3. IMAGE ENHANCEMENT (2)

Image enhancement on the human retina

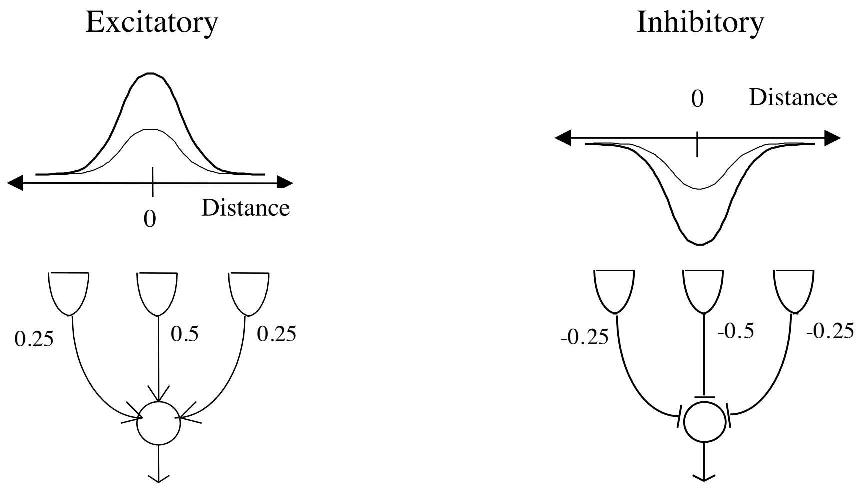
Light reaching receptors in the eye does not cast a perfect image on the retina. Certain mechanisms have evolved to compensate for the distortions introduced in this biological image formation process. Many computational processes are modelled on their biological counterparts.

Spatial response to light

Before describing the processes which enhance retinal images, we shall look at the basic processes "implemented" by the retinal neurons.

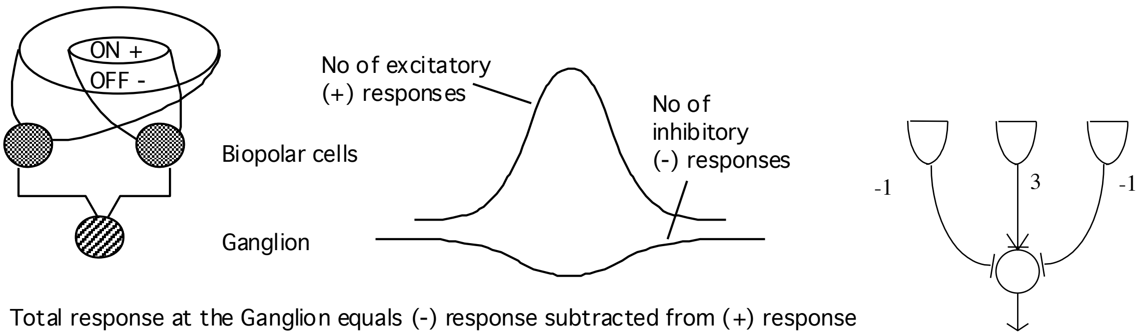
Cell responses

The neighbouring cells are connected vi *lateral connections*. The connections can be *excitatory* (positive) or *inhibitory* (negative). The strength of both type of the responses decreases with distance and increases with brightness of light falling on the retina. The lateral connections help to "smooth" the gaps and spatial brightness distortions.



Architecture of the lateral connections

The cells are arranged into units called *centre-surround*, where the centre response is excitatory and the surround response is inhibitory. There are also "symmetrical" units, with opposite structure. The shape of the response of the unit is made of two contributions. This architecture enables the incoming image to be appear sharper.



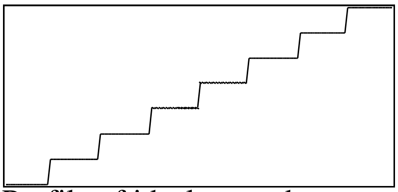
Architecture of on-centre-off-surround connections

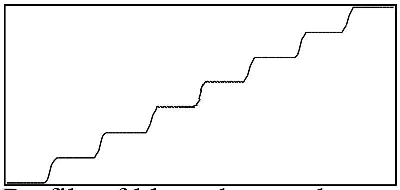
Digital filtering via convolution

An input digital image (like its biological counterpart) is never perfect, but is always "noisy" and distorted to some extent by:

- non-linearities of electronic components of sensors
- sampling and quantisation errors
- data transmission artefacts

Noise is normally characterised by *high frequency* changes. It corrupts the image and obscures the detail. The opposite problem is when the image is blurred, boundaries between objects in the image are indistinct and of low contrast. In terms of frequency characteristics the image lacks high frequency components in the signal.


Profile of ideal step edges


Profile of blurred step edges

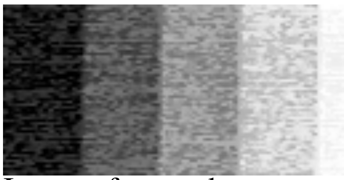
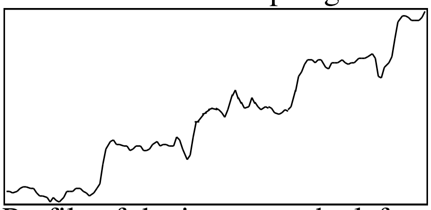


Image of step edges corrupted by noise


Profile of the image on the left.
Note both noise and blur in the profile.

Frequency filtering operations - a tool for reducing frequency distortions

Frequency filtering operations separate frequency components within an image into different frequency ranges, most often into:

- low frequency components
- high frequency components

Following this separation, unwanted components can be rejected or, low and high frequency components can be processed separately and then re-combined.

Removing noise

One of the simplest methods (also "implemented" in nature) to remove spurious signals is to average the incoming signal over the number of sensors. The spatial averaging produces visually smoother image; this is because the unwanted high frequencies are eliminated by averaging. The process of *smoothing* works by removing rapid changes in brightness (high frequency component) but at the same time retaining slow changes (low frequency component). Averaging of pixel values is an example of a simple digital image processing technique for reducing high frequency component of an image.

f1	f2	f3
f4	f5	f6
f7	f8	f9

Here the average pixel value is calculated for 3x3 pixel area of the image.

$$\frac{(f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + f9)}{9} =$$

$$= \frac{1}{9}\ f1 + \frac{1}{9}\ f2 + \frac{1}{9}\ f3 + \frac{1}{9}\ f4 + \frac{1}{9}\ f5 + \frac{1}{9}\ f6 + \frac{1}{9}\ f7 + \frac{1}{9}\ f8 + \frac{1}{9}\ f9$$

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

This is equivalent to multiplying image values f1 - f9 by values stored in a small 3x3 matrix called a *convolution kernel* (or a convolution mask or a filter). A convolution kernel can also be thought of as a set of spatial weights applied to an input signal, analogous to using a "network" with weights attached to connections.

Convolution - definition

Convolution is a very important image processing operation. It is an example of a so-called *linear transformation* because the only operations necessary are that of addition and multiplication. The convolution of two (discrete) functions f and g is a function h of a displacement y defined as:

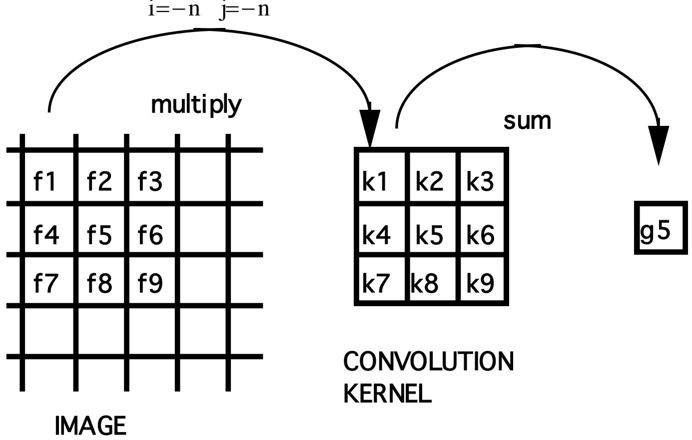
$$h(y) = f * g = \Sigma f(x)\, g(y - x)$$

This equation simply says that the value of the convolution at any displacement is the sum of the products (multiplications) of the relatively displaced function values.

Convolution in spatial domain

Convolution operation is defined as:

$$g(x,y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} k(n+i, n+j)\, f(x+i, y+j)$$



g5 = f1*k1 + f2*k2 + f3*k3 + f4*k4 + f5*k5 + f6*k6 + f7*k7 + f8*k8 + f9*k9

Depending on the weights in the convolution kernel many different "low level" processes can be modelled using convolution. Both the lateral and the feed-forward processes can be modelled.

Outline of implementation:

```
k = kernel_size / 2;

for( x=k; x<IMAGE_WIDTH-k; x++ )
     for( y=k; y<IMAGE_HEIGHT-k; y++ )
    {
            sum = 0;

            for( i= -k; i<= k; i++ )
                    for( j= -k; j<= k; j++ )
                     sum = sum + image[x+i][y+j]*kernel[k+i][k+j];

             new_image[x][y] = sum;
```

```
    }
```

Notes:

Low pass filters in spatial domain

The so called “low pass filters” *remove* high frequency components and pass through low frequency components. Most low frequency filters suffer from the side effect: blur.

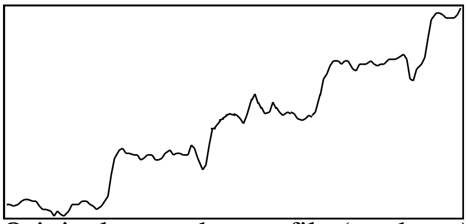
In spatial domain low pass kernel coefficients sum up to 1 and are all positive numbers.

Examples of low pass filters in spatial domain

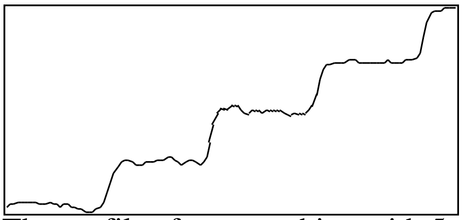
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/10	1/10	1/10
1/10	1/5	1/10
1/10	1/10	1/10

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16



Original step edge profile (as above)

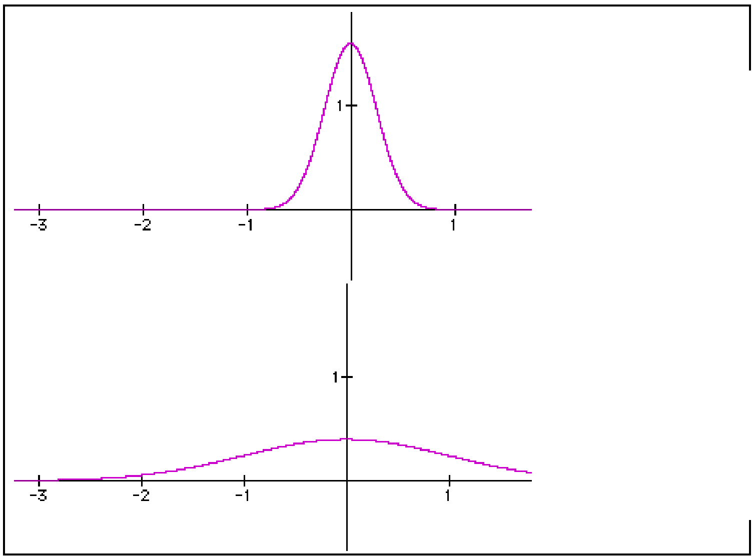


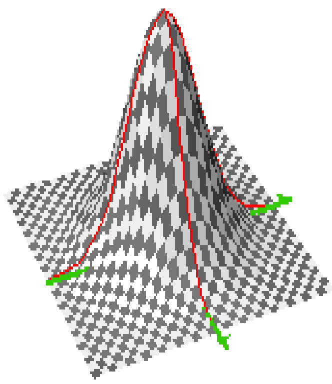
The profile after smoothing with 5x5 gaussian filter.

A two-dimensional Gaussian function makes a flexible smoothing kernel. The formula is:

$$G(x,y,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The amount of smoothing depends on the value of σ: the greater the σ the stronger the effect of smoothing.





Top: 1-dimensional Gaussian with σ = 0.25 2-dimensional Gaussian with σ = 0.25
Bottom: 1-dimensional Gaussian with σ = 1.0

Smoothing by non-linear filtering

Non-linear filters are so called because their operation cannot be implemented using addition and multiplication. For the image fragment:

0	1	1
2	5	2
1	1	2

Median filter:

replace each pixel by the median value of its neighbourhood

0 1 1 1 <u>1</u> 2 2 2 5 => 1

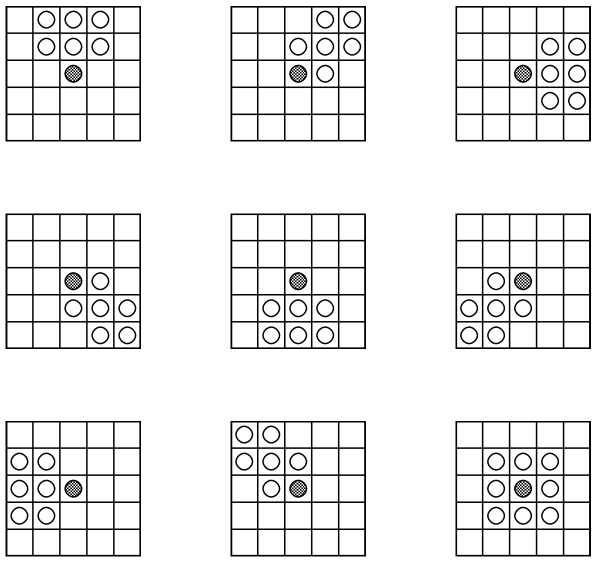
Order-n filter:

replace each pixel by the n-th ordered value of its neighbourhood

Order 0: <u>0</u> 1 1 1 1 2 2 2 5 => 0
Order 7: 0 1 1 1 1 2 <u>2</u> 2 5 => 2

Edge-preserving smoothing

replace each pixel by the average value of a portion of its neighbourhood with the smallest standard deviation



Removing blur

Possible causes of blur can be, e.g. out of focus image, light scatter, motion.

High pass filters

Blurred images lack high frequency component. The so called “high pass filters” *emphasise* high frequency components but at the same time pass through low frequency components. The side effect of all sharpening filters is that they enhance noise. In spatial domain high pass filter coefficients sum up to 1; smaller coefficients surround a larger positive centre.

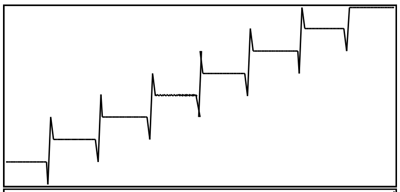
Examples of high pass filters in spatial domain

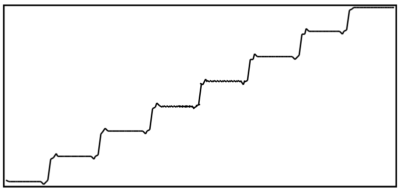
-1	-1	-1
-1	9	-1
-1	-1	-1

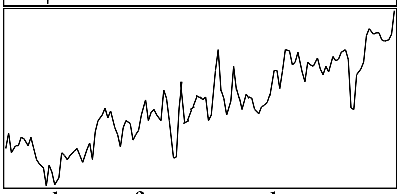
0	-1	0
-1	5	-1
0	-1	0

1	-2	1
-2	5	-2
1	-2	1

The response of a sharpening filter applied to:







the perfect step edges profile

the blurred step edges

the noisy image

Further reading and exploration

Sonka, M.et al, 4.3.1
Umbaugh, 4.3, 4.4, 3.1, 3.2
Gonzalez, R.C. & Woods, R.E., Ch. 4.

HIPR

Worksheets->Digital Filters:
- Mean Filter
- Median Filter
- Gaussian Smoothing
- Conservative Smoothing
- Crimmins Speckle Removal

CVIP

Enhancement -> Smoothing -> Mean filter
Enhancement -> Smoothing -> Median filter
Enhancement -> Smoothing -> Gaussian blur

Enhancement -> Sharpening
(try all the methods in "spatial domain filtering" division)

EXERCISES

1. Load the image **bham.tif**. Convolve it with a small (3 x 3) Gaussian kernel and observe the results. Keep a copy of the convolved image (G1).

2. Convolve the original image with a larger (e.g. 7 x 7) Gaussian kernel, keep a copy (G2). Can you predict the result of subtraction: G1 - G2?
Check your predictions by carrying out the subtraction (you may need to enhance the result using for example histogram stretch).

3. Apply a sharpening filter to the image **corr_ex.tif** and observe the results. Is it possible to obtain an improved result, i.e. the image with edges looking "sharper", through image processing operations?

4. Compare biological and computational "implementations" of smoothing and sharpening. Can both be described by the same mathematical model?