Spatial filtering fundamentals

by Gleb V. Tcheslavski: gleb@ee.lamar.edu
http://ee.lamar.edu/gleb/dip/index.htm

Mechanics of spatial filtering

Considering frequency domain filtering, the effect of LPF applied to an image is to blur (smooth) it. Similar smoothing effect can be achieved by using spatial filters (spatial masks, kernels, templates, or windows).

We discussed that a spatial filter consists of a neighborhood and a pre-defined operation performed on the image pixels defining the neighborhood. The result of filtering – a new pixel with coordinated of the neighborhood’s center and the value defined by the operation. If the operation is linear, the filter is said to be a linear spatial filter.
Mechanics of spatial filtering

Assuming a 3 x 3 neighborhood, at any point \((x,y)\) in the image, the response of the spatial filter is

\[ g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \ldots + w(0, 0)f(x, y) + \ldots + w(1, 1)f(x + 1, y + 1) \]

Filter coefficient  Pixel intensity

In general:

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

Mechanics of spatial filtering

Here a mask size is \(m \times n\).

\[ m = 2a + 1 \]

\[ n = 2b + 1 \]

Where \(a\) and \(b\) are some integers.

For a 3 x 3 mask
Spatial correlation and convolution

**Correlation** is a process of moving the filter mask over the image and computing the sum of products at each location as previously described.

**Convolution** is the same except that the filter is first rotated by 180°.

For a 1D case, we first zero-pad \( f \) by \( m-1 \) zeros on each size. We compute a sum of products in both cases...

---

Spatial correlation and convolution

Correlation is a function of displacement of the filter.

A function containing a single 1 with the rest being zeros is called a **discrete unit impulse**. Correlation of a function with a discrete unit impulse yields a rotated version of a function at the location of the impulse.

To perform a convolution, we need to pre-rotate the filter by 180° and perform the same operation as in correlation.

---
Spatial correlation and convolution

In a 2D case, for a filter of size $m \times n$, we pad the image with $m-1$ rows of zeros at the top and bottom and $n-1$ columns of zeros on the left and right.

For convolution, we pre-rotate the mask and perform the sliding sum of products.

---

Spatial correlation and convolution

Correlation of a filter $w(x,y)$ of size $m \times n$ with an image $f(x,y)$ is

$$w(x,y) \ast f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x + s, y + t)$$

Convolution of a filter $w(x,y)$ of size $m \times n$ with an image $f(x,y)$ is

$$w(x,y) \ast f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x - s, y - t)$$
Vector representation of linear filtering

It is convenient sometimes to represent a sum of products as

\[ R = \sum_{k=1}^{mn} w_k z_k = w^T z \]

For example, for a 3 x 3 filter:

\[ R = \sum_{k=1}^{9} w_k z_k = w^T z \]

Generating spatial filter masks

Generating an \( m \times n \) linear spatial filter requires specification of \( mn \) mask coefficients. These coefficients are selected based on what the filter is supposed to do keeping in mind that all we can do with linear filtering is to implement a sum of products.

Assuming that we need to replace the pixels in an image with the average pixel intensities of a 3x3 neighborhood centered on those pixels. If \( z_i \) are the intensities, the average is

\[ R = \frac{1}{9} \sum_{i=1}^{9} z_i \]

Which is:

\[ R = \sum_{i=1}^{9} w_i z_i = w^T z; \quad w_i = \frac{1}{9} \]
Smoothing spatial filters

Smoothing filters are used for blurring and noise reduction. Blurring may be implemented in preprocessing tasks to remove small details from an image prior to large object extraction.

The output of a smoothing (averaging or lowpass) linear spatial filter is the average of the pixels contained in the neighborhood of the filter mask.

By replacing the value of every pixel in an image by the average of the intensity levels in the neighborhood defined by a filter mask, the resulting image will have reduced “sharp” transitions in intensities. Since random noise typically corresponds to such transitions, we can achieve denoising.

However, edges (characterized by sharp intensity transitions) will be blurred.

Examples of such masks:

1) A box filter – spatial averaging filter 3x3;
2) Weighted average filter – attempt to reduce blurring:

$$g(x, y) = \frac{\sum_{s=a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=a}^{a} \sum_{t=-b}^{b} w(s, t)}$$
Smoothing spatial filters

The effect of filter size.

The original 500x500 image

And the results of smoothing with a square averaging filter of sizes \( m = 3, 5, 9, 15, 25, \) and 35 pixels.

Smoothing spatial filters

Frequently, blurring is desired for ease of object detection: an original Hubble image, the result of applying a 15x15 averaging mask to it and the result of thresholding with a threshold of 25% of the highest intensity.
Order-statistic (nonlinear) filters

Order-statistic filter are nonlinear spatial filters whose response is based on ordering (Ranking) the pixels in the neighborhood and then replacing the value of the center pixel by the value determined by the ranking result.

The median filters are quite effective against the impulse noise (salt-and-pepper noise). The median of a set of values is such that half the values in the set are greater than the median and half is lesser than it:

Ex: the 3x3 neighborhood has values (10, 20, 20, 20, 15, 20, 100, 25, 20). These values are ranked as (10, 15, 20, 20, 20, 20, 20, 25, 100). The median will be 20.

There are also max and min filters.

Original image with salt-and-pepper noise
Noise reduction with a 3x3 averaging mask
Noise reduction with a 3x3 median mask
Sharpening spatial filters: foundations

The main objective of sharpening is to highlight transitions in intensity. Since averaging is analogous to spatial integration, we can assume that sharpening is analogous to differentiation in space.

The derivatives of a digital function are defined in differences.

The first derivative must be:
1) Zero in areas of constant intensity;
2) Non-zero at the onset and end of an intensity step or ramp;
3) Non-zero along ramps of constant slope.

The second derivative must be:
1) Zero in areas of constant intensity;
2) Non-zero at the onset and end of an intensity step or ramp;
3) Zero along ramps of constant slope.

Sharpening spatial filters: foundations

The first-order derivative:
\[
\frac{\partial f}{\partial x} = f(x + 1) - f(x)
\]

The second-order derivative:
\[
\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x)
\]

It can be verified that these definitions satisfy the conditions for derivatives.
Sharpening spatial filters: foundations

The circles indicate the onset or end of intensity transitions.
The sign of the second derivative changes at the onset and end of a step of ramp.
The second derivative enhances fine details much better than the first derivative. This is suitable for sharpening.

Using the second derivative for image sharpening – the Laplacian

We consider isotropic filters – the response is independent of the direction of the discontinuity in the image. Such filters are rotation invariant.
The simplest isotropic derivative operator is the Laplacian:

\[ \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \]

Therefore:

\[ \nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \]

The Laplacian is a linear operator since derivatives are linear operators.
Using the second derivative for image sharpening – the Laplacian

The Laplacian can be implemented by these filter masks:

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

Since the Laplacian is a derivative operator, its use highlights intensity discontinuities in the image and deemphasize regions with slow varying intensity levels. It tends to produce images having grayish edge lines and other discontinuities, and a dark, feature-less background.

Using the second derivative for image sharpening – the Laplacian

Background features can be preserved together with the sharpening effect of the Laplacian by adding the Laplacian image to the original.

If the definition of the Laplacian has a negative central coefficient, the Laplacian image must be subtracted rather than added to obtain a sharpening result. In general:

\[ g(x, y) = f(x, y) + c \left[ \nabla^2 f(x, y) \right] \]

-1 – if the center is negative; +1 - otherwise
Using the second derivative for image sharpening – the Laplacian

The Laplacian

The original (blurred) image

The image sharpened with mask 1

Laplacian with scaling

The image sharpened with mask 2

Unsharp masking and highboost filtering

An approach used for many years to sharpen images is:
1. Blur the original image;
2. Subtract the blurred image from the original (the result is called the mask):

\[ g_{mask}(x, y) = f_k(x, y) - \bar{f}(x, y) \]

3. Add the mask to the original:

\[ g(x, y) = f(x, y) + k \cdot g_{mask}(x, y) \]

Here \( k \) is a weight.
Unsharp masking and highboost filtering

When \( k = 1 \) – **unsharp masking**;
\( k > 1 \) – **highboost filtering**;
\( k < 1 \) – de-emphasize the contribution of a mask.

The shown intensity profile can be viewed as a horizontal scan through a vertical edge transition from a dark to a light region.

This approach is similar to Laplacian method.
Gradient method

First derivatives can be implemented for nonlinear image sharpening using the magnitude of the gradient:

\[ \nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \]

The gradient vector points in the direction of the greatest rate of change of \( f \) at location \((x,y)\). The magnitude (length) of gradient

\[ M(x, y) = |\nabla f| = \sqrt{g_x^2 + g_y^2} \]

Is the value of rate of change at \((x,y)\) in the direction of gradient.

Gradient method

\( M(x,y) \) is an image of the same size as the original and is called the gradient image. Magnitude makes \( M(x,y) \) non-linear. It is more suitable in some applications to use:

\[ M(x, y) \approx |g_x| + |g_y| \]

For an image where \( z_5 \) represent the pixel \( f(x,y) \) and \( z_7 \) represent the pixel \( f(x-1,y-1) \), the simplest (Roberts) definitions for gradients are:

\[
\begin{align*}
M(x, y) &= \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2} \\
M(x, y) &\approx |z_9 - z_5| + |z_8 - z_6|
\end{align*}
\]

However, Roberts cross-gradient operators lead to masks of even sizes, which is inconvenient.
Gradient method

The smallest masks with central symmetry (ones we are interested in) are 3x3. The gradient can be approximated for such masks as following:

\[ g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \]

\[ g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \]

Therefore, the mask could be:

\[
\begin{array}{ccc}
-1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 1 & 0 \\
\end{array}
\]

Roberts operators

\[
\begin{array}{ccc}
-1 & 0 & -1 \\
0 & 1 & 0 \\
1 & 0 & 1 \\
\end{array}
\]

Sobel operators.

The coefficients in all masks shown sum to zero. This indicates that mask will give a zero response in an area of constant intensity as expected of a derivative operator.
Combining spatial enhancement techniques

Frequently, a combination of several methods is used to enhance an image...

1) Original image – 2) Laplacian – 3) image sharpened by Laplacian – 4) Sobel gradient of the original image – 5) Sobel image smoothed with a 5x5 averaging filter – 6) product of Sobel image with its smoothed version – 7) sharpened image (a sum of the original and 6) – 8) power-law transformation.