**Project**

**High-boost filter**

CPE 214

Signals and Systems

Instructor

Professor ‎[Bundit Thipakorn](https://www.facebook.com/bundit.thipakorn?fref=nf)

Professor Suthathip Manee

Author Name

Mr. Siwat Kaolueng ID 56070503434

Mr. NaiprakanNakornkhet ID 56070503419

Mr. Saurawit Sanguangphokai ID 56070503437

Mr. Saran Keawkajee ID 560705034XX

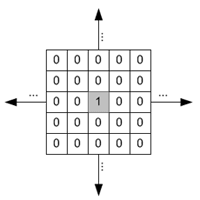
SECTION C

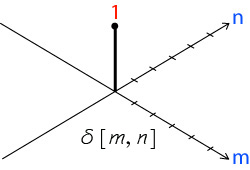
Department of Computer Engineering

King Mongkut’s University of Technology Thonburi

**Theory**

All pass filter

* We know that if we have input. We then convolute it with delta function. Output will be the same as input. Therefore, it calls all pass filter

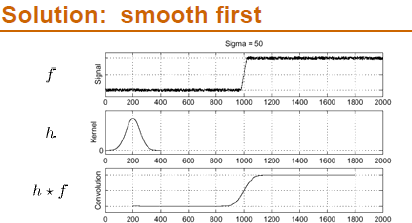


\begin{displaymath}W_{ap}=\left[ \begin{array}{ccc} 0 & 0 & 0  0 & 1 & 0 \\
0 & 0 & 0 \end{array} \right] \end{displaymath}

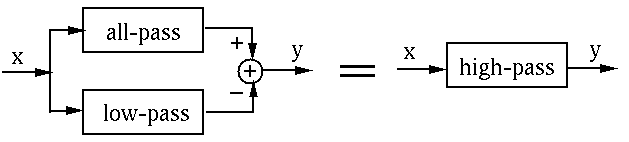
When an image is convolved with this delta function is not changed:

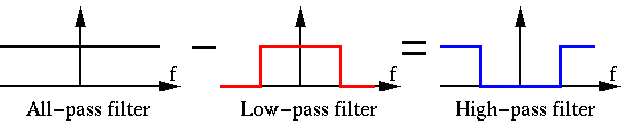
\begin{displaymath}W_{ap} * I_o = I_o \end{displaymath}

Low pass filter

* **A low-pass filter**, also called a "blurring" or "smoothing" filter, averages out rapid changes in intensity.
* 

High boosting filter

* In image processing, it is often desirable to emphasize high frequency components representing the image details without eliminating low frequency components (such as sharpening). The **high-boost filter** can be used to enhance high frequency component.
* High boost filter is composed by an **all pass filter** and a **edge detection(high pass)** filter (**laplacian filter**). Thus, it emphasizes edges and results in image sharpener.
* As the opposite of low-pass filtering for image smoothing and noise reduction, high-pass filtering can sharpen the image, thereby enhancing and emphasizing the detailed information (high spatial frequency components) in the image.
* High-pass filtering can be carried out by subtracting the low-pass filtered image from its original version, which can be considered as all-pass filtered by a delta function kernel:
* As convolution is a linear operation, we have
* \begin{displaymath}I_{hp}&=& I_{ap}-I_{lp}=W_{ap}*I_o-W_{lp}*I_o
  =(W_{ap}-W_{lp})*I_o=W_{hp}*I_o
  \end{displaymath}
* where $W_{hp}=W_{ap}-W_{lp}$ is the high-pass kernel corresponding to the low-pass kernel $W_{lp}$. Equivalently the frequency domain filtering is shown in the figure:
* 



We can therefore obtain a high-pass filtering kernel corresponding to each of the low-pass filter kernels by subtracting the low-pass kernel from the all-pass kernel. The resulting kernels are various forms of high-pass filtering kernels, also called the Laplace operators. \begin{displaymath}W_{hp}=W_{ap}-W_{lp}
=\left[ \begin{array}{ccc} 0 & 0 & 0 \\...
...0 & -1 & 0  -1 & 4 & -1 \\
0 & -1 & 0 \end{array} \right]
\end{displaymath}

\begin{displaymath}W_{hp}=W_{ap}-W_{lp}
=\left[ \begin{array}{ccc} 0 & 0 & 0 \\...
... -1 & -1  -1 & 8 & -1 \\
-1 & -1 & -1 \end{array} \right]
\end{displaymath}

\begin{displaymath}W_{hp}=W_{ap}-W_{lp}
=\left[ \begin{array}{ccc} 0 & 0 & 0  ...
...-2 & -1  -2 & 12 & -2 \\
-1 & -2 & -1 \end{array} \right]
\end{displaymath}

**Code**

**lena4.m**

%http://www.aquaphoenix.com/lecture/matlab10/page3.html

%niaokaia = imread('NiaoKai.JPG');

lena = imread('Lenna.PNG');

c=1;

subplot(2,2,1), image(lena), title('Lena original');

gaussianFilter = [1,4,7,4,1;4,20,33,20,4;7,33,55,33,7;4,20,33,20,4;1,4,7,4,1];

gaussianFilter = gaussianFilter / sum(sum(gaussianFilter));

%mean = [1/16 1/16 1/16 1/16; 1/16 1/16 1/16 1/16; 1/16 1/16 1/16 1/16; 1/16 1/16 1/16 1/16];

bright = [1/4 1/4; 1/4 1/4];

gaussianLenna = imfilter(lena, gaussianFilter);

subplot(2,2,2), image(gaussianLenna), title('Gaussian Lenna');

laplcian = [-1 -1 -1; -1 8 -1; -1 -1 -1];

gaussiantolaplcian = imfilter(gaussianLenna,laplcian);

a = imfilter(lena,laplcian);

%subplot(2,2,2), image(a, title('Edge detection (Laplacian) without Blurring');

subplot(2,2,3), image(gaussiantolaplcian), title('Gaussian-> Laplcian Lenna = (GOS)Edge detection');

sharpLena = gaussianLenna+gaussiantolaplcian;

subplot(2,2,4), image(sharpLena), title('Sharp Lena');

**lena5.m**

lena = imread('Niaokai.jpg');

c=5;

subplot(2,2,1), image(lena), title('Lena original');

%highpass= [0 0 0; 0 1 0; 0 0 0] - 1/9\*[1 1 1; 1 1 1; 1 1 1]; %all pass - low pass

%sharpLena = imfilter(lena, gaussianFilter);

original = imfilter(lena, [0 0 0; 0 1 0; 0 0 0]);

blur = imfilter(lena, 1/9\*[1 1 1; 1 1 1; 1 1 1]);

subplot(2,2,2), image(blur), title('Blur or Smooth');

edge=c\*(original-blur);

%d=[0 0 0; 0 1 0; 0 0 0]-(1/9\*[1 1 1; 1 1 1; 1 1 1]);

%sharp = imfilter(lena,d);

%e=imcompliment(e); black to white, white to black

subplot(2,2,3), image(edge), title('Edge = (Original - Blur)');

sharp= lena+edge;

subplot(2,2,4), image(sharp), title('Sharp');

**lena.6m**

lena = imread('Lenna.png');%Lenna.png NiaoKai.JPG

subplot(2,2,1), image(lena), title('Lena original');

c=2;

lena = imfilter(lena,gaussianFilter);

allpass = [0 0 0; 0 1 0; 0 0 0];

highpass = [-1 -1 -1; -1 8 -1; -1 -1 -1];

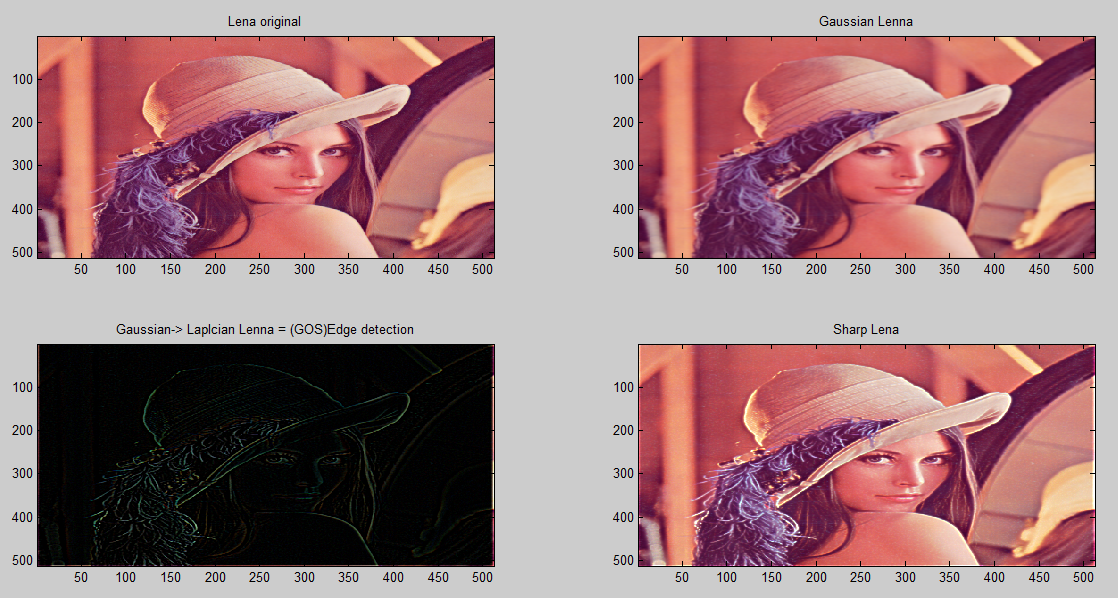
ahighpass = allpass+ c\*highpass;

y = imfilter(lena, ahighpass);

subplot(2,2,2), image(y), title('High-boost filtering');

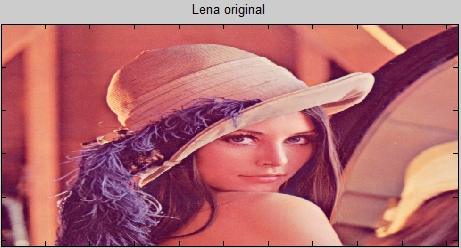
**Result**

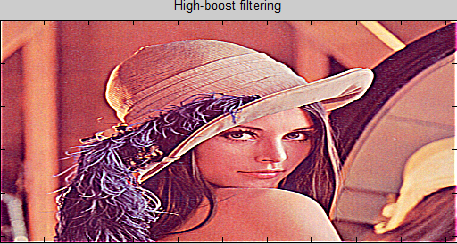
**lena4.m**



**lena4.m**

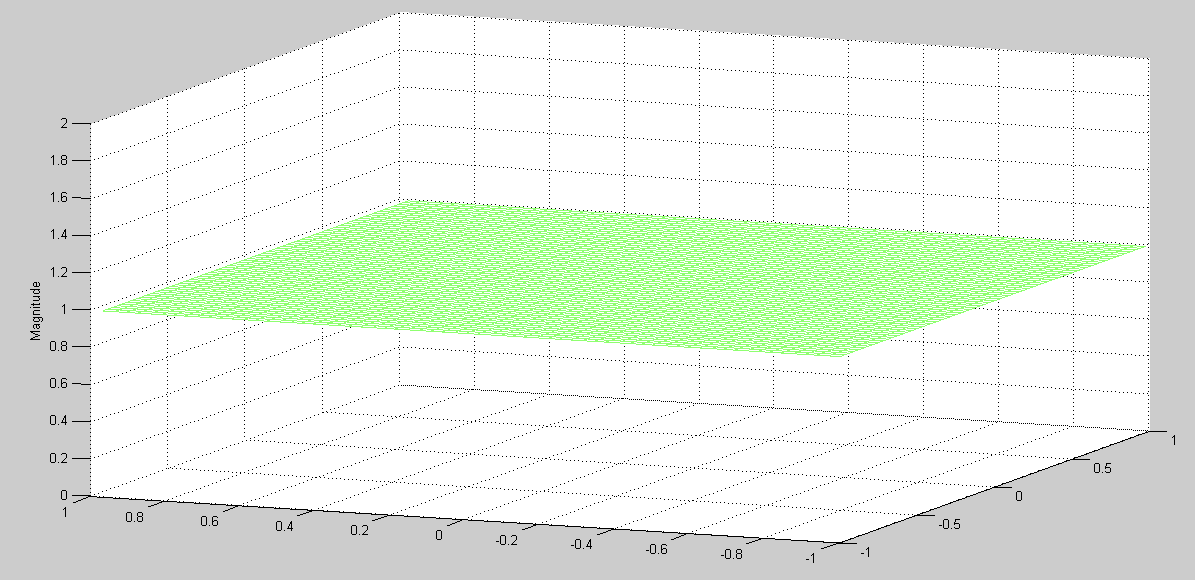
**lena6.m**

****

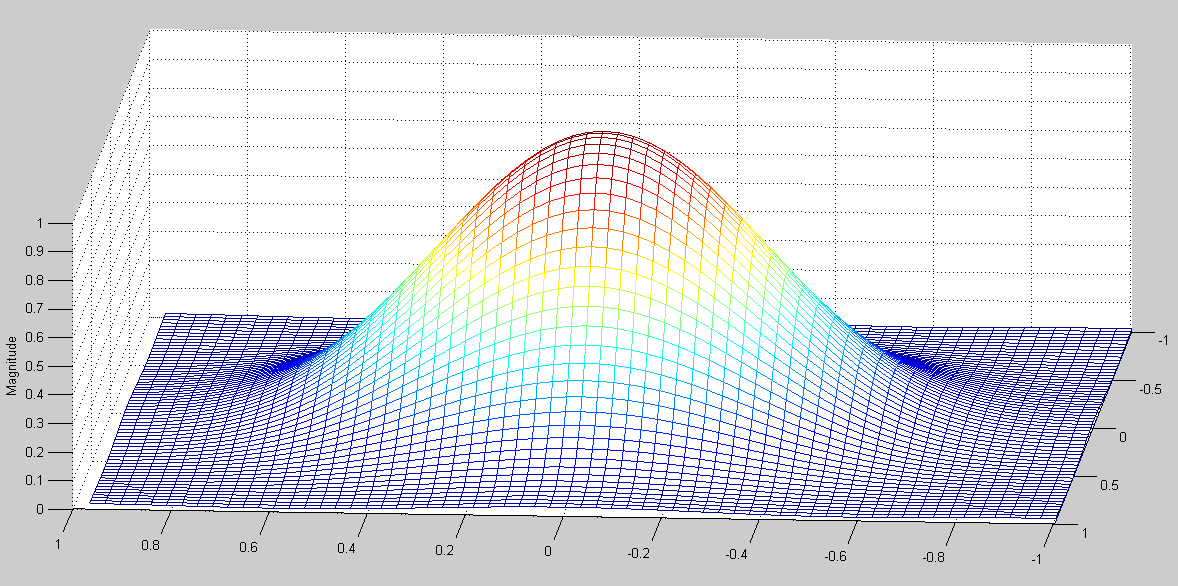
****

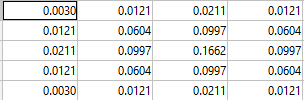
**Coefficient = 2**

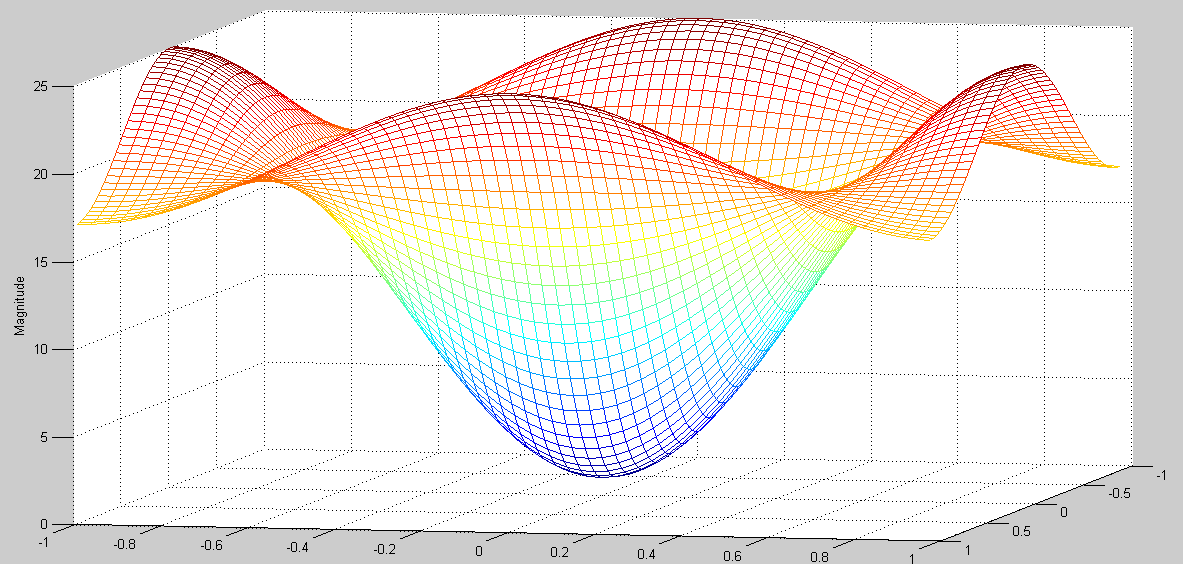
**Frequency response**

****

**All pass (Same picture) **

****

**Low pass (Gaussian For blur)** ****

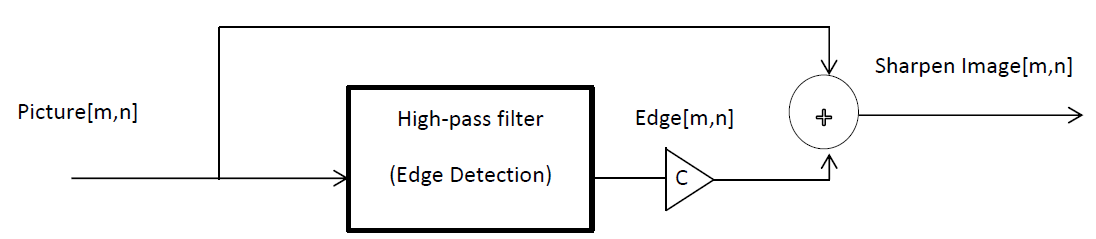
****

**High pass = Delta function + Coefficent\* Laplacian **

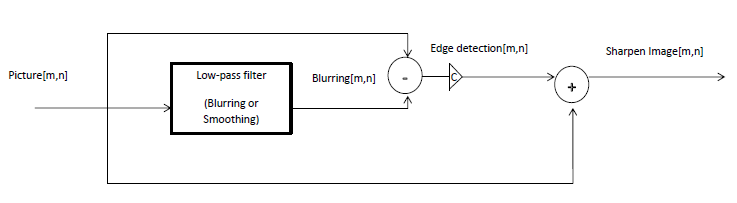
**= Delta function + Coefficient\* (Delta function – Gaussian)**

**Block diagram**

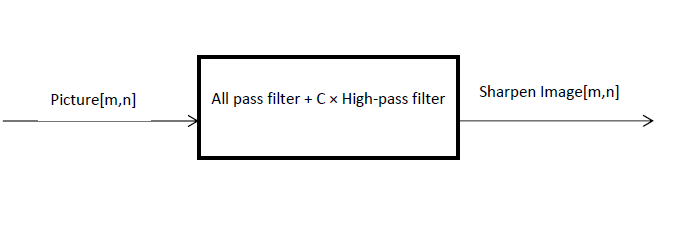
**lena4.m**

****

**lena5.m**

****

**lena6.m**

****

**\*If input picture have noise, we must reduce it by using blurring or smoothing. Because after we find edge or high pass, all noises will be emphasized, that is incorrect. On the other hands if we have clean picture without noise, if we blur or smoothen it, all edge can disappear, that is not good. Finally, we have to consider picture must be pre image processing or not.**

**Reference**

* **http://fourier.eng.hmc.edu/e161/lectures/gradient/node7.html**
* **http://oscar.iitb.ac.in/OSCARPP/Electrical%20Engineering/UploadedStoryboards/High%20Boost%20filtering.ppt**
* **http://alumni.media.mit.edu/~maov/classes/vision09/lect/09\_Image\_Filtering\_Edge\_Detection\_09.pdf**