

# Digital Image Processing

Image Segmentation:  
Thresholding

So far we have been considering image processing techniques used to transform images for human interpretation

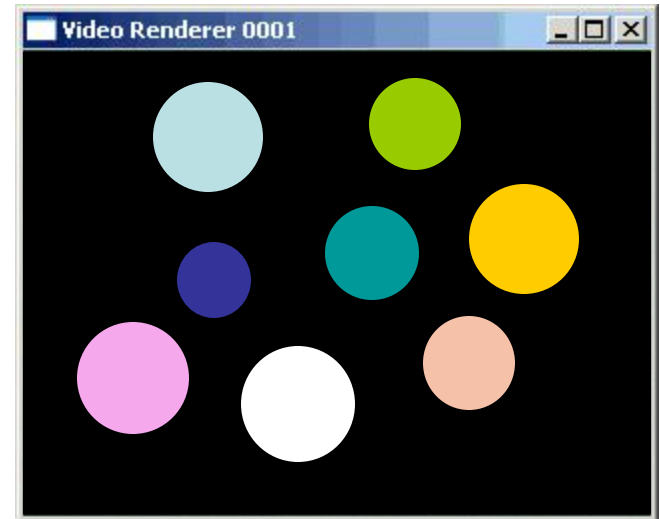
Today we will begin looking at automated image analysis by examining the thorny issue of image segmentation:

- The segmentation problem
- Finding points, lines and edges

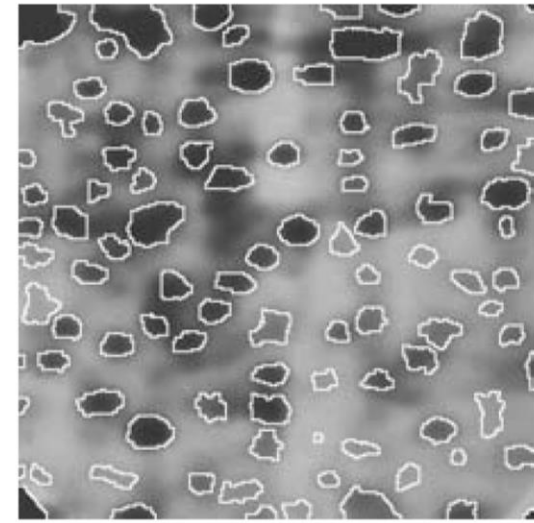
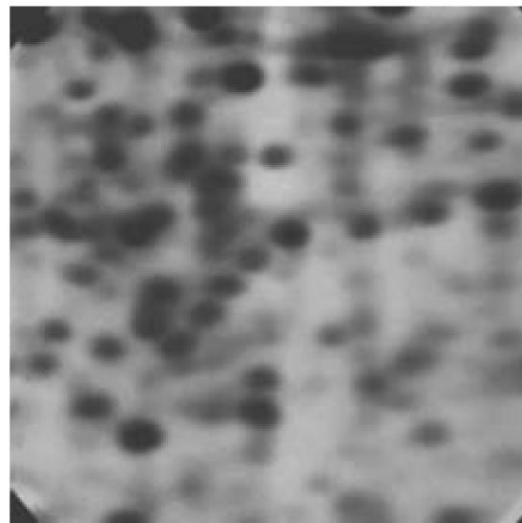
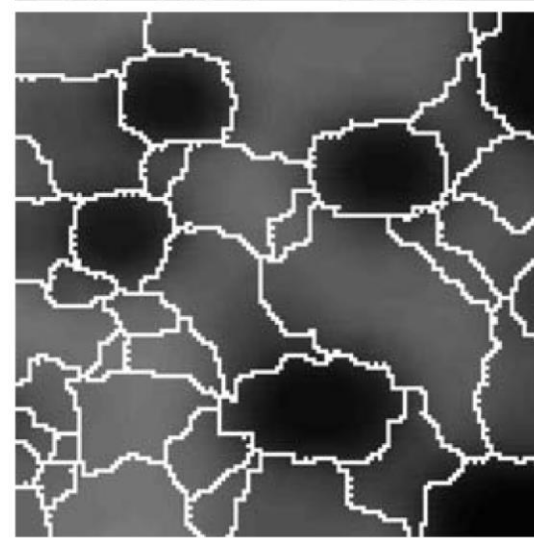
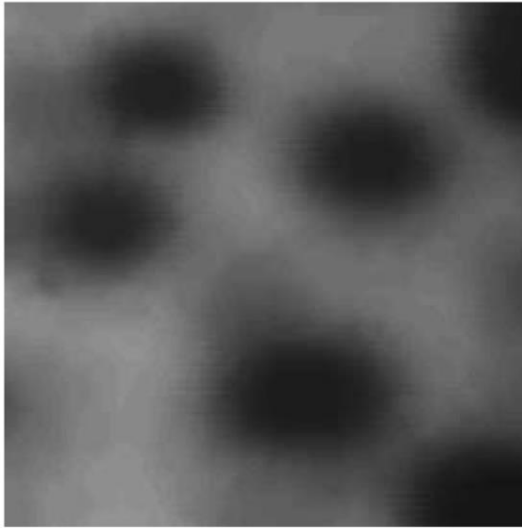
# The Segmentation Problem

Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image

Typically the first step in any automated computer vision application



# Segmentation Examples



# Detection Of Discontinuities

There are three basic types of grey level discontinuities that we tend to look for in digital images:

- Points
- Lines
- Edges

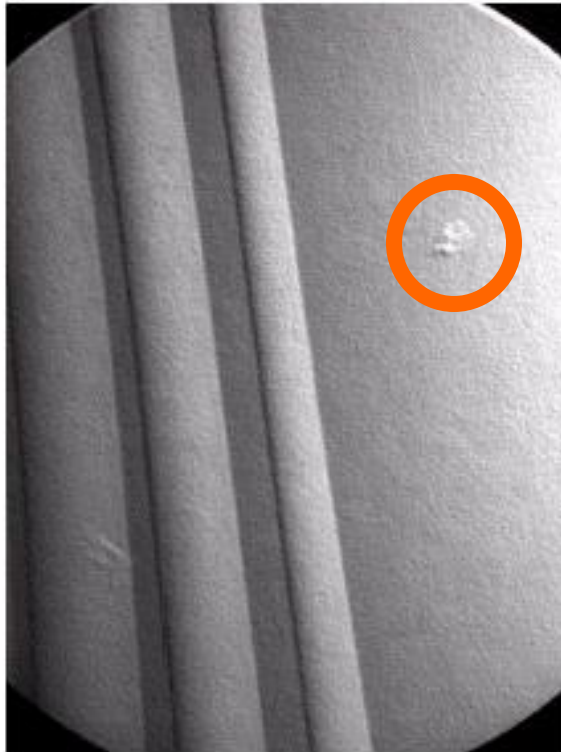
We typically find discontinuities using masks and correlation

Point detection can be achieved simply using the mask below:

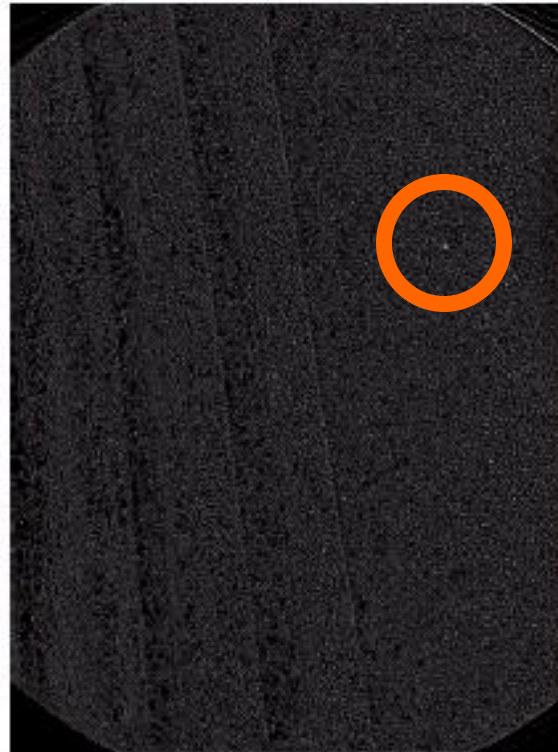
-1	-1	-1
-1	8	-1
-1	-1	-1

Points are detected at those pixels in the subsequent filtered image that are above a set threshold

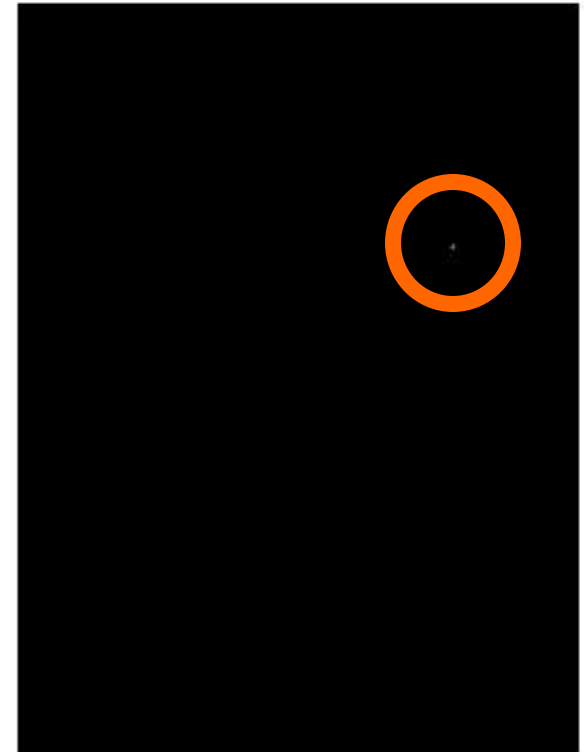
# Point Detection (cont...)



X-ray image of  
a turbine blade



Result of point  
detection



Result of  
thresholding

The next level of complexity is to try to detect lines

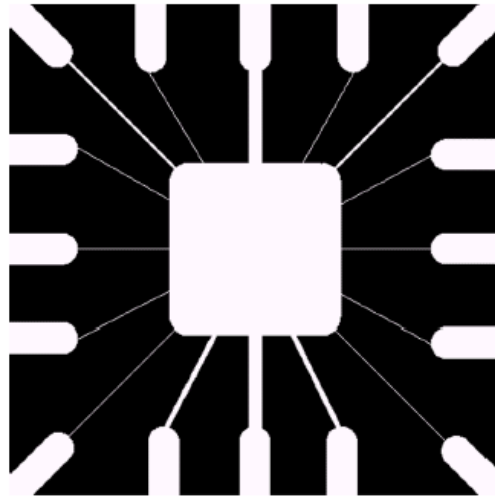
The masks below will extract lines that are one pixel thick and running in a particular direction

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		



# Line Detection (cont...)

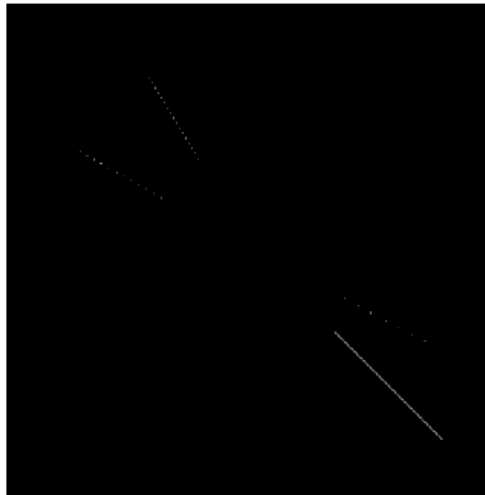
Binary image of a wire  
bond mask



After  
processing  
with  $-45^\circ$  line  
detector

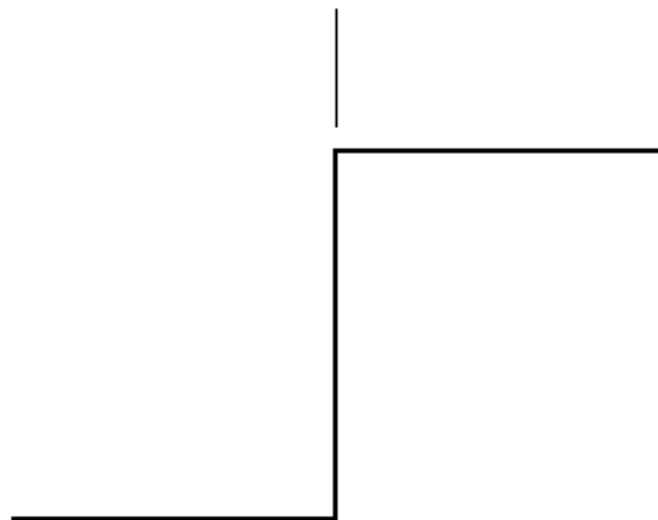


Result of  
thresholding  
filtering result



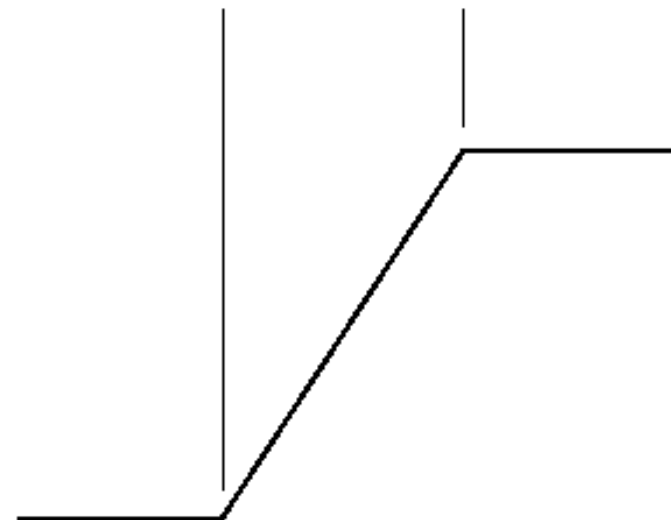
An edge is a set of connected pixels that lie on the boundary between two regions

Model of an ideal digital edge



Gray-level profile  
of a horizontal line  
through the image

Model of a ramp digital edge



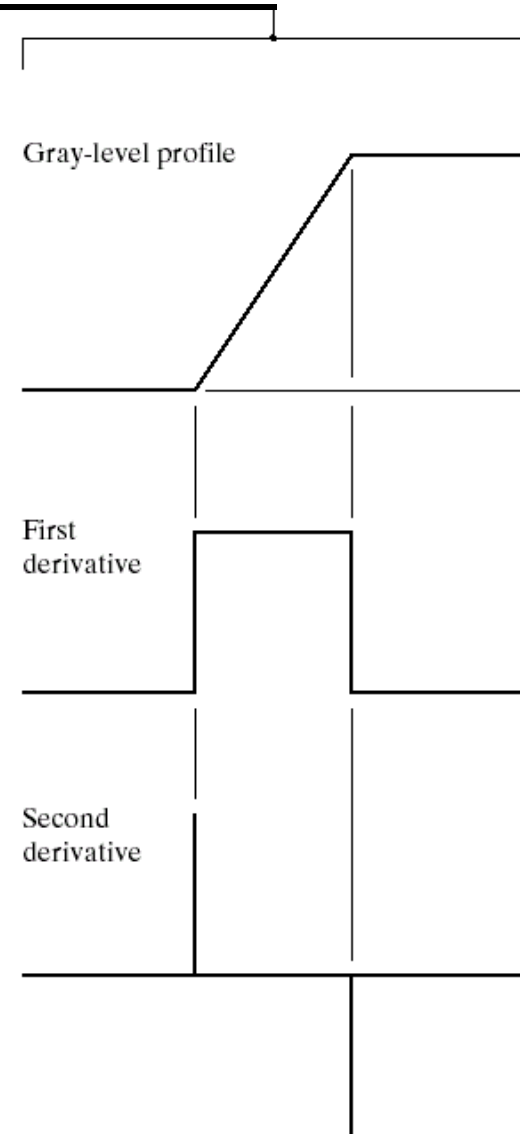
Gray-level profile  
of a horizontal line  
through the image

# Edges & Derivatives

We have already spoken about how derivatives are used to find discontinuities

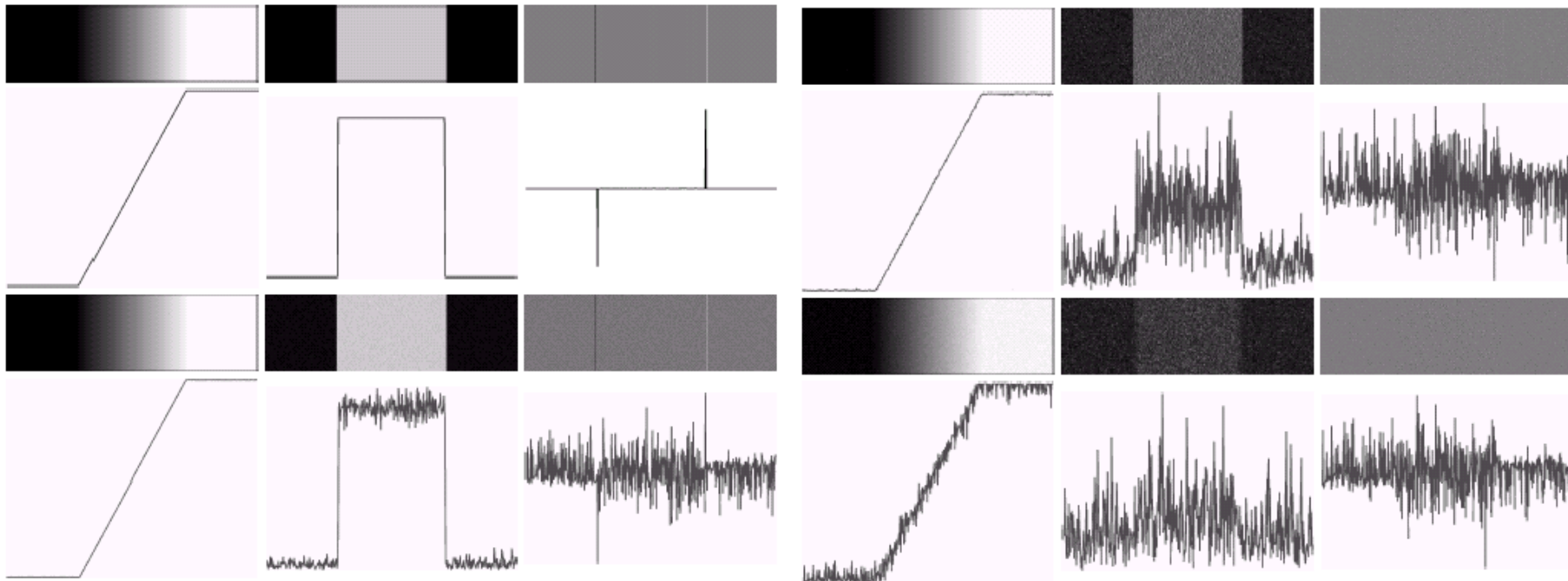
1<sup>st</sup> derivative tells us where an edge is

2<sup>nd</sup> derivative can be used to show edge direction



Derivative based edge detectors are extremely sensitive to noise

We need to keep this in mind



# Common Edge Detectors

Given a 3\*3 region of an image the following edge detection filters can be used

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt

-1	0
0	1

0	-1
1	0

Roberts

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

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

Sobel

# Edge Detection Example

Original Image



Horizontal Gradient Component



Vertical Gradient Component



Combined Edge Image

# Edge Detection Example



# Edge Detection Example





# Edge Detection Example



# Edge Detection Example



# Edge Detection Problems

Often, problems arise in edge detection in that there are is too much detail

For example, the brickwork in the previous example

One way to overcome this is to smooth images prior to edge detection

# Edge Detection Example With Smoothing

Original Image



Horizontal Gradient Component



Vertical Gradient Component



Combined Edge Image

# Laplacian Edge Detection

We encountered the 2<sup>nd</sup>-order derivative based Laplacian filter already

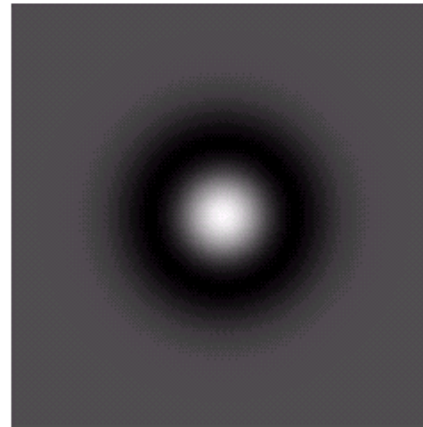
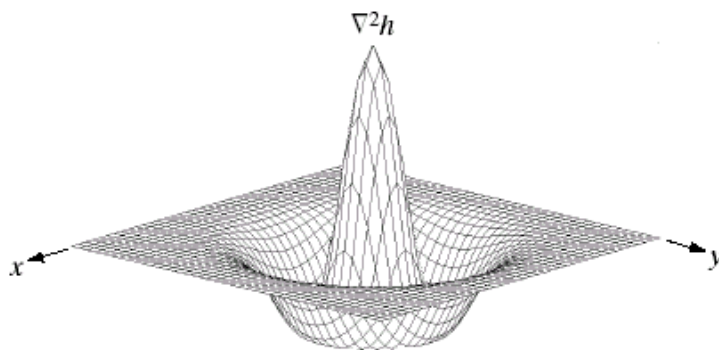
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

The Laplacian is typically not used by itself as it is too sensitive to noise

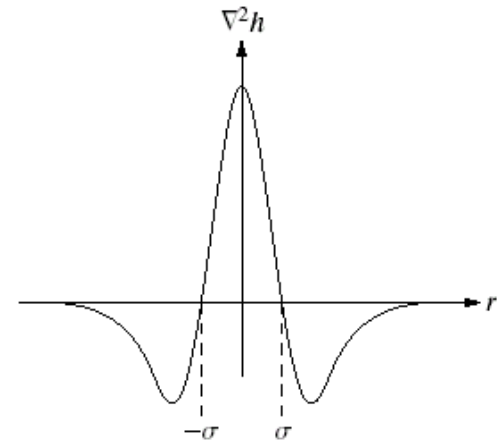
Usually when used for edge detection the Laplacian is combined with a smoothing Gaussian filter

# Laplacian Of Gaussian

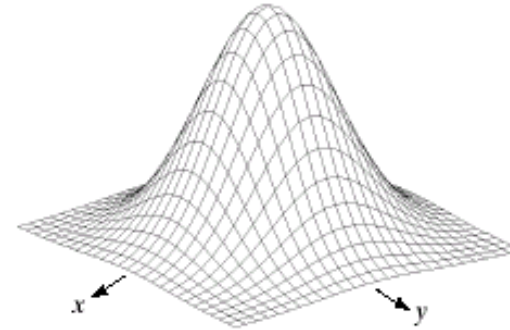
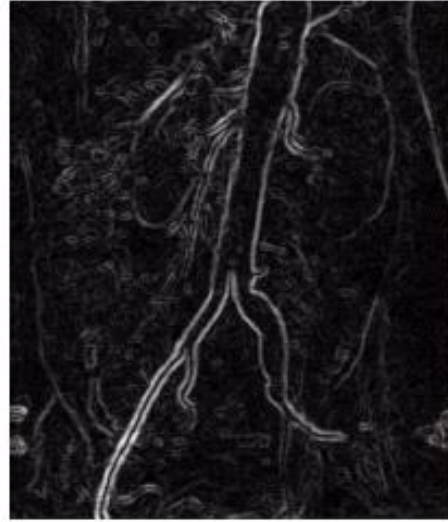
The Laplacian of Gaussian (or Mexican hat) filter uses the Gaussian for noise removal and the Laplacian for edge detection



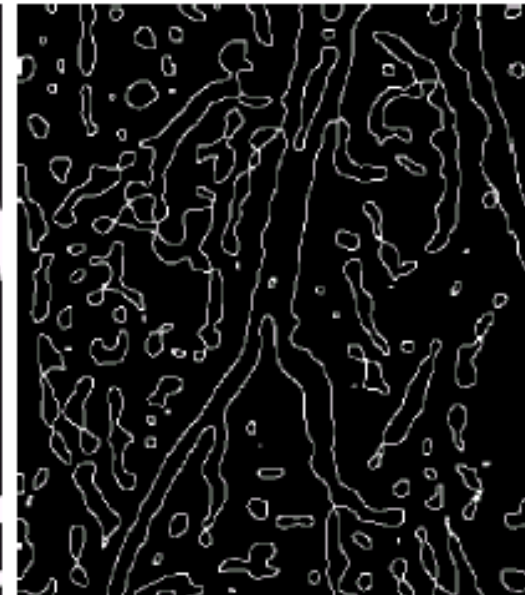
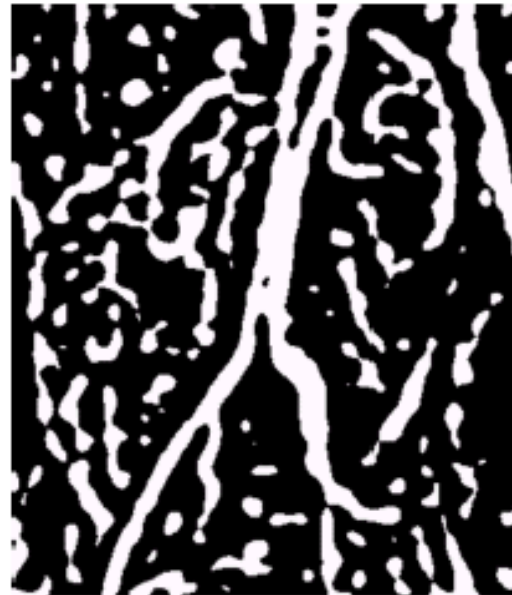
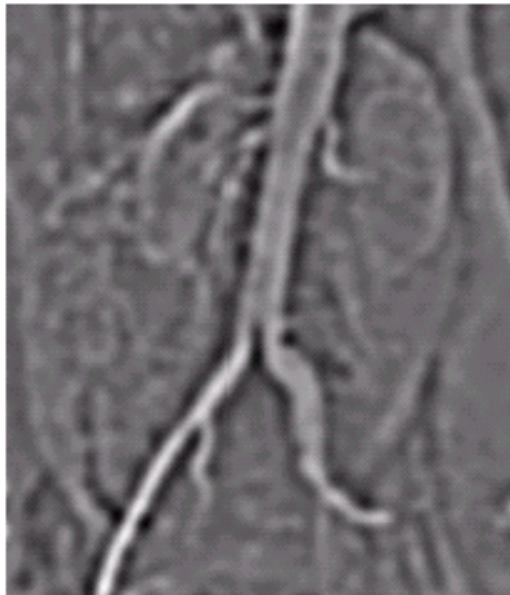
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



# Laplacian Of Gaussian Example



-1	-1	-1
-1	8	-1
-1	-1	-1



In this lecture we have begun looking at segmentation, and in particular edge detection

Edge detection is massively important as it is in many cases the first step to object recognition