Output (cosine)
real component

Output (sine)
imaginary component

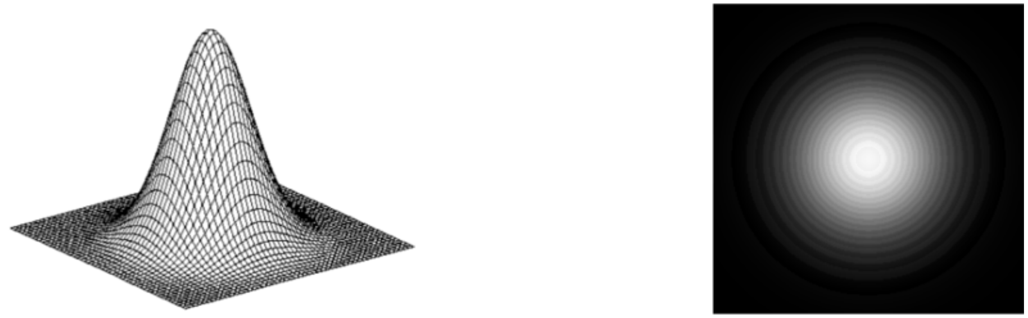
Output
complex magnitude



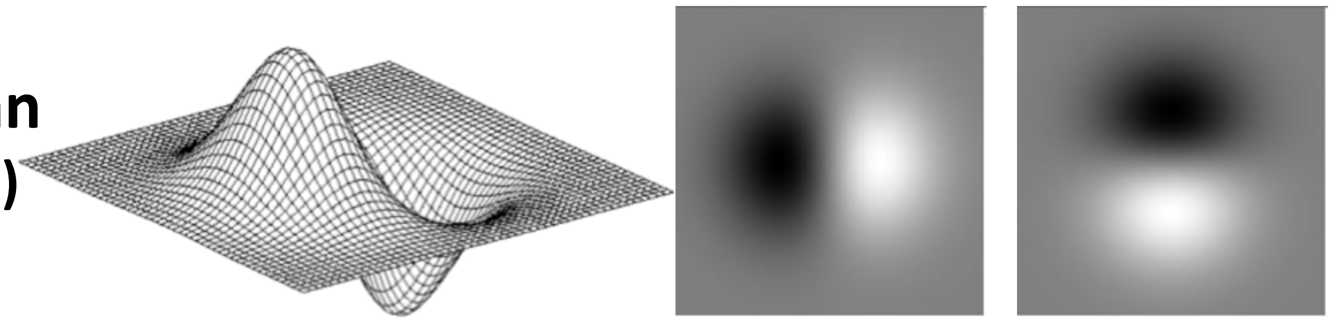
Movellan, Tutorial on Gabor filters

Revisiting Some Filter Kernels

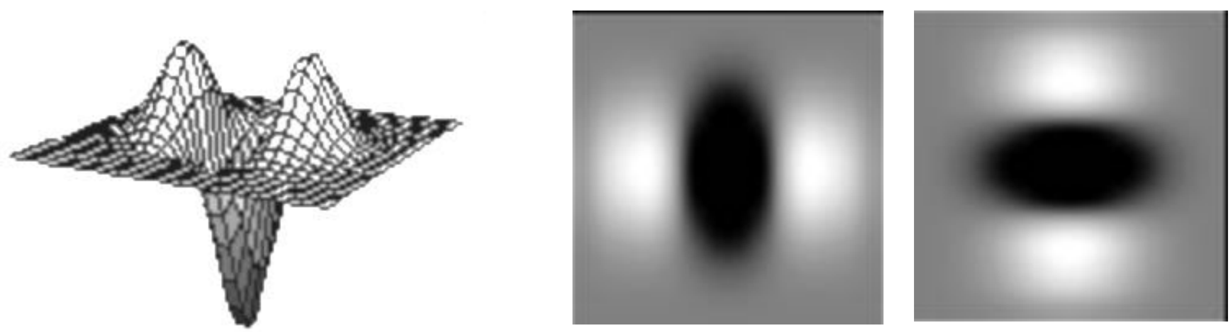
Gaussian
(in all directions)



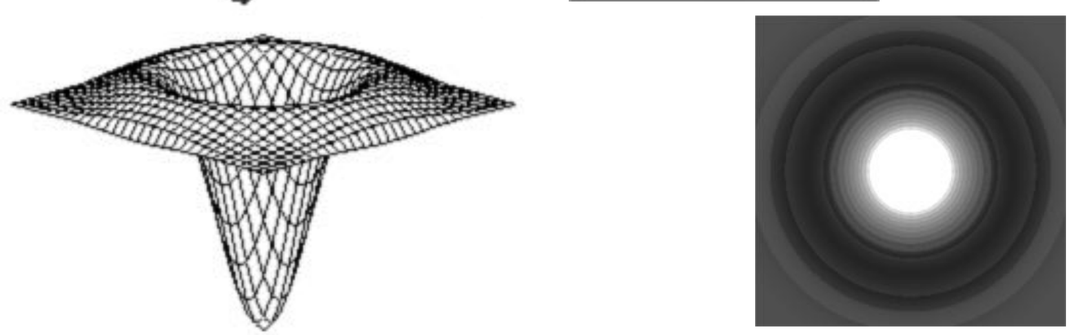
1st Derivative of Gaussian
(Gaussian in other directions)



2nd Derivative of Gaussian
(Gaussian in other direction)

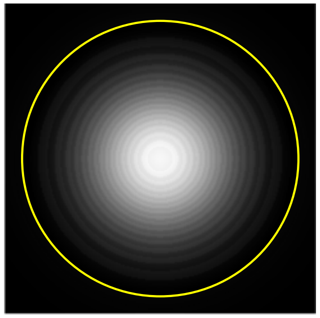


Laplacian of Gaussian
(sum of 2nd derivatives in all directions)

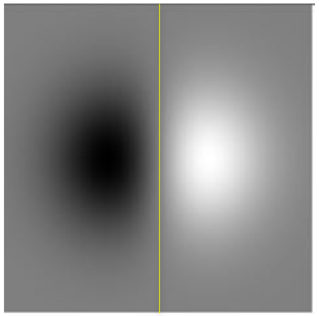


Filter Bank Composition

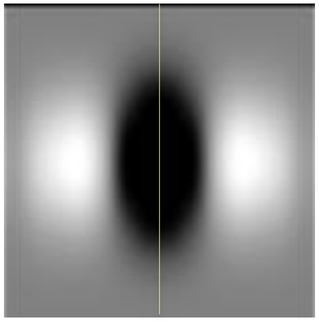
- Radially symmetric filters
  - Difference of Gaussian
  - Laplacian of Gaussian
  - Gaussian



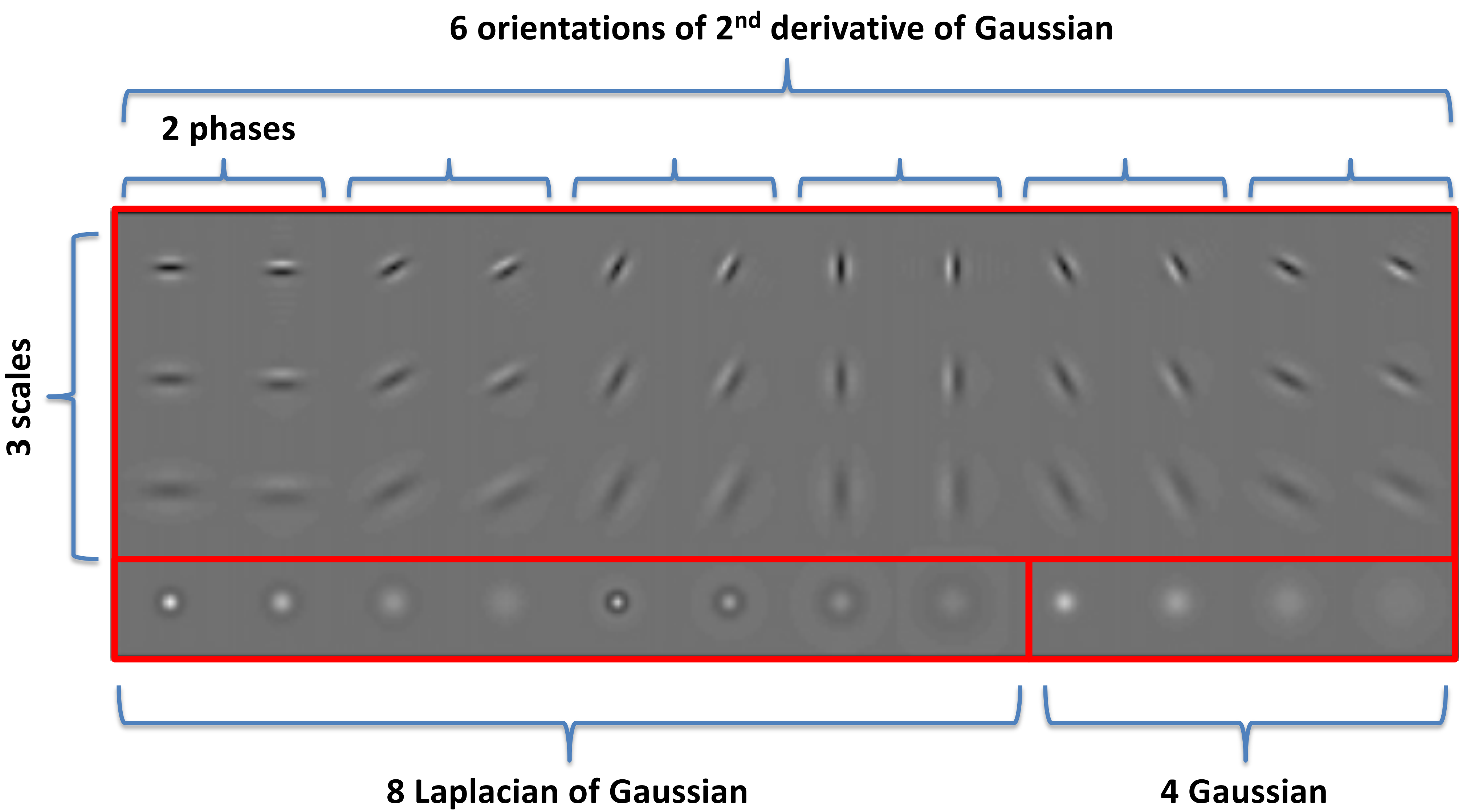
- Oriented odd-symmetric filters
  - Derivative of Gaussian
  - Gabor sine component



- Oriented even-symmetric filters
  - Second derivative of Gaussian
  - Gabor cosine component



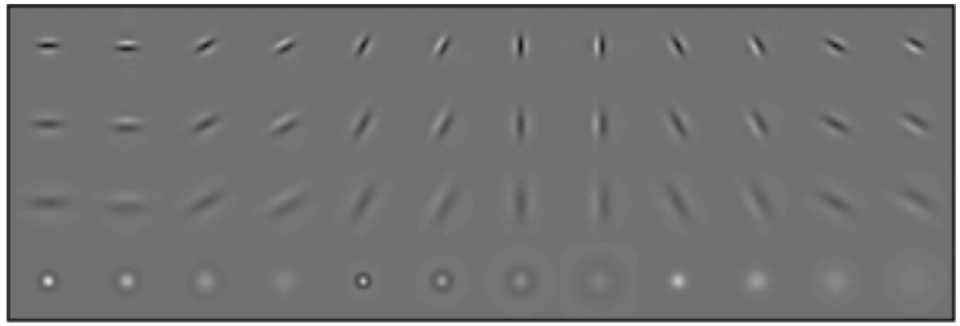
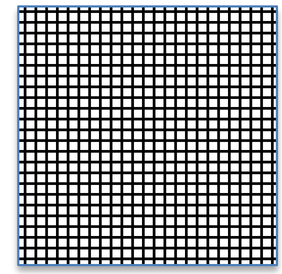
Example Filter Bank



Total 48 filters

Leung and Malik, IJCV 2001

Textons: Elementary Units of Texture

 =  

$\mathbf{r}$

48 dimensional output

$\mathbf{F}$

48 x 625 matrix

48 oriented filters

(6 orientations x 3 scales x 2 phases + 8 Laplacian + 4 Gaussian)

$\mathbf{p}$

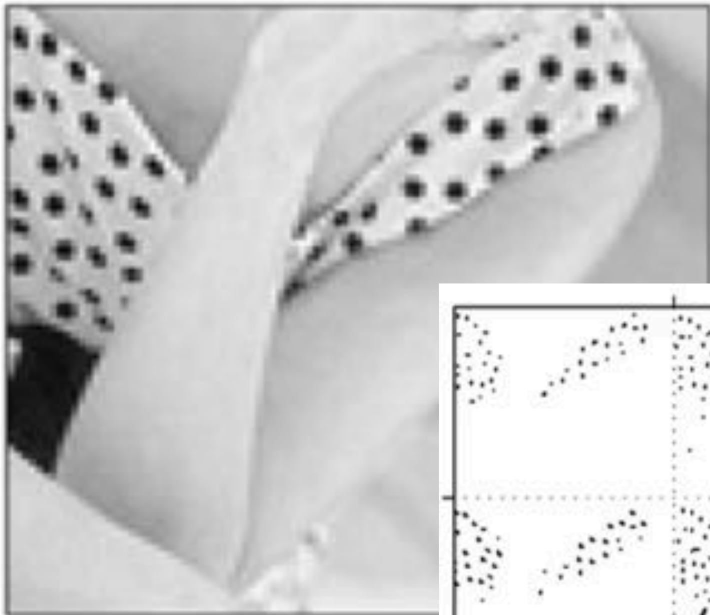
625 pixel image patch (25x25) linearized into a column vector

$$\mathbf{r} = \mathbf{F}\mathbf{p}$$

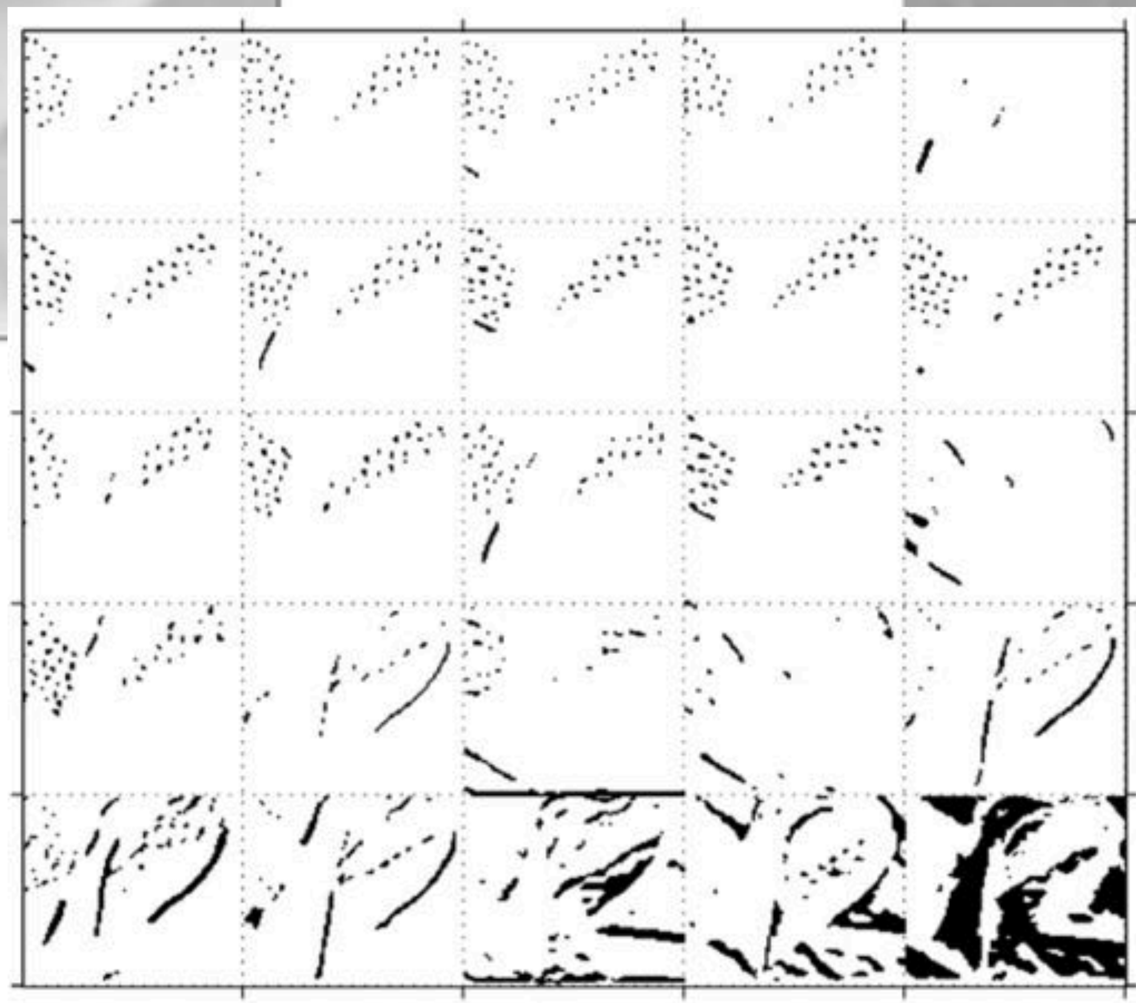
Over an image (or set of images), the vector of filter bank outputs, $\mathbf{r}$, can be clustered by *K*-means

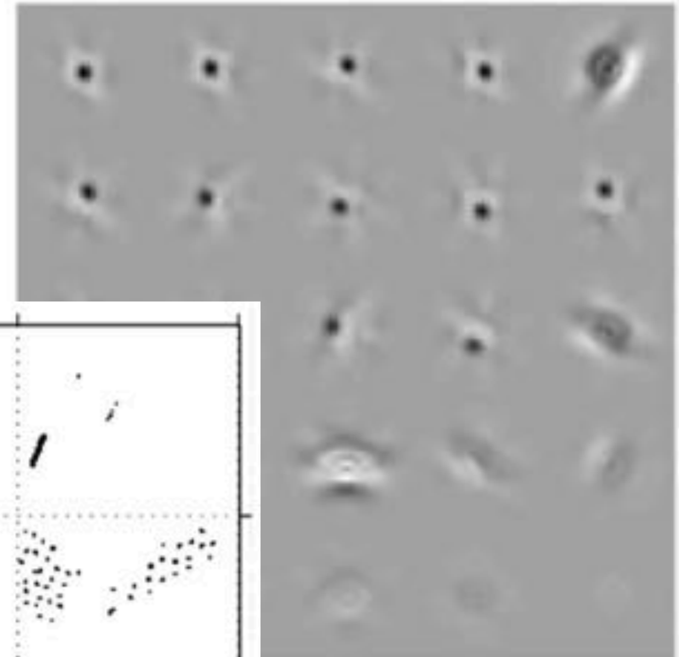
We can use the pseudoinverse of $\mathbf{F}$ (in a least squares sense) to go from k cluster centers to prototypical image patches, which are called *textons* (analogous to phonemes in speech)

Examples of Textons



Original image





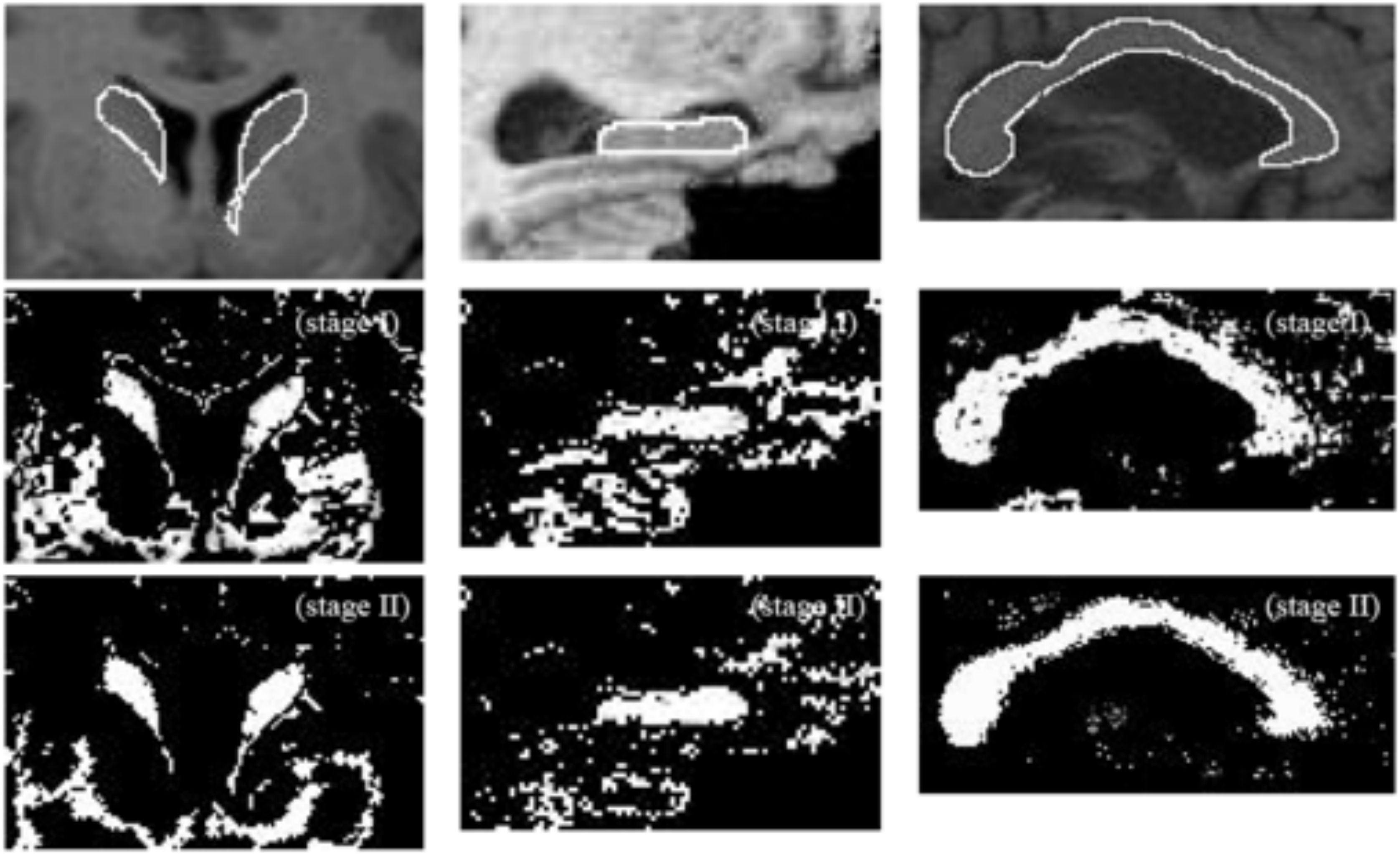
Top 25 textons

Texture Segmentation

True Structure

Stage I Results

Stage II Results



Caudate Nucleus

Hippocampus

Corpus Callosum

T1-weighted MR → Haralick,Gabor,etc. → neural net → ~90% accuracy

Pitiot et al, WCCI-IJCNN 2002

Structural Analysis

Fractals

Fractal Dimension and Lacunarity

Fractal Dimension

$$N = K\varepsilon^{-D}$$

$$\ln N = -D\ln\varepsilon + \ln K$$

Lacunarity

$$\lambda_\varepsilon = \left(\frac{\sigma_\varepsilon}{\mu_\varepsilon}\right)^2$$

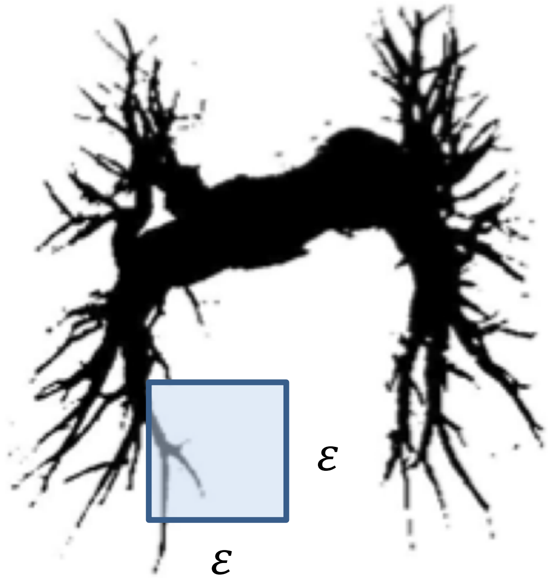
$$\ln\lambda_\varepsilon = m\ln\varepsilon + b$$

N = number of non-empty boxes
ε = size of boxes
K = constant
D = fractal dimension

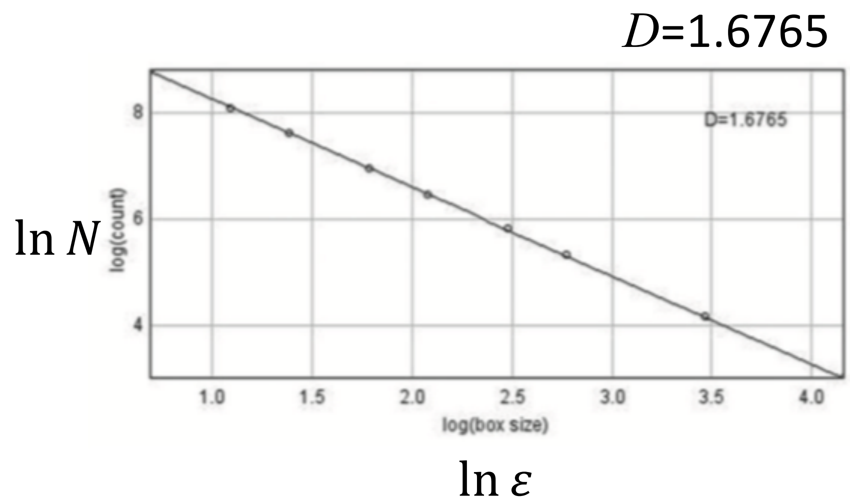
λ_ε = lacunarity
σ_ϵ = st dev of # foreground pixels
μ_ϵ = mean of # foreground pixels
m = slope of lacunarity
b = intercept

Fractal Dimension of Arterial Tree
Pulmonary Hypertension

Pulmonary Artery
Pulmonary Hypertension (PH)



Binary View of
Coronal Projection

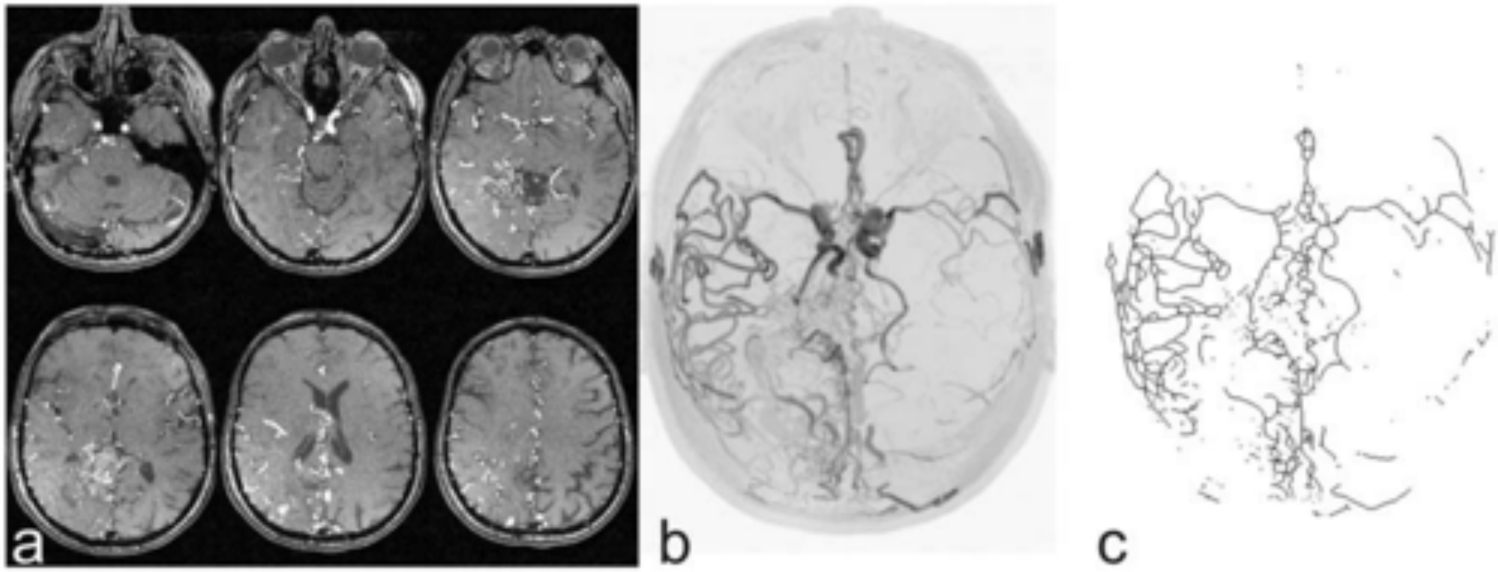


PH: D = 1.64 ± 0.04

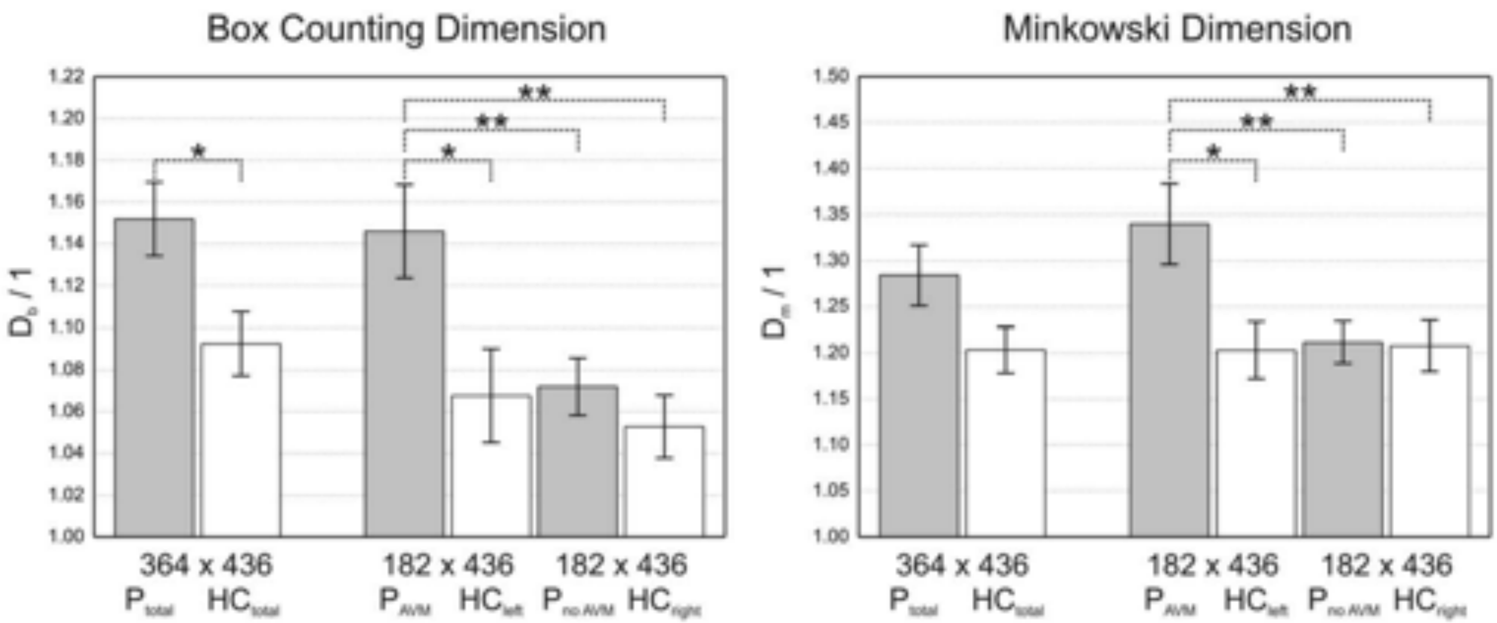
No PH: D = 1.54 ± 0.04

Haitao et al 2011

Fractal Dimension and Vessel Complexity



TOF MRI **K-means** **Medial Axis**



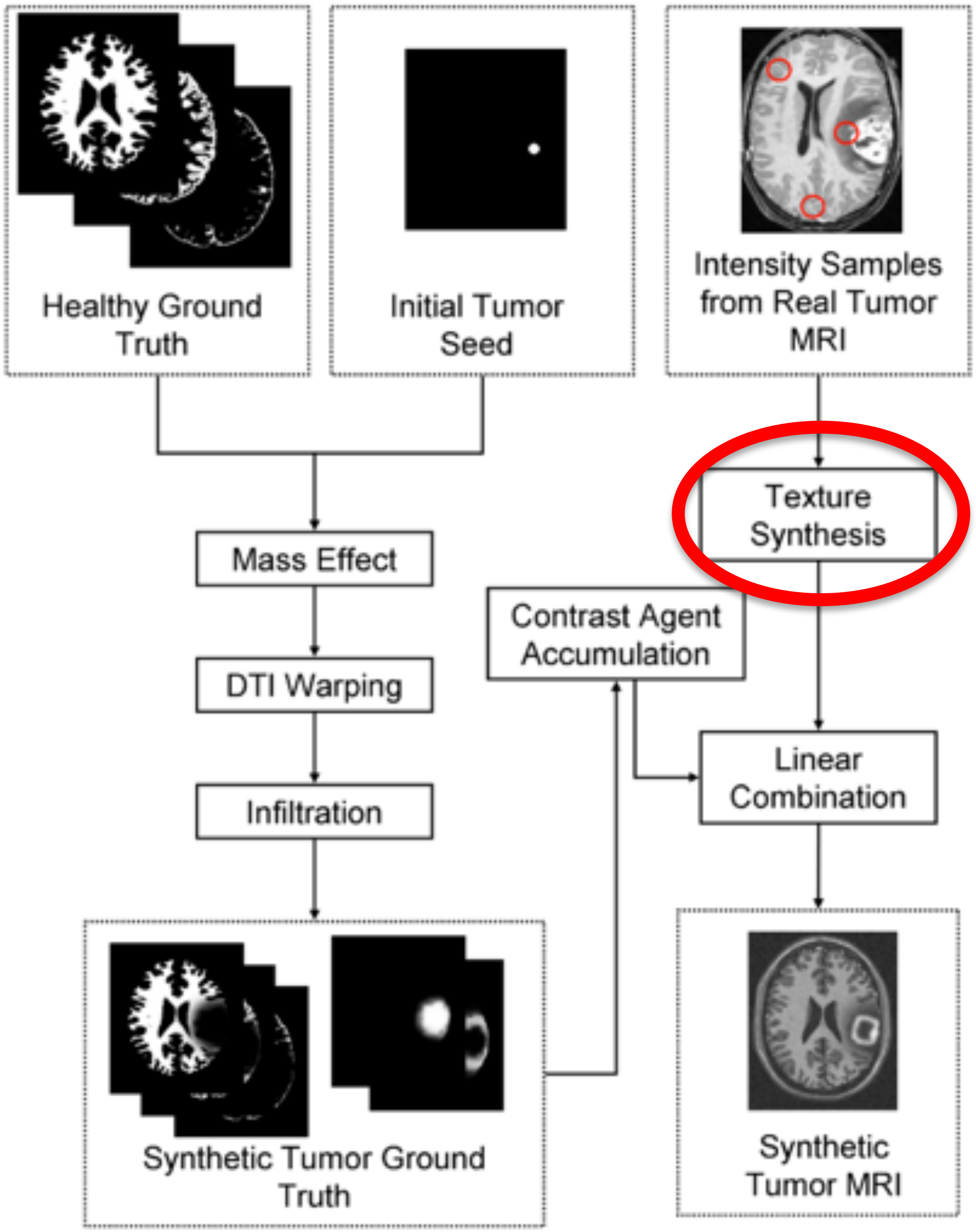
Arteriovenous Malformation

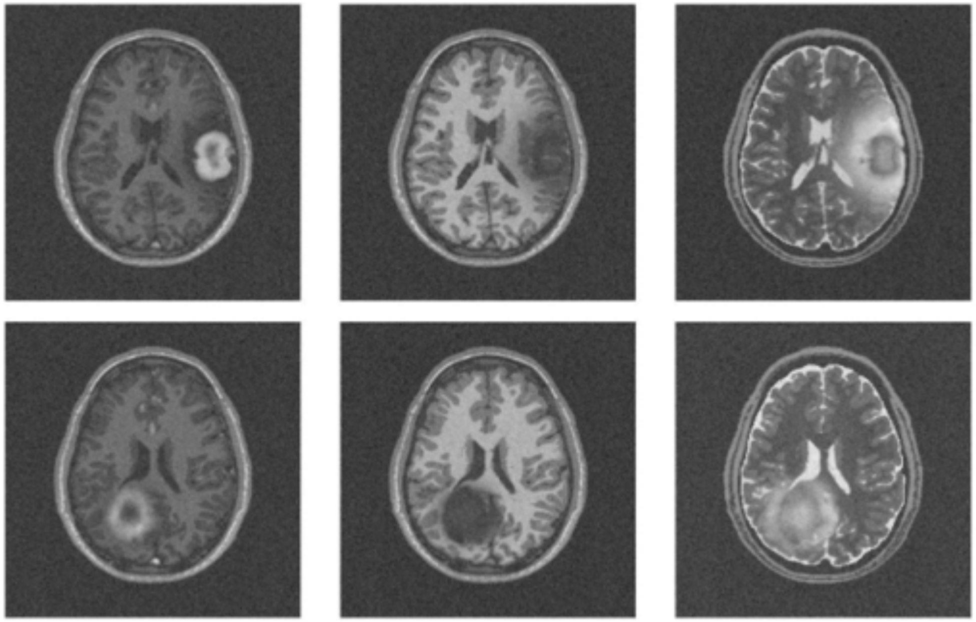
Healthy Control

Reishofer et al, 2012

Applications

Texture Synthesis



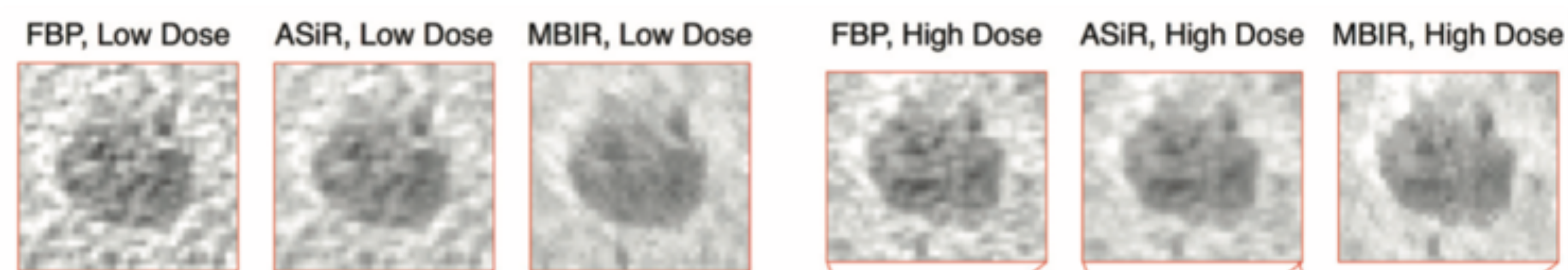


Texture model: Markov Random Field

Prastawa et al, Med Image Anal 2009

Caveat Emptor

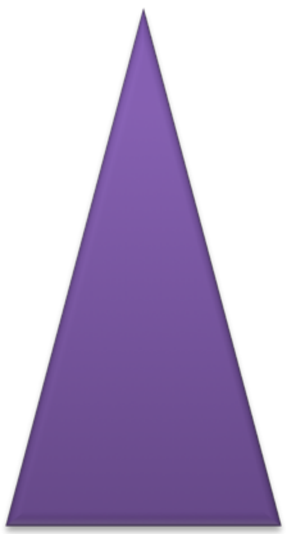
- Factors that affect texture that are not related to pathology
  - Image reconstruction algorithm
  - Scanner manufacturer
  - Contrast
  - Low-dose acquisition
  - Motion
  - Other imaging acquisition artifacts



Solomon et al 2016

How does Texture Fit in with Other Image Features?

Local



Global

- Point-wise Image Features
  - Intensity, Location
- Texture Features
  - Histogram Analysis
  - Haralick/GDCM
  - Gabor
  - Fractal
- Shape Features
  - Geometric Measures, Medial Axis
- Global Features
  - Histogram, Fourier Transform

Image features reduce the dimensionality of a full image (millions of pixels) down to a feature vector (tens to hundreds)

What does it mean for me?

- Topics:
  - Defining Texture
  - 1st and 2nd Order Statistical Features
  - Transforms
  - Fractal Analysis
  - Applications
- Textures are a rich set of features useful for tissue classification and other clinical tasks
- Fill a niche between pixels and shapes

Next Lecture:

Image Registration