CIS 467
Image Analysis and Processing
Lecture 25, 05/10/2005

Li Shen
Computer and Information Science
UMass Dartmouth

Outline

- Chapter 1: Introduction
- Chapter 2: Digital Image Fundamentals
- Chapter 3: Image Enhancement in the Spatial Domain
- Chapter 4: Image Enhancement in the Frequency Domain
- Chapter 5: Image Restoration
- Chapter 7: Wavelets and Multiresolution Processing
- Chapter 8: Image Compression
- Chapter 9: Morphological Image Processing
- Chapter 10: Image Segmentation

- One page brief notes, calculator

Chapter 1: Introduction

- Source of energy for images
  - Electromagnetic (EM) energy spectrum
  - Acoustic, ultrasonic, electronic, synthetic

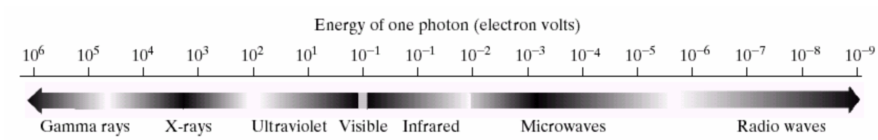


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

Chapter 2: Fundamentals

- Image sensing and acquisition
- Image sampling and quantization
  - Spatial and gray-level resolution
  - Zooming and shrinking
- Some basic relationships between pixels
  - Adjacency, distance, path
- Linear and nonlinear operations

Chapter 3: Image Enhancement in the Spatial Domain

- Basic gray level transformations
  - Linear, logarithmic, power-law
- Histogram processing
  - Equalization
- Enhancement using arithmetic/logic operations
  - Averaging, subtraction
- Basics of spatial filtering
  - Linearity, near image border
- Smoothing spatial filters
  - Averaging, median
- Sharpening spatial filters
  - Laplacian, Sobel

Point versus mask processing

Smoothing versus sharpening

Chapter 4: Image Enhancement in the Frequency Domain

- 1D Fourier transform
  - DFT, frequency domain, frequency component
- 2D Fourier transform
  - Spectrum, phase angle, power spectrum, visualization, dc component
- Filtering in the frequency domain
  - Procedure, notch filter, lowpass, highpass
- Correspondence between filtering in the spatial and frequency domains
  - Convolution theorem, Gaussian filter

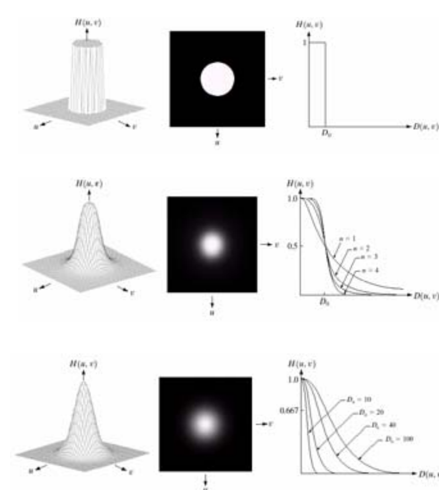
Chapter 4: Image Enhancement in the Frequency Domain

- Smoothing frequency-domain filters
  - Ideal lowpass filters
  - Butterworth lowpass filters
  - Gaussian lowpass filters
- Sharpening frequency-domain filters
  - Ideal highpass filters
  - Butterworth highpass filters
  - Gaussian highpass filters
  - High-boost filtering

Smoothing Filters

- Smoothing
  - Attenuating a specified range of high-frequency components in the transform of an image
- Three types
  - Ideal: very sharp
  - Butterworth: transition
    - Parameter: the filter order
  - Gaussian: very smooth

Ringing behavior



Sharpening Filters

- Reverse operation of lowpass filters
  - $H_{hp}(u,v) = 1 – H_{lp}(u,v)$
- Highpass filters
  - Ideal
  - Butterworth
  - Gaussian
- Unsharp masking
  - $f_{hp}(x,y) = f(x,y)-f_{lp}(x,y)$

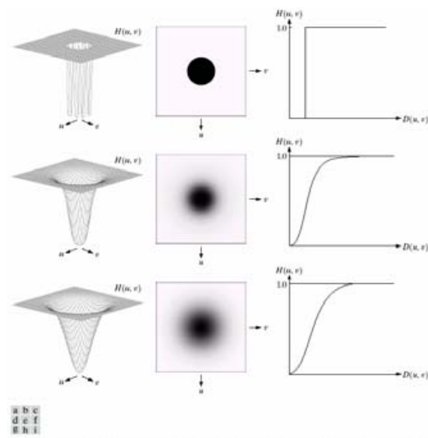


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Chapter 5: Image Restoration

- A model of the image degradation/restoration process
- Noise models
- Restoration in the presence of noise only-spatial filtering
- Linear, position-invariant degradation
- Estimating the degradation function
- Inverse filtering
- Minimum mean square error (Wiener) filtering

Model of Degradation/Restoration

- Input: g(x,y), knowledge about H and η
- Objective: an estimate of f(x,y)

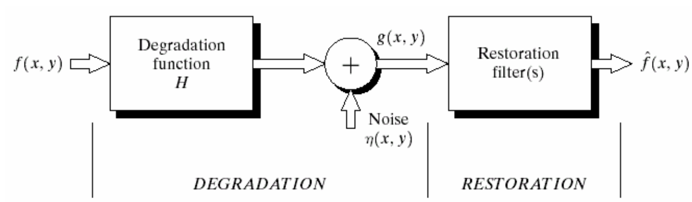
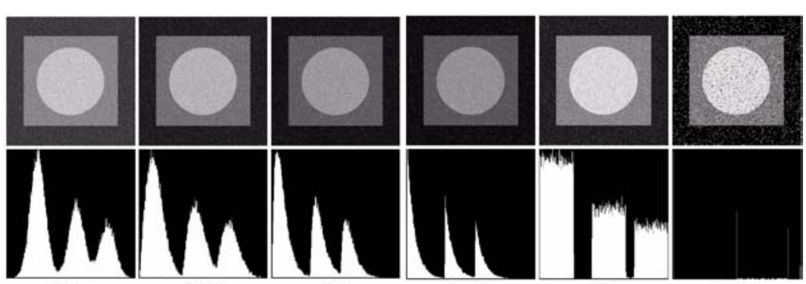


FIGURE 5.1 A model of the image degradation/restoration process.

h: linear, position-invariant
g(x,y) = h(x,y) * f(x,y) + η(x,y)
G(x,y) = H(u,v) F(u,v) + N(u,v)

Noise Models



- Histogram and PDF
- Salt-and-pepper: peak at white end
- First 5 are visually similar
- Estimate a noise model

Noise Reduction Using Spatial Filtering

- Mean filters
  - Arithmetic and geometric
  - Harmonic and Contra-harmonic
- Order-statistics filters
  - Median, min, max
  - Alpha-trimmed filter
- Adaptive filters
  - Adaptive local noise reduction filter
  - Adaptive median filter

Adaptive LNR Filters

- Zero noise variance, return g(x,y)
  - g(x,y) = f(x,y)
- High local variance, return a value similar to g(x,y)
  - Edges
- Two variances are equal, return arithmetic mean
  - Noise reduction by averaging
- Estimate noise variance
- Negative gray levels

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

$$\sigma_\eta^2 \le \sigma_L^2$$

Linear, Position-Invariant Degradations

- A linear, spatially-invariant degradation system with additive noise can be modeled in the spatial domain as the convolution of the degradation function with an image, followed by the addition of noise.
- In the frequency domain, the same process can be expressed as the product of the transforms of the image and degradation, followed by the addition of the transform of the noise.

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

Inverse Filtering

- G(u,v)/H(u,v)
- Apply to the ratio
- Butterworth lowpass function of order 10
- Cutoff distances 40, 70, 85

Wiener Filtering

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

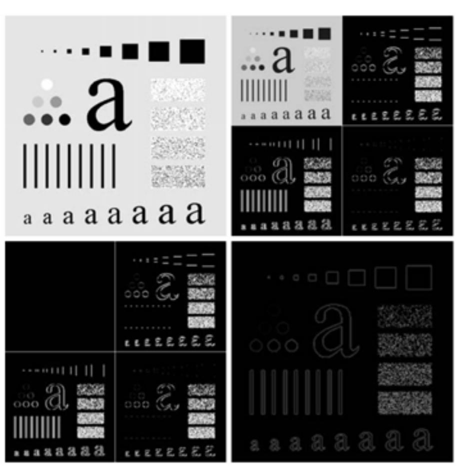
Chapter 7: Wavelets and Multi-Resolution Processing

- Background
  - Image pyramids
  - Subband coding
  - The Haar transform
- Multi-resolution expansions
  - Scaling function, wavelets
- Wavelet transforms
- Applications

Multi-Resolution Expansion

- Scaling function
  - A series of approximations
  - Each differing by a factor of 2
- Wavelets
  - Encode difference between adjacent approximations

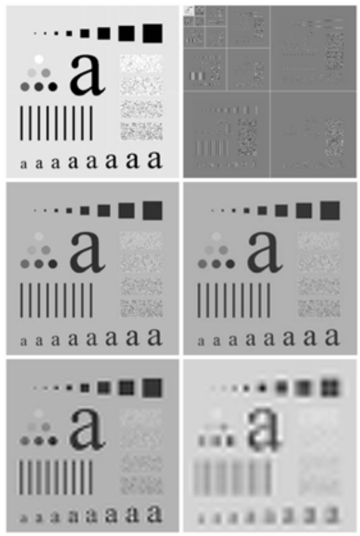
DWT Application: Edge Detection



a b
c d

FIGURE 7.7 Wavelets in edge detection: (a) A simple test image; (b) its wavelet transform; (c) the transform modified by zeroing all approximation coefficients; and (d) the edge image resulting from computing the absolute value of the inverse transform.

DWT Application: Image Smoothing



a b
c d
e f

FIGURE 7.8 Wavelet-based image smoothing: (a) A test image; (b) its wavelet transform; (c) the inverse transform after zeroing the first-level detail coefficients; and (d) through (f) similar results after zeroing the second-, third-, and fourth-level details.

Chapter 8: Image Compression

- Fundamentals
  - Coding redundancy
  - Interpixel redundancy
  - Psychovisual redundancy
  - Fidelity Criteria
- Image compression models
  - The source encoder and decoder
  - The channel encoder and decoder
- Error-free compression
  - Variable-length coding

IGS code
RMS error
RMS SNR

Image Compression Models

- Three reduction techniques: combined to form compression systems
- Overall characteristics + general model
- Source encoder/decoder: remove redundancies
- Channel encoder/decoder: increase noise immunity

Mapper, quantizer, symbol encoder

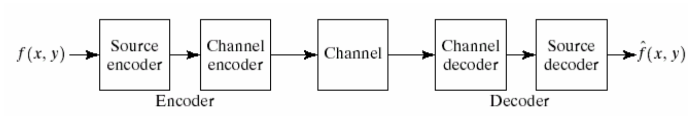


FIGURE 8.5 A general compression system model.

Hamming Code

Variable-Length Codes

TABLE 8.5 Variable-length codes.

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	100000	C10C00	111 101	00 1110
a_{14}	0.03	01101	01111	100001	C10C01	111 110	00 1111
Block 3							
a_{15}	0.03	01110	01100	100010	C10C10	111 111 000	00 00 10
a_{16}	0.02	01111	010000	100011	C10C11	111 111 001	00 00 11
a_{17}	0.02	10000	010001	100100	C11C00	111 111 010	00 00 110
a_{18}	0.02	10001	001010	100101	C11C01	111 111 011	00 00 100
a_{19}	0.02	10010	001011	100110	C11C10	111 111 100	00 00 101
a_{20}	0.02	10011	011010	100111	C11C11	111 111 101	00 00 1110
a_{21}	0.01	10100	011011	101000	C00C00C00	111 111 110	00 00 1111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

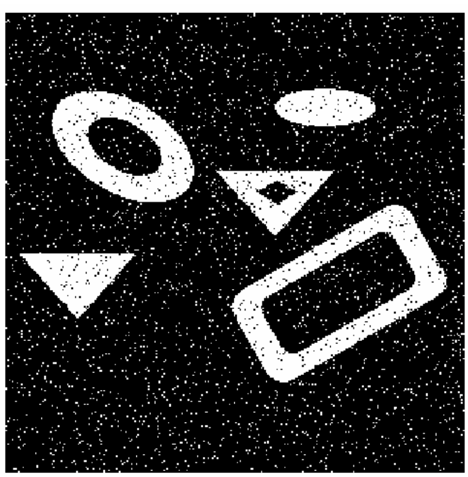
Arithmetic Coding

Block versus nonblock

Chapter 9: Morphological Image Processing

- Preliminaries
  - Set theory and logic operations
- Fundamental operations
  - Dilation and erosion
  - Opening and closing
  - The hit-or-miss transformation
- Some basic morphological algorithms
  - Boundary extraction, region filling, extraction of connected components, convex hull
  - Thinning, thickening, skeletons, pruning

Numbers of Components, Holes?



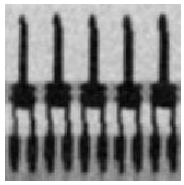
Chapter 10: Segmentation

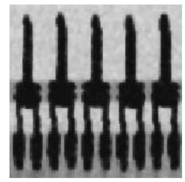
- Point detection
- Line detection
- Basic global thresholding
- Basic adaptive thresholding

Uneven illumination

HW5 Review

Problem 5.10. Arithmetic and geometric mean filters





$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Less blurred

Thicker

Problem 5.11. Contraharmonic filter

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

- Q>0, pepper noise
- Q<0, salt noise
- Poor result when the wrong polarity is chosen for Q
- Q=-1
- Any Q, constant gray levels

Problem 5.26. Telescope images

- Telescope images are blurry
- Cannot conduct controlled lab experiments with the lenses and imaging sensors
- Formulate a digital image processing solution

One possible solution: (1) Average images to reduce noise. (2) obtain blurred image of a bright, single star to simulate an impulse (the star should be as small as possible in the field of view of the telescope to simulate an impulse as closely as possible. (3) The Fourier transform of this image will give $H(u, v)$. (4) Use a Wiener filter and vary K until the sharpest image possible is obtained.

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

Problem 7.9. Haar transform

- Compute the Haar transform

  F = [3 -1; 6 2]

$$\begin{aligned} \mathbf{T} &= \mathbf{HFH} = \left(\frac{1}{\sqrt{2}}\right)^2 \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 5 & 4 \\ -3 & 0 \end{bmatrix}. \end{aligned}$$

- Inverse Haar transform

$$\begin{aligned} \mathbf{F} &= \mathbf{H}^{-1}\mathbf{T}\mathbf{H}^{-1} \\ &= \left(\frac{1}{\sqrt{2}}\right)^2 \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 5 & 4 \\ -3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix}. \end{aligned}$$

Problem 8.1. Histogram equalized image and compression

- Variable-length coding: compress a histogram equalized image

- What about interpixel redundancies?

Problem 8.4. RMS error and RMS signal-to-noise ratio

Data	6-bit Code	Sum	IGS Code	Decoded IGS	Error	Sq. Error
		000000				
12	001100	001100	001	8	4	16
12	001100	010000	010	16	-4	16
13	001101	001101	001	8	5	25
13	001101	010010	010	16	-3	9
10	001010	001100	001	8	2	4
13	001101	010001	010	16	-3	9
57	111001	111001	111	56	1	1
54	110110	110111	110	48	6	36

- RMS error = 14.5

$$SNR_{rms} = \sqrt[2]{\frac{6400}{116}} = 7.43$$

Problem 8.5. Hamming code

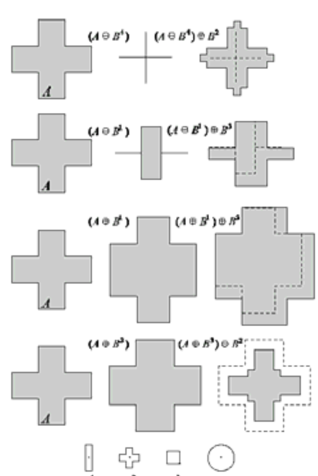
- 0110: 1100110
- 1001: 0011001
- 1000: 1110000
- 1111: 1111111

$$
\begin{aligned}
h_1 &= b_3 \oplus b_2 \oplus b_0 = 0 \oplus 1 \oplus 0 = 1 \\
h_2 &= b_3 \oplus b_1 \oplus b_0 = 0 \oplus 1 \oplus 0 = 1 \\
h_3 &= b_3 = 0 \\
h_4 &= b_2 \oplus b_1 \oplus b_0 = 1 \oplus 1 \oplus 0 = 0 \\
h_5 &= b_2 = 1 \\
h_6 &= b_1 = 1 \\
h_7 &= b_0 = 0.
\end{aligned}
$$

- 1100111: 111, 0110
- 1100110: 000, 0110
- 1100010: 101, 0110

$$
\begin{aligned}
c_1 &= h_1 \oplus h_3 \oplus h_5 \oplus h_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1 \\
c_2 &= h_2 \oplus h_3 \oplus h_6 \oplus h_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1 \\
c_4 &= h_4 \oplus h_5 \oplus h_6 \oplus h_7 = 0 \oplus 1 \oplus 1 \oplus 1 = 1
\end{aligned}
$$

Problem 9.7. Morphological operations



Project 05-01: Noise generator

```
function im2 =
      add_gaussian(im,MU,SIGMA,PROB)

d = size(im);
v = rand(d);
idx = find(v<PROB); % affect only PROB*100%
      pixels

noise = normrnd(MU,SIGMA,d(1),d(2));

im2 = im; im2(idx) =
      uint8(double(im(idx))+noise(idx));

figure; subplot(1,2,1); imshow(im);
subplot(1,2,2); imshow(im2);



function r = normrnd(mu,sigma,m,n);
r = randn(m,n) .* sigma + mu;
```

```
function im2 =
      add_saltpepper(im,PROB1,PROB2)

d = size(im); im2 = im;
% PROB1+PROB2 pixels are affected
v = rand(d); idx = find(v<PROB1+PROB2);

% PROB1 pixels are affected by salt noise
n = length(idx); v = rand(1,n);
ix1 = find(v<PROB1/(PROB1+PROB2)); idx1 =
      idx(ix1);
im2(idx1) = 255;

% PROB2 pixels are affected by pepper noise
ix2 = find(v>=PROB1/(PROB1+PROB2)); idx2 =
      idx(ix2);
im2(idx2) = 0;

figure; subplot(1,2,1); imshow(im);
subplot(1,2,2); imshow(im2);
```


Project 05-02: Noise reduction using a median filter

```
im = imread('Fig5.07(a).jpg','jpg');

SALT=0.2; PEPPER=0.2;
im2 =
   add_saltpepper(im,SALT,PEP
   PER);

im3 = noisered_med(im2);
```

```
function im2 = noisered_med(im)

im2 = medfilt2(im);

figure;
subplot(1,2,1); imshow(im);
subplot(1,2,2); imshow(im2);

return;
```

Project 08-01: Objective fidelity criteria

```
im = imread('Fig5.03.jpg','jpg');
im2 = floor(double(im)*(16/256)); %
    reduce the number of gray levels from
    256 to 16
im2 = uint8(im2*16); % change it back to
    256;
[rmse, snr] = rms_err_snr(im,im2);
```

```
function  [rmse, snr] = rms_err_snr(im,im2)

d = size(im); im = double(im); im2 =
    double(im2);

rmse = sqrt(sum(sum((im-
    im2).^2))/prod(d));

snr = sum(sum(im2.^2))/sum(sum((im2-
    im).^2));

disp(sprintf('Root-mean-square error: %f;
    signal-to-noise ration: %f',rmse,snr));

return;
```


Project 09-02: Boundary extraction

```
function im2 = boundary(im)

im2 = imerode(im,ones(3,3));
im2 = uint8(double(im) - double(im2));

figure;
subplot(1,2,1); imshow(im);
subplot(1,2,2); imshow(im2);
title('Boundary');

return;
```

Project 05-04: Parametric Wiener filter

```
function pwf(im)

d = size(im);

% create a blurring filter
h = fspecial('motion', 50, 50); im2 = imfilter(im, h, 'circular');

% create Gaussian noise
m = 0; v = 3; noise = normrnd(m,v,d(1),d(2));
im3 = uint8(double(im2) + noise);

% noise-to-power ratio
tim = double(im); NSR = sum(noise(:).^2)/sum(tim(:).^2);

% apply wiener filter
im4 = deconvwnr(im3,h,NSR);

figure;
subplot(2,2,1); imshow(im); title('Original');
subplot(2,2,2); imshow(im2); title('Blurred');
subplot(2,2,3); imshow(im3); title('Gaussian noise added');
subplot(2,2,4); imshow(im4); title('Result');

return;
```

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v)$$

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

Project 09-04: Morphological solution to Problem 9.27

```
function particles(im)

% make it binary
im1 = zeros(size(im)); idx = find(im>128);
      im1(idx) = 1;

% make border to be 1s
im1([1 end],:) = 1; im1(:,[1 end]) = 1;

% connected component analysis
[L,num] = bwlabel(im1,8);

border = [];
overlap = [];
nonoverlap = [];

for i=1:num
   comp{i} = find(L==i);
   len(i) = length(comp{i});
   if len(i)>7000
      border = [border; comp{i}];
   elseif len(i)>400
      overlap = [overlap; comp{i}];
   else
      nonoverlap = [nonoverlap; comp{i}];
   end
End

im2 = zeros(size(im)); im2(border) = 1;
im3 = zeros(size(im)); im3(overlap) = 1;
im4 = zeros(size(im)); im4(nonoverlap) =
      1;
```

Thank You

Good luck with the finals

Have a great summer!