

## E3 - Filtering

Integrative filtering

```
I = imread('lena_gray_512.tif');
h = [1 1 1; 1 1 1; 1 1 1];
h = h/sum(sum(h)); % filtros integrativos; suma=1
J = imfilter(I,h);
montage({I,J});
```



Gaussian filtering

```
h = fspecial('gaussian',10,10/5);
J = imfilter(I,h);
montage({I,J});
```



## Non-linear filtering

```
In = imnoise(I,'salt & pepper',0.02);
h = fspecial('gaussian',5,5/5);
Ig = imfilter(In,h);
Im = medfilt2(In,[3,3]);
montage({I,In,Ig,Im});
```



## High pass filtering

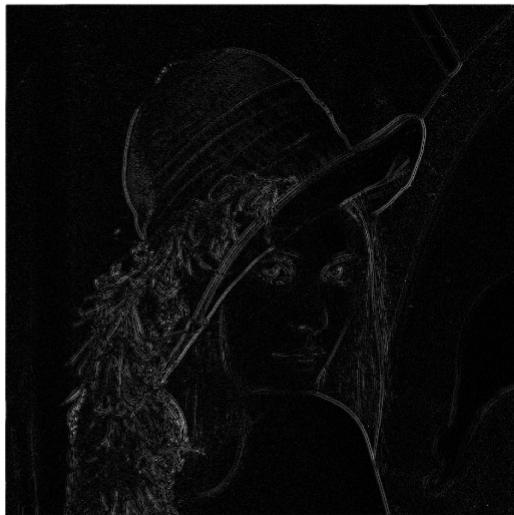
```
I = imread('lena_gray_512.tif');
h = fspecial('gaussian',5,5/5);
J = imfilter(I,h);
D = I - J;
imshow(D, [])
```



```
D2 = imcomplement(I-J);
imshow(D2, [])
```



```
D3 = abs(double(I) - double(J));
imshow(D3, [])
```

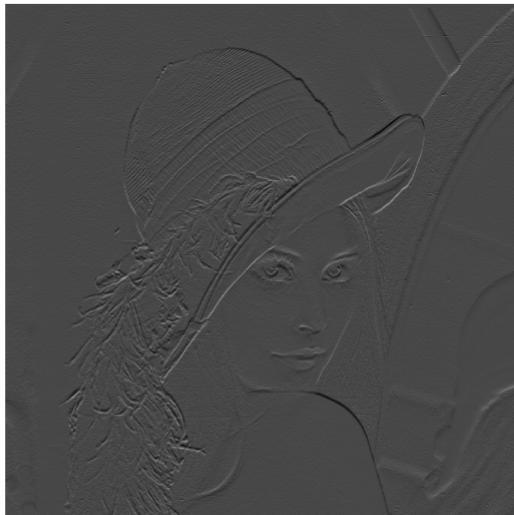


#### Derivative filtering

```
%gradient in x  
hx = [1, -1];  
Gx = imfilter(double(I),hx);  
imshow(Gx, [])
```

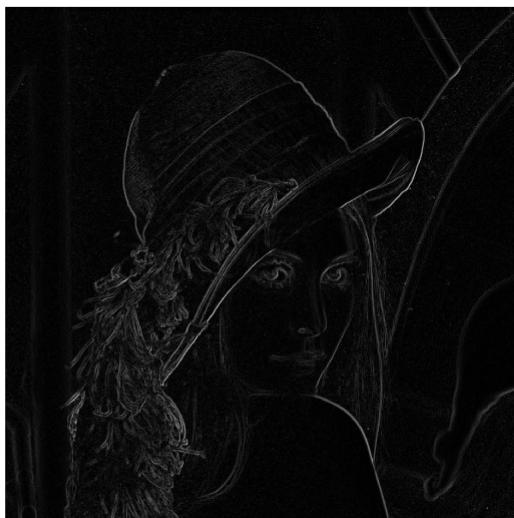


```
%gradient in y  
hy = [1; -1];  
Gy = imfilter(double(I),hy);  
imshow(Gy, [])
```



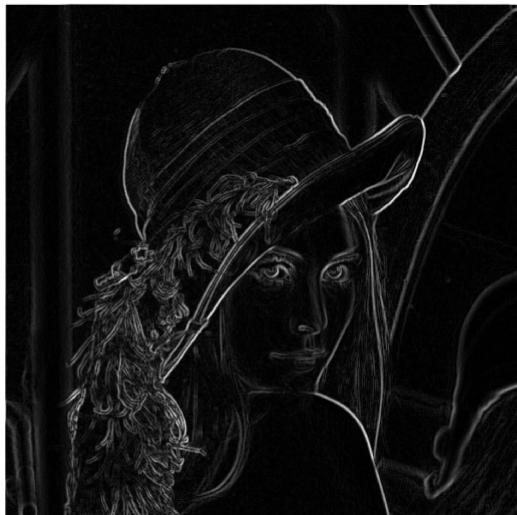
Module between gradient in axis

```
Mod = abs(Gx) + abs(Gy);  
imshow(Mod, [])
```



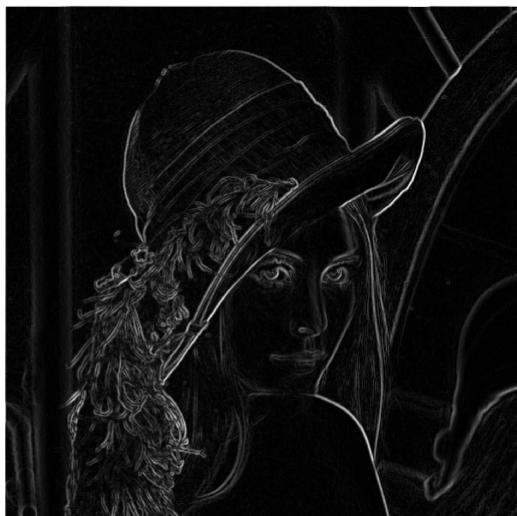
Prewitt operator

```
hx = [1 0 -1; 1 0 -1; 1 0 -1];  
Gx = imfilter(double(I),hx);  
hy = hx';  
Gy = imfilter(double(I),hy);  
Prewitt = (abs(Gx) + abs(Gy))/2;  
imshow(Prewitt, [])
```



Sobel operator

```
hx = [1 0 -1; 2 0 -2; 1 0 -1];
Gx = imfilter(double(I),hx);
hy = hx';
Gy = imfilter(double(I),hy);
Sobel = (abs(Gx) + abs(Gy));
imshow(Sobel, [])
```



Laplacian operator

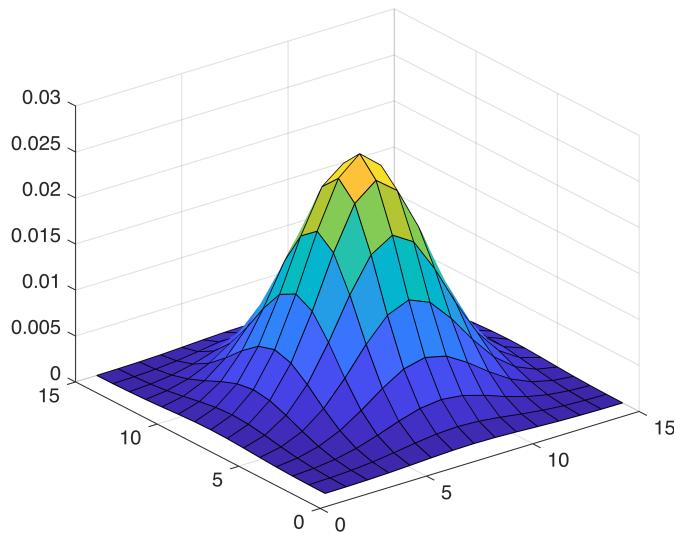
```
hL = [0 1 0; 1 -4 1; 0 1 0];
L = imfilter(double(I), hL);
```

```
imshow(L, [])
```



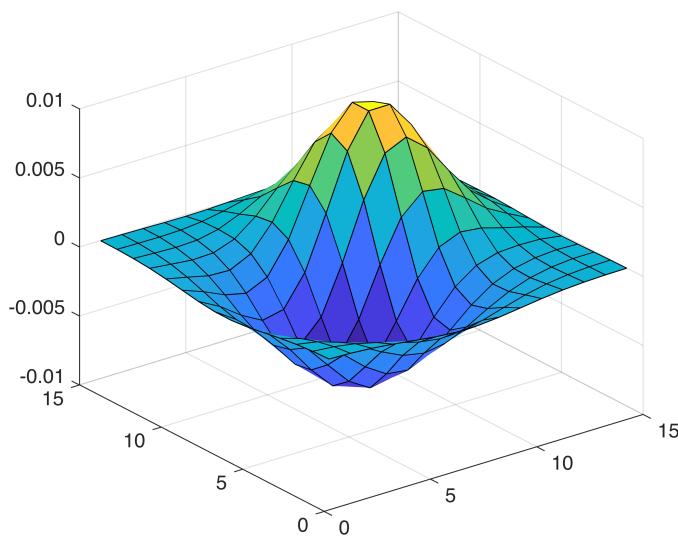
Laplacian of Gaussian

```
%Gaussian funtions  
N = 15;  
h = fspecial('gaussian',N,N/6);  
[X, Y] = meshgrid(1:N,1:N);  
surf(X,Y,h);
```

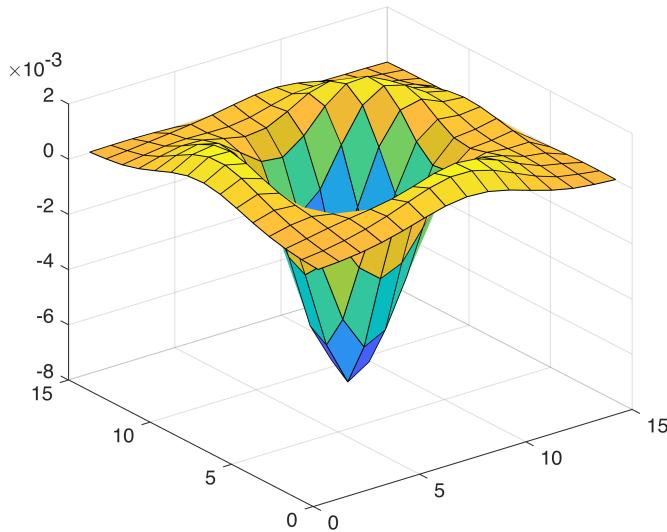


```
%First derivatives  
hx = [1 -1];  
hy = hx';  
dx = imfilter(h,hx,'replicate');
```

```
dy = imfilter(h,hy,'replicate');
d1 = dx+dy;
surf(X,Y,d1);
```



```
%Second derivatives
dx2 = imfilter(dx,hx,'replicate');
dy2 = imfilter(dy,hy,'replicate');
hL = dx2+dy2;
surf(X,Y,hL);
```



```
%Laplacian Lena
I = imread('lena_gray_512.tif');
LL = imfilter(double(I),hL);
```

```
imshow(LL, [])
```



### Canny operator

```
C = edge(I, 'canny'); % Canny's algorithm  
Z = edge(I, 'zerocross', 0.005); % Zero crossing detector  
montage({C,Z});
```



### How to detect outliers

One approach may be calculating the median of the colors, and those that are very large or very small in comparison with the median, may have a high probability of being an outlier.

Another approach may be calculating the distance between 2 colors, and if the distance of the colors between 2 points are large, the algorithm may detect those 2 points as an outlier.