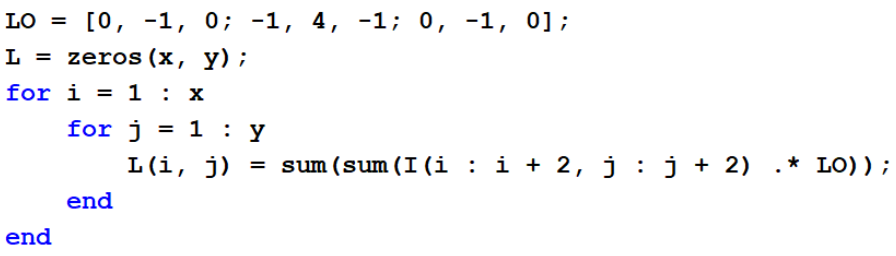
CPS566 Image Processing
University of Dayton
Department of Computer Science
Spring 2019
Project-3

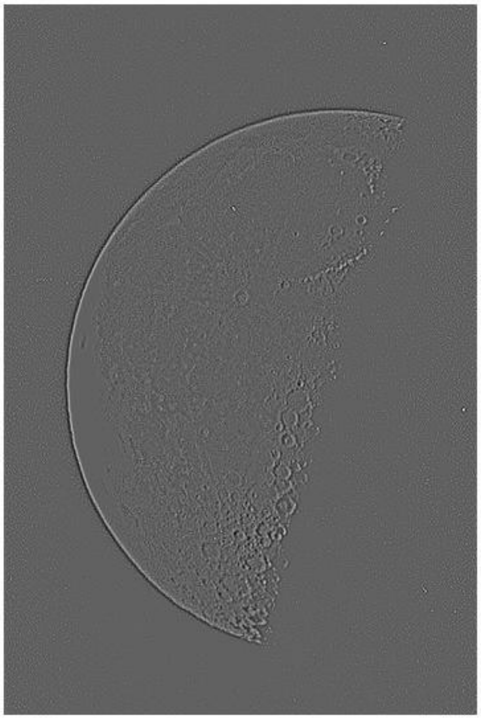
Submitter: Dhaval Kadia [101622808]

Original Image

1. Laplacian (second order derivative) of the image

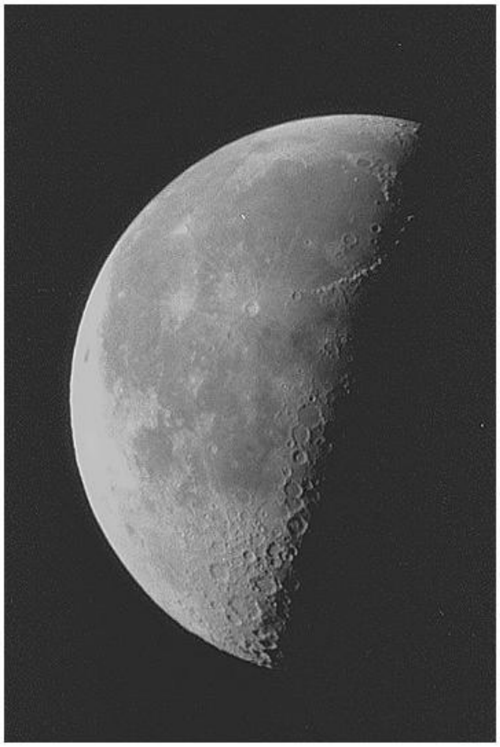


```
LO = [0, -1, 0; -1, 4, -1; 0, -1, 0];
L = zeros(x, y);
for i = 1 : x
    for j = 1 : y
        L(i, j) = sum(sum(I(i : i + 2, j : j + 2) .* LO));
    end
end
```

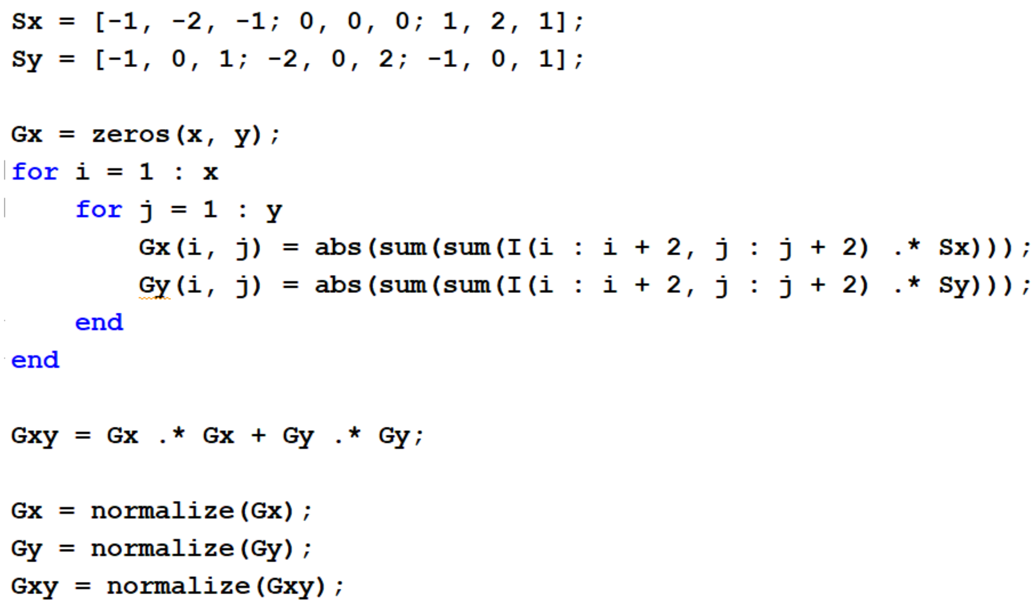


2. Sharpened image by adding Laplacian with the input

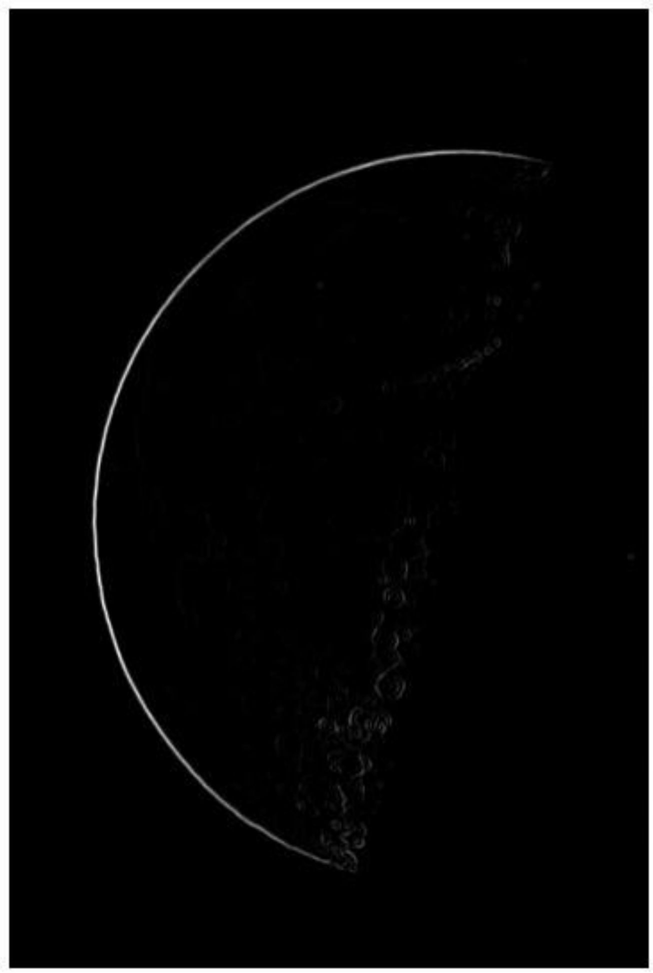
```
S = img + L;
```



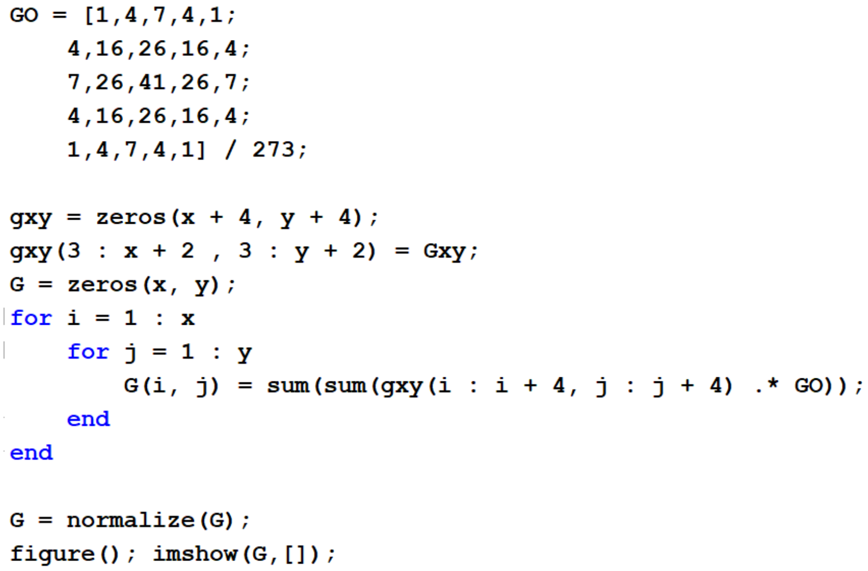
3. Gradient image of the input by using Sobel operators



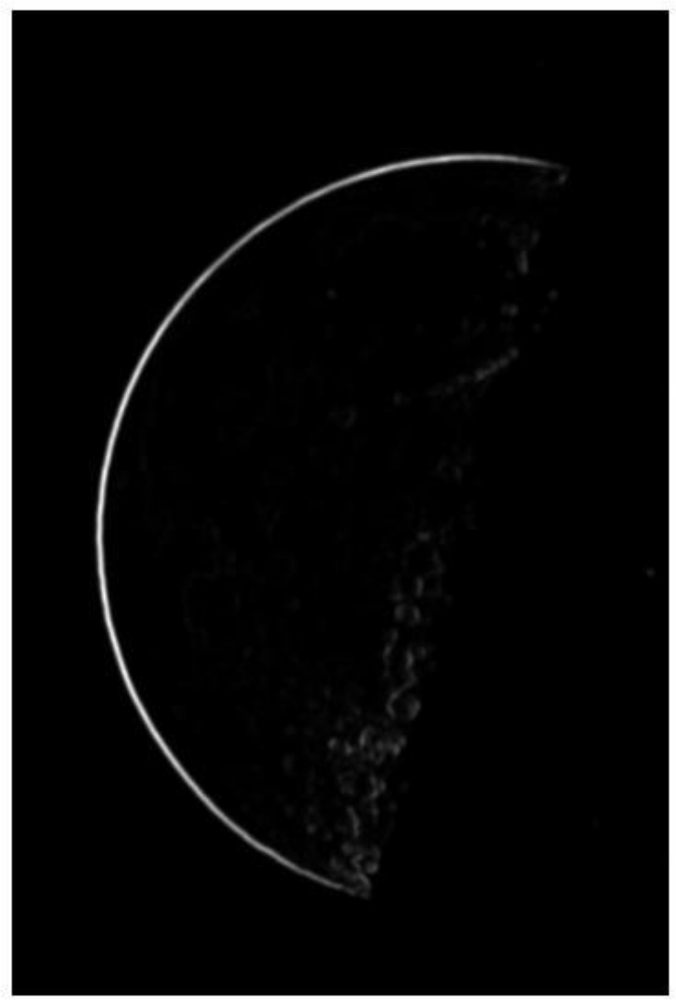
```
Sx = [-1, -2, -1; 0, 0, 0; 1, 2, 1];
Sy = [-1, 0, 1; -2, 0, 2; -1, 0, 1];

Gx = zeros(x, y);
for i = 1 : x
    for j = 1 : y
        Gx(i, j) = abs(sum(sum(I(i : i + 2, j : j + 2) .* Sx)));
        Gy(i, j) = abs(sum(sum(I(i : i + 2, j : j + 2) .* Sy)));
    end
end

Gxy = Gx .* Gx + Gy .* Gy;

Gx = normalize(Gx);
Gy = normalize(Gy);
Gxy = normalize(Gxy);
```



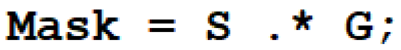
4. Smoothened gradient image by using a 5×5 window



```
GO = [1,4,7,4,1;
    4,16,26,16,4;
    7,26,41,26,7;
    4,16,26,16,4;
    1,4,7,4,1] / 273;

gxy = zeros(x + 4, y + 4);
gxy(3 : x + 2 , 3 : y + 2) = Gxy;
G = zeros(x, y);
for i = 1 : x
    for j = 1 : y
        G(i, j) = sum(sum(gxy(i : i + 4, j : j + 4) .* GO));
    end
end

G = normalize(G);
figure(); imshow(G,[]);
```

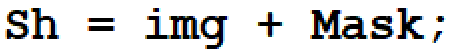


5. Mask image by multiplying the results in steps 2 and 4



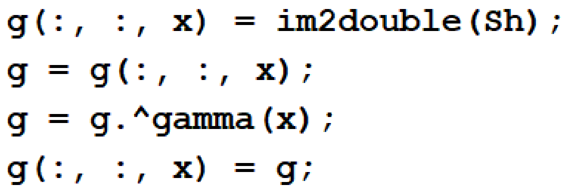
```
Mask = S .* G;
```

6. Sharpened image by adding the input with the Mask image

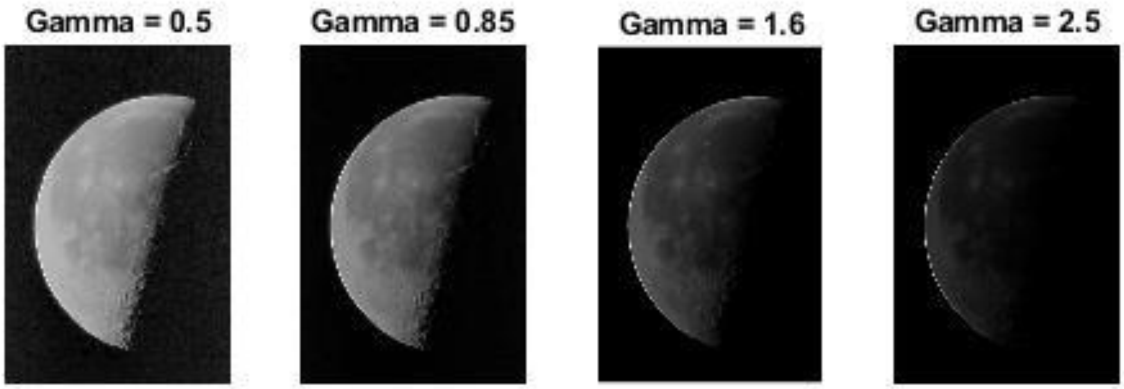


```
Sh = img + Mask;
```

7. Power-law transformation (Gamma = 0.5, 0.85, 1.6, 2.5)



```
g(:, :, x) = im2double(Sh);
g = g(:, :, x);
g = g.^gamma(x);
g(:, :, x) = g;
```



Adjusting the image

```
function N = normalize(n)
    N = n;
    minR = min(N(:));
    maxR = max(N(:));
    N = (N - minR) / (maxR - minR);
end
```