Input image:

1.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 100 & 100 & 0 \\ 0 & 100 & 100 & 100 & 0 \\ 0 & 100 & 100 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Subtracting Laplacian:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Result of filtering (convolution):

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -100 & -100 & -100 & 0 & 0 \\ 0 & -100 & 300 & 200 & 300 & -100 & 0 \\ 0 & -100 & 200 & 100 & 200 & -100 & 0 \\ 0 & -100 & 300 & 200 & 300 & -100 & 0 \\ 0 & 0 & -100 & -100 & -100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

The resulting image has enhanced edges, but also has negative values and ringing or over/undershoot (increased positive and negative values) around edges.

The subtracting Laplacian is an edge-enhancement operator. However, being based on the second derivative (Laplacian) it causes ringing or over/undershoot around edges. The operator maintains the overall intensity (average value) of the image. (The sum of the mask coefficients = 1.)

2. Unsharp masking filter $\begin{bmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 2 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{bmatrix}$

$$g(m,n) = 2f(m,n) - \tfrac{1}{8}\,[f(m-1,n-1) + f(m-1,n) + f(m-1,n+1) + f(m,n-1) + f(m,n+1) + f(m+1,n-1) + f(m+1,n) + f(m+1,n+1)]$$

$$\text{MTF} \quad H(k,l) = \frac{G(k,l)}{F(k,l)} = \frac{FT[g(m,n)]}{FT[f(m,n)]}$$

now, $FT[f(m-1,n-1)] = F_1(k,l) =$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-1,n-1)\exp\left[-j\frac{2\pi}{N}(mk+nl)\right]$$

using a $N \times N$ DFT array with zero padding as needed, and ignoring the $1/N$ scale factor.

Let $m-1 = a$ and $n-1 = b$.

Ignoring the effects at the edges,

$$F_1(k,l) = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\exp\left[-j\frac{2\pi}{N}\{(a+1)k+(b+1)l\}\right]$$

$$= \exp\left[-j\frac{2\pi}{N}(k+l)\right]\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\exp\left[-j\frac{2\pi}{N}(ak+bl)\right]$$

$$= \exp\left[-j\frac{2\pi}{N}(k+l)\right]F(k,l).$$

similarly, $FT[f(m+1,n)] = \exp\left[j\frac{2\pi}{N}k\right]F(k,l)$,

$FT[f(m-1,n)] = \exp\left[-j\frac{2\pi}{N}k\right]F(k,l)$, etc.

2. Continued:

$$\therefore G(k,l) = 2F(k,l) - \frac{1}{8}F(k,l)\{$$

$$\exp\left[-j\frac{2\pi}{N}(k+l)\right] + \exp\left[+j\frac{2\pi}{N}(k+l)\right]$$

$$+ \exp\left[-j\frac{2\pi}{N}(k)\right] + \exp\left[+j\frac{2\pi}{N}(k)\right]$$

$$+ \exp\left[-j\frac{2\pi}{N}(l)\right] + \exp\left[+j\frac{2\pi}{N}(l)\right]$$

$$+ \exp\left[-j\frac{2\pi}{N}(k-l)\right] + \exp\left[+j\frac{2\pi}{N}(k-l)\right]\}$$

$$H(k,l) = G(k,l)/F(k,l)$$

$$= 2 - \frac{1}{4}\left\{\cos\frac{2\pi}{N}(k+l) + \cos\frac{2\pi}{N}k + \cos\frac{2\pi}{N}l + \cos\frac{2\pi}{N}(k-l)\right\}$$

DC gain $= H(0,0) = 2 - \frac{1}{4}\{1+1+1+1\} = 1.$

Folding frequency = half the sampling rate occurs at $k = N/2$ and/or $l = N/2$.

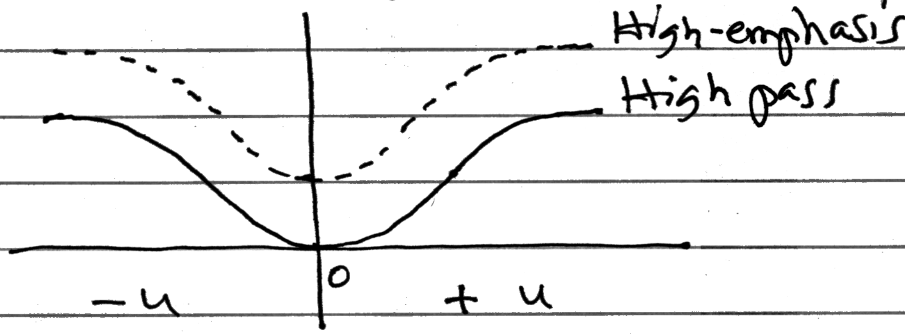
At $k = l = N/2$, the gain is

$$2 - \frac{1}{4}\{\cos 2\pi + \cos\pi + \cos\pi + \cos 0\}$$

$$= 2 - \frac{1}{4}(1 - 1 - 1 + 1) = 2$$

$\therefore$ The gain increases with frequency, with a nonzero gain at DC. The filter is a high-frequency emphasis filter, and performs edge enhancement.

3. The transfer functions of both the highpass and high-emphasis filters increase with frequency. However, the highpass filter has a zero gain at DC or (0, 0) frequency, whereas the high-emphasis filter has a nonzero gain at DC.



Sectional view of the transfer functions of Butterworth highpass and high-emphasis filters.

In the image domain, a highpass filter yields edges only. The result will have positive and negative values, with an average value of zero over the entire image. Intensity information is lost. The filter performs edge extraction.

A high-emphasis filter performs edge enhancement. Intensity information is maintained. Edges relate to high-frequency components, and are enhanced (sharpened). The resulting image could have negative pixel values.

4. i) Take the 2D FT of the image. Pad the images prior to the use of an FFT algorithm with zeros to an $N \times N$ array such an $N = 2^k$, k = an integer.

ii) Fold or shift the FT array to have the DC or (0,0) frequency at the center of the array.

iii) Multiply every point in the FT array with the corresponding value of the filter transfer function. Note that the filter function must also have its (0,0) component at the center of its array. Note that the FT array is complex; most filter functions are real.

iv) Unfold or shift the FT array.

v) Take the inverse FT of the result.

vi) Take the <u>real part</u> of the result.

vii) Select the appropriate part of the result if zero-padding was performed in step (i).

<u>Note</u>: If the filter is given as a space-domain mask, its transfer function must be obtained by taking the FT of an $N \times N$ zero-padded array. The result must be folded or shifted [prior to step (iii)].

5. Sobel masks

$$G_x : \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y : \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

To detect edges of any orientation in $f(m,n)$:

i) get $\left.\begin{aligned} g_x(m,n) &= f(m,n) * G_x \\ g_y(m,n) &= f(m,n) * G_y \end{aligned}\right\}$ $* = $ 2D convolution

ii) compose vectorial (complex) gradient

$$\underline{g}(m,n) = g_x(m,n) + j\, g_y(m,n)$$

iii) get magnitude of gradient

$$\| \underline{g}(m,n) \| = \left[g_x^2(m,n) + g_y^2(m,n) \right]^{1/2}$$

iv) get angle of gradient

$$\angle \underline{g}(m,n) = \tan^{-1}\left[g_y(m,n) / g_x(m,n) \right]$$

v) Threshold the magnitude image to detect significant edges. The angle image gives the corresponding orientation.

Limitations: Local operators; limited spatial scope of operation; no scale variability; edges could be broken/disjoint; noise enhanced.
To improve: use edge linking to fill in holes in edges. Reject isolated small segments.