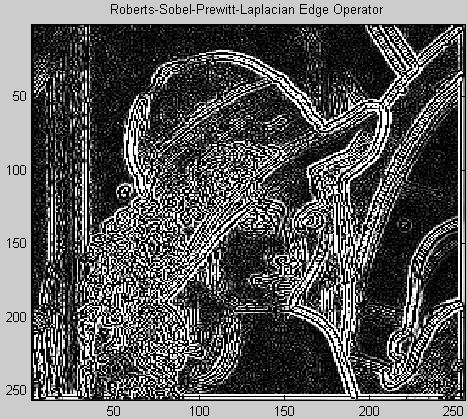
EECS 5330 Image Analysis and Computer Vision

Assignment 2

Edge Detection Operators

By Edris Amin

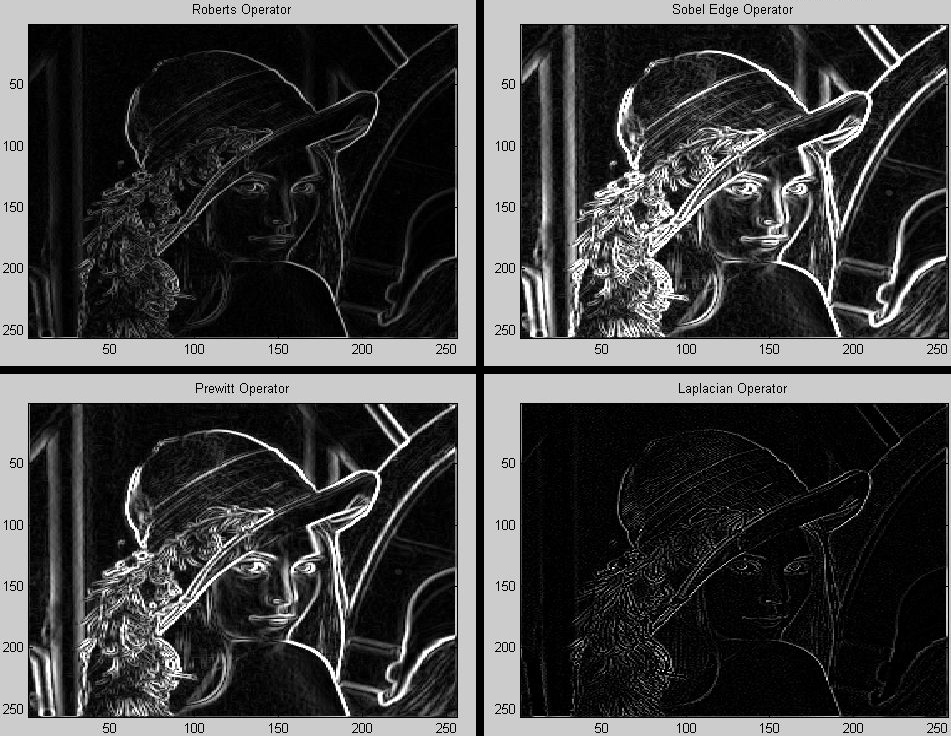


For this assignment different edge operators were applied to the original Lenna image. The four assigned operators were 1. Robert’s, 2. Sobel, 3. Prewitt, and 4. Laplacian. The image on the cover of this report is a result of performing all four edge operators to the original Lenna image in the order listed. The images in the figure below are the resulting images after performing each of the four operators separately. The MATLAB code for producing the resulting images is provided at the end of this report. The code for each operator is proceeded by a comment formatted like so:

%=======================|

%XXXXXXX Edge Operator |

%=======================|



For the Roberts operator the last column and row (right most, bottom) of pixels were left unchanged because they didn’t have valid neighbors to perform the operation. For the other three operators the pixels around the border of the image were left unchanged for the same reason. MATLAB considers the top left corner of an image as the origin of the (i,j) axis system. With those considerations the following equations were used to produce the results shown.

The code given below was run in MATLAB to produce the images with detected edges.

clear all;

load lenna;

colormap(gray(256));

%=======================|

%Roberts Edge Operator |

%=======================|

roberts1 = zeros(256, 256);%image for squareroot method

%last row and column are ignored for this operator (i = 256; j = 256)

%because for them no "next" pixel exists

%just copy them from lenna

for i=1:256 %copy row 256

roberts1(i,256) = lenna(i,256);

end

for j=1:256 %copy column 256

roberts1(256, j) = lenna(256, j);

end

for i=1:255

for j=1:255

roberts1(i,j) = sqrt((lenna(i+1, j+1)-lenna(i, j))^2 + (lenna(i, j+1)-lenna(i+1, j))^2);

%take the gradient of the downward slope - the upward slope

% sqrt( \^2 + /^2 )

end

end

subplot(2, 2, 1); image(roberts1);

title('Roberts Operator');

%=======================|

%Sobel Operator |

%=======================|

sobel1 = zeros(256, 256);%image for squareroot method

%outer row and column are ignored for this operator (i = 1; j = 1; i = 256; j = 256)

%because they do not have proper neighbors for the operations.

%So they will just be copied with out changes from lenna

for i=1:256 %copy row 1 & 256

sobel1(i,1) = lenna(i,1);

sobel1(i,256) = lenna(i,256);

end

for j=1:256 %copy column 1 & 256

sobel1(1, j) = lenna(1, j);

sobel1(256, j) = lenna(256, j);

end

%maskX

% |-1 0 1 |

% |-2 p 2 |

% |-1 0 1 |

% Dxf(i,j) = (-lenna(i-1,j-1)-2\*lenna(i-1, j)-lenna(i-1, j+1)

% +lenna(i+1,j-1)+2\*lenna(i+1, j)+lenna(i+1, j+1))

Dxf = zeros(256,256);

for i = 2:255

for j = 2:255

Dxf(i,j) = (-lenna(i-1,j-1)-2\*lenna(i-1, j)-lenna(i-1, j+1)+lenna(i+1,j-1)+2\*lenna(i+1, j)+lenna(i+1, j+1));

end

end

%maskY

% |-1 -2 -1

% | 0 p 0

% | 1 2 1

% Dyf(i,j) = (-lenna(i-1,j-1)-2\*lenna(i,j-1)-lenna(i+1,j-1)

% +lenna(i-1,j+1)-2\*lenna(i,j+1)-lenna(i+1,j+1))

Dyf = zeros(256, 256);

for i = 2:255

for j = 2:255

Dyf(i,j) = (-lenna(i-1,j-1)-2\*lenna(i,j-1)-lenna(i+1,j-1)+lenna(i-1,j+1)+2\*lenna(i,j+1)+lenna(i+1,j+1));

end

end

%Sobel1 gradient = sqrt( (Dxf)^2 + (Dyf)^2 )

for i=2:255

for j=2:255

sobel1(i,j) = sqrt( (-lenna(i-1,j-1)-2\*lenna(i-1, j)-lenna(i-1, j+1)+lenna(i+1,j-1)+2\*lenna(i+1, j)+lenna(i+1, j+1))^2 + (-lenna(i-1,j-1)-2\*lenna(i,j-1)-lenna(i+1,j-1)+lenna(i-1,j+1)+2\*lenna(i,j+1)+lenna(i+1,j+1))^2);

%sobel1(i,j) = sqrt( (Dxf(i,j))^2 + (Dyf(i,j))^2);

end

end

subplot(2, 2, 2); image(sobel1);

title('Sobel Edge Operator');

%=======================|

%Prewitt Edge Operator |

%=======================|

Prewitt1 = zeros(256, 256);%image for squareroot method

%outer row and column are ignored for this operator (i = 1; j = 1; i = 256; j = 256)

%because they do not have proper neighbors for the operations.

%So they will just be copied with out changes from lenna

for i=1:256 %copy row 1 & 256

Prewitt1(i,1) = lenna(i,1);

Prewitt1(i,256) = lenna(i,256);

end

for j=1:256 %copy column 1 & 256

Prewitt1(1, j) = lenna(1, j);

Prewitt1(256, j) = lenna(256, j);

end

%maskX

% |-1 0 1 |

% |-1 p 1 |

% |-1 0 1 |

% Dxf(i,j) = (-lenna(i-1,j-1)-lenna(i-1, j)-lenna(i-1, j+1)

% +lenna(i+1,j-1)+lenna(i+1, j)+lenna(i+1, j+1))

Dxf = zeros(256,256);

for i = 2:255

for j = 2:255

Dxf(i,j) = (-lenna(i-1,j-1)-lenna(i-1, j)-lenna(i-1, j+1)+lenna(i+1,j-1)+lenna(i+1, j)+lenna(i+1, j+1));

end

end

%maskY

% |-1 -1 -1

% | 0 p 0

% | 1 1 1

% Dyf(i,j) = (-lenna(i-1,j-1)-lenna(i,j-1)-lenna(i+1,j-1)

% +lenna(i-1,j+1)-lenna(i,j+1)-lenna(i+1,j+1))

Dyf = zeros(256, 256);

for i = 2:255

for j = 2:255

Dyf(i,j) = (-lenna(i-1,j-1)-lenna(i,j-1)-lenna(i+1,j-1)+lenna(i-1,j+1)+lenna(i,j+1)+lenna(i+1,j+1));

end

end

%Prewitt1 gradient = sqrt( (Dxf)^2 + (Dyf)^2 )

for i=2:255

for j=2:255

Prewitt1(i,j) = sqrt( (-lenna(i-1,j-1)-lenna(i-1, j)-lenna(i-1, j+1)+lenna(i+1,j-1)+lenna(i+1, j)+lenna(i+1, j+1))^2 + (-lenna(i-1,j-1)-lenna(i,j-1)-lenna(i+1,j-1)+lenna(i-1,j+1)+lenna(i,j+1)+lenna(i+1,j+1))^2);

%Prewitt1(i,j) = sqrt( (Dxf(i,j))^2 + (Dyf(i,j))^2);

end

end

subplot(2, 2, 3); image(Prewitt1);

title('Prewitt Operator');

%=======================|

%Laplacian Edge Operator|

%=======================|

Laplacian1 = zeros(256, 256);%image for squareroot method

%outer row and column are ignored for this operator (i = 1; j = 1; i = 256; j = 256)

%because they do not have proper neighbors for the operations.

%So they will just be copied with out changes from lenna

for i=1:256 %copy row 1 & 256

Laplacian1(i,1) = lenna(i,1);

Laplacian1(i,256) = lenna(i,256);

end

for j=1:256 %copy column 1 & 256

Laplacian1(1, j) = lenna(1, j);

Laplacian1(256, j) = lenna(256, j);

end

%mask1

% |0 1 0 |

% |1 -4 1 |

% |0 1 0 |

% Df(i,j) = (lenna(i,j-1)+lenna(i-1,j)+lenna(i+1,j)

% +lenna(i,j+1)-4\*lenna(i,j))

Df = zeros(256,256);

for i = 2:255

for j = 2:255

Df(i,j) = (lenna(i,j-1)+lenna(i-1,j)+lenna(i+1,j)+lenna(i,j+1)-4\*lenna(i,j));

end

end

for i=2:255

for j=2:255

Laplacian1(i,j) = (lenna(i,j-1)+lenna(i-1,j)+lenna(i+1,j)+lenna(i,j+1)-4\*lenna(i,j));

%Laplacian1(i,j) = (lenna(i,j-1)+lenna(i-1,j)+lenna(i+1,j)+lenna(i,j+1)-4\*lenna(i,j));

%Laplacian1(i,j) = sqrt( (Df1(i,j))^2);

end

end

subplot(2, 2, 4); image(Laplacian1);

title('Laplacian Operator');