Project Description

Students are to write a computer program to perform edge detections on images. The name of the computer program is bmp_image.cpp. The computer program reads an image (*image*.bmp) as input; *image* is the name of the .bmp image as specified by the user. *Image***.**bmp must be in the workspace directory. The word "*image*" should be replaced with the appropriate image file name. The image file must be in .bmp format. If the image is color, the image is saved in grey level as "*image_grey*.bmp".

In addition to the file name, the computer program takes in an odd integer as input. The odd integer represents the side-length of a M x M mask used in the adaptive-threshold technique.

The computer program performs edge detection operations on "*image*_grey.bmp". The computer program applies the Roberts, Sobel, Prewitt, and Robinson operators on "*image*_grey.bmp". *Image*_Roberts.bmp, *image*_Sobel.bmp, *image*_Prewitt.bmp, and *image*_Robinson.bmp are generated. Then, the computer program thresholds each new image in two ways; globally and locally. Edge pixels are black and non-edge pixels are white. Black-and-white edge images are generated for various thresholds for each grey-image.

The computer program performs edge detections using the Laplacian of the Gaussian. The computer program applies the Laplacian of the Gaussian with mask sizes of 11x11 and 21x21 on the images "actress.bmp", "coins.bmp", and "pattern2.bmp". For each mask size, the computer program computes for various values of σ. The computer program uses zero crossings to detect the edges of the Laplacian of the Gaussian. The computer program generates images.

Project Background

Template Matching
The Roberts, Sobel, Prewitt, and Robinson operators are differential operators used to detect edges. The Roberts operator uses diagonally adjacent pixels to approximate a pixels gradient. This differs from the Sobel, Prewitt, and Robinson operators which use two 3x3 kernels to approximate a pixels gradient. Yet, all four operators are similar; they all compute an approximate gradient for each pixel. The Sobel, Prewitt, and Robinson operators each have two unique 3x3 kernels.

Global Thresholding
Global thresholds are applied to *image*_Roberts.bmp, *image*_Sobel.bmp, *image*_Prewitt.bmp, and *image*_Robinson.bmp. A global threshold is used to clip a gray-scale image into a binary image. If a pixel's intensity exceeds the threshold, set that pixel to black. If a pixel's intensity succeeds the threshold, set that pixel to white.

Global Threshold Selection
To select a global threshold a series of experiments is performed in which the thresheld image is examined as the threshold is adjusted, and the best results ascertained by eye.

Local Thresholding
Local thresholds are applied to *image*_Prewitt.bmp. Local thresholding involves analyzing intensities in the neighborhood of each pixel to determine the optimal local threshold level. Local thresholding employs a local window. Both the threshold and the local window size are variables. The local window is of size NxN where N is an odd integer.

Local Threshold Selection
To select a local threshold a series of experiments is performed in which the thresheld images are examined as both the threshold and local window size are adjusted, and the best results ascertained by eye.

Laplacian of Gaussian (LoG)
The LoG is composed of two parts; first the Gaussian, second the Laplacian. Applying the Gaussian to an image will smooth the image. The Gaussian has a tendency to blur images. This means that for impulse noises with spikes, the Gaussian will smear spikes over a sizable number of pixels (Gaussian smoothing). The Laplacian highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian of an image is a measure of the sum of the second partial derivatives. Edges are highly correlated to the partial second derivatives of the Gaussian.

Zero-crossing the LoG is an edge detection method. A pixel crosses-zero if it is significantly darker than the other pixels in the neighborhood. With LoG images, the sets of zero-crossing pixels tend to be the image-edges. Therefore, computer programs detect edges using the zero-crossing of the LoG of an image.

Algorithm Description

Template Matching

The Sobel, Prewitt, and Robinson operators uses two 3x3 kernels. The Roberts operator uses two 2x2 kernels. In either case, the two kernels are convoluted with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define **A** as the source image, $\mathbf{G}_x$ (horizontal gradient) as a point in an image formed by convolving with the first kernel, and $\mathbf{G}_y$ (vertical gradient) as a point in an image formed by convolving with the second kernel, then, for each pixel in **A,** there exists a gradient, **G**:

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

The only difference between the Sobel, Prewitt, Robinson, and Roberts operators are in the definitions of $\mathbf{G_x}$ and $\mathbf{G_y}$

Sobel:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

Prewitt:

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * \mathbf{A}$$

Robinson:

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix} * \mathbf{A}$$

Roberts:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} * \mathbf{A}$$

Global Thresholding
Threshold **G** to obtain the edge images. This computer program thresholds **G** globally or locally. To compute the global threshold of **G**,

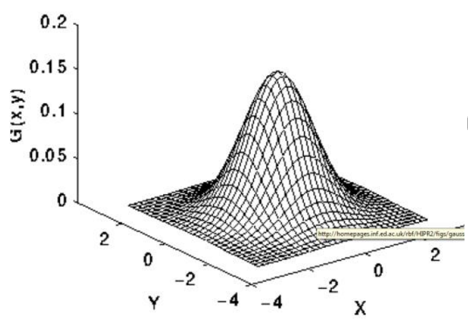
```
// G' is the thresheld image of G
// thresh is the global threshold
// thresh is adjusted via trial and error experimentation (Bisection Method)
for all pixels in image G do {
        [ [  if  ( G>thresh )  G' = 0;  else  G' = 1;  ] ]
}
```

Local Thresholding
To compute the local threshold of **G**, the computer program implements an adaptive thresholding algorithm:

```
// G' is the thresheld image of G.
// minrange is the minimum range.
// minrange is adjusted by the user via trial and error.
// T is the adaptive threshold.
// The size of the local-window can be adjusted by user
for all pixels in image G do {
        find minimum and maximum of local intensity distribution;
        range = maximum – minimum;
        if ( range > minrange )
                T = (minimum + maximum) / 2;
        else
                T = maximum – minrange / 2;
        If ( G > T ) G' = 255; else G' = 0;
}
```

The Laplacian of Gaussian

The Gaussian ("Bell Curve") may be used to reduce noise by blurring images.
Define the Gaussian of a xy-point to be:



$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

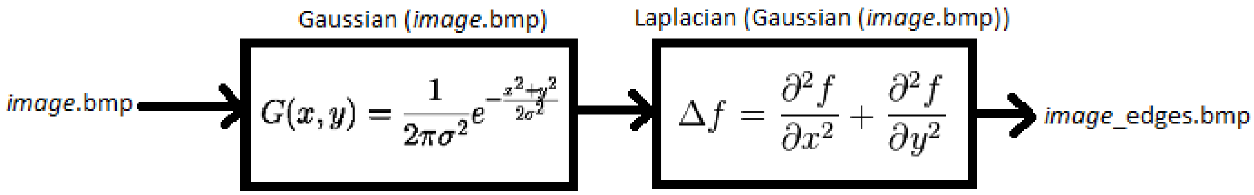
The Laplacian highlights regions of rapid intensity change; exploiting color-spatial locality to detect edges.
In mathematics, the Laplacian is a differential operator given by the divergence of the gradient of a function.

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The Laplacian of the Gaussian (LoG) closely defines the edges of an image. The LoG kernel can be pre calculated so only one convolution needs be performed per image.

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where the point (x,y) lies within the range of the LoG kernel. This mean that LoG is of two variables: σ (the standard deviation) and the-kernel-size. The zero-crossing of the LoG binarizes the LoG into edge and non-edge pixels where edge pixels are black and non-edge pixels are white. Moreover, a zero-crossing applies a threshold of value equal to zero.



Compute the Laplacian of Gaussian kernels for various kernel sizes and σ. Convolve *image*.bmp with each kernel, visually-selecting the most appealing combination of kernel size and σ.

Pseudo code: Compute LoG.

```
double sigma = 3.0;
double pi = 3.14159265;

// Perform the LoG in advance to obtain mask, LoG.
double** LoG = new double* [WSIZE];
for ( i = 0 ; i < WSIZE ; i++ ) {
        *(LoG + i) = new double[WSIZE];
}
// For each element in the LoG mask.
// half is the kernel-size (WSIZE) divided by 2.
for (k=-half; k<=half; k++) {
        for (m=-half; m<=half; m++) {
                // Perform the LoG(x,y)
                LoG[k+half][m+half] = ((-1)/(pi*sigma*sigma*sigma*sigma))*
                                      (1-(k*k+m*m)/(2*sigma*sigma))*
                                      (exp((-1)*(k*k+m*m)/(2*sigma*sigma)));
        }
}
```

Pseudo code: Zero-Crossing

```
double sum = 0;
// For each pixel
for (int j=0; j<yd; j++) {
for (int i=0; i<xd; i++) {
        sum = 0;
        for (k=-half; k<=half; k++) {
                for (m=-half; m<=half; m++) {
                        if (i+k>=0 && i+k<xd && j+m>=0 && j+m<yd) {
                        /* if pixel k,m of the window centered at i,j is
                           within [0,xd-1],[0,yd-1] */
                                sum += (LoG[k+half][m+half] * array1[RED][i+k][j+m]);
                        }
                }
        }
        if (sum < 0) *indexr = 255;
        else *indexr = 0;
        indexr++;
}
}
```

Show the values of the LoG masks:

11 x 11 σ = 1										
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0005	0.0007	0.0005	0.0001	0.0000	0.0000	0.0000
0.0000	0.0000	0.0003	0.0026	0.0085	0.0124	0.0085	0.0026	0.0003	0.0000	0.0000
0.0000	0.0001	0.0026	0.0175	0.0392	0.0431	0.0392	0.0175	0.0026	0.0001	0.0000
0.0000	0.0004	0.0086	0.0392	0.0000	-0.0965	0.0000	0.0392	0.0086	0.0004	0.0000
0.0000	0.0007	0.0123	0.0431	-0.0965	-0.3183	-0.0965	0.0431	0.0123	0.0007	0.0000
0.0000	0.0004	0.0086	0.0392	0.0000	-0.0965	0.0000	0.0392	0.0086	0.0004	0.0000
0.0000	0.0001	0.0026	0.0175	0.0391	0.0431	0.0391	0.0175	0.0026	0.0001	0.0000
0.0000	0.0000	0.0003	0.0026	0.0086	0.0124	0.0086	0.0026	0.0003	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.0005	0.0007	0.0005	0.0001	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

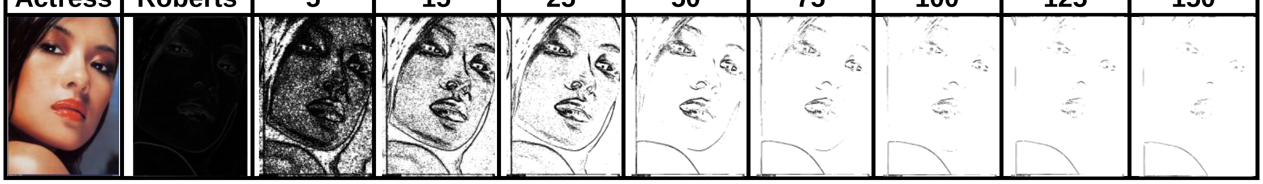
21 x 21 σ = 1																				
1e-042	1e-038	6e-035	1e-031	6e-029	1e-026	1e-024	4e-023	4e-022	2e-021	3e-021	2e-021	4e-022	4e-023	1e-024	1e-026	6e-029	1e-031	6e-035	1e-038	1e-042
1e-038	2e-034	7e-031	1e-027	7e-025	2e-022	1e-020	4e-019	5e-018	2e-017	3e-017	2e-017	5e-018	4e-019	1e-020	2e-022	7e-025	1e-027	7e-031	2e-034	1e-038
6e-035	7e-031	3e-027	5e-024	3e-021	7e-019	5e-017	2e-015	2e-014	8e-014	1e-013	8e-014	2e-014	2e-015	5e-017	7e-019	3e-021	5e-024	3e-027	7e-031	6e-035
1e-031	1e-027	5e-024	8e-021	5e-018	1e-015	8e-014	2e-012	3e-011	1e-010	2e-010	1e-010	3e-011	2e-012	8e-014	1e-015	5e-018	8e-021	5e-024	1e-027	1e-031
6e-029	7e-025	3e-021	5e-018	3e-015	5e-013	4e-011	1e-009	1e-008	5e-008	8e-008	5e-008	1e-008	1e-009	4e-011	5e-013	3e-015	5e-018	3e-021	7e-025	6e-029
1e-026	2e-022	7e-019	1e-015	5e-013	1e-010	8e-009	2e-007	2e-006	9e-006	1e-005	9e-006	2e-006	2e-007	8e-009	1e-010	5e-013	1e-015	7e-019	2e-022	1e-026
1e-024	1e-020	5e-017	8e-014	4e-011	8e-009	5e-007	1e-005	1e-004	5e-004	7e-004	5e-004	1e-004	1e-005	5e-007	8e-009	4e-011	8e-014	5e-017	1e-020	1e-024
4e-023	4e-019	2e-015	2e-012	1e-009	2e-007	1e-005	3e-004	3e-003	9e-003	1e-002	9e-003	3e-003	3e-004	1e-005	2e-007	1e-009	2e-012	2e-015	4e-019	4e-023
4e-022	5e-018	2e-014	3e-011	1e-008	2e-006	1e-004	3e-003	2e-002	4e-002	4e-002	4e-002	2e-002	3e-003	1e-004	2e-006	1e-008	3e-011	2e-014	5e-018	4e-022
2e-021	2e-017	8e-014	1e-010	5e-008	9e-006	5e-004	9e-003	4e-002	-0e+000	-1e-001	-0e+000	4e-002	9e-003	5e-004	9e-006	5e-008	1e-010	8e-014	2e-017	2e-021
3e-021	3e-017	1e-013	2e-010	8e-008	1e-005	7e-004	1e-002	4e-002	-1e-001	-3e-001	-1e-001	4e-002	1e-002	7e-004	1e-005	8e-008	2e-010	1e-013	3e-017	3e-021
2e-021	2e-017	8e-014	1e-010	5e-008	9e-006	5e-004	9e-003	4e-002	-0e+000	-1e-001	-0e+000	4e-002	9e-003	5e-004	9e-006	5e-008	1e-010	8e-014	2e-017	2e-021
4e-022	5e-018	2e-014	3e-011	1e-008	2e-006	1e-004	3e-003	2e-002	4e-002	4e-002	4e-002	2e-002	3e-003	1e-004	2e-006	1e-008	3e-011	2e-014	5e-018	4e-022
4e-023	4e-019	2e-015	2e-012	1e-009	2e-007	1e-005	3e-004	3e-003	9e-003	1e-002	9e-003	3e-003	3e-004	1e-005	2e-007	1e-009	2e-012	2e-015	4e-019	4e-023
1e-024	1e-020	5e-017	8e-014	4e-011	8e-009	5e-007	1e-005	1e-004	5e-004	7e-004	5e-004	1e-004	1e-005	5e-007	8e-009	4e-011	8e-014	5e-017	1e-020	1e-024
1e-026	2e-022	7e-019	1e-015	5e-013	1e-010	8e-009	2e-007	2e-006	9e-006	1e-005	9e-006	2e-006	2e-007	8e-009	1e-010	5e-013	1e-015	7e-019	2e-022	1e-026
6e-029	7e-025	3e-021	5e-018	3e-015	5e-013	4e-011	1e-009	1e-008	5e-008	8e-008	5e-008	1e-008	1e-009	4e-011	5e-013	3e-015	5e-018	3e-021	7e-025	6e-029
1e-031	1e-027	5e-024	8e-021	5e-018	1e-015	8e-014	2e-012	3e-011	1e-010	2e-010	1e-010	3e-011	2e-012	8e-014	1e-015	5e-018	8e-021	5e-024	1e-027	1e-031
6e-035	7e-031	3e-027	5e-024	3e-021	7e-019	5e-017	2e-015	2e-014	8e-014	1e-013	8e-014	2e-014	2e-015	5e-017	7e-019	3e-021	5e-024	3e-027	7e-031	6e-035
1e-038	2e-034	7e-031	1e-027	7e-025	2e-022	1e-020	4e-019	5e-018	2e-017	3e-017	2e-017	5e-018	4e-019	1e-020	2e-022	7e-025	1e-027	7e-031	2e-034	1e-038
1e-042	1e-038	6e-035	1e-031	6e-029	1e-026	1e-024	4e-023	4e-022	2e-021	3e-021	2e-021	4e-022	4e-023	1e-024	1e-026	6e-029	1e-031	6e-035	1e-038	1e-042

Results and Analysis

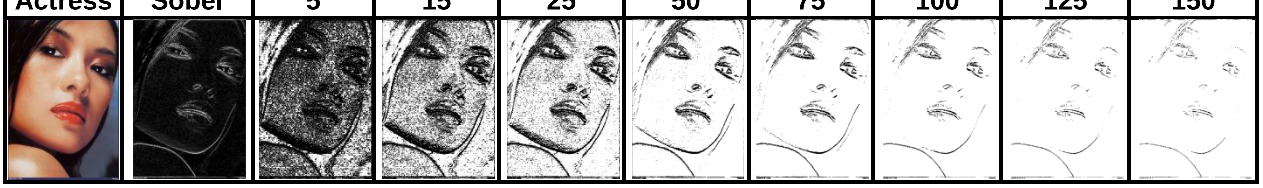
Template Matching

	Actress Edge Detection				
	Roberts	Sobel	Prewitt	Robinson-3	Robinson-5
Global Thresh	✓	✓	✓	✓	✓
Local Thresh	✗	✗	✓	✗	✗

Actress.bmp Roberts Global Threshold									
Actress	Roberts	5	15	25	50	75	100	125	150



Actress.bmp Sobel Global Threshold									
Actress	Sobel	5	15	25	50	75	100	125	150



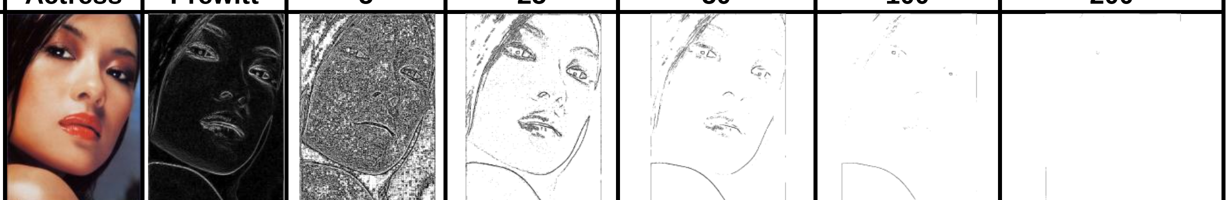
EDGE DETECTION

Chris Upton

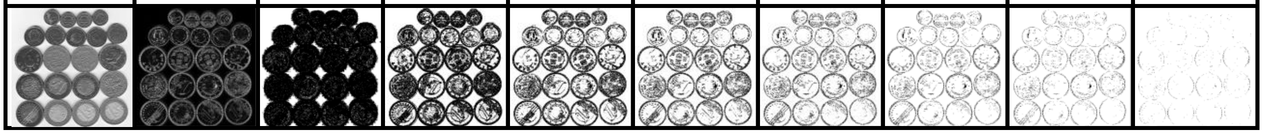
Actress.bmp Prewitt Global Threshold									
Actress	Prewitt	5	15	25	50	75	100	125	150

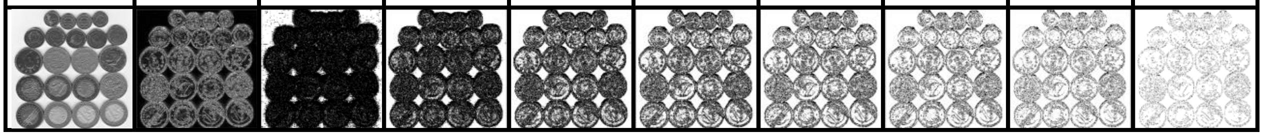
Actress.bmp Robinson3 Global Threshold									
Actress	Robinson3	5	15	25	50	75	100	125	150

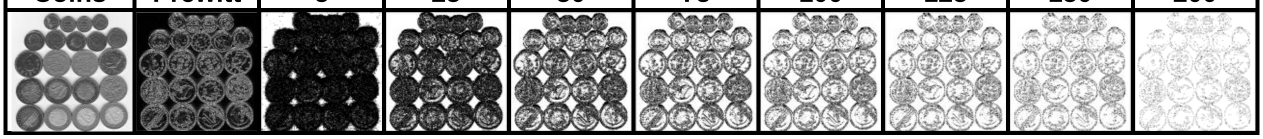
Actress.bmp Robinson5 Global Threshold									
Actress	Robinson5	5	15	25	50	75	100	125	150

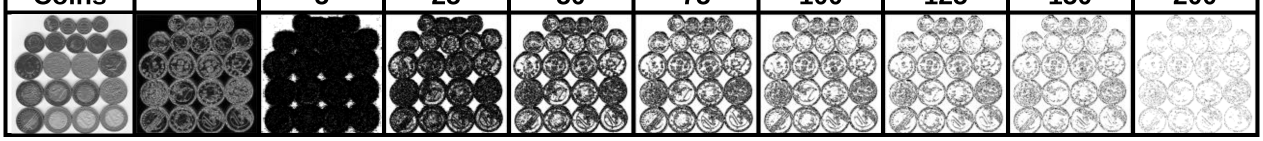
	Actress.bmp Prewitt Local Threshold (minrange)						
	Actress	Prewitt	5	25	50	100	200
5 x 5							
9 x 9							
13 x 13							

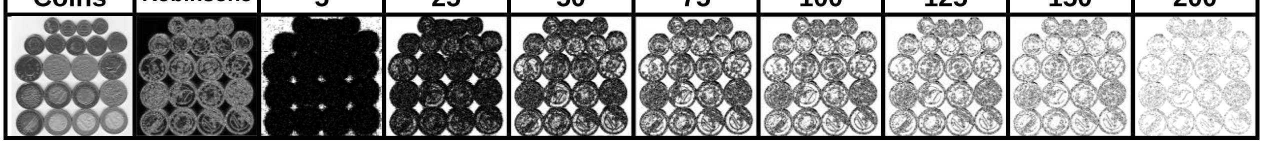
	Coins Edge Detection				
	Roberts	Sobel	Prewitt	Robinson-3	Robinson-5
Global Thresh	✓	✓	✓	✓	✓
Local Thresh	✗	✗	✓	✗	✗

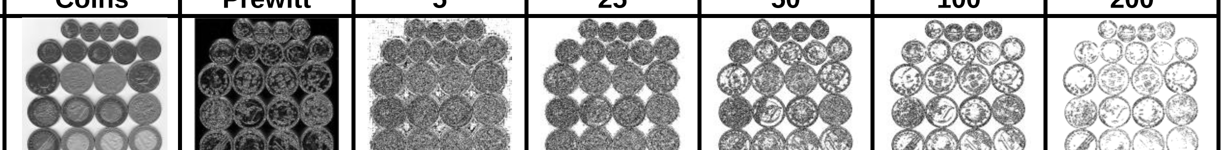
Coins.bmp Roberts Global Threshold									
Coins	Roberts	5	25	50	75	100	125	150	200
									

Coins.bmp Sobel Global Threshold									
Coins	Sobel	5	25	50	75	100	125	150	200
									

Coins.bmp Prewitt Global Threshold									
Coins	Prewitt	5	25	50	75	100	125	150	200
									

Coins.bmp Robinson3 Global Threshold									
Coins	Robinson3	5	25	50	75	100	125	150	200
									

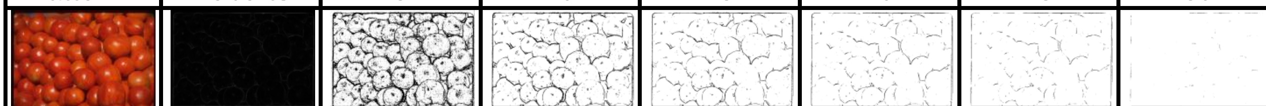
Coins.bmp Robinson5 Global Threshold									
Coins	Robinson5	5	25	50	75	100	125	150	200
									

	Coins.bmp Prewitt Local Threshold (minrange)						
	Coins	Prewitt	5	25	50	100	200
5 x 5							
9 x 9							

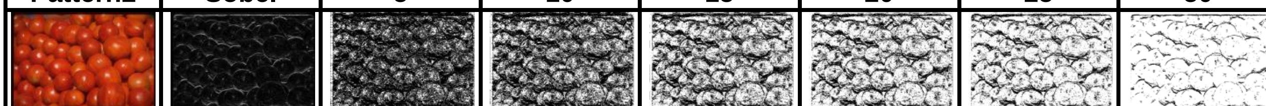
13 x 13

	Pattern2 Edge Detection				
	Roberts	Sobel	Prewitt	Robinson-3	Robinson-5
Global Thresh	✓	✓	✓	✓	✓
Local Thresh	✗	✗	✓	✗	✗

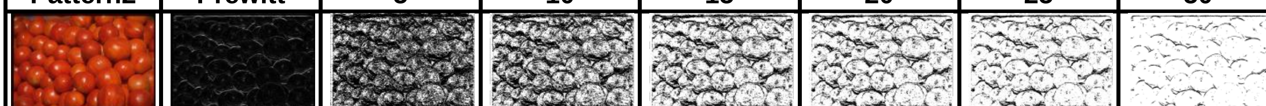
Pattern2.bmp Roberts Global Threshold							
Pattern2	**Roberts**	**5**	**10**	**15**	**20**	**25**	**50**



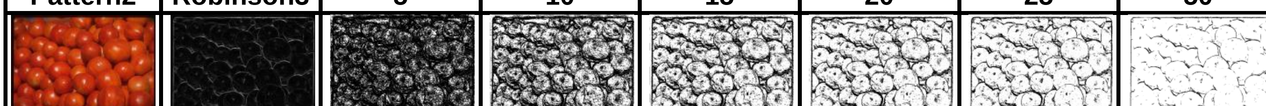
Pattern2.bmp Sobel Global Threshold							
Pattern2	**Sobel**	**5**	**10**	**15**	**20**	**25**	**50**



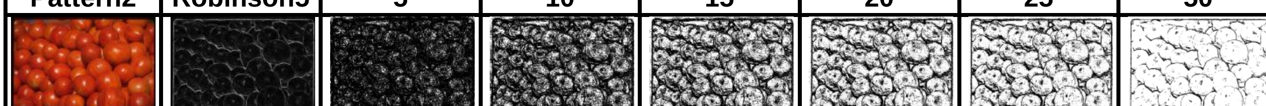
Pattern2.bmp Prewitt Global Threshold							
Pattern2	**Prewitt**	**5**	**10**	**15**	**20**	**25**	**50**



Pattern2.bmp Robinson3 Global Threshold							
Pattern2	**Robinson3**	**5**	**10**	**15**	**20**	**25**	**50**

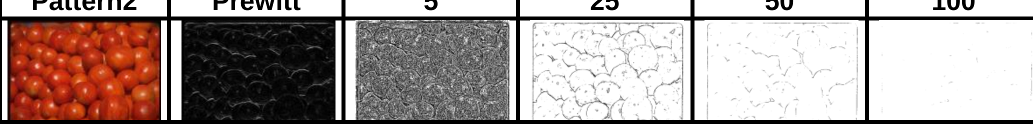


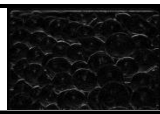
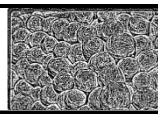
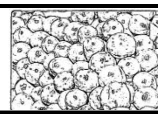
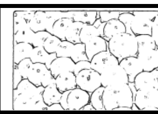
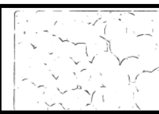
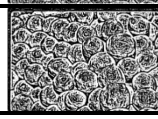
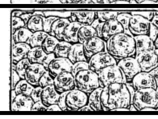
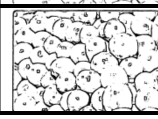
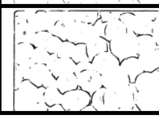
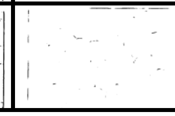
Pattern2.bmp Robinson5 Global Threshold							
Pattern2	**Robinson5**	**5**	**10**	**15**	**20**	**25**	**50**



Pattern2.bmp Prewitt Local Threshold (minrange)							
	Pattern2	**Prewitt**	**5**	**25**	**50**	**100**	**200**

5 x 5



9 x 9							
13 x 13							

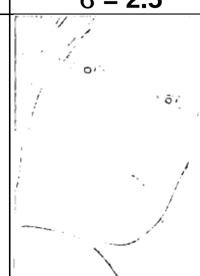
Observations (Global Thresholding):

- The Roberts, Sobel, Prewitt, Robinson3 and Robinson5 operations are real edge detectors.
- The Roberts, Sobel, Prewitt, Robinson3 and Robinson5 operations are unique edge detectors.
- The Robinson5 operator is a better edge detector than Robinson3.
- The Robinson3 operator is a better edge detector than Sobel.
- The Sobel operator is the least effective at eliminating noise.
- The Roberts operator is the most effective at eliminating noise.
- The Roberts operator is the least effective at detecting edges.
- The Robinson5 operator is the most effective at detecting edges.
- Robinson3 is less noisy than Robinson5, however Robinson5's edges are more defined than Robinson3's.
- The threshold that produces the optimal edge detection is a variable of the image.

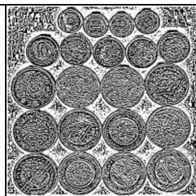
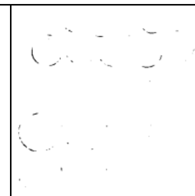
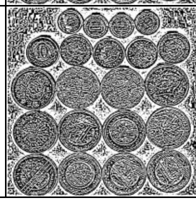
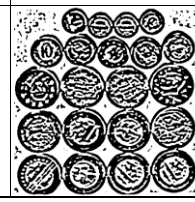
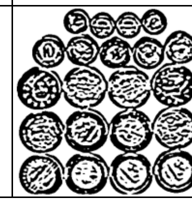
Observations (Local Thresholding):

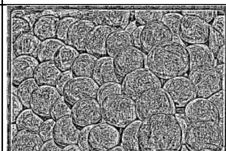
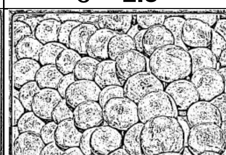
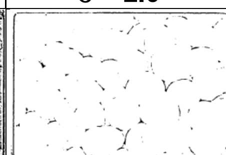
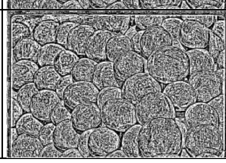
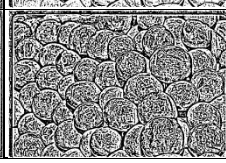
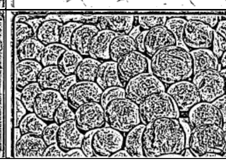
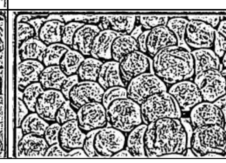
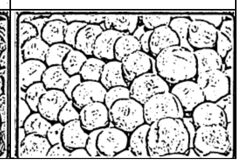
- Compared to global thresholding, adaptive (local) thresholding is an effective edge detection technique.
- The optimal window-size is a variable of the image.
- The optimal minrange is a variable of the image.
- Increasing the window-size thickens the edges.

Laplacian of Gaussian

	Actress.bmp				
	σ = 1.0	**σ = 1.5**	**σ = 2.0**	**σ = 2.5**	**σ = 3.0**
11 x 11					
21 x 21					

	Coins.bmp				
	σ = 1.0	**σ = 1.5**	**σ = 2.0**	**σ = 2.5**	**σ = 3.0**

11 x 11					
21 x 21					

	Pattern2.bmp				
	σ = 1.0	**σ = 1.5**	**σ = 2.0**	**σ = 2.5**	**σ = 3.0**
11 x 11					
21 x 21					

Observations:
- The zero-crossing of the Laplacian of the Gaussian is an edge detection mechanism.
- Increasing the mask size results in thicker edges and thicker noise.
- Increasing σ eliminates noise; however, increasing σ too much will eliminate edges.
- There exists an optimal pair (**σ,**mask-size) for each and every image.

Conclusion

Because local thresholding is a more effective edge detecting method than global thresholding, we may conclude that exploiting spacial locality can only improve edge detection; in the worst case, local (adaptive) thresholding will be at least as effective as global thresholding.

Template Matching and the Laplacian of the Gaussian are two edge detecting methods. With local thresholding, increasing the window size increases the width of the edges. With the Laplacian of the Gaussian, increasing the window size increases the width of the edges.

With the Laplacian of the Gaussian edge detection method; increasing the mask size results in thicker edges and thicker noise, and increasing σ eliminates noise, however, increasing σ too much will eliminate edges.