Computational Vision

Lecture 3.2: Noise Filtering

Hamid Dehghani

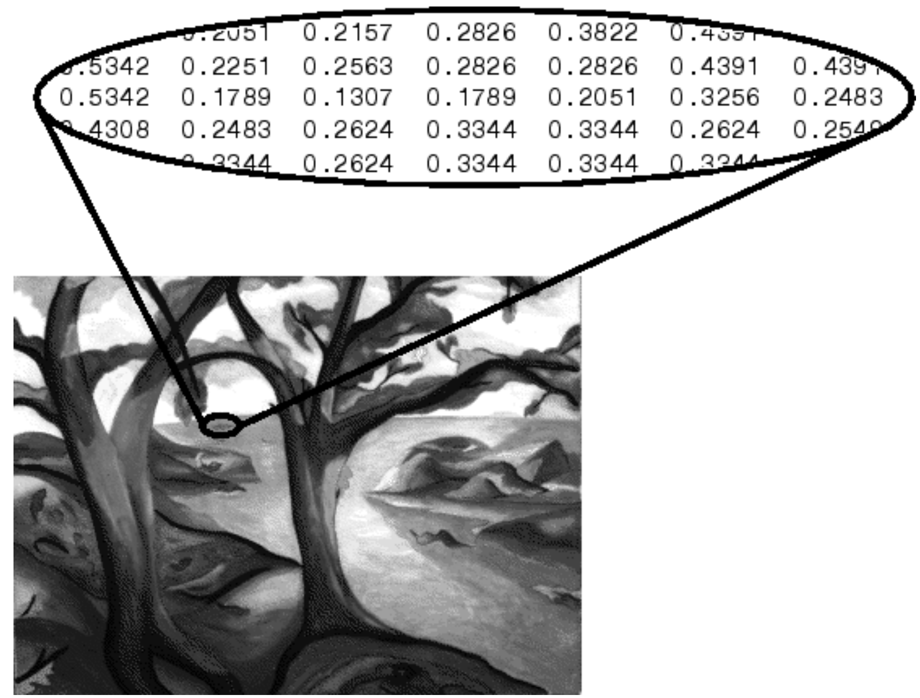
Office: UG38

Aims

- Intensity Images
- Noise Reduction
- Edge Detection!

Intensity Images

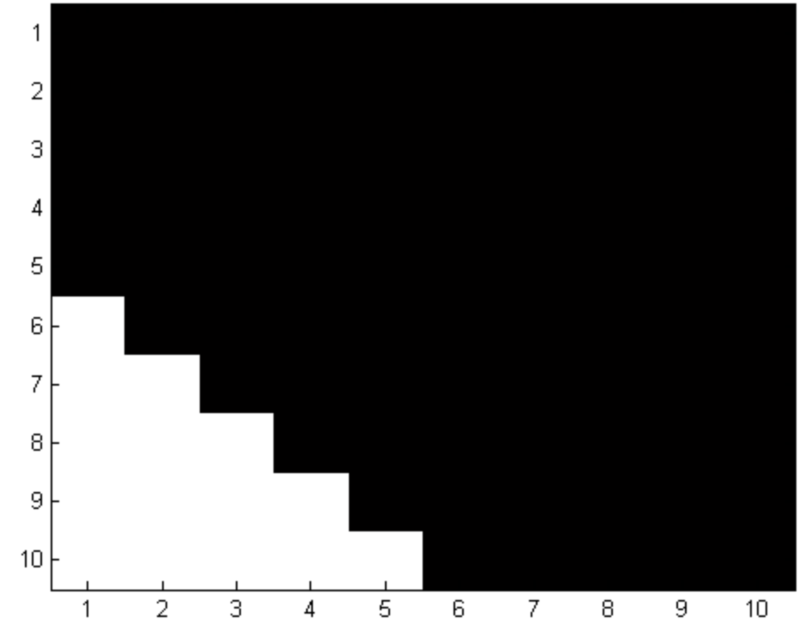
- An intensity image is a data matrix, whose values represent intensities within some range.
- represented as a single matrix, with each element of the matrix corresponding to one image pixel
- In matlab: To display an intensity image, use the imagesc ("image scale") function



Intensity gradients

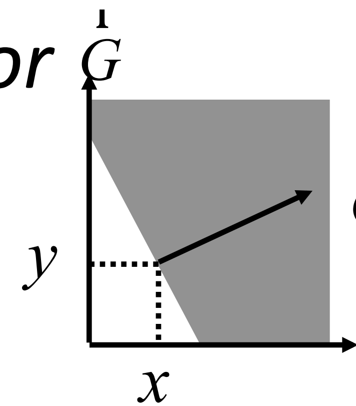
- The image is a function mapping coordinates to intensity *f(x,y)*

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1



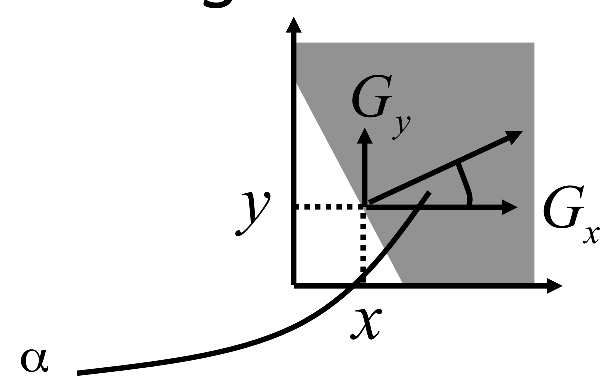
Intensity gradients

- The image is a function mapping coordinates to intensity *f(x,y)*
- *The gradient of the intensity is a vector*



$$\vec{G}[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$

- *We can think of the gradient as having an x and a y component*



$$M(\vec{G}) = \sqrt{G_x^2 + G_y^2}$$

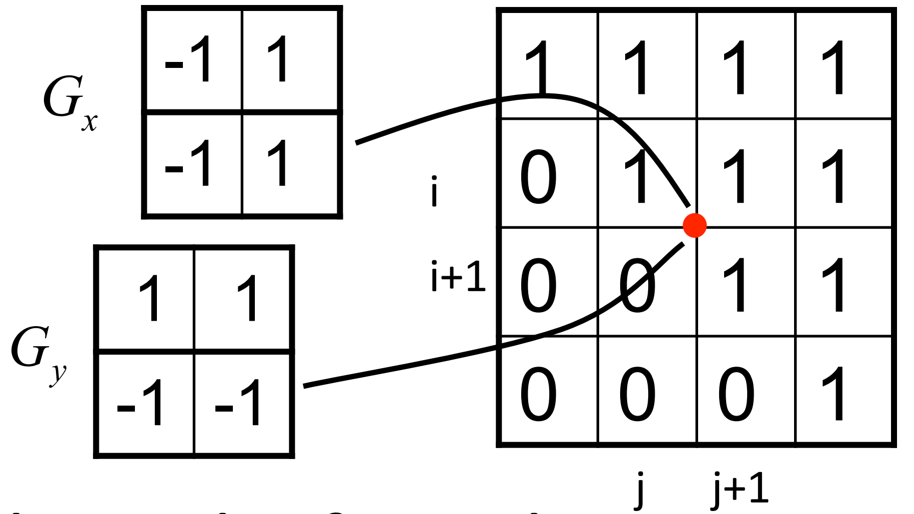
magnitude

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

direction

Approximating the gradient

- So we use a 2x2 mask instead



- For each mask of weights you multiply the corresponding pixel by the weight and sum over all pixels

Other edge detectors

- Roberts

G_x

1	0
0	-1

G_y

0	-1
1	0

- Sobel

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

1	2	1
0	0	0
-1	-2	-1

Approximating the gradient

- Sobel

G_x

-1	0	1
-2	0	2
-1	0	1

	1	1	1	1
i	0	1	1	1
i+1	0	0	1	1
	0	0	0	1
		j	j+1	

G_y

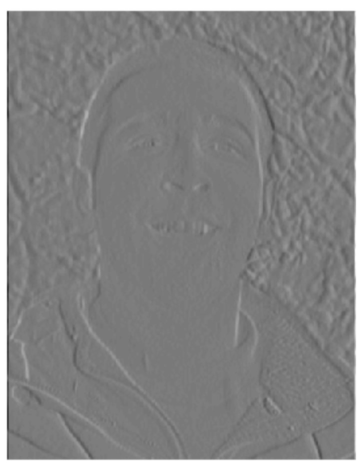
1	2	1
0	0	0
-1	-2	-1

What do these filters do

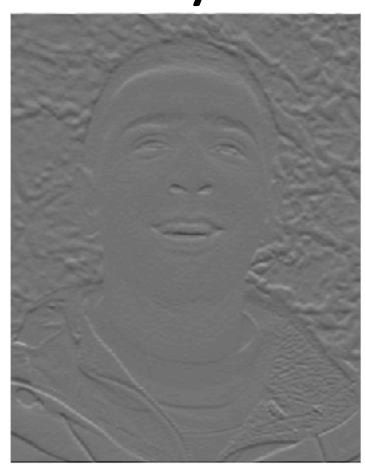
- Steps:
  - Take image
  - Convolve mask with image for each direction
    - Calculate derivatives Gx and Gy
  - Calculate magnitude = $M(\vec{G}) = \sqrt{G_x^2 + G_y^2}$

Original

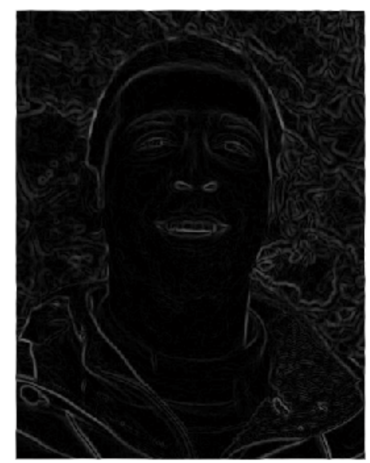
Gx



Gy



M



Noise

- It turns out we will need to remove noise
- There are many noise filters
- We can implement most of them using the idea of convolution again
- e.g. Mean filter

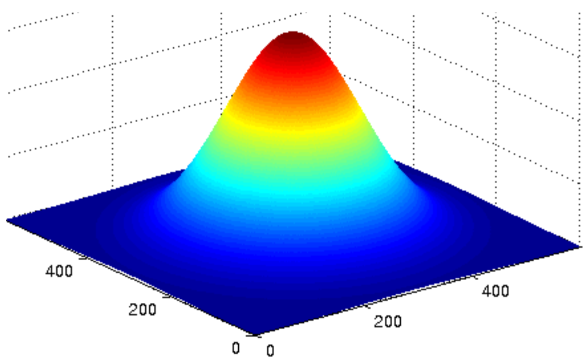
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

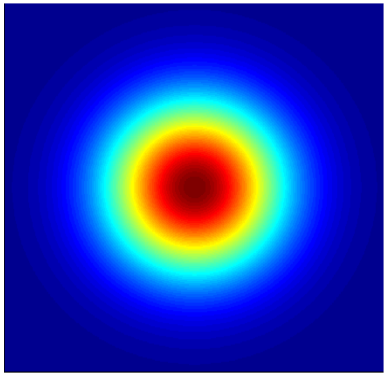
Noise filtering

- We can use convolution to remove noise as we mentioned, e.g. mean filter

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

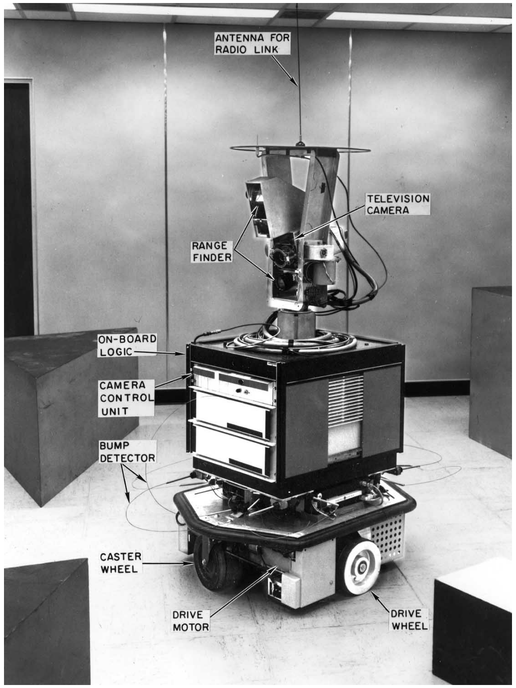
- This is a linear filter
- The most widely used is Gaussian filtering



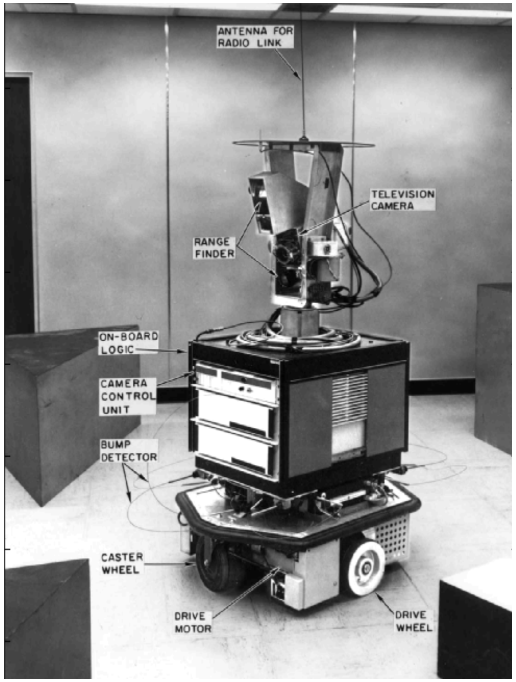


0	.01	.02	.01	0
.01	.06	.11	.06	.01
.02	.11	.16	.11	.02
.01	.06	.11	.06	.01
0	.01	.02	.01	0

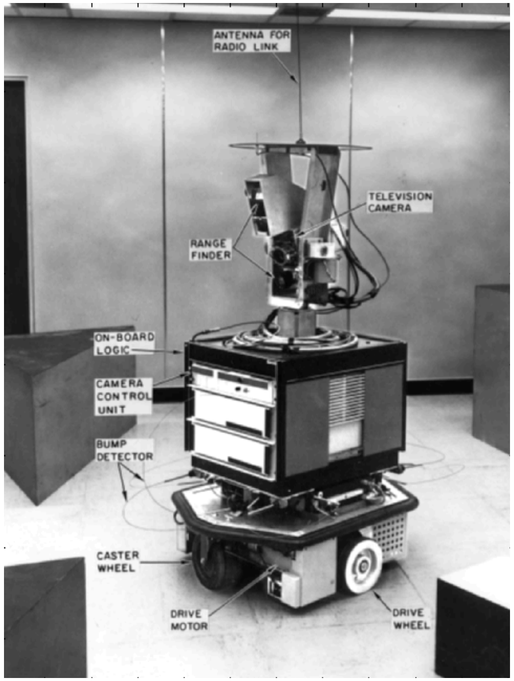
Effect of mean filtering



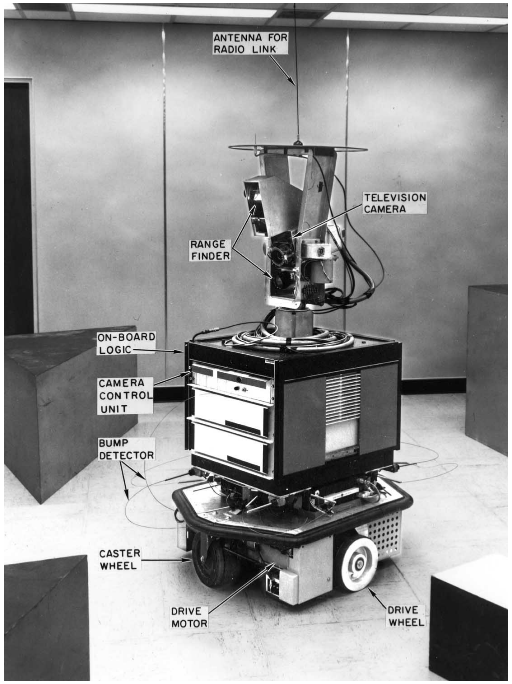
Original



3x3 filter



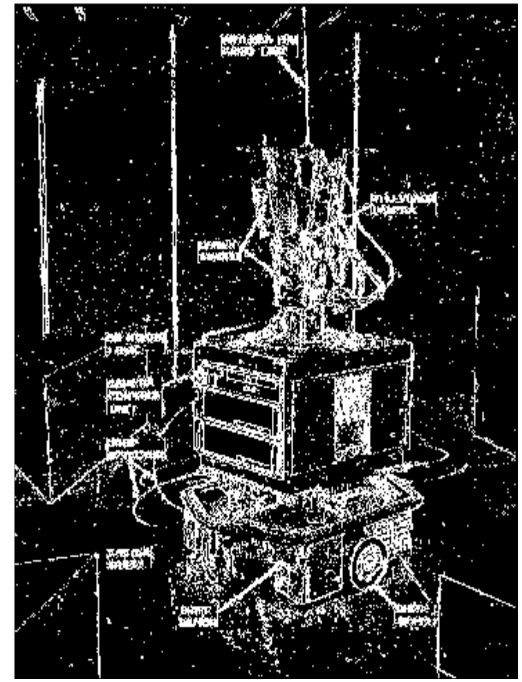
5x5 filter

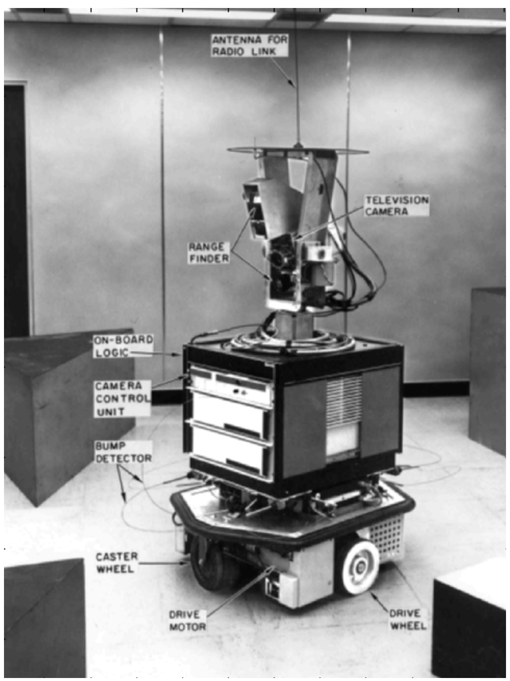


Horizontal Sobel operator

Abs(G_x)

Threshold=30



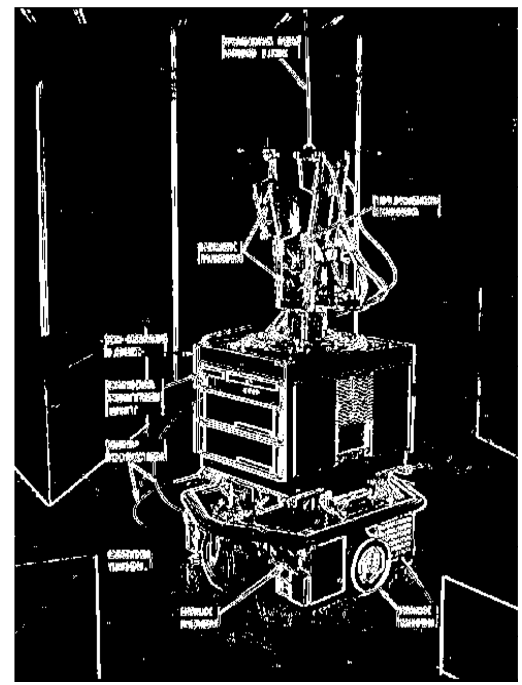


5x5 Mean Filter

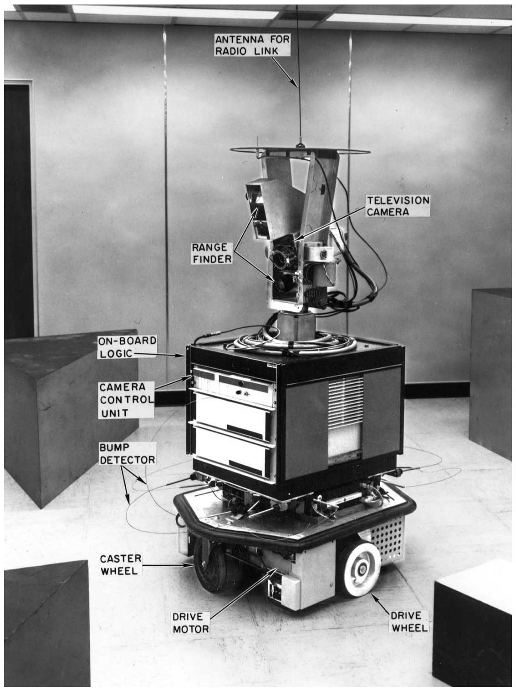
Horizontal Sobel operator

Abs(G_x)

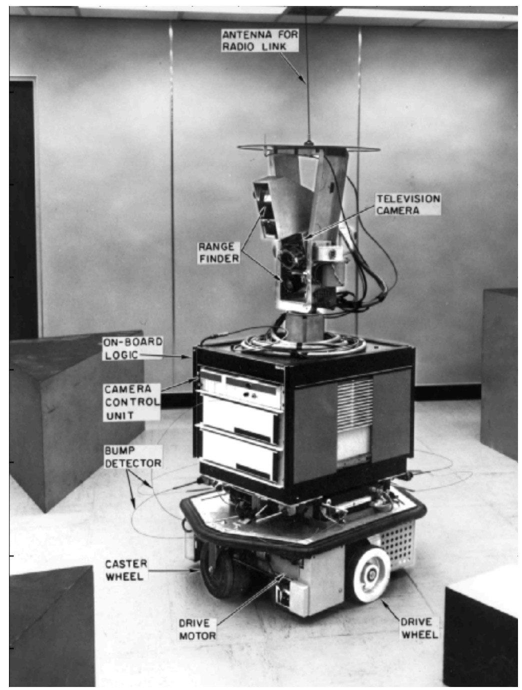
Threshold=30



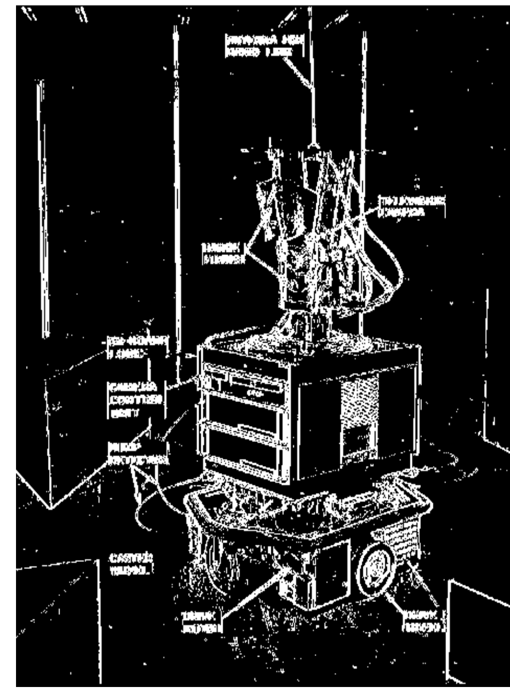
Effect of Gaussian filtering



Original



5x5 filter



Horizontal
Sobel Operator

Abs(G_x)

Threshold = 30

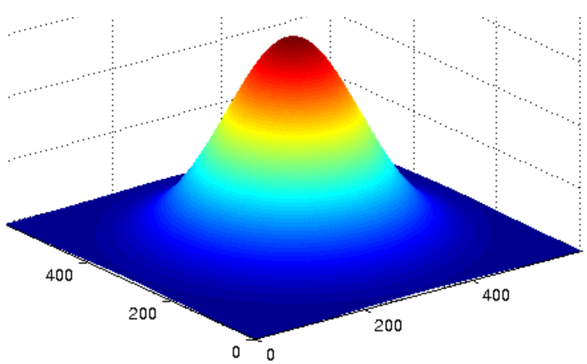
Sequenced filters

We can replace a 2D Gaussian filter with 2, 1D Gaussian filters in sequence

0.003	.0133	.0219	.0133	0.003
.0133	.0596	.0983	.0596	.0133
.0219	.0983	.1621	.0983	.0219
.0133	.0596	.0983	.0596	.0133
0.003	.0133	.0219	.0133	0.003

.0545	.2442	.4026	.2442	.0545

.0545
.2442
.4026
.2442
.0545



Laplacian Operator

Highly Directed Work

- Second order operators
- Laplacian
- Laplacian of Gaussian
- Gaussian (Canny) edge detection
- Thresholding