Environmental Remote Sensing
GEOG 2021

Lecture 5

Spatial information in remote sensing

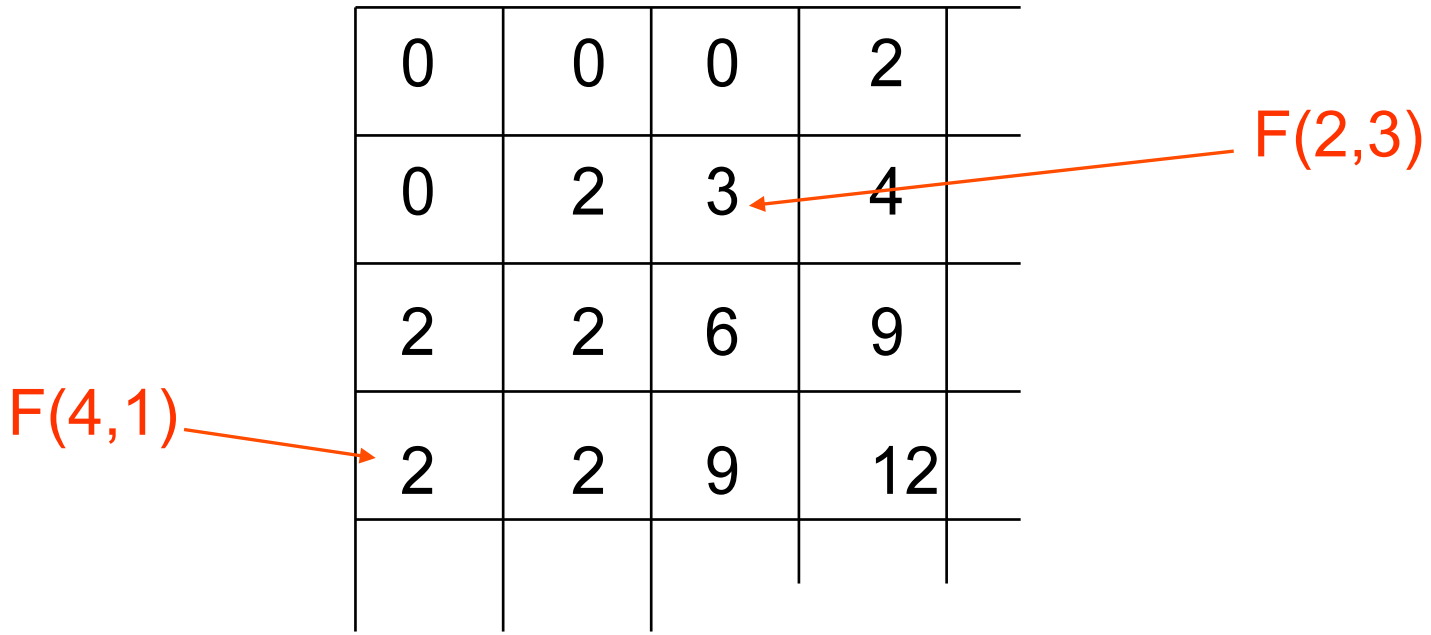
Aim

- Recognising spatial information in EO images
- Improve our ability to interpret remote sensing data by *image processing*
- Typical operations
  - smoothing
  - edge detection
  - segmentation
- and some terminology

Why process images ?

- Improve interpretability
  - Enhance image, smooth, remove noise, improve contrast….
- Detect particular spatial features
  - edges of fields or woodland
  - ship wakes on the ocean surface
  - power lines ? Airport runways ?

- Images are presented as 2-d arrays. Each pixel (array element) has a location (x,y) and associated with it a digital number (DN).
- Can think of the DN as the value of a function F(x,y)

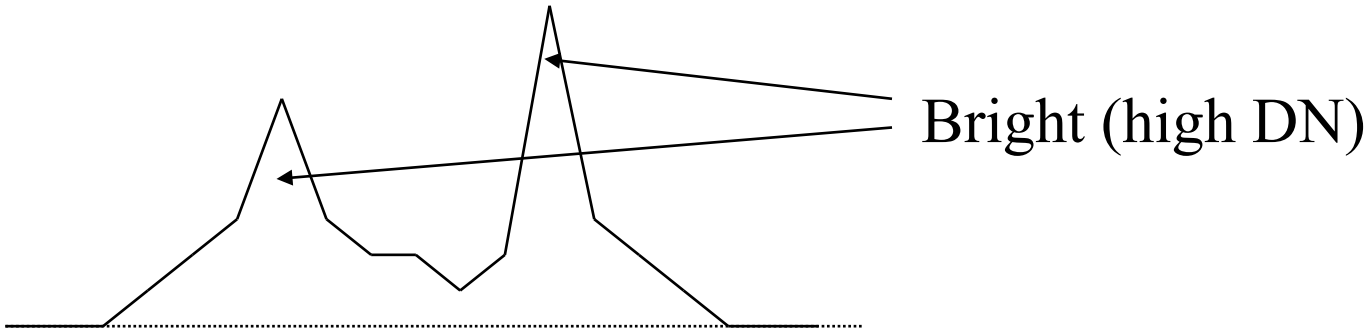


- Consider a single line of an image:-

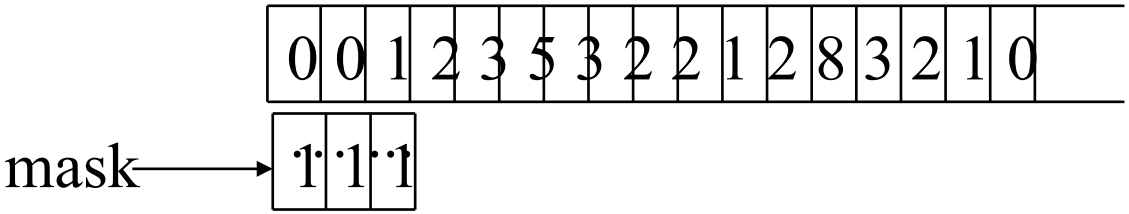
0	0	1	2	3	5	3	2	2	1	2	8	3	2	1	0	

… …

- which can be represented as

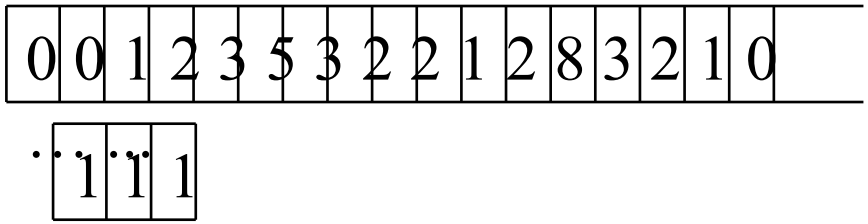


- Consider following set of operations (which we'll do first, and then think about afterwards…)
- Place a mask (3 pixels long) over the start of the sequence
  - multiply the numbers in the array by the numbers in the mask
  - add them together
  - divide by the number of cells in the mask:-



  - 0*1 + 0*1 + 1*1 = 1
  - divide by 3 → 1/3

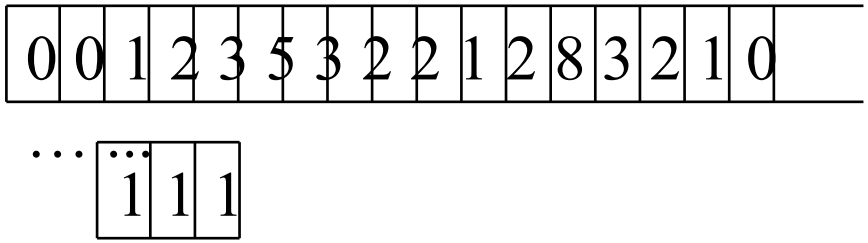
– move the mask along one and repeat...



– 0*1 + 1*1 + 2*1 = 3

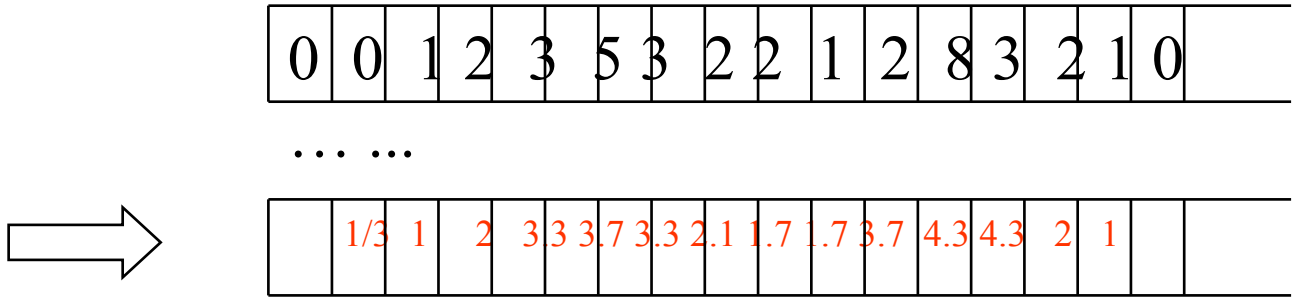
– divide by 3 → 1

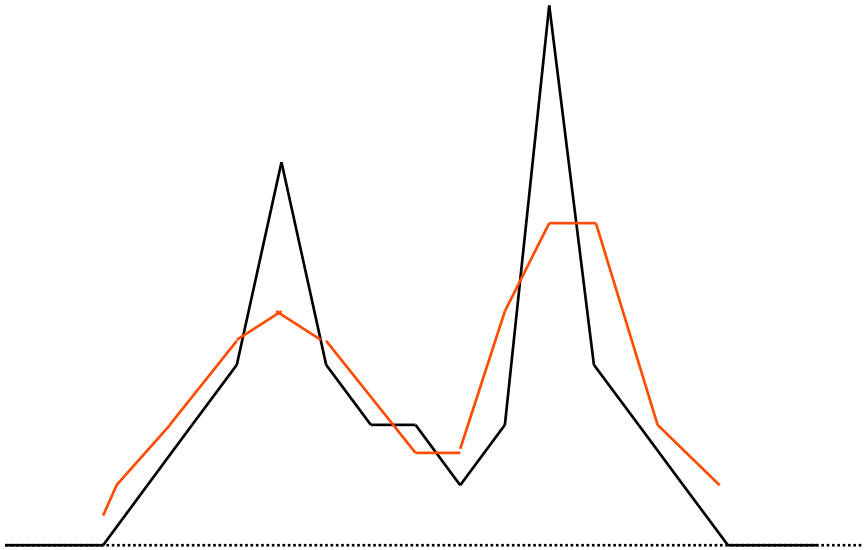
– and repeat…



– 1*1 + 2*1 + 3*1 = 6

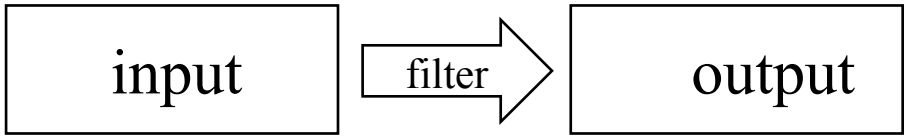
– divide by 3 → 2

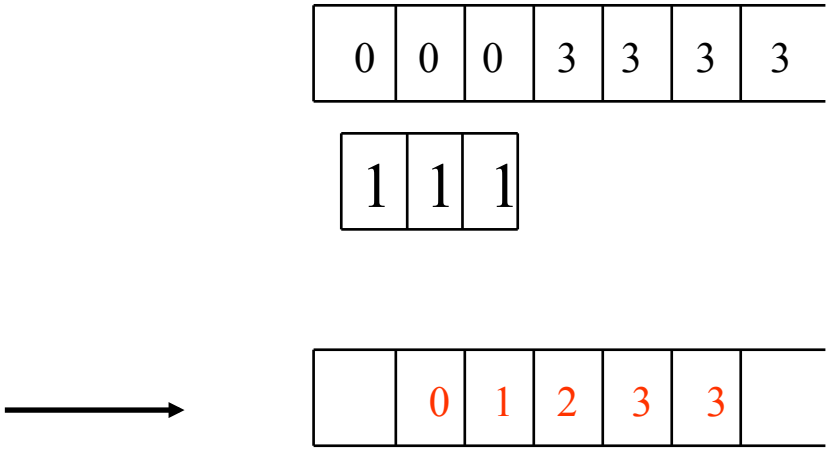

0 0 1 2 3 5 3 2 2 1 2 8 3 2 1 0
1/3 1 2 3 3 3.7 3.3 2.1 1.7 1.7 3.7 4.3 4.3 2 1



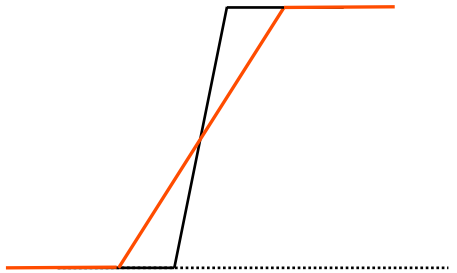
- The series of operations we have carried out is called "**convolution filtering**"

- Convolution transforms an input function (in this case a 1D array) into an output function (in this case another 1D array) using an operator function (the filter – also a 1D array)





The filtering we've done seems to have "smoothed" the 1D profile

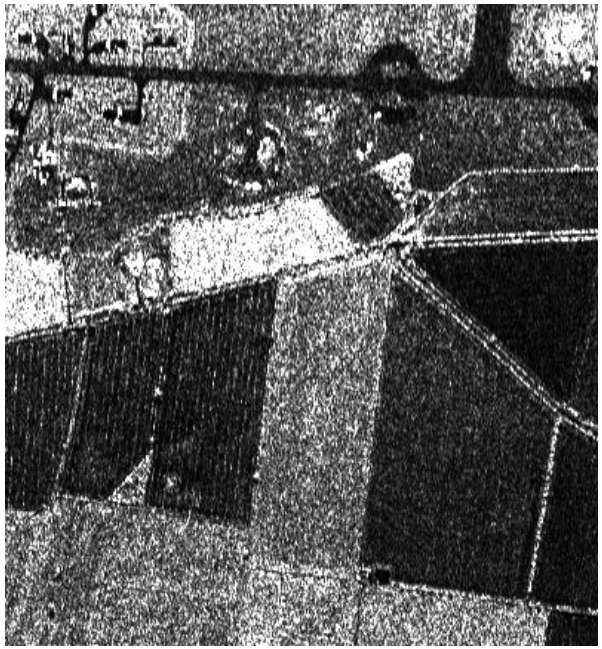


- This is an example of a “smoothing” filter. It proceeds by calculating average (mean) values. Therefore it will smooth away deviations
- Sometimes called “running mean” or “moving average”.
- Note that smoothing filters “blur” the features in an image
- → loss of spatial resolution
  - So why do we do it??

- The generalisation to 2D arrays (images) works in exactly the same way…
- except that we now use a mask that's also a 2D array:-

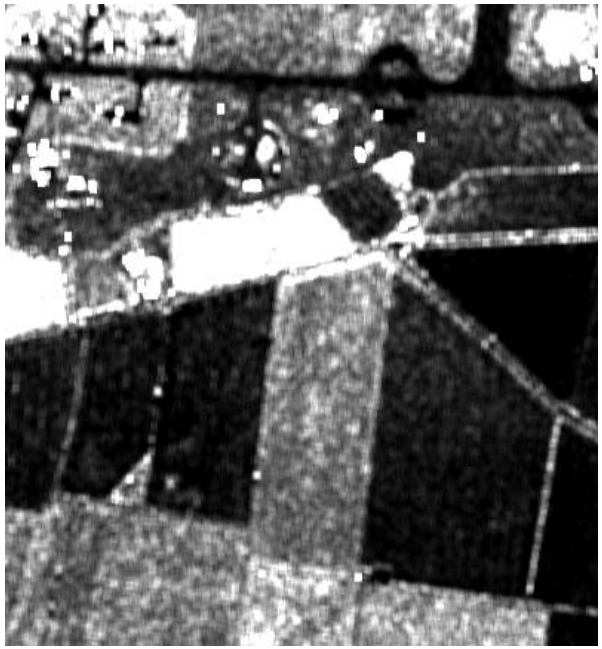
1	1	1
1	1	1
1	1	1

0	0	0	2	
0	2	3	4	
2	2	6	9	
2	2	9	12	



unsmoothed

running mean 3x3



running mean 5x5

- Notice that we have a buffer around the edge, where we cannot quite apply the mask

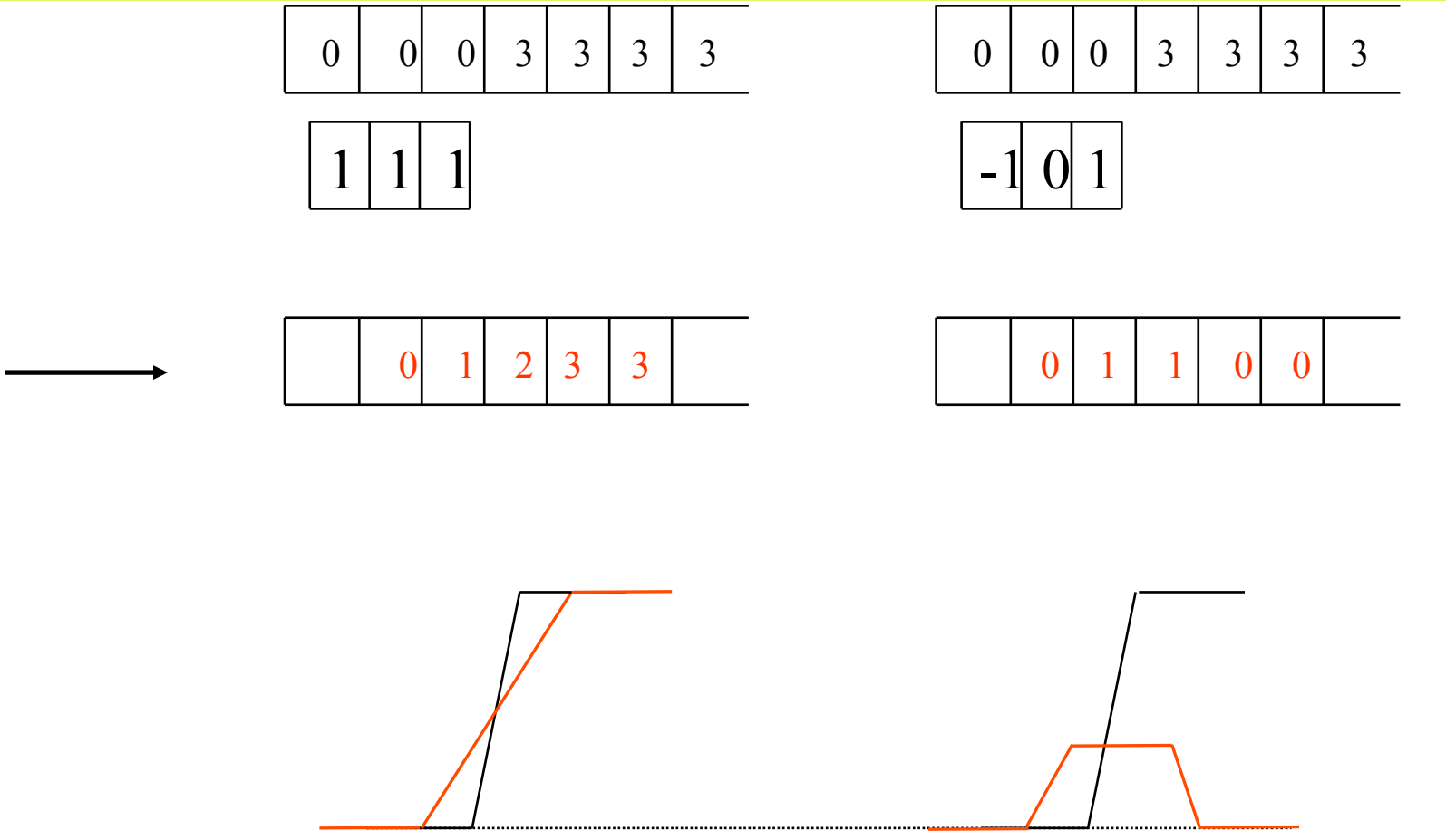
- There is another important way of calculating the mean value - median.

- What does a median filter do?
- What advantages are there compared with the mean?

- The mask we are using is called the **kernel**
- So far are simple
  - Value of 1 in every cell of a 1x3 or 3x3 kernel
- But we can very easily complicate the kernel
- In this case the output of the filter will be very different

- e.g. what happens with this kernel?

-1	0	1



Mean filter

"1st derivative" filter

- 1st derivative filter calculates **gradient**.
- Where DN values constant (gradient=0) filter =0
- Only starts returning values where there is a gradient (i.e. where DN values change from one pixel to the next)
- Returns high values wherever *change* in DN is high
- Use to detect ***edges*** since these are often visible in an image where there is a change in brightness

- Can again apply to 2D images
- For the edge detecting kernel, we might have

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

or

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

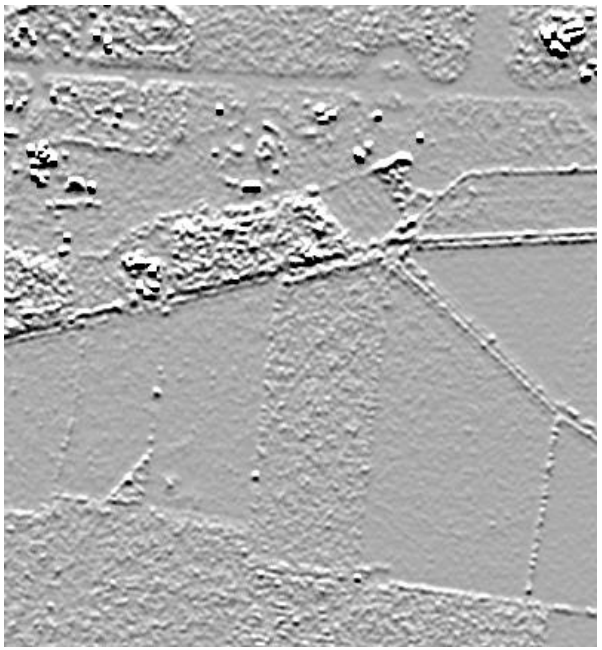
- What is the different between these, and what effects are they likely to have ??

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

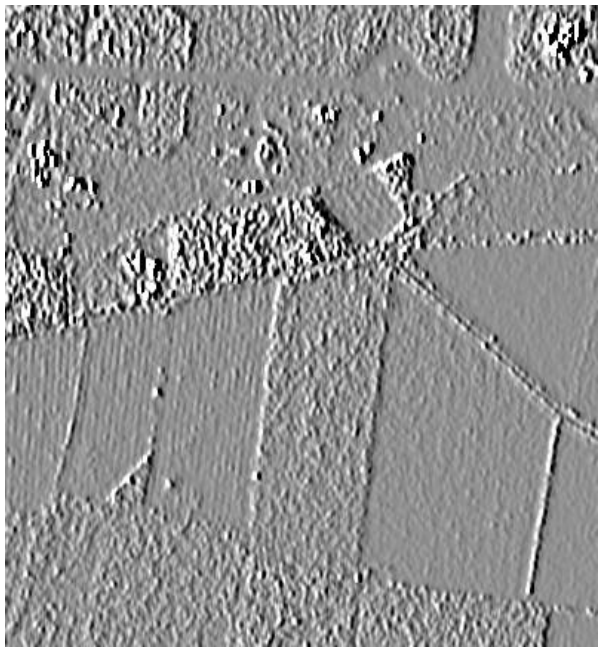
- This filter has
  - smoothing in the y direction
  - gradient operator in the x-direction
  - “combination” of two filters
- It won’t do the same thing as the second kernel

Directionality

- Note that when we work with 2D images, we can introduce filters that are not isotropic (that is, the kernels have different numbers in the x-y directions)
- horizontal and vertical smoothing
- horizontal and vertical edge detectors



horizontal edge detection



vertical edge detection

Convolution

- One way of describing the filtering operations we have been doing is to use the term “convolution”
- Convolution of an image with a kernel (moving the mask around on the image and multiplying them together)

$$O = F \otimes I$$

- where I represents the input, and O is the output.

- Note that we are free to do more than one filtering operation:-

  - $O = (F_2 \otimes (F_1 \otimes I))$

- which effectively means “do the filtering operation 1 on the input I, and then on the output of this, do another filtering operation (2) to give the final output O”.

Sobel

- Another edge detecting filter is the Sobel filter:-

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

or

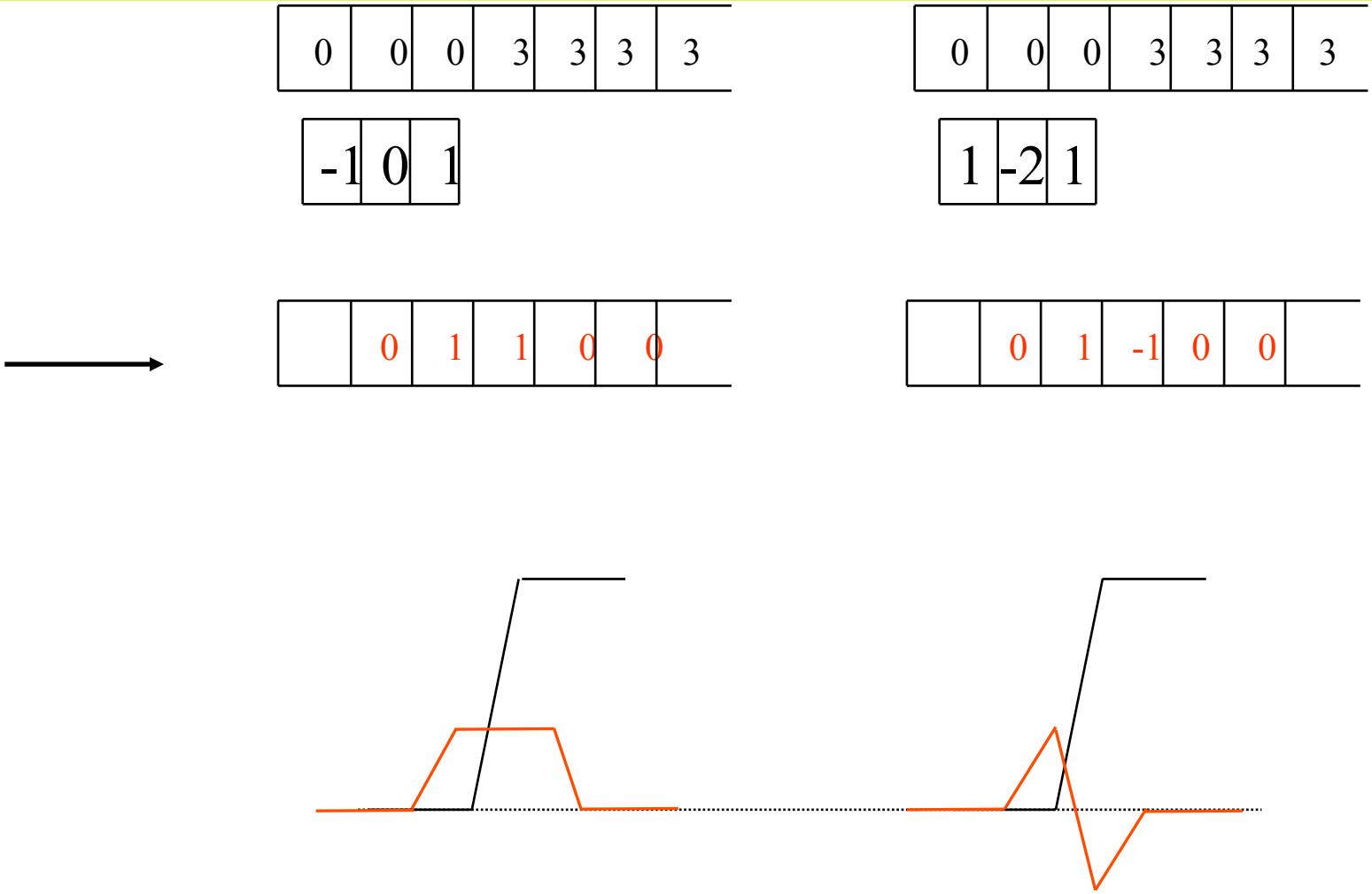
-1	-2	-1
0	0	0
1	2	1

- Combinations of smoothing and gradient operator
- Again there is a vertical version, and a horizontal version

- If 1st order derivative filter calculates gradients in an image, we can also calculate gradients in the gradient image (2nd order)

- Rather than doing two separate filtering operations, we can instead use a single filter to do the whole job

- Called a “Laplacian”

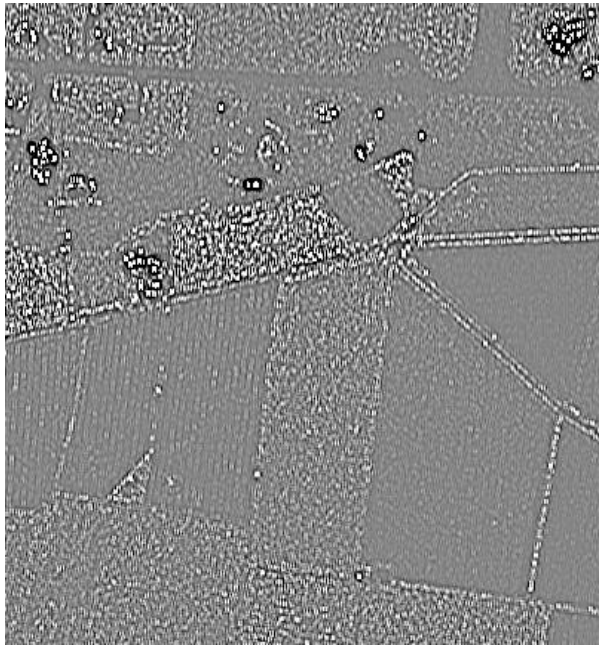


"1st derivative" filter

"2nd derivative"

- Laplacians detect “the edges of the edges”

0	1	0
1	-4	1
0	1	0



Low- and High-Pass Filtering

- When we look at spatial structure, we can usually see a characteristic length scale (e.g. size of fields, width and length of roads, etc)

- Images usually have features on lots of different length scales

- Sometimes, instead of talking about “length scale”, we talk instead about “spatial frequency”.

- **High frequency variation** == changes in DN over small distance
- **Low frequency variation** == changes in DN over large distance

- e.g. Smoothing filters

- remove small scale features but maintain large scale features
- i.e. remove (smooth out) high spatial frequencies from the image, but keep low spatial frequencies

- → “low pass” filter

- Similarly, edge detectors keep small scale features, but remove large scales

- → “high pass” filter