CAP6411
Computer Vision Systems
Lecture 3

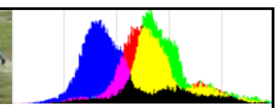
Alper Yilmaz
Office: CSB 250
Email: yilmaz@cs.ucf.edu
Web: http://www.cs.ucf.edu/courses/**cap6411/cap6411**/spring2006

Recap
Object Representations

- Shape of objects
  - Point Representations
  - Primitive Geometric Shapes
  - Object silhouette and contour
  - Articulated shape models
  - Skeletal models

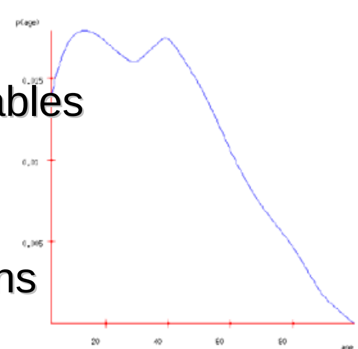
Recap
Object Representations

- Appearance of objects
  - PDFs
  - Templates
  - Multi-view appearance methods



Recap
Probability Distributions

- Probability distribution
- Probability density
- Independence of variables
- Marginal distributions
  - Projection onto axis
- Conditional distributions
- Bayes' rule



Recap
Eigenspace Decomposition

- Encodes different views of an object
- Selection of few eigenvectors is enough
- Highlights differences within the class

Visual Features

Properties

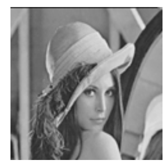
- Uniqueness
- Related to the object representation
  - Color for histogram based representation
  - Edges for contour based representation
- Combination of features improve performance of vision algorithms

Typical Visual Features

- Color
- Edges
- Optical flow
- Texture

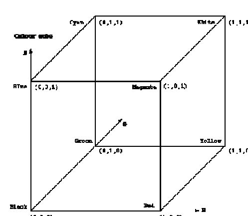
Color Feature

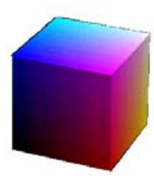
- Multiple bands
  - RGB, HSV, etc.
- Single band
  - Binary
  - Gray scale, heat index, etc.

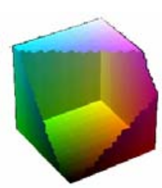


RGB Color Space (Red, Green, Blue)

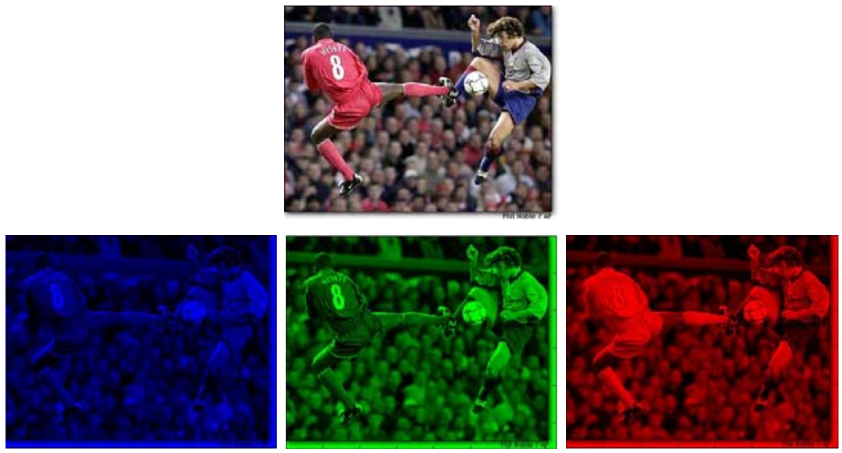
- Color cube
- Range from 0-255
- Any color is specified by r, g, b triple.
- The diagonal line of the cube represents all the grays
- Simply a linear scaling of a unit color cube
- Lies within our perceptual space
- Represents fewer colors than we can see.







RGB



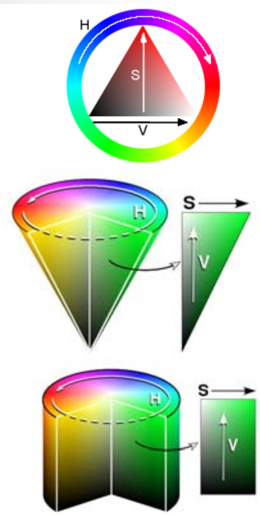
HSV Color Space (Hue, Saturation, Value)

- A common alternative to RGB
- Hue is the color type
  - Ranges from 0→360
- Saturation is amount of gray in color
  - Ranges from 0→100%
- Value is the brightness
  - Ranges from 0→100%

$$V = MAX \qquad \max = MAX(R,G,B)$$

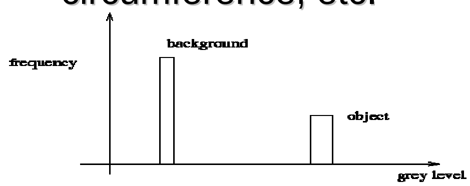
$$S = \frac{MAX - MIN}{MAX} \qquad \min = MIN(R,G,B)$$

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

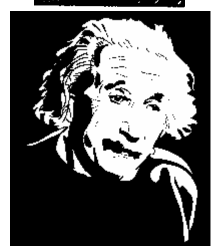


Single Band Images
Binary

- Two possible intensity values
- Produced by thresholding
- Used in shape analysis
  - Elongation, area, compactness, circumference, etc.



Histogram of a binary image

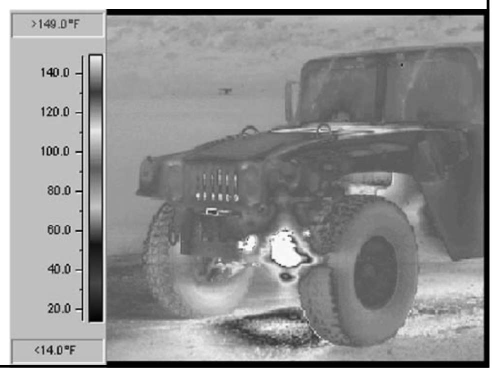


Single Band Images
Gray Level Image

- Only colors are shades of gray
- Less storage space
- Are sufficient to many tasks
  - Many low-level vision algorithms use gray level images

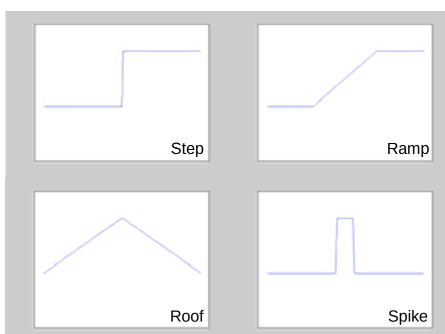
Single Band Images
Heat Index

- Expensive sensors
- Used in military
- Black→Cold
- White→Hot



Edge Feature

- Discontinuity of intensities in the image
- Edge models
  - Step
  - Roof
  - Ramp
  - Spike

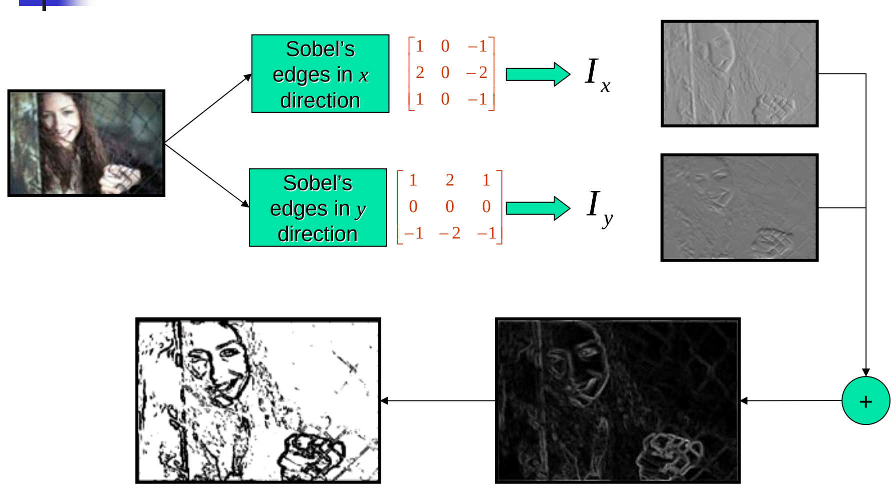


Edge Feature

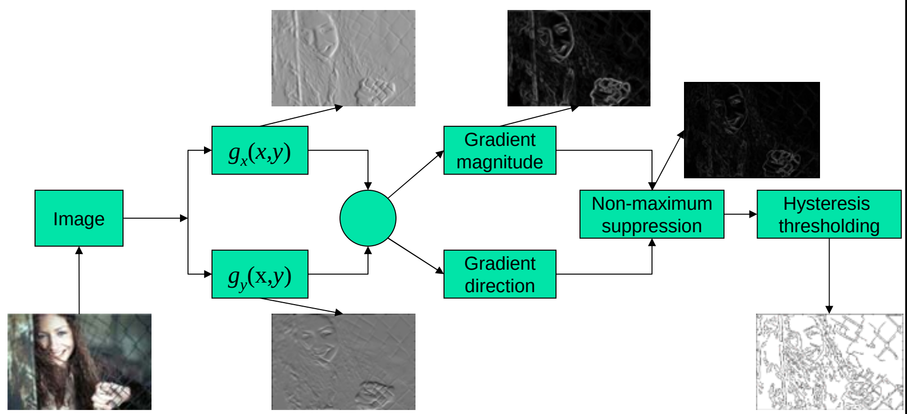
- Less sensitive to illumination changes
- Commonly used in contour based image representation
- Extracted from gray level images
- Used in context of:
  - Object recognition
  - Tracking
  - Image retrieval

Bowyer, K., Kranenburg, C., and Dougherty, S. 2001. Edge detector evaluation using empirical ROC curve. CVIU 10, 77–103.

Sobel Edge Detector



Canny Edge Detector



Optical Flow Feature

- Dense field of displacement vectors
  - Translation of each pixel in a region
- Computed from a set of frames
- Computed from brightness constancy constraint
  - Intensity of a moving pixel does not change over-time

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t)$$

Common Uses

- Motion based segmentation
- Object tracking
- Video indexing and retrieval
- Correction of camera jitter
- Image alignment
- Structure from motion
- Video compression

Optical Flow

- Flow vector in image space (2D)
- Taylor series expansion of right side around Δt

$$I(x,y,t) = \boxed{I(x+\Delta x, y+\Delta y, t+\Delta t)}$$

$$\boxed{I(x+\Delta x, y+\Delta y, t+\Delta t)} = I(x,y,t) + \Delta x \frac{\partial I}{\partial x} + \Delta y \frac{\partial I}{\partial y} + \Delta t \frac{\partial I}{\partial t}$$

$$I(x,y,t) = I(x,y,t) + \Delta x I_x + \Delta y I_y + \Delta t I_t$$

$$0 = \Delta x I_x + \Delta y I_y + \Delta t I_t \Rightarrow 0 = \frac{\Delta x}{\Delta t} I_x + \frac{\Delta y}{\Delta t} I_y + I_t$$

Optical Flow

$$0 = \frac{\Delta x}{\Delta t} I_x + \frac{\Delta y}{\Delta t} I_y + I_t$$

$$\boxed{u = \frac{\Delta x}{\Delta t}} \qquad \boxed{v = \frac{\Delta y}{\Delta t}}$$

$$\boxed{uI_x + vI_y + I_t = 0}$$

Brightness constancy equation

$$v = u\frac{I_x}{I_y} + \frac{I_t}{I_y}$$

Equation of a line in (u,v) space

Computing Optical Flow Lucas & Kanade

- Line fitting
  - Define an energy functional
  - Take derivatives equate it to 0
  - Rearrange and construct an observation matrix

$$E = \sum (uI_x + vI_y + I_t)^2$$

$$\frac{\partial E}{\partial u} = \sum 2I_x(uI_x + vI_y + I_t) = 0$$

$$\frac{\partial E}{\partial v} = \sum 2I_y(uI_x + vI_y + I_t) = 0$$

Computing Optical Flow Lucas & Kanade

$$\boxed{\frac{\partial E}{\partial u} = \sum 2 I_x (u I_x + v I_y + I_t) = 0}$$

$$\boxed{\frac{\partial E}{\partial v} = \sum 2 I_y (u I_x + v I_y + I_t) = 0}$$

$$\sum u I_x^2 + \sum v I_x I_y + \sum I_x I_t = 0$$
$$u \sum I_x^2 + v \sum I_x I_y = -\sum I_x I_t$$
$$\left[\sum I_x^2 \quad \sum I_x I_y\right] \begin{bmatrix} u \\ v \end{bmatrix} = -\sum I_x I_t$$

$$\sum u I_x I_y + \sum v I_y^2 + \sum I_y I_t = 0$$
$$u \sum I_x I_y + v \sum I_y^2 = -\sum I_y I_t$$
$$\left[\sum I_x I_y \quad \sum I_y^2\right] \begin{bmatrix} u \\ v \end{bmatrix} = -\sum I_y I_t$$

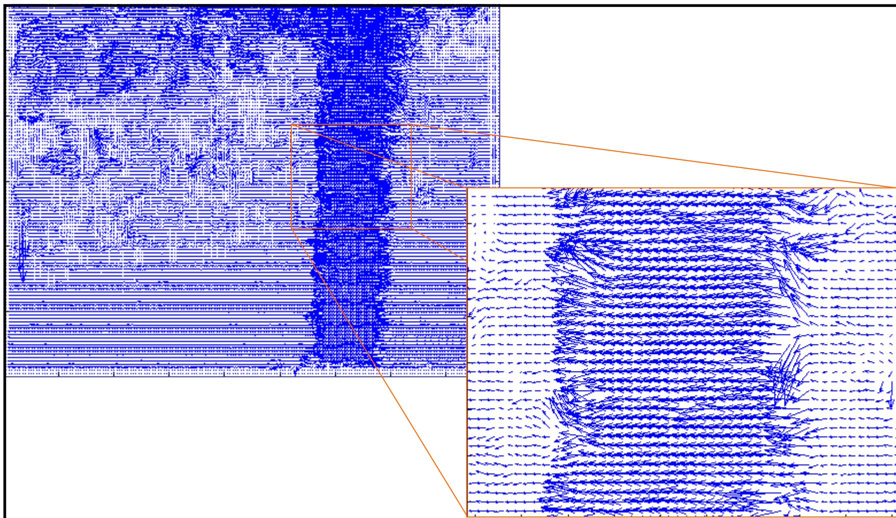
$$\boxed{\overbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} u \\ v \end{bmatrix}}^{u} = \overbrace{\begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}}^{B}}$$

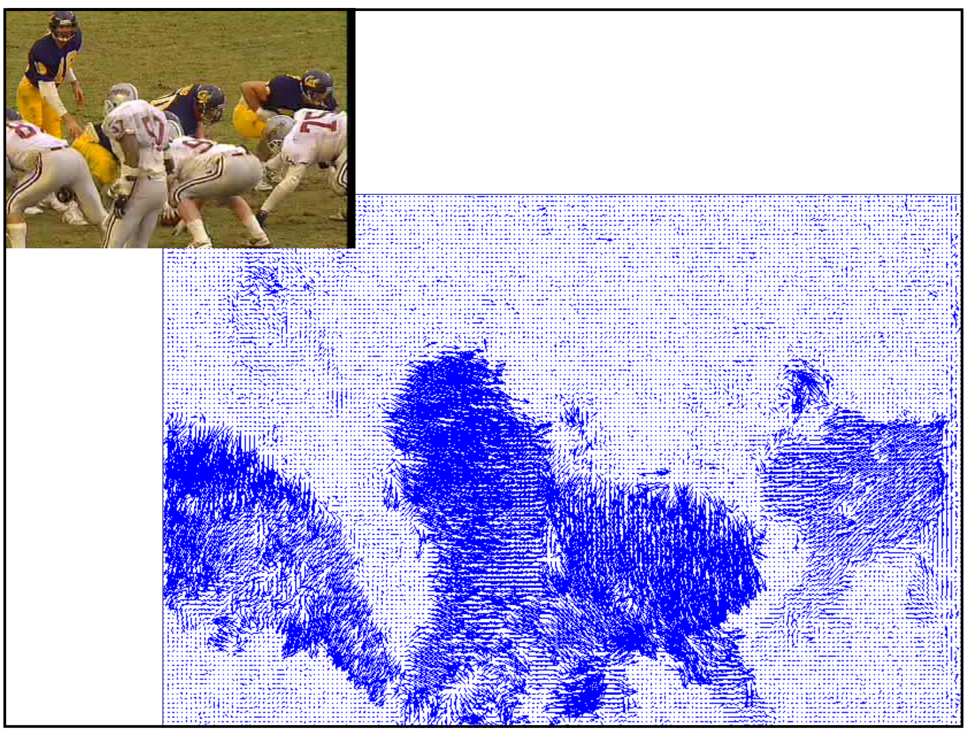
Computing Optical Flow Lucas & Kanade

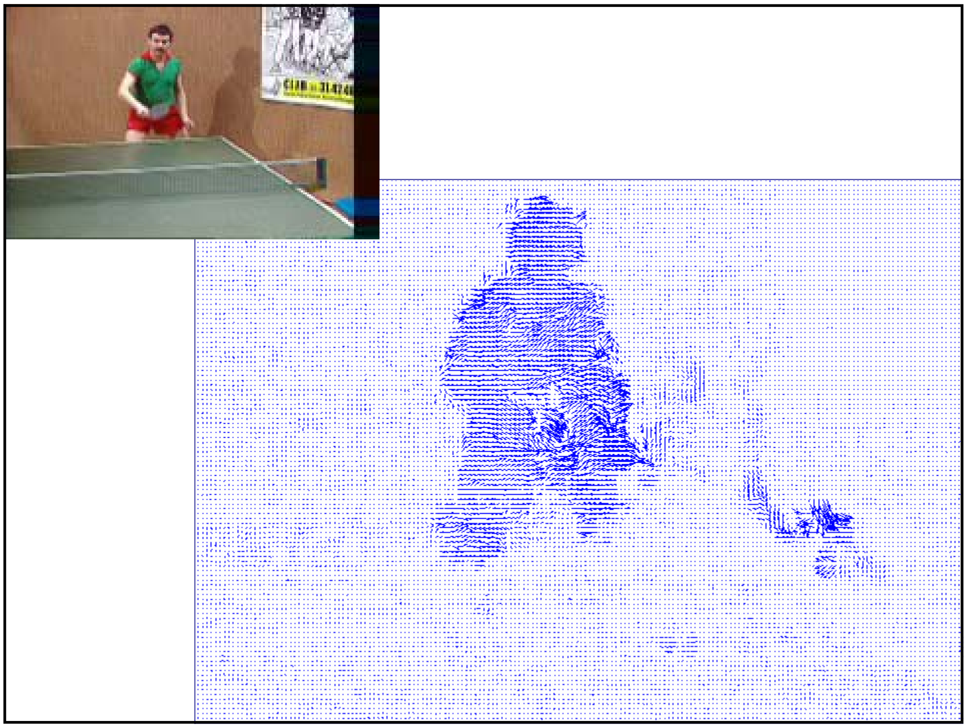
$$Au = B \qquad A^{-1}Au = A^{-1}B \qquad Iu = A^{-1}B \qquad u = A^{-1}B$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\boxed{\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum I_x^2 \sum I_y^2 - \left(\sum I_x I_y\right)^2} \begin{bmatrix} \sum I_y^2 & -\sum I_x I_y \\ -\sum I_x I_y & \sum I_x^2 \end{bmatrix} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}}$$







Texture Feature

- Measure of the intensity variation of a surface
- Quantifies smoothness and regularity.
- Requires a processing step to generate the descriptors.

Texture Measures

- GLCM (Gray level Co-occurrence Matrices)
- Law's Texture Energy Measures
- Wavelets
- Steerable Pyramids

GLCMs

- 2D histogram of image intensities
- $P(i,j,d,\theta)$: Count of occurrence of gray level i with j at distance d and in direction θ.

GLCMs

50	51	52	50
53	51	51	52
51	50	51	52
52	53	53	52

Intensity image patch

	50	51	52	53
50	0	2	0	0
51	1	1	3	0
52	1	0	0	1
53	0	1	1	1

$P(d,\theta)$, $d=1$, $\theta=0^o$

GLCMs

- Too many parameters
- Computationally Expensive
- Not suitable for coarse texture
- Susceptible to noise

Law's Texture Measures

- Feature Extraction scheme based on gradient operators
- 25 Masks by convolution of 5 1-D vectors
  - Level L5 = [1 4 6 4 1]
  - Edge E5 = [-1 -2 0 2 1]
  - Spot S5 = [-1 0 2 0 -1]
  - Wave W5= [-1 2 0 -2 1]
  - Ripple R5= [1 -4 6 -4 1]

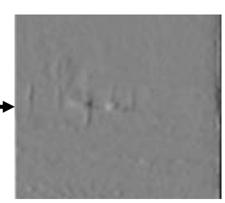
Law's Texture Energy Measures

$$(L5^t)E5 = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

$$L5 = [1 \quad 4 \quad 6 \quad 4 \quad 1]$$
$$E5 = [-1 \quad -2 \quad 0 \quad 2 \quad 1]$$

Input Image

$(L5^t)E5$

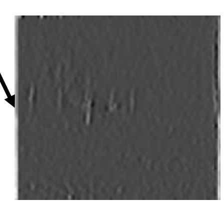


Output Image

$(L5^t)S5$

$$(L5^t)S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

$$S5 = [-1 \quad 0 \quad 2 \quad 0 \quad -1]$$



Output Image

Wavelet Analysis

- Tool for multi-resolution analysis
- Provides localization in both spatial and frequency domain
- Every decomposition contains information of a specified scale and orientation

Wavelet Analysis

- Wavelet transform decomposes f(x) onto a basis of wavelet functions:

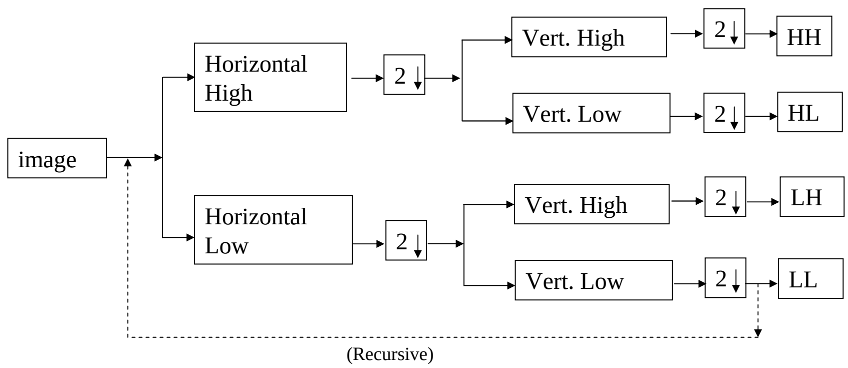
$$(W_a f)(b) = \int f(x)\varphi_{a,b}(x)dx$$

where

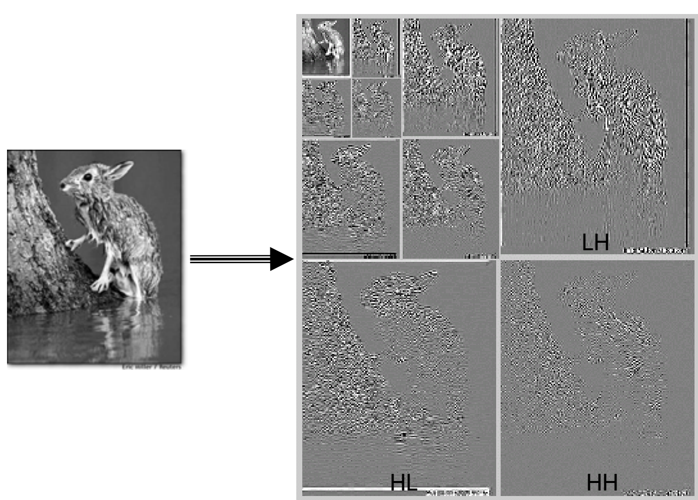
$$\varphi_{a,b}(x) = \frac{1}{\sqrt{a}}\varphi\left(\frac{x-b}{a}\right)$$

Wavelet Analysis

- 2-D Wavelet decomposition is obtained by separable filter bank.



Wavelet Analysis



Gabor Wavelets

- An effective strategy for extracting textural information from images
- optimal filters[1] both for
  - orientation and spatial frequency content
  - 2-D position

[1] Daugman, ``Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. **2**, pp. 1160-1169, 1985.

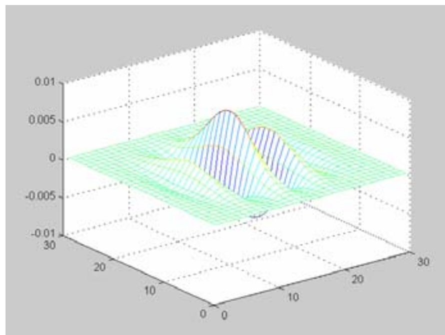
Gabor Filters

- Two dimensional Gabor filters over the image domain (x,y) have the following functional form

$$G(x,y) = e^{-\pi\left[(x-x_0)^2/\alpha^2 + (y-y_0)^2/\beta^2\right]} e^{-2\pi j\left[u_0(x-x_0)+v_0(y-y_0)\right]}$$

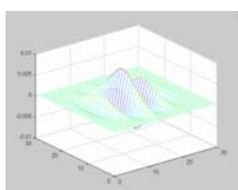
(x_0,y_0) specify position in the image
(α,β) specify effective width and length
(u_0,v_0) specify modulation

Gabor Filters

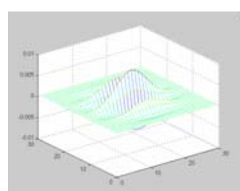


Real Part of a gabor filter

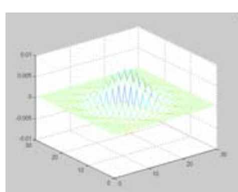
Gabor Filters with different orientations



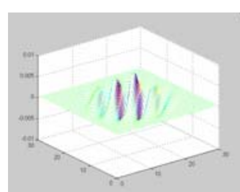
$$G(x,y) = g(x)\cos(2\pi[u_0(x-x_0)])$$



$$G(x,y) = g(x)\cos(2\pi[v_0(y-y_0)])$$



$$G(x,y) = g(x)\cos(2\pi[u_0(x-x_0)+v_0(y-y_0)])$$

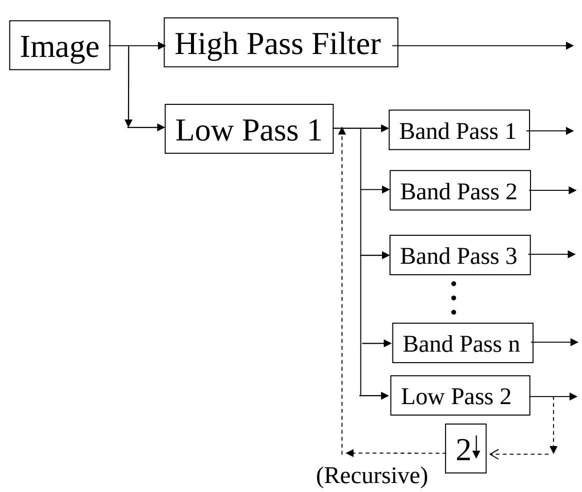


$$G(x,y) = g(x)\cos(2\pi[u_0(x-x_0)-v_0(y-y_0)])$$

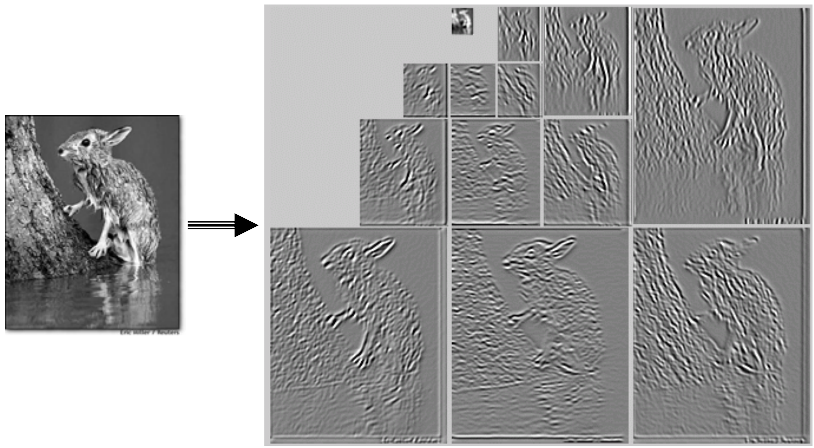
Steerable Pyramids

- Linear multi-scale, multi-orientation image decomposition
- Basis functions are directional derivative operators in different sizes and orientations
- Type of over-complete wavelet transform
- Steerable orientation decomposition

Steerable Pyramids



Steerable Pyramids



Texture Measures

- Energy

$$e_i = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I_i^2(x, y)$$

- Entropy

$$Entropy_i = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I_i(x, y) \log I_i(x, y)$$

Texture Measures

- Kurtosis

$$k = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_i(x,y)-\mu\right]^4}{M \times N \times \sigma^4} - 3$$

- Skew

$$Skew = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_i(x,y)-\mu\right]^3}{M \times N \times \sigma^3}$$

- Variance

$$\sigma^2 = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_i(x,y)-\mu\right]^2}{M \times N}$$