Introduction to spatial filtering

In spatial filtering pixel colours are modified according to the colours of the pixels located in its neighbourhood—called the mask (filter, kernel). The filtering procedure is a discrete convolution process performed with moving mask.

\[ g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x-t, y-s)w(t, s) \]

The result of the convolution is a filtered value at the pixel below center of the mask.

The convolution is a linear operation so the filter is linear.
Usually the central point of the mask is a center of local coordinate system. If we simplify the notation

\[
\begin{bmatrix}
  w(1,1) & w(1,0) & w(1,-1) \\
  w(0,1) & w(0,0) & w(0,-1) \\
  w(-1,1) & w(0,-1) & w(-1,-1)
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  w_1 & w_2 & w_3 \\
  w_4 & w_5 & w_6 \\
  w_7 & w_8 & w_9
\end{bmatrix}
\]

The convolution can be write down as a simple weighted sum taken from pixels covered by the mask.

\[
g(m,n) = w_1 f(m-1, n-1) + w_2 f(m-1, n) + w_3 f(m-1, n+1) \\
+ w_4 f(m, n-1) + w_5 f(m, n) + w_6 f(m, n+1) \\
+ w_7 f(m+1, n-1) + w_8 f(m+1, n) + w_9 f(m+1, n+1)
\]

For popular masks the weighting coefficients are real and symmetrical according to the center of the mask. In that case the convolution give the same result as correlation.

The weights coefficients in the mask decide about average brightness of the output image

if 1) \( \sum_{x,y \in K} w(x, y) = 1 \) brightness do not change

2) \( \sum_{x,y \in K} w(x, y) > 1 \) brightness increases

3) \( \sum_{x,y \in K} w(x, y) < 1 \) brightness decreases

In most cases the output image is normalized by sum of weighting coefficients. The resulted image brightness do not change.

\[
g(x,y) = \frac{\sum_{t,s \in K} f(x-t, y-s)w(t,s)}{\sum_{t, s \in K} w(t,s)} = \frac{\sum_{t,s \in K} f(x-t, y-s)w(t,s)}{\sum_{t \in K} w(t,s)}
\]
If all coefficients $w(x,y)=1$ filtering is a simple averaging (smoothing) operation.

Systematic analysis of image averaging:

1. Image of size 500 x 500 pixels, averaged with the masks of increasing sizes: $n=3,5,9,15,35$ pixels respectively
2. Sizes of the elements of the picture:
3. Squares: sizes 3,5,9,15,25,35,45,55, space 25 pixels
4. Circles: diameter 25, distance 15 pixels, colour decreasing by 20%
5. Bars: sizes 5 x 100, distance 20 pixels
6. Letters: from 10 to 24 pkt, increment 2 pkt, large „a” 60 pkt
7. Noise samples in rectangles of size 50 x 120 pixels
8. Background 10% of black (example: for 8bits – 230)
Averaging filters removes small perturbances and distortions in the image, smooths small edge disruptions, removes effects of colour oscillations of background and figure.

Unfortunately – the result of image averaging is blurring of the image.

Some modifications of the averaging masks: (parameters $a$ and $b$ can be choosen a priori)

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Central coefficient $w=8+a$

<table>
<thead>
<tr>
<th>1</th>
<th>b</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>$b^2$</td>
<td>b</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>1</td>
</tr>
</tbody>
</table>

Sum of weighting coefficients $\sum w=(b+2)^2$

Some interesting result we obtain if $a=0$

Most of the averaging filters have the important drawback – removing small disturbances in the picture they removes also small details of the figures and blur the edges of the figures. The most disadvantages of these filters can be avoided if we use nonlinear (median) filters.

Definition: The median of a finite list of numbers can be found by arranging all the observations from lowest value to highest value and picking the middle one. If there is an even number of observations, the median is not unique, so one often takes the mean of the two middle values. Then the median is described as the number separating the higher half of a sample, a population, or a probability distribution from the lower half.

$$
\begin{bmatrix}
1 & 13 & 19 \\
12 & 198 & 17 \\
17 & 16 & 13 \\
\end{bmatrix}
$$

Example: Median taken from the above presented mask can be found in the following way: first the numbers from the mask are arranged into increasing sequence of the numbers to obtain $\{1, 12, 13, 13, 16, 17, 17, 19, 198\}$ and then we pick the middle value 16. The arithmetic mean is equal to 34 in that case.
Median filters are the best tools to remove the local noise. Median filters do not blur the images what is done by convolution filter.

In the figures on the right the comparison of median filters (first column) and convolution filters (second column).

First example is a dark pixel surrounded by bright pixels.
Second example is a edge between dark and bright regions.

In both examples window size is equal 3 pixels.

Is median filtering is the perfect one?

The effect of corner distruction (erosion) as a result of median filtering. To show the influence of the size of median filter on figure distorsion for masks of sizes 3 x 3, 5 x 5, 7 x 7 and 9 x 9 pixels.
Median filters almost perfectly removes impulse noise called „salt and pepper“

Example of the „salt and pepper“ noise reduction: a) oryginal image, b) averaging convolutional filtering with mask of size 3 x 3  c) median filtering with mask of size 3 x 3.

Sharpening filters are used to deblurring operations. They sharpen the image, highlight the edges and enhance details (and noise unfortunately).

The most popular sharpening filters based on discrete first- and second-order derivatives. Their values are sensible on rapid changes in the signal or image.

Analyze the profile along the image on the right.

\[
\frac{\partial f}{\partial x} = f_x = f(x+1, y) - f(x, y)
\]

\[
\frac{\partial^2 f}{\partial x^2} = f_{xx} = f(x+1, y) - 2f(x, y) + f(x-1, y)
\]

It is worth to notice that second-order derivative splits the narrow edges (in 1D isolated point and step)
The simplest sharpening masks are constructed using a discrete gradient.

So called Robert's masks

\[
\begin{bmatrix}
1 & -1 \\
0 & 0 \\
1 & 0 \\
-1 & 0
\end{bmatrix}
\quad \text{or}
\quad
\begin{bmatrix}
1 & 0 \\
0 & -1 \\
0 & 1 \\
-1 & 0
\end{bmatrix}
\]

Prewitt masks

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{bmatrix}
\]

Sobel masks

\[
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
\]

45° rotated masks

\[
\begin{bmatrix}
0 & 1 & 1 \\
-1 & 0 & 1 \\
-1 & -1 & 0
\end{bmatrix}
\]

90° rotated masks

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}
\]

The directional derivatives are evaluated as

\[
f_x = f(x+1, y) - f(x, y)
\]

\[
f_y = f(x, y+1) - f(x, y)
\]

or better

\[
f_x = \frac{f(x+1, y) - f(x-1, y)}{2}
\]

\[
f_y = \frac{f(x, y+1) - f(x, y-1)}{2}
\]

There can be eight Prewitt and Sobel masks – they are rotated versions of the basic mask.

Edges in the image are usually evaluated as a magnitude of the gradient.

\[
|\nabla f| = \sqrt{(f_x)^2 + (f_y)^2}
\]

\[
|\nabla f| = |f_x| + |f_y|
\]
The sharpening filters base on first-order derivatives are direction sensitive. Below the Sobel gradient (horizontal and vertical)

The unification of both directional gradients is not directionally sensitive

Because there can be eight rotated versions of the basic mask we can obtain eight directional gradients of the original image.

The images are normalized to full range of colors.

Of course we can build the magnitude-similar image using not two but eight images.
The simplest isotropic derivative operator is the Laplace operator. It is based on Laplacian

\[ L[f(x, y)] = \nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \]

where the second order derivatives can be evaluated as

\[ \frac{\partial^2 f}{\partial x^2} = f(x-1, y) - 2f(x, y) + f(x+1, y) \]
\[ \frac{\partial^2 f}{\partial y^2} = f(x, y-1) - 2f(x, y) + f(x, y+1) \]

The possible masks are as follows:

First mask implements only horizontal and vertical directions:

\[
\begin{bmatrix}
0 & 0 & 0 \\
1 & -2 & 1 \\
0 & 0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 1 & 0 \\
0 & -2 & 0 \\
0 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
0 & -4 & 1 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

The second implementation of the Laplacian:

\[
\begin{bmatrix}
-1 & -1 & -1 \\
2 & 2 & 2 \\
-1 & -1 & -1
\end{bmatrix} + \begin{bmatrix}
-1 & 2 & -1 \\
-1 & 2 & -1 \\
-1 & 2 & -1
\end{bmatrix} = \begin{bmatrix}
-2 & 1 & -2 \\
1 & 4 & 1 \\
-2 & 1 & -2
\end{bmatrix}
\]

The basic Laplacian mask:

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

can be extend to the diagonal direction

\[
\nabla^2 = \begin{bmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
0 & -4 & 0 \\
0 & 1 & 0
\end{bmatrix} + \begin{bmatrix}
1 & 0 & 1 \\
0 & -4 & 0 \\
1 & 0 & 1
\end{bmatrix}
\]

and combine with an original image to build so called composite Laplacian mask

\[ g(x, y) = f(x, y) - \nabla^2 f(x, y) \]

On the left
a) North pool of the Moon
b) after Laplacian filtering
c) image b) scaled into interval [0,255]
d) after imposition on the original figure
Some modification of the previously discussed method in so called high-boost filtering.

\[ g(x, y) = Af(x, y) - \nabla^2 f(x, y) \]

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>0</th>
<th></th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>A+4</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>A+8</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

If the factor \( A \) increases the influence of the Laplacian on the image decreases. For large \( A \) the image tends to be identical with the original image except the significant increment of the brightness.

Scanning electron microscopy (SEM) image of the tungsten filament following the thermal failure. The image is relatively dark. The result of application of the Laplacian mask \( (A=0) \), image after application of the composite Laplacian mask \( (A=1) \), an image after high boost filtering \( (a=1.7) \).