Lecture 18

Texture Classification by Gabor Filters

ECEN 5283 Computer Vision

Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goals

- To review two basic approaches for texture analysis.
- To study a basic technique of texture classification.
- To introduce Project 3 on texture classification.

Oriented and Non-oriented Texture Analysis

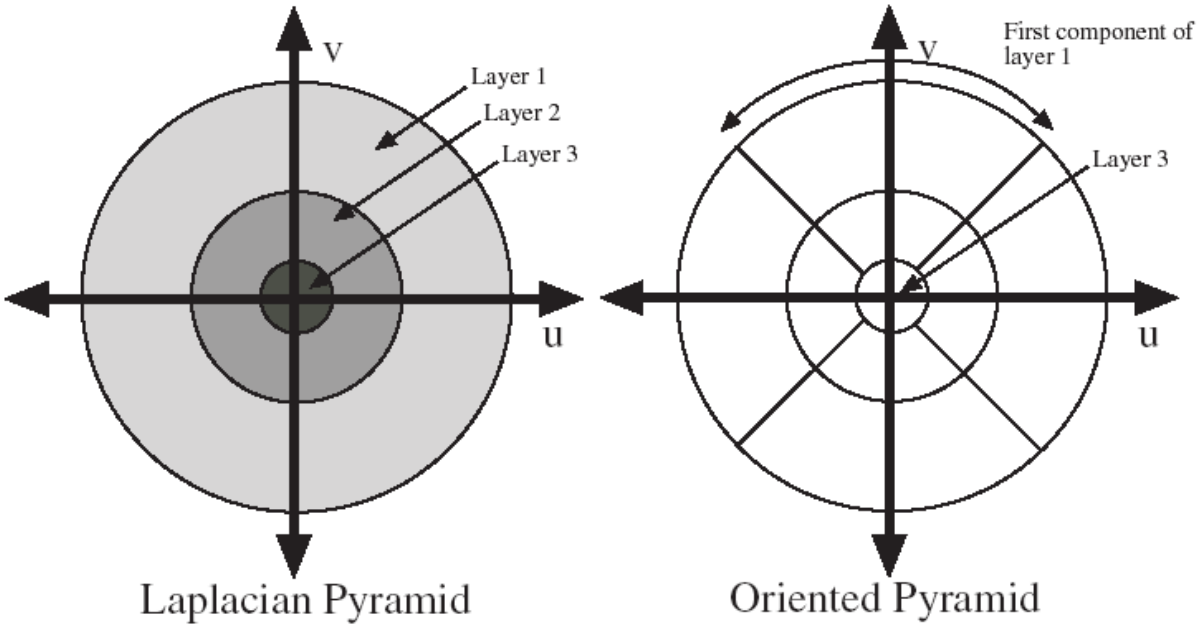
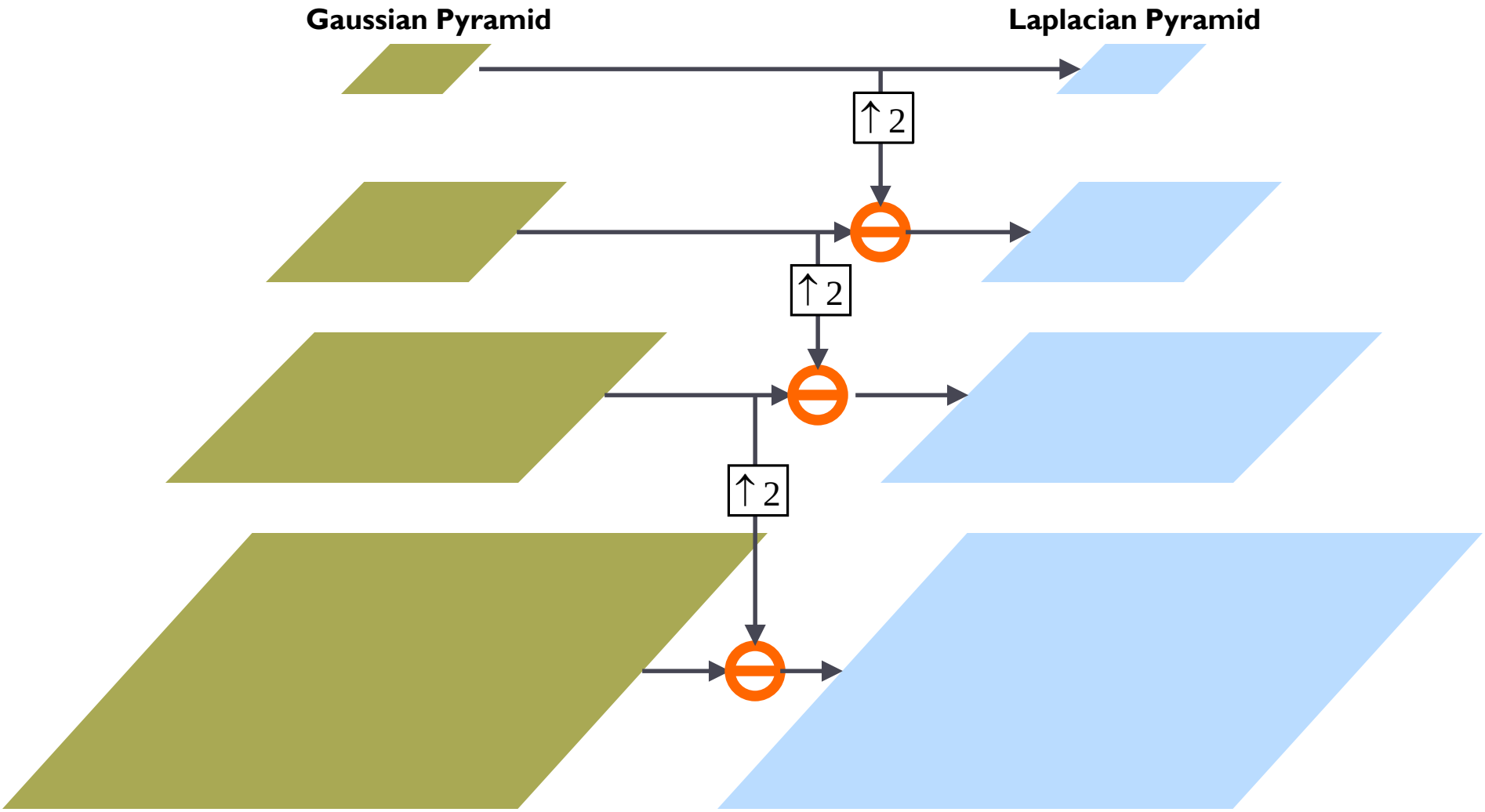
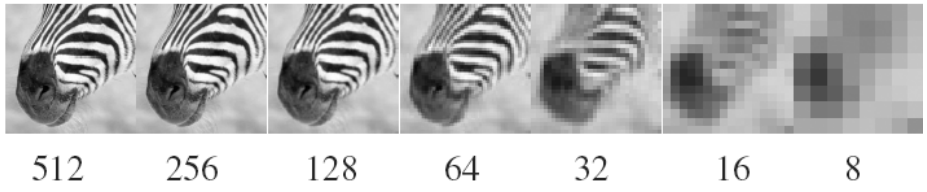


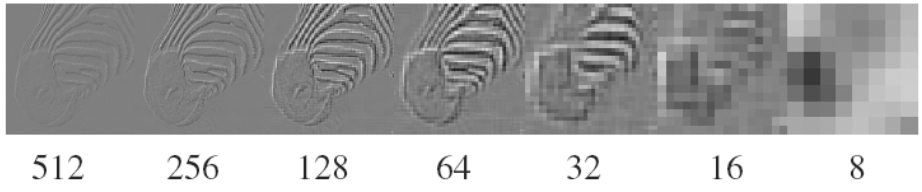
Figure 11.8. Each layer of the Laplacian pyramid consists the elements of a smoothed and resampled image that are not represented by the next smoother layer. Assuming that a Gaussian is a sufficiently good smoothing filter, each layer can be thought of as representing the image components within a range of spatial frequencies — this means that the Fourier transform of each layer is an annulus of values from the Fourier transform space (u, v) space (recall that the magnitude of (u, v) gives the spatial frequency). The sum of these annuluses is the Fourier transform of the image, so that each layer cuts an annulus out of the image's Fourier transform. An oriented pyramid cuts each annulus into a set of wedges. If (u, v) space is represented in polar coordinates, each wedge corresponds to an interval of radius values and an interval of angle values (recall that $\arctan(u/v)$ gives the orientation of the Fourier basis element).

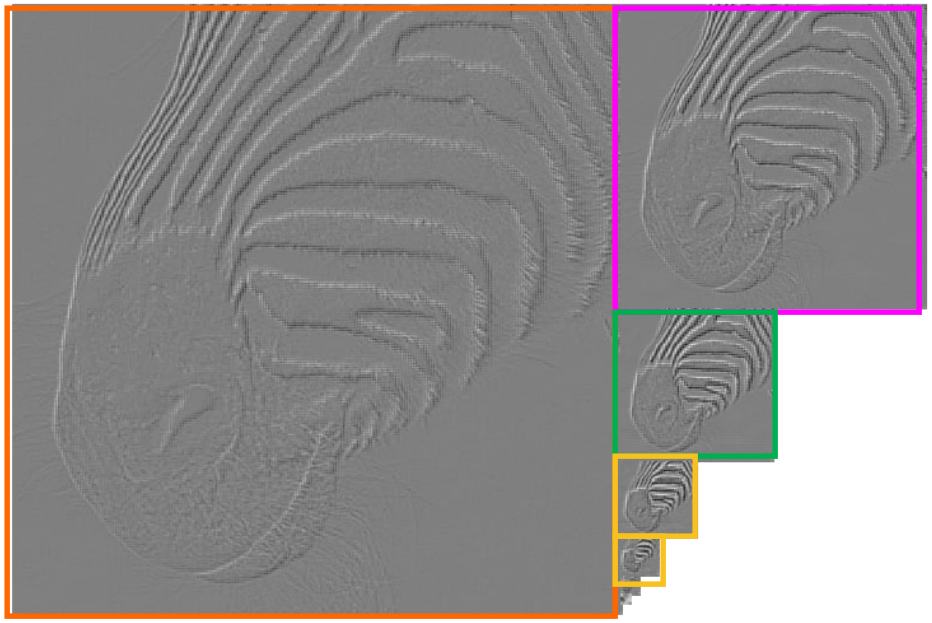
Laplacian and Gaussian Pyramids



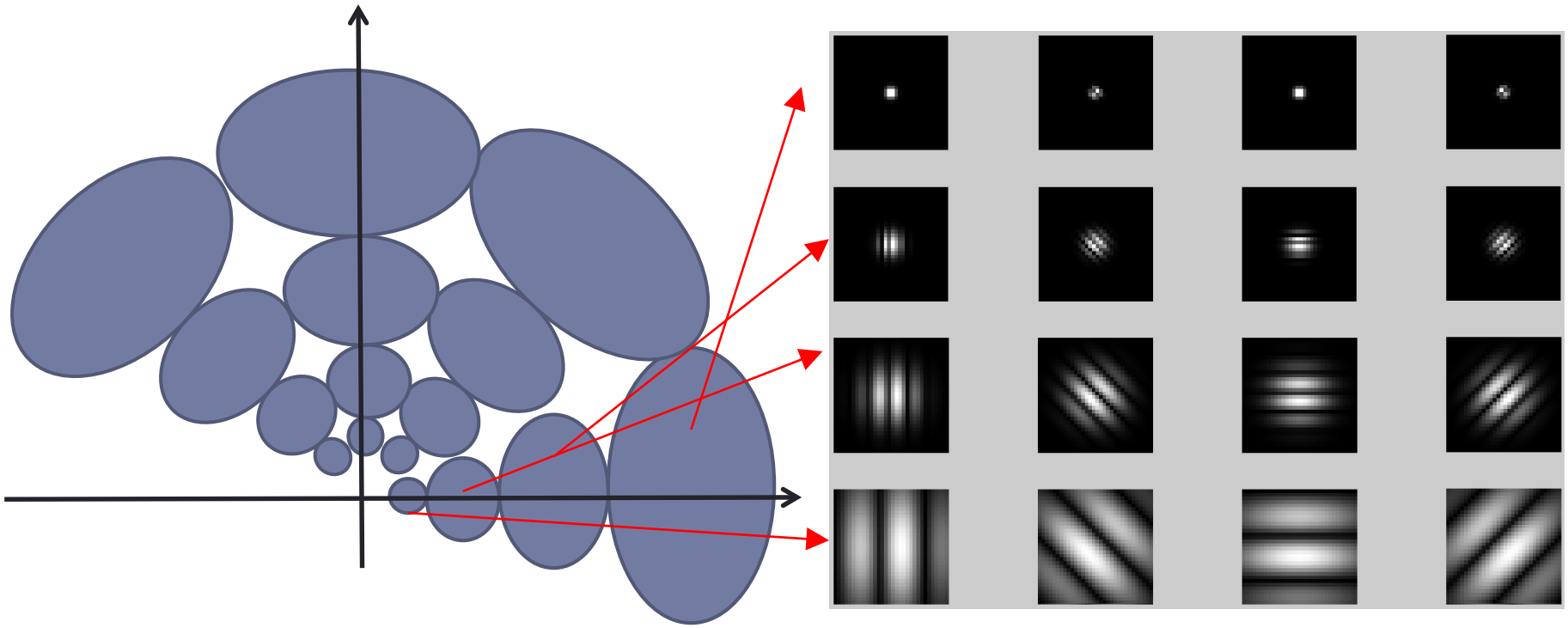
Laplacian-based Texture Analysis







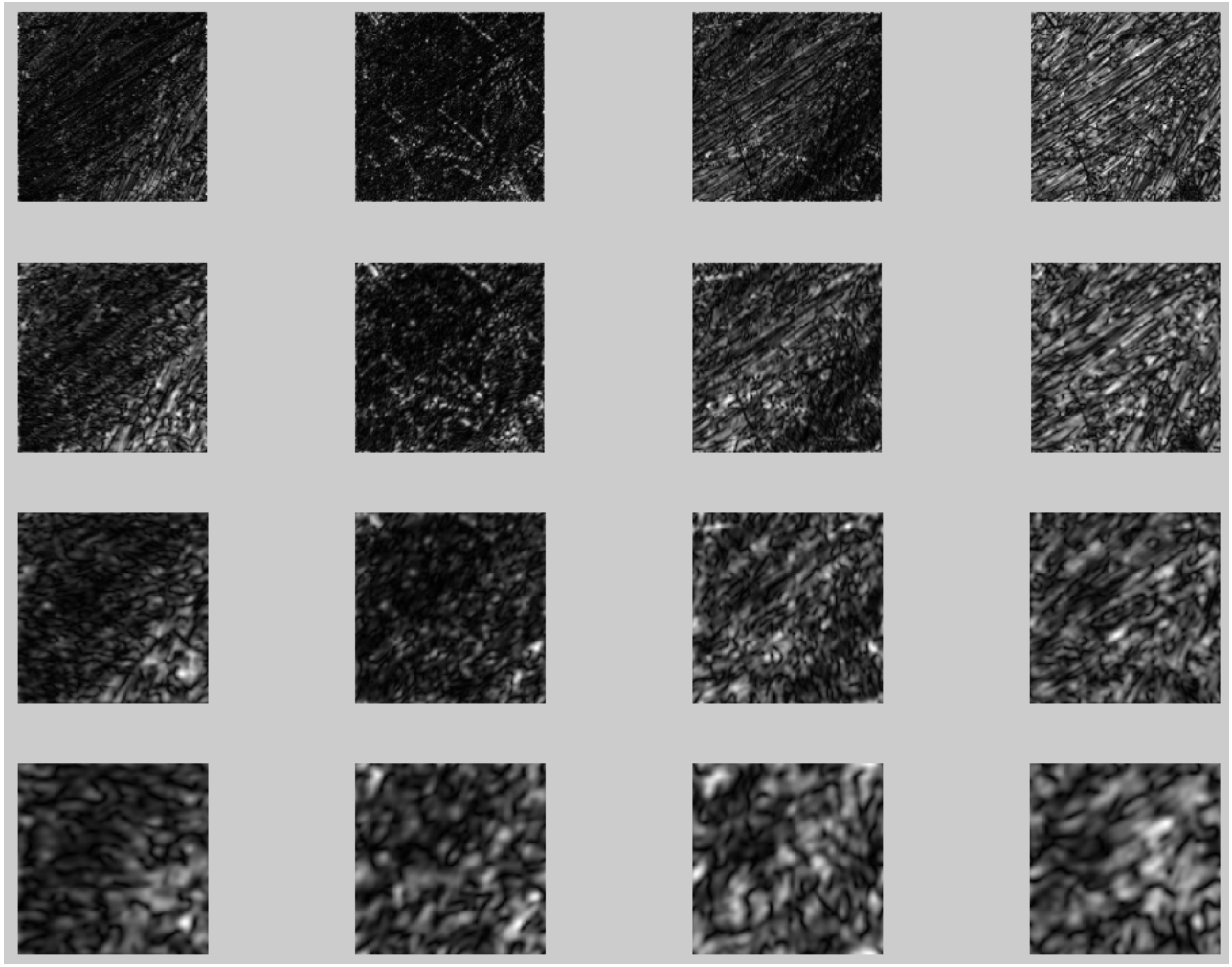
Gabor Filter Kernels
(Frequency Domain vs. Spatial Domain)

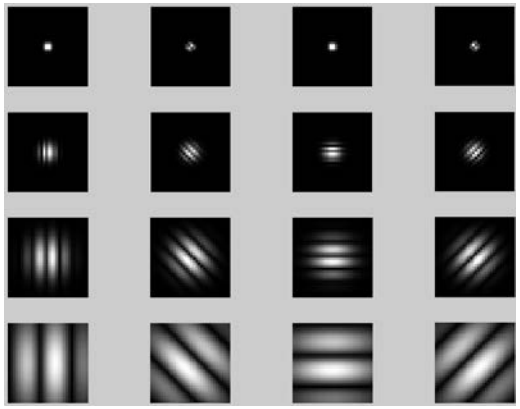


Frequency representation
of the Gabor filter design

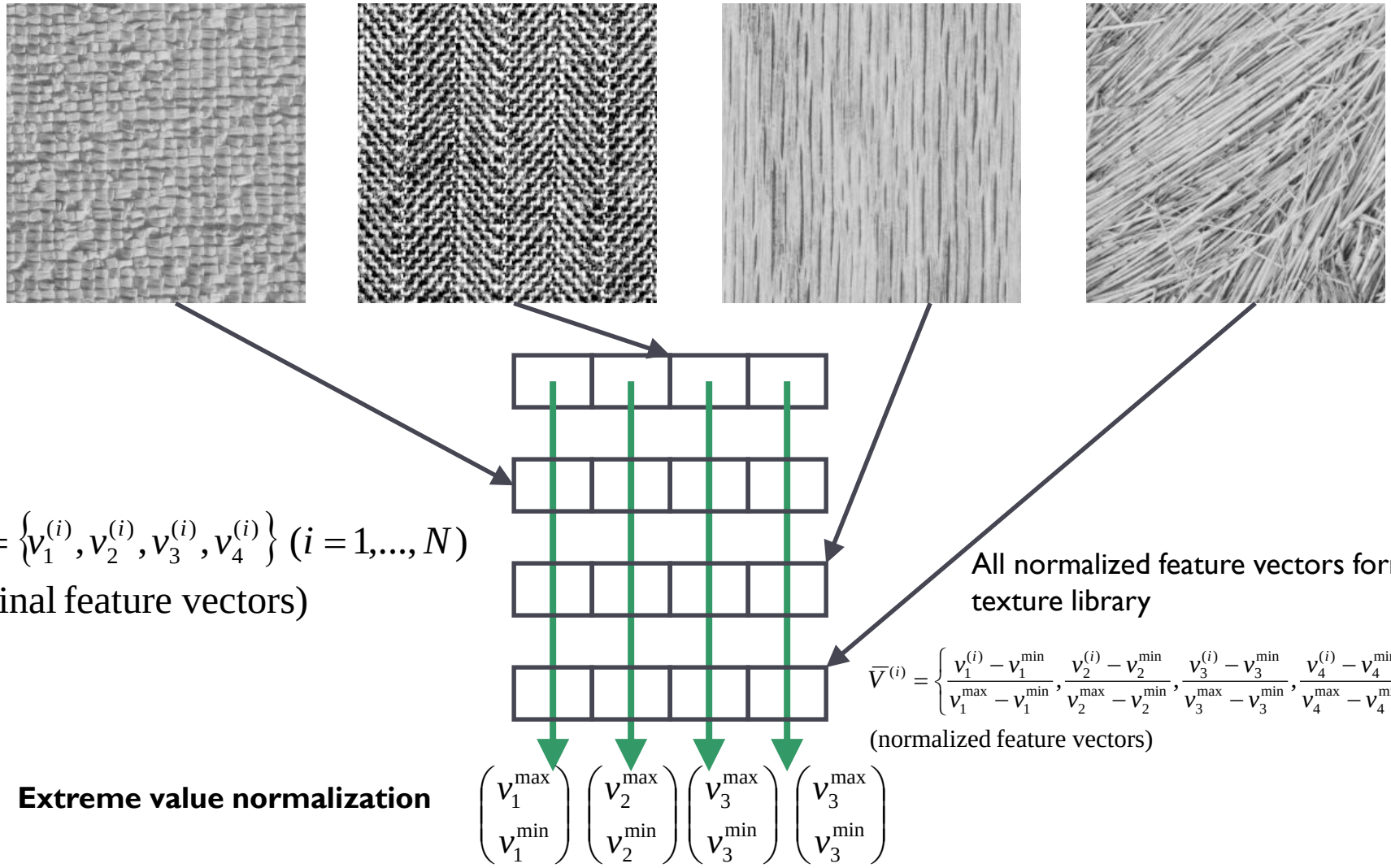
Spatial representation of
of Gabor filter kernels

Gabor Filtering of Brodaz Texture D15





Texture Analysis: Training



$$V^{(i)} = \left\{ v_1^{(i)}, v_2^{(i)}, v_3^{(i)}, v_4^{(i)} \right\} \; (i = 1, \ldots, N)$$

(original feature vectors)

All normalized feature vectors form a texture library

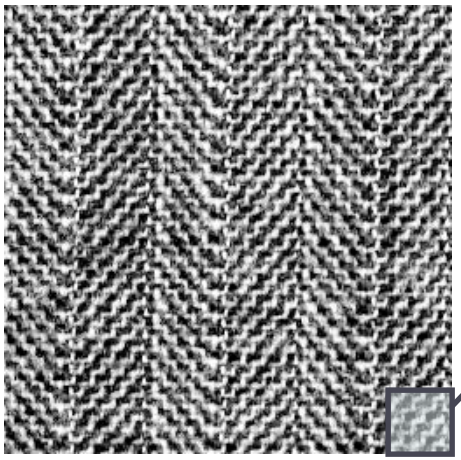
$$\overline{V}^{(i)} = \left\{ \frac{v_1^{(i)} - v_1^{\min}}{v_1^{\max} - v_1^{\min}}, \frac{v_2^{(i)} - v_2^{\min}}{v_2^{\max} - v_2^{\min}}, \frac{v_3^{(i)} - v_3^{\min}}{v_3^{\max} - v_3^{\min}}, \frac{v_4^{(i)} - v_4^{\min}}{v_4^{\max} - v_4^{\min}} \right\}$$

(normalized feature vectors)

Extreme value normalization

$$\begin{pmatrix} v_1^{\max} \\ v_1^{\min} \end{pmatrix} \begin{pmatrix} v_2^{\max} \\ v_2^{\min} \end{pmatrix} \begin{pmatrix} v_3^{\max} \\ v_3^{\min} \end{pmatrix} \begin{pmatrix} v_3^{\max} \\ v_3^{\min} \end{pmatrix}$$

Texture Analysis: Classification



$$U = \{u_1, u_2, u_3, u_4\}$$

$$\overline{U} = \left\{ \frac{u_1 - v_1^{\min}}{v_1^{\max} - v_1^{\min}}, \frac{u_2 - v_2^{\min}}{v_2^{\max} - v_2^{\min}}, \frac{u_3 - v_3^{\min}}{v_3^{\max} - v_3^{\max}}, \frac{u_4 - v_4^{\min}}{v_4^{\max} - v_4^{\min}} \right\}$$

$$\text{Class}(\overline{U}) = \arg_{c \in C} \min \left| \overline{V}_c - \overline{U} \right|$$

What is the major assumption for this classification scheme?

Normalized feature vectors in the library

The assumption for this scheme is that the visual dissimilarity between two textures can be represented by the Euclidean distance of their feature vectors.

Texture Classification using Gabor Filters

Setup the parameters of a *M*-scale and *N*-orientation Gabor filter bank (totally *M*x*N* channels)

For each texture image, perform Gabor filtering and construct a feature vector by computing certain statistics, (e.g., mean, variance, skewness, kurtosis) of the filtering output for each channel.

For all texture images, we normalize their feature vectors by scaling each coefficient by the maximum and minimum values in that channel across all textures. This will construct a texture library with all normalized feature vectors.

For a texture image we divide it into NxN blocks and for each block we compute and normalize the corresponding feature vector and find the best match in the library.

Texture Classification using the Laplacian Pyramid

Choose the number of scales in the Laplacian pyramid and the smoothing filter used prior to each down-sampling

For each of the given texture images, we develop a M-scale Laplacian pyramid and construct a feature vector by computing certain statistics (e.g., mean, variance, skewness, kurtosis) for each scale in the Laplacian pyramid

For all texture images, we normalize their feature vectors by scaling each coefficient by the maximum and minimum values in that channel across all textures. This will construct a texture library with all normalized feature vectors.

For a texture image we divide it into NxN blocks and for each block we compute and normalize the corresponding feature vector and find the best match in the library.

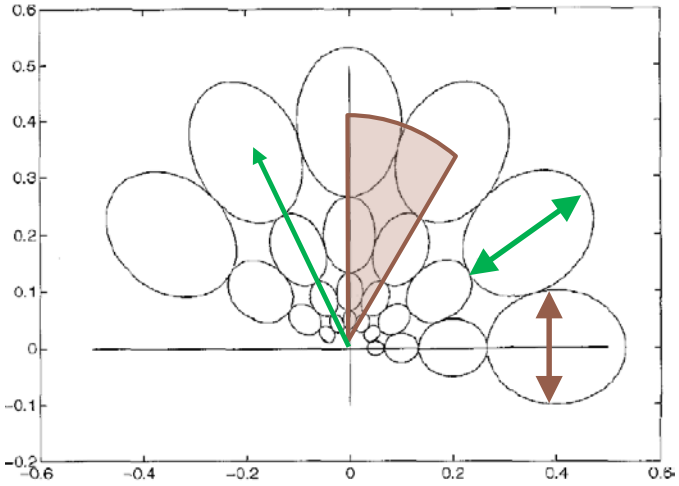
Project 3. Texture Analysis

- You are given 59 texture images for Project 3.
  - First, obtain the feature vectors (e.g., mean, variance, skewness, kurtosis) for all textures.
  - Normalize (using extremes) all feature vectors in each dimension across all textures, and save the normalized feature vectors in a texture library.
  - For each texture image (640x640), divide it into 100 blocks of size 64x64, for which a normalized feature vector is computed.
  - Classify all blocks by finding the closet feature vector in the texture library, and compute the percentage of correct classification (PCC).
- To test and optimize the Laplacian-based texture analysis method by adjusting the parameters the of Laplacian pyramid.
- To test and optimize Gabor filter-based texture analysis method by using the given Gabor filter Matlab function.
- Compared the two texture analysis methods in terms the performance of texture classification.

Matlab Programming (1)

- What are the I/O parameters for the Gabor filter function?

```
% GABORCONVOLVE - function for convolving image with log-Gabor filters
% EO = gaborconvolve(im,  nscale, norient, minWaveLength, mult, sigmaOnf, dThetaOnSigma)
%
%
%
%
%
%
%  Variable              Suggested   Description
%  name                  value
%  ---------------------------------------------------------
%   im                                Image to be convolved.
%   nscale               = 4;         Number of wavelet scales.
%   norient              = 6;         Number of filter orientations.
%   minWaveLength        = 3;         Wavelength of smallest scale filter.
%   mult                 = 2;         Scaling factor between successive filters.
%   sigmaOnf             = 0.65;      Ratio of the standard deviation of the Gaussian describing the Gabor filter's
%                                     transfer function in the frequency domain to the filter center frequency.
%   dThetaOnSigma        = 1.5;       Ratio of angular interval between filter orientations and the standard
%                                     deviation of the angular Gaussian function used to construct filters in the
%                                     frequency plane.  A value of dThetaOnSigma = 1.5 results in approximately
%                                     the minimum overlap needed to get even spectral coverage.
```



http://www.csse.uwa.edu.au/~pk/research/matlabfns/

Matlab Programming (2)

- How to read many texture images (D1.bmp, D2.bmp…, D59.bmp)?

```
Gau_Kernel=5;                                   % the dimension of the smoothing kernel
Gau_Sigma=0.75;                                 % the dimension of the smoothing kernel
Gau_Layer=4;                                    % the layer number of the Laplacian pyramid
Texture_Num=59;                                 % the total number of texture images
S=64;                                           % the block size for texture classification

Ti=cell(Texture_Num,1);                         % to save all texture images
for i=1:Texture_Num
    N=num2str(i);                               % create the file name for each texture
    Ti{i}=imread(['D',N,'.bmp']);               % read a texture image into a cell
    Fi(i,:)=Laplacian_Pyramid(Ti{i},Gau_Layer,Gau_Sigma,Gau_Kernel);
                                                % Laplacian feature extraction
End

for i=1:Gau_Layer                               % normalize each dimension of all feature vectors
    Max_Var(i)=max(Fi(:,i));
    Min_Var(i)=min(Fi(:,i));
    Ni(:,i)=(Fi(:,i)-Min_Var(i))/(Max_Var(i)-Min_Var(i)); % create the texture library for all training images
end
```

Matlab Programming (3)

- How to visualize the Gabor filtering outputs for a texture?

```
clear all;
texture=imread('D7','bmp');

Num_scale=4;
Num_orien=6;

E0=gaborconvolve(texture,Num_scale,Num_orien,3,2,0.65,1.5);

for i=1:Num_scale
    for j=1:Num_orien
        ind=(i-1)*Num_orien+j;
        subplot(4,6,ind);              % Create a multi-figure plot
        imshow(abs(E0{i,j}),[]);       % Show the magnitude of each channel
    end
end
```