

ECGR 6118
Computer Project: Feature Selection
Student Name: _____

(copyright T. Weldon, 2006)

For this project, you may use mathcad or NetBeans

Project tasks:

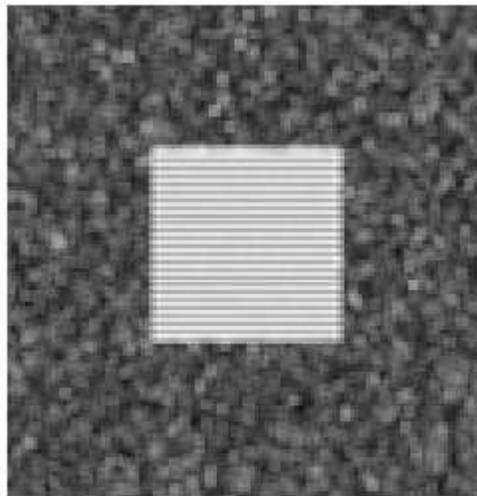
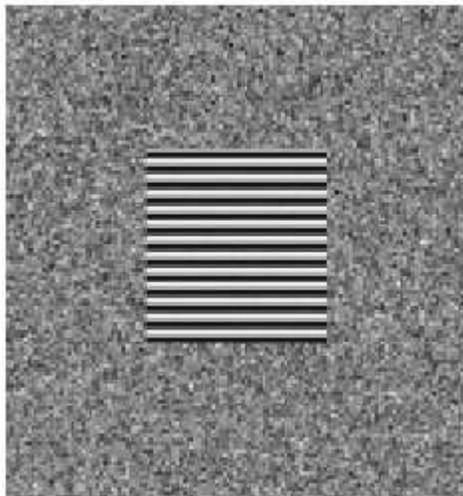
Complete the tasks below and turn in a project report.

Turn in a 7-page report,
with page-1 cover sheet,
page 2 explaining the methods you used,
pages 3 - 7 being 6x6 inch printouts of your
images and answers to questions for each of the tasks below.

Make sure that you caption each image, clearly stating what the image is.
Include your Mathcad or Java as an Appendix.

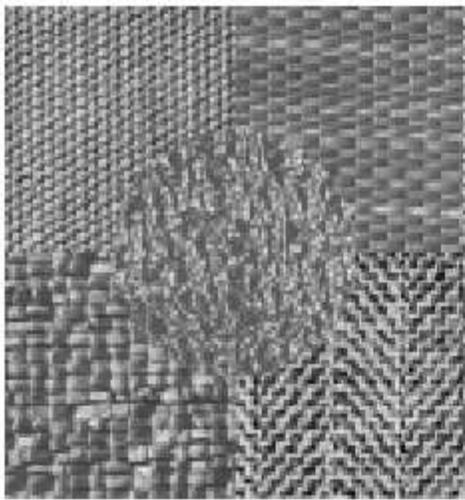
1. For the image nsinb.gif, write a program to compute the standard deviation over a 3x3 window (Pratt Eq. 16.2-2), with the resulting image rescaled to between 0 and 255 gray-levels. Plot the image.

Below, the input image nsinb.gif is illustrated on the left, and a standard deviation over a 3x3 window is illustrated on the right. Do not use the image below, download the original image from the website.



2. For the preceding image nsinb.gif, write a program to compute the standard deviation over a 3x3 window (Pratt Eq. 16.2-2), then apply a 3x3 window average twice in succession, then threshold the image to segment it as illustrated below for the nat5c image. Choose the best threshold that you can.

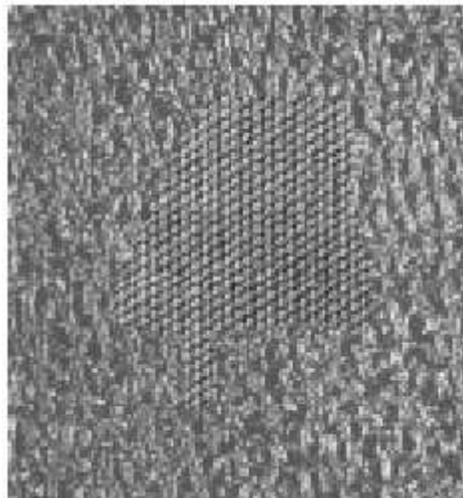
Note that the image example below **IS NOT** for nsinb.gif



3. For the image d24d77b.gif, shown below, write a program to compute the Laws 1 feature (Pratt Fig. 16.6-8 and Fig. 16.6-9), and do not forget to apply the standard deviation computation after the Laws mask filtering (read Pratt Sec. 16.6.6). Use a 3x3 window for computing this standard deviation, although you may wish to experiment with others. Rescale the resulting image rescaled to between 0 and 255 gray-levels. Plot the image.

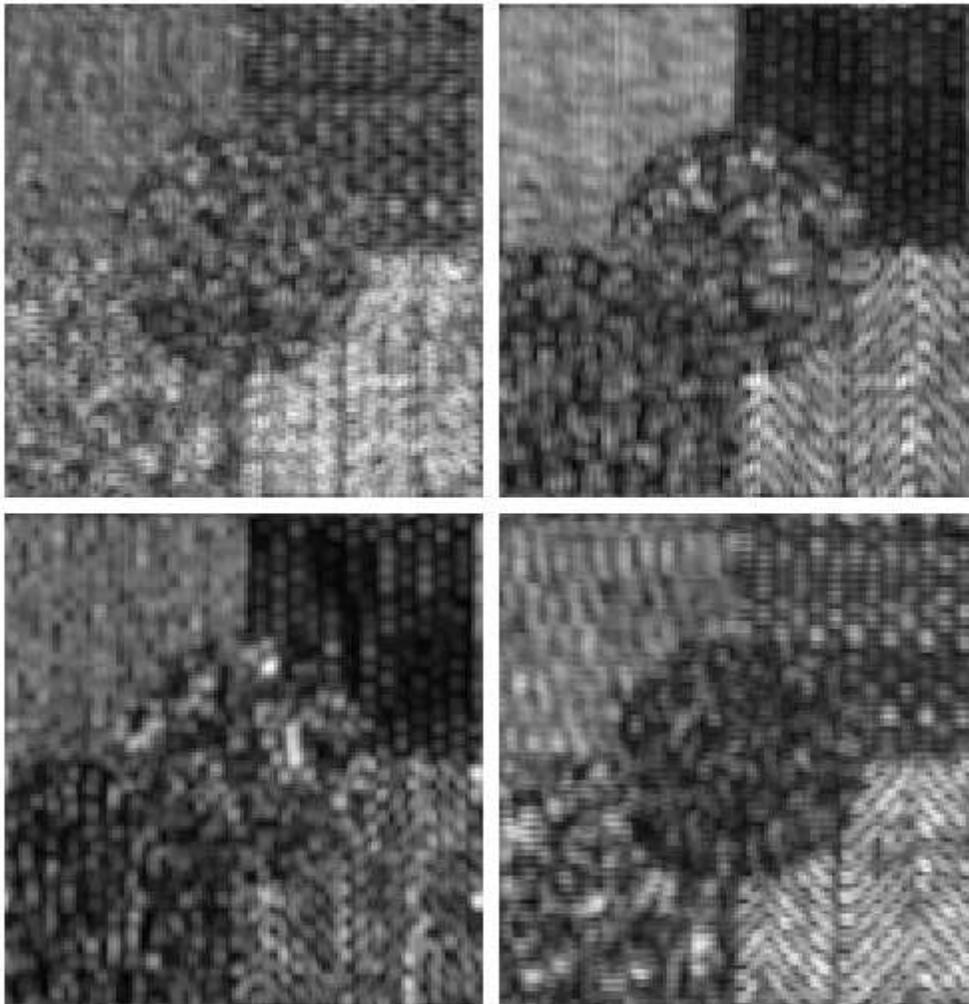
Laws 1 through Laws 4 masks (you may wish to add scaling factors from Pratt Fig. 16.6-9)

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



4. Repeat problem 3 for the image d24d77b.gif and for Laws 1, Laws 2, Laws 3, and Laws 4 filters, and plot the resulting set of 4 Laws images, as illustrated below. Do not forget to apply the standard deviation computation after the Laws mask filtering. Use a 3x3 window for computing this standard deviation, although you may wish to experiment with others. Rescale the resulting image rescaled to between 0 and 255 gray-levels. Plot the 4 images, as illustrated below.

Note that the image example below **IS NOT** for the proper image



Place all the answers to the following questions on a separate sheet of paper

5. Using Laws 1, Laws 2, Laws 3, and Laws 4 filters, form a 4-feature vector (Pratt Fig. 16.6-8) and find the B-distance (Pratt Eq. 16.1-3) between canvasb.gif and leather1b.gif . Also compute the error bound (Pratt Eq. 16.1-2) assuming that the two classes are equally likely ($P(S1) = P(S2) = 0.5$).

B-distance = ?

Error bound = ?

hints:

$$u := \begin{pmatrix} \text{mean}(\text{laws}_0) \\ \text{mean}(\text{laws}_1) \\ \text{mean}(\text{laws}_2) \\ \text{mean}(\text{laws}_3) \end{pmatrix}$$

$$R_{a,b} := \text{mean}(\overrightarrow{\text{laws}_a \text{ laws}_b})$$

$$\Sigma := R - u \cdot u^T$$

6. Using Laws 1 and Laws 2 filters, form a 2-feature vector (Pratt Fig. 16.6-8) and find the B-distance (Pratt Eq. 16.1-3) between canvasb.gif and leather1b.gif . Also compute the error bound (Pratt Eq. 16.1-2) assuming that the two classes are equally likely ($P(S1) = P(S2) = 0.5$).

B-distance = ?

Error bound = ?

7. Why is the answer in question 6 less than the answer in question 5?

8. Change the center frequency of the following Gabor filter to $(r_0, c_0) = (40, 25)$, and plot the new Gabor filtered output image **and** also plot the frequency spectrum of this Gabor filtered image (using the log magnitude as in previous projects). Notice the huge change in the filtered image output.

```
infile := READBMP("nat5.gif")    rr := 0..rows(infile) - 1    cc := 0..cols(infile) - 1
```

```
X := cfft(infile)
```

Gabor filter parameters: (r_0, c_0) = center frequency, sigma=std deviation of Gabor Gaussian

```
r0 := 25    c0 := 25    sigma := 10
```

$$g_{rr, cc} := e^{-\left[\frac{(rr-r_0)^2 + (cc-c_0)^2}{\sigma^2} \right]}$$

$Y := \overrightarrow{(g_{ab} \cdot X)}$ here we do the filtering in frequency domain

```
ygab := icfft(Y)    ygab :=  $\overrightarrow{|ygab|}$     maxy := max(ygab)
```

```
ygab :=  $\frac{ygab \cdot 250}{maxy}$     ygab :=  $\overrightarrow{\text{floor}(ygab)}$ 
```

infile

ygab