DIGITAL IMAGE PROCESSING

18EC733

LECTURE NOTES

Module -2

IMAGE ENHANCEMENT

Image Enhancement in Spatial domain

Image enhancement approaches fall into two broad categories: spatial domain methods and frequency domain methods. The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.

Frequency domain processing techniques are based on modifying the Fourier transform of an image. Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. Image enhancement has very good applications. It is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c. As indicated previously, the term spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression.

$$g(x,y) = T[f(x,y)]$$

where f(x, y) is the input image, g(x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular subimage area centered at (x, y), as Fig. 2.1 shows. The center of the subimage is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.

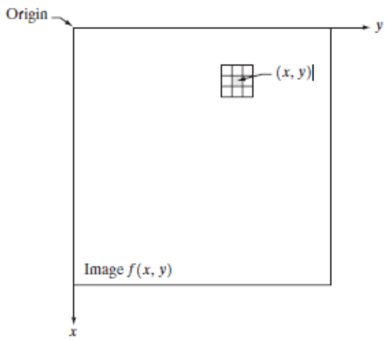


Fig.: 3x3 neighborhood about a point (x,y) in an image.

The simplest form of T is when the neighborhood is of size 1*1 (that is, a single pixel). In this case, g depends only on the value of f at (x, y), and T becomes a gray-level (also called an intensity or mapping) transformation function of the form

$$s = T(r)$$

where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of 'r' to each value of 's'.

For example, if T(r) has the form shown in Fig. 2.2(a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation function into a narrow range of s, toward black.The opposite effect takes place for values of r above m.

In the limiting case shown in Fig. 2.2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.

One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig. 2.1, in which the values of the mask coefficients determine the nature of the process, such as image sharpening. Enhancement techniques based on this type of approach often are referred to as mask processing or filtering.

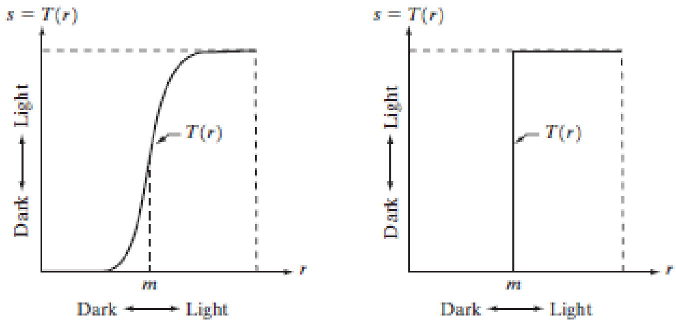


Fig. 2.2 Gray level transformation functions for contrast enhancement.

Image enhancement can be done through gray level transformations which are discussed below.

BASIC GRAY LEVEL TRANSFORMATIONS:

- Image negative
- Log transformations
- Power law transformations
- Piecewise-Linear transformation functions

LINEAR TRANSFORMATION:

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transformation has been discussed in our tutorial of image transformation, but a brief description of this transformation has been given here.

Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below:

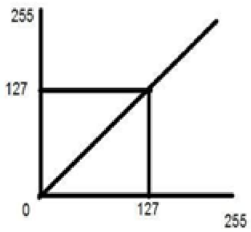


Fig. Linear transformation between input and output.

NEGATIVE TRANSFORMATION:

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image

IMAGE NEGATIVE: The image negative with gray level value in the range of [0, L-1] is obtained by negative transformation given by $S = T(r)$ or

$$S = L - 1 - r$$

Where r= gray level value at pixel (x,y)

L is the largest gray level consists in the image

It results in getting photograph negative. It is useful when for enhancing white details embedded in dark regions of the image.

The overall graph of these transitions has been shown below.

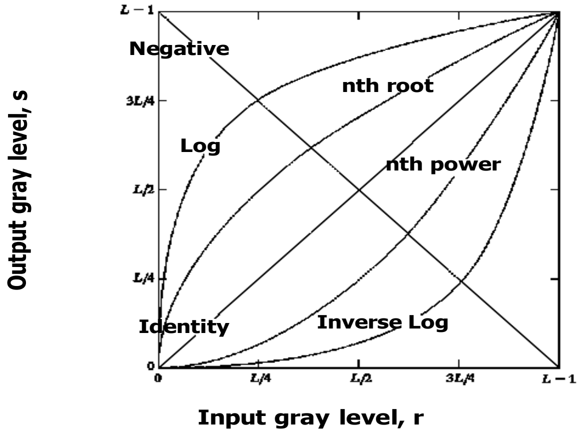


Fig. Some basic gray-level transformation functions used for image enhancement.

In this case the following transition has been done.

$$s = (L - 1) - r$$

since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this

$$s = 255 - r$$

So each value is subtracted by 255 and the result image has been shown above. So what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

It has been shown in the graph below.

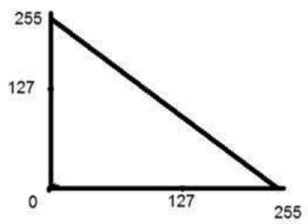


Fig. Negative transformations.

LOGARITHMIC TRANSFORMATIONS:

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

LOG TRANSFORMATIONS:

The log transformations can be defined by this formula

$$s = c \log(r + 1).$$

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.

An another way of representing LOG TRANSFORMATIONS: Enhance details in the darker regions of an image at the expense of detail in brighter regions.

$$T(f) = C * \log(1+r)$$

- Here C is constant and $r \geq 0$.

- The shape of the curve shows that this transformation maps the narrow range of low gray level values in the input image into a wider range of output image.
- The opposite is true for high level values of input image.

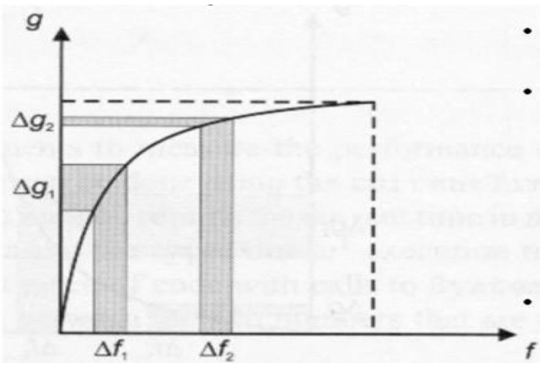


Fig. log transformation curve input vs output

POWER – LAW TRANSFORMATIONS:

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

$$s=cr^{\gamma}$$

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

where c and g are positive constants. Sometimes Eq. (6) is written as $S = C(r+\varepsilon)^{\gamma}$ to account for an offset (that is, a measurable output when the input is zero). Plots of s versus r for various values of γ are shown in Fig. 2.10. As in the case of the log transformation, power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels. Unlike the log function, however, we notice here a family of possible transformation curves obtained simply by varying γ.

In Fig that curves generated with values of $\gamma>1$ have exactly The opposite effect as those generated with values of $\gamma<1$. Finally, we Note that Eq. (6) reduces to the identity transformation when $c=\gamma=1$.

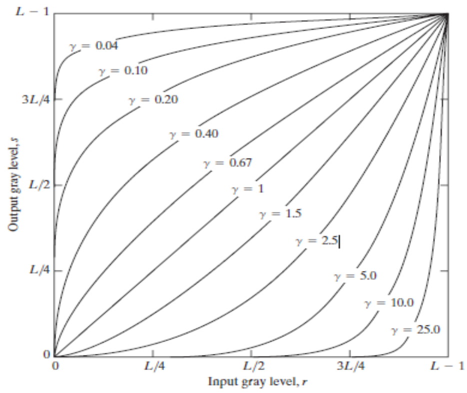


Fig. 2.13 Plot of the equation $S = cr^{\gamma}$ for various values of γ (c =1 in all cases).

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Varying gamma (γ) obtains family of possible transformation curves $S = C^* r^{\gamma}$

Here C and γ are positive constants. Plot of S versus r for various values of γ is

$\gamma > 1$ compresses dark values

Expands bright values

$\gamma < 1$ (similar to Log transformation)

Expands dark values

Compresses bright values

When $C = \gamma = 1$, it reduces to identity transformation .

CORRECTING GAMMA:

$s=cr^{\gamma}$

$s=cr^{(1/2.5)}$

The same image but with different gamma values has been shown here.

Piecewise-Linear Transformation Functions:

A complementary approach to the methods discussed in the previous three sections is to use piecewise linear functions. The principal advantage of piecewise linear functions over the types of functions we have discussed thus far is that the form of piecewise functions can be arbitrarily complex.

The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

Contrast stretching: One of the simplest piecewise linear functions is a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition.

$$S = T(r)$$

Figure x(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation

Function. If r1=s1 and r2=s2, the transformation is a linear function that produces No changes in gray levels. If r1=r2, s1=0and s2= L-1, the transformation Becomes a thresholding function that creates a binary image, as illustrated In fig. 2.2(b).

Intermediate values of ar1, s1b and ar2, s2b produce various degrees Of spread in the gray levels of the output image, thus affecting its contrast. In general, r1≤ r2 and s1 ≤ s2 is assumed so that the function is single valued and Monotonically increasing.

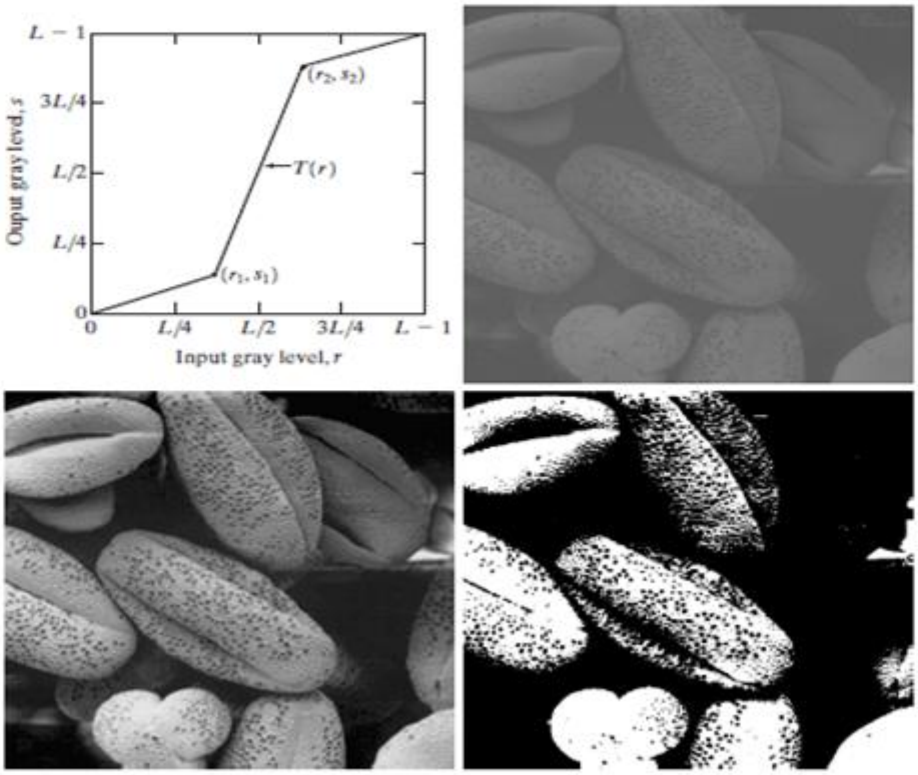


Fig. x Contrast stretching. (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding (original image courtesy of Dr.Roger Heady, Research School of Biological Sciences, Australian National University Canberra Australia.

Figure x(b) shows an 8-bit image with low contrast. Fig. x(c) shows the result of contrast stretching, obtained by setting (r1, s1)=(rmin, 0) and (r2, s2)=(r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively.Thus, the transformation function stretched the levels linearly from their original range to the full range

[0, L-1]. Finally, Fig. x(d) shows the result of using the thresholding function defined previously,

with r1=r2=m, the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.

There are several ways of doing level slicing, but most of them are variations of two basic themes.One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.

This transformation, shown in Fig. y(a), produces a binary image. The second approach, based on the transformation shown in Fig.y (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure y (c) shows a gray-scale image, and Fig. y(d) shows the result of using the transformation in Fig. y(a).Variations of the two transformations shown in Fig. are easy to formulate.

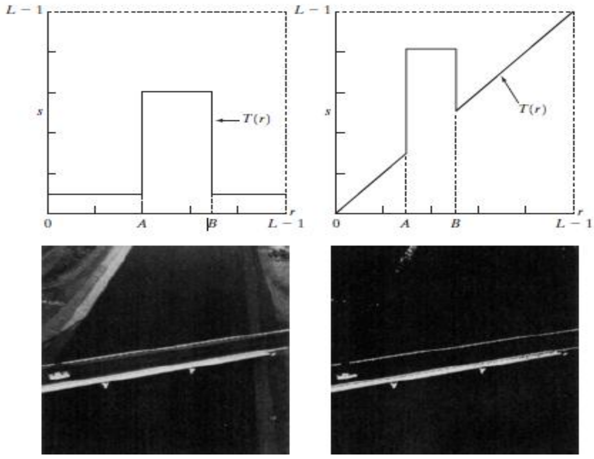


Fig. y (a)This transformation highlights range [A,B] of gray levels and reduces all others to a constant level (b) This transformation highlights range [A,B] but preserves all other levels. (c) An image . (d) Result of using the transformation in (a).

BIT-PLANE SLICING:

Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for the most significant bit. In terms of 8-

bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.

Figure 3.12 illustrates these ideas, and Fig. 3.14 shows the various bit planes for the image shown in Fig. 3.13. Note that the higher-order bits (especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

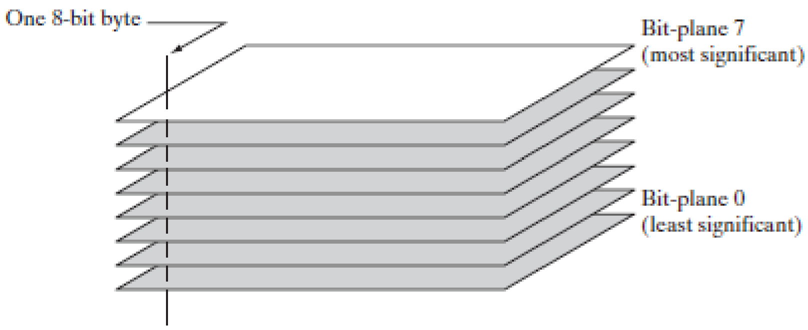


FIGURE Bit-plane representation of an 8-bit image.

In terms of bit-plane extraction for an 8-bit image, it is not difficult to show that the (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation function that (1) maps all levels in the image between 0 and 127 to one level (for example, 0); and (2) maps all levels between 129 and 255 to another (for example, 255).The binary image for bit-plane 7 in Fig. 3.14 was obtained in just this manner. It is left as an exercise

(Problem 3.3) to obtain the gray-level transformation functions that would yield the other bit planes.

Histogram Processing:

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function of the form

$$H(r_k)=n_k$$

where rk is the kth gray level and nk is the number of pixels in the image having the level rk.. A normalized histogram is given by the equation

$$p(r_k)=n_k/n \text{ for } k=0,1,2,\ldots,L-1$$

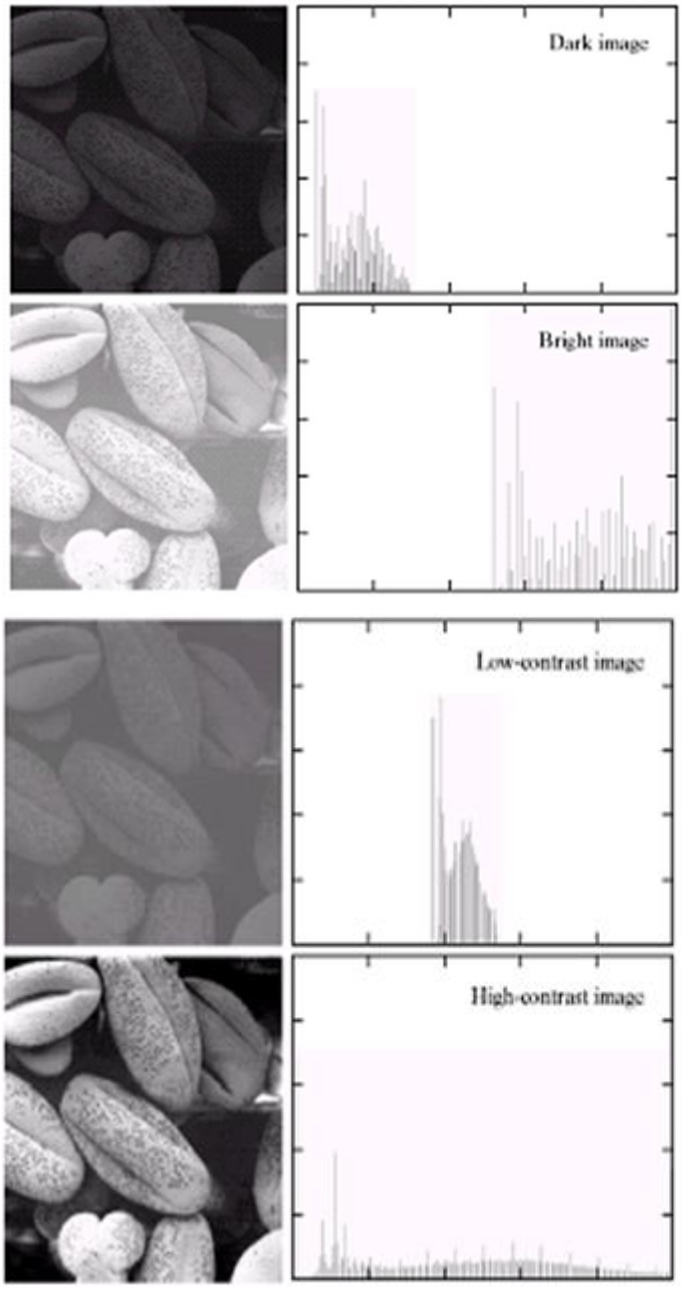
P(rk) gives the estimate of the probability of occurrence of gray level rk.

The sum of all components of a normalized histogram is equal to 1.

The histogram plots are simple plots of H(rk)=nk versus rk.

In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale. In case of bright image the histogram components are baised towards the high side of the gray scale. The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.

The components of the histogram in the high contrast image cover a broad range of the gray scale. The net effect of this will be an image that shows a great deal of gray levels details and has high dynamic range.



Histogram Equalization:

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail are compressed into the dark end of the histogram. If we could ‘stretch out’ the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.

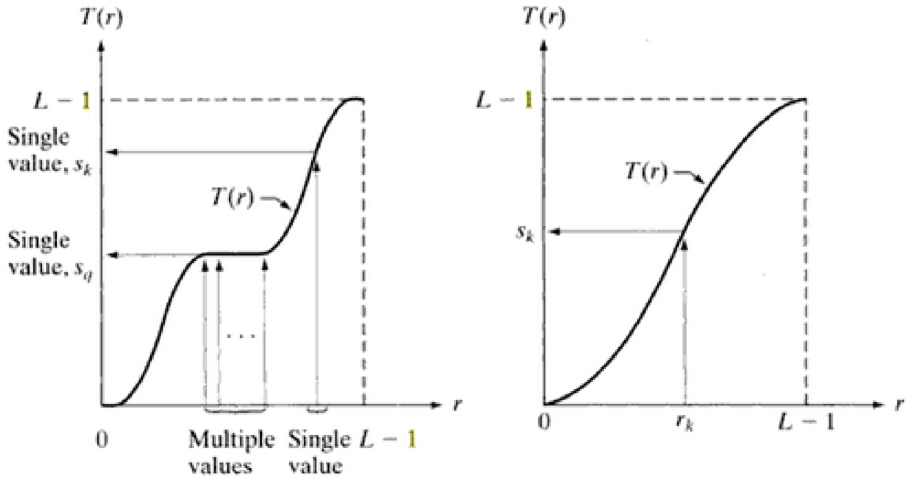
Let there be a continuous function with r being gray levels of the image to be enhanced. The range of r is [0, 1] with r=0 repressing black and r=1 representing white. The transformation function is of the form

$$S=T(r) \text{ where } 0<r<1$$

It produces a level s for every pixel value r in the original image.

a b

FIGURE (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.



The transformation function is assumed to fulfill two condition T(r) is single valued and monotonically increasing in the internal 0<T(r)<1 for 0<r<1.The transformation function should be single valued so that the inverse transformations should exist. Monotonically increasing condition preserves the increasing order from black to white in the output image. The second conditions guarantee that the output gray levels will be in the same range as the input levels. The gray levels of the image may be viewed as random variables in the interval [0.1]. The most fundamental descriptor of a random variable is its probability density function (PDF) Pr(r) and Ps(s) denote the probability density functions of random variables r and s respectively. Basic results from an elementary probability theory states that if Pr(r) and Tr are known and T-1(s) satisfies conditions (a), then the probability density function Ps(s) of the transformed variable is given by the formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Thus the PDF of the transformed variable s is the determined by the gray levels PDF of the input image and by the chosen transformations function.

A transformation function of a particular importance in image processing

$$s = T(r) = (L - 1)\int_0^r p_r(w)\,dw$$

This is the cumulative distribution function of r.

L is the total number of possible gray levels in the image.

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

BLURRING/NOISE REDUCTION: Noise characterized by sharp transitions in image intensity. Such transitions contribute significantly to high frequency components of Fourier transform. Intuitively, attenuating certain high frequency components result in blurring and reduction of image noise.

IDEAL LOW-PASS FILTER:

Cuts off all high-frequency components at a distance greater than a certain distance from origin (cutoff frequency).

$$H(u,v) = \begin{cases} 1, & \text{if } D(u,v) \le D_0 \\ 0, & \text{if } D(u,v) > D_0 \end{cases}$$

Where D0 is a positive constant and D(u,v) is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle; that is

$$D(u,v) = [(u-P/2)^2 + (v-Q/2)^2]^{1/2}$$

Where as P and Q are the padded sizes from the basic equations

Wraparound error in their circular convolution can be avoided by padding these functions with zeros,

VISUALIZATION: IDEAL LOW PASS FILTER:

Aa shown in fig.below

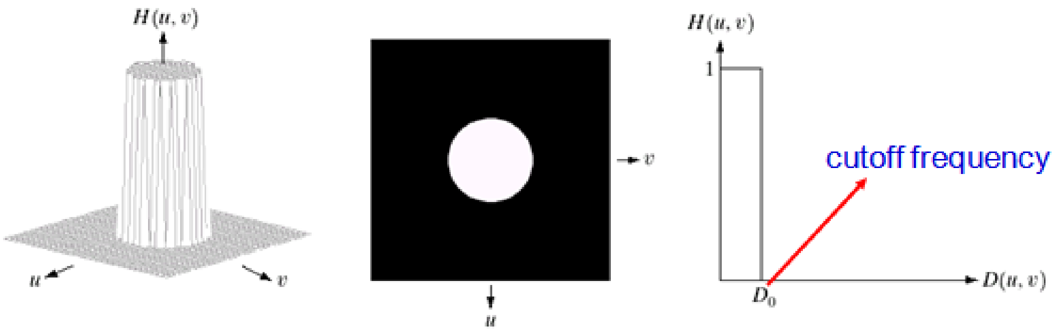


Fig: ideal low pass filter 3-D view and 2-D view and line graph.

EFFECT OF DIFFERENT CUTOFF FREQUENCIES:

Fig.below(a) Test pattern of size 688x688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160 and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8 and 99.2% of the padded image power respectively.

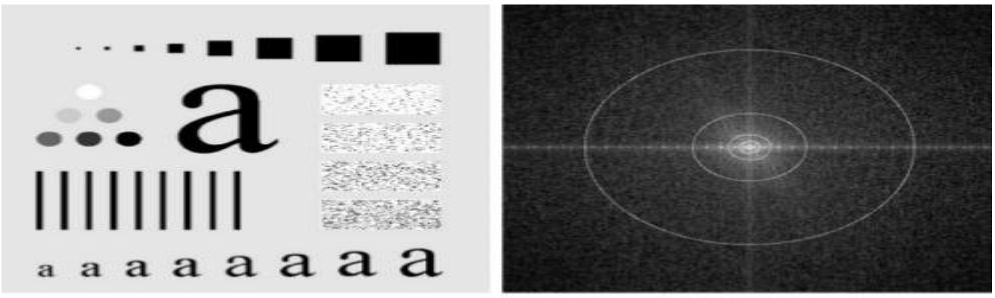


Fig: (a) Test patter of size 688x688 pixels (b) its Fourier spectrum

Fig: (a) original image, (b)-(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160 and 460, as shown in fig.2.2.2(b). The power removed by these filters was 13, 6.9, 4.3, 2.2 and 0.8% of the total, respectively.

As the cutoff frequency decreases,

- image becomes more blurred
- Noise becomes increases
- Analogous to larger spatial filter sizes

The severe blurring in this image is a clear indication that most of the sharp detail information in the picture is contained in the 13% power removed by the filter. As the filter radius is increases less and less power is removed, resulting in less blurring. Fig. (c) through (e) are characterized by “ringing” , which becomes finer in texture as the amount of high frequency content removed decreases.

WHY IS THERE RINGING?

Ideal low-pass filter function is a rectangular function

The inverse Fourier transform of a rectangular function is a sinc function.

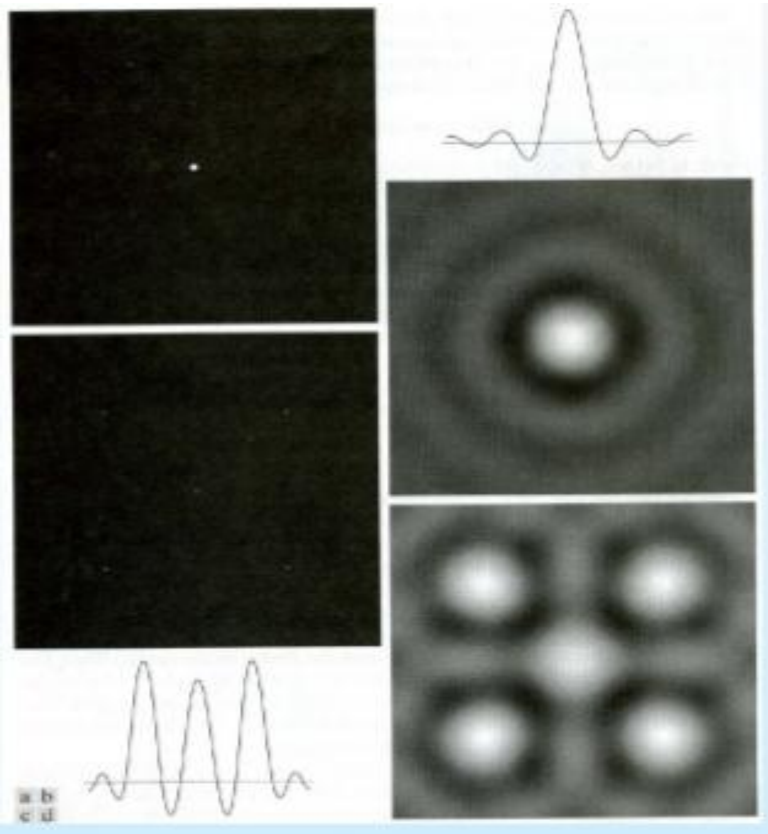


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity profiles through the center of the filters(the size of all cases is 1000x1000 and the cutoff frequency is 5), observe how ringing increases as a function of filter order.

BUTTERWORTH LOW-PASS FILTER:

Transfor funtion of a Butterworth lowpass filter (BLPF) of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as

- $$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

Transfer function does not have sharp discontinuity establishing cutoff between passed and filtered frequencies.

Cut off frequency D_0 defines point at which H(u,v) = 0.5

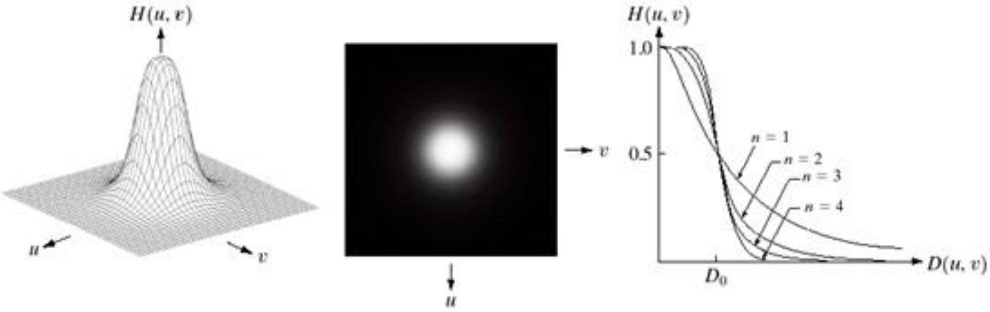


Fig. (a) perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c)Filter radial cross sections of order 1 through 4.

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that gives a clear cutoff between passed and filtered frequencies.

BUTTERWORTH LOW-PASS FILTERS OF DIFFEREN T FREQUENCIES:

Fig. (a) Original image.(b)-(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii

Fig. shows the results of applying the BLPF of eq. to fig.(a), with n=2 and D0 equal to the five radii in fig.(b) for the ILPF, we note here a smooth transition in blurring as a function of increasing cutoff frequency. Moreover, no ringing is visible in any of the images processed with this particular BLPF, a fact attributed to the filter's smooth transition between low and high frequencies.

A BLPF of order 1 has no ringing in the spatial domain. Ringing generally is imperceptible in filters of order 2, but can become significant in filters of higher order.

Fig.shows a comparison between the spatial representation of BLPFs of various orders (using a cutoff frequency of 5 in all cases). Shown also is the intensity profile along a horizontal scan line through the center of each filter. The filter of order 2 does show mild ringing and small negative values, but they certainly are less pronounced than in the ILPF. A butter worth filter of order 20 exhibits characteristics similar to those of the ILPF (in the limit, both filters are identical).

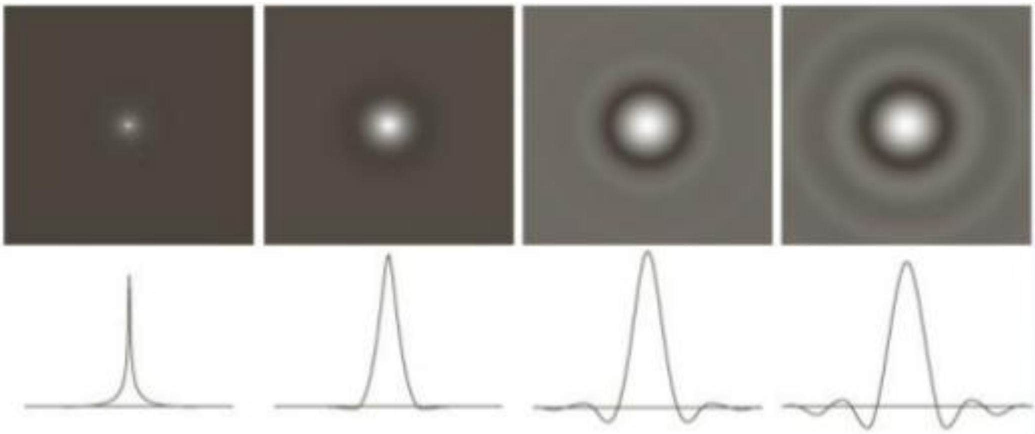


Fig.2.2.7 (a)-(d) Spatial representation of BLPFs of order 1, 2, 5 and 20 and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 x 1000 and the cutoff frequency is 5) Observe how ringing increases as a function of filter order.

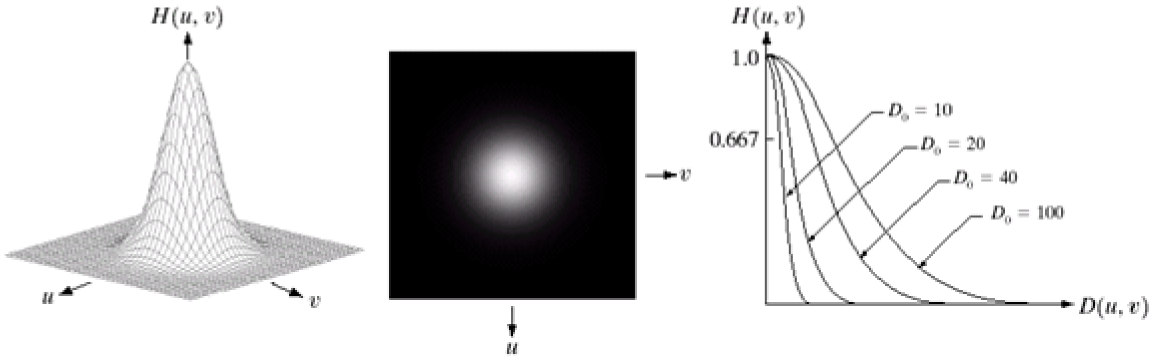
GAUSSIAN LOWPASS FILTERS:

The form of these filters in two dimensions is given by

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

- This transfer function is smooth , like Butterworth filter.
- Gaussian in frequency domain remains a Gaussian in spatial domain
- Advantage: No ringing artifacts.

Where D_0 is the cutoff frequency. When $D(u,v) = D_0$, the GLPF is down to 0.607 of its maximum value. This means that a spatial Gaussian filter, obtained by computing the IDFT of above equation., will have no ringing. Fig..shows a perspective plot, image display and radial cross sections of a GLPF function.



a b c

FIGURE (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Fig. (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c). Filter radial cross sections for various values of D_0

Fig.(a) Original image. (b)-(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in fig.2.2.2. compare with fig.2.2.3 and fig.2.2.6

a b c

Fig. (a) Original image (784x 732 pixels). (b) Result of filtering using a GLPF with D0 = 100. (c) Result of filtering using a GLPF with D0 = 80. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

Fig. shows an application of lowpass filtering for producing a smoother, softer-looking result from a sharp original. For human faces, the typical objective is to reduce the sharpness of fine skin lines and small blemished.

IMAGE SHARPENING USING FREQUENCY DOMAIN FILTERS:

An image can be smoothed by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in intensities are associated with high-frequency components, image sharpening can be achieved in the frequency domain by high pass filtering, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

The filter function H(u,v) are understood to be discrete functions of size PxQ; that is the discrete frequency variables are in the range u = 0,1,2,…….P-1 and v = 0,1,2,…….Q-1.

The meaning of sharpening is

- Edges and fine detail characterized by sharp transitions in image intensity
- Such transitions contribute significantly to high frequency components of Fourier transform
- Intuitively, attenuating certain low frequency components and preserving high frequency components result in sharpening.

Intended goal is to do the reverse operation of low-pass filters

- When low-pass filter attenuated frequencies, high-pass filter passes them

- When high-pass filter attenuates frequencies, low-pass filter passes them.

A high pass filter is obtained from a given low pass filter using the equation.

$$H_{hp}(u,v) = 1 - H_{tp}(u,v)$$

Where $H_{lp}(u,v)$ is the transfer function of the low-pass filter. That is when the low-pass filter attenuates frequencies, the high-pass filter passed them, and vice-versa.

We consider ideal, Butter-worth, and Gaussian high-pass filters. As in the previous section, we illustrate the characteristics of these filters in both the frequency and spatial domains. Fig.. shows typical 3-D plots, image representations and cross sections for these filters. As before, we see that the Butter-worth filter represents a transition between the sharpness of the ideal filter and the broad smoothness of the Gaussian filter. Fig.discussed in the sections the follow, illustrates what these filters look like in the spatial domain. The spatial filters were obtained and displayed by using the procedure used.

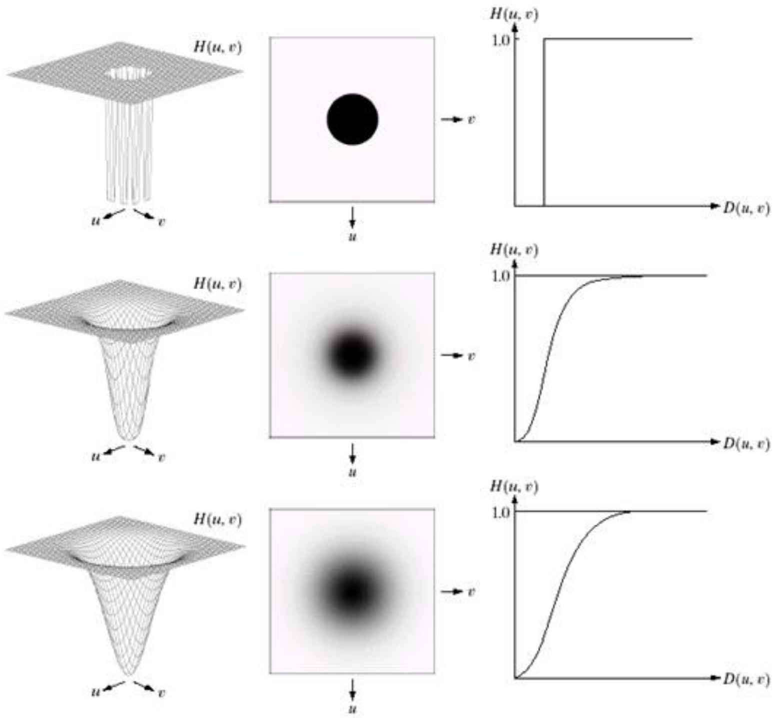


Fig: Top row: Perspective plot, image representation, and cross section of a typical ideal high-pass filter. Middle and bottom rows: The same sequence for typical butter-worth and Gaussian high-pass filters.

IDEAL HIGH-PASS FILTER:

A 2-D ideal high-pass filter (IHPF) is defined as

$$H(u,v) = \begin{cases} 0, & \text{if } D(u,v) \le D_0 \\ 1, & \text{if } D(u,v) > D_0 \end{cases}$$

Where D_0 is the cutoff frequency and D(u,v) is given by eq. As intended, the IHPF is the opposite of the ILPF in the sense that it sets to zero all frequencies inside a circle of radius D_0 while passing, without attenuation, all frequencies outside the circle. As in case of the ILPF, the IHPF is not physically realizable.

SPATIAL REPRESENTATION OF HIGHPASS FILTERS:

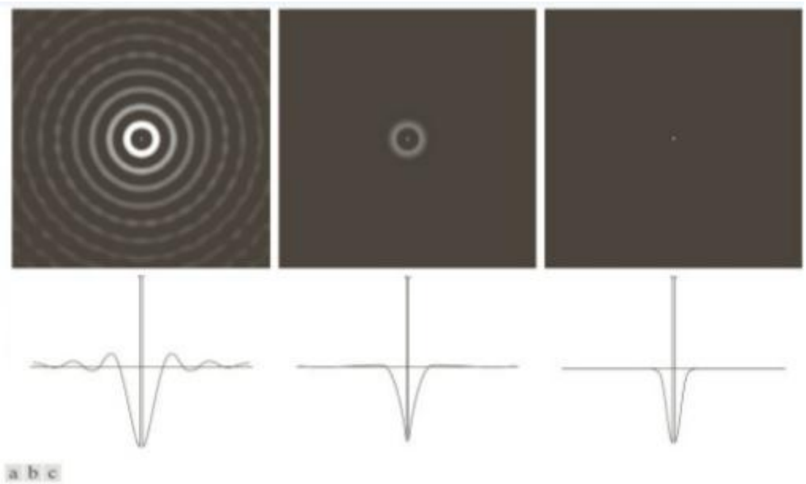


Fig.. Spatial representation of typical (a) ideal (b) Butter-worth and (c) Gaussian frequency domain high-pass filters, and corresponding intensity profiles through their centers.

We can expect IHPFs to have the same ringing properties as ILPFs. This is demonstrated clearly in Fig.. which consists of various IHPF results using the original image in Fig.(a) with D_0 set to 30, 60,and 160 pixels, respectively. The ringing in Fig. (a) is so severe that it produced distorted, thickened object boundaries (e.g.,look at the large letter "a"). Edges of the top three circles do not show well because they are not as strong as the other edges in the image (the intensity of these three objects is much closer to the background intensity, giving discontinuities of smaller magnitude).

FILTERED RESULTS: IHPF:

Fig.. Results of high-pass filtering the image in Fig.(a) using an IHPF with D_0 = 30, 60, and 160.

The situation improved somewhat with D0 = 60. Edge distortion is quite evident still, but now we begin to see filtering on the smaller objects. Due to the now familiar inverse relationship between the frequency and spatial domains, we know that the spot size of this filter is smaller than the spot of the filter with D0 = 30. The result for D0 = 160 is closer to what a high-pass filtered image should look like. Here, the edges are much cleaner and less distorted, and the smaller objects have been filtered properly.

Of course, the constant background in all images is zero in these high-pass filtered images because highpass filtering is analogous to differentiation in the spatial domain.

BUTTER-WORTH HIGH-PASS FILTERS:

A 2-D Butter-worth high-pass filter (BHPF) of order n and cutoff frequency D_0 is defined as

$$H(u,v) = \frac{1}{1 + \left[D_0 / D(u,v) \right]^{2n}}$$

Where D(u,v) is given by Eq.(3). This expression follows directly from Eqs.(3) and (6). The middle row of Fig.2.2.11. shows an image and cross section of the BHPF function.

Butter-worth high-pass filter to behave smoother than IHPFs. Fig.2.2.14.shows the performance of a BHPF of order 2 and with D0 set to the same values as in Fig.2.2.13. The boundaries are much less distorted than in Fig.2.2.13. even for the smallest value of cutoff frequency.

FILTERED RESULTS: BHPF:

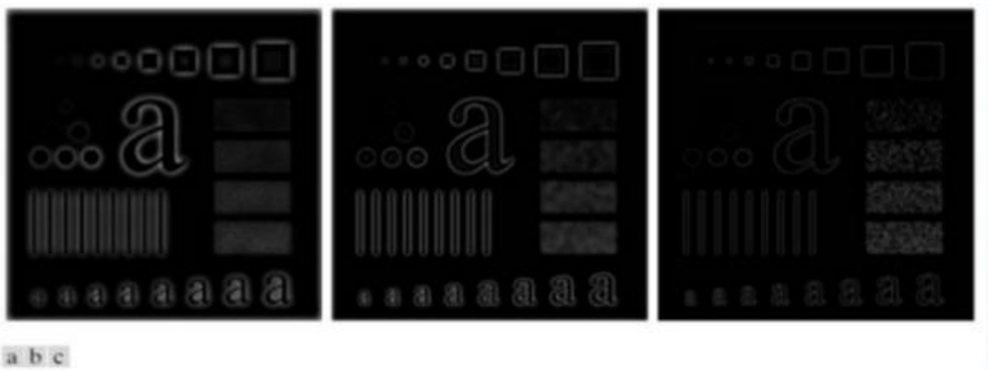


Fig. Results of high-pass filtering the image in Fig.2.2.2(a) using a BHPF of order 2 with D_0 = 30, 60, and 160 corresponding to the circles in Fig.2.2.2(b). These results are much smoother than those obtained with an IHPF.

GAUSSIAN HIGH-PASS FILTERS:

The transfer function of the Gaussian high-pass filter(GHPF) with cutoff frequency locus at a distance D_0 from the center of the frequency rectangle is given by

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Where D(u,v) is given by Eq.(4). This expression follows directly from Eqs.(2) and (6). The third row in Fig.2.2.11. shows a perspective plot, image and cross section of the GHPF function. Following the same format as for the BHPF, we show in Fig.2.2.15. comparable results using GHPFs. As expected, the results obtained are more gradual than with the previous two filters.

FILTERED RESULTS:GHPF:

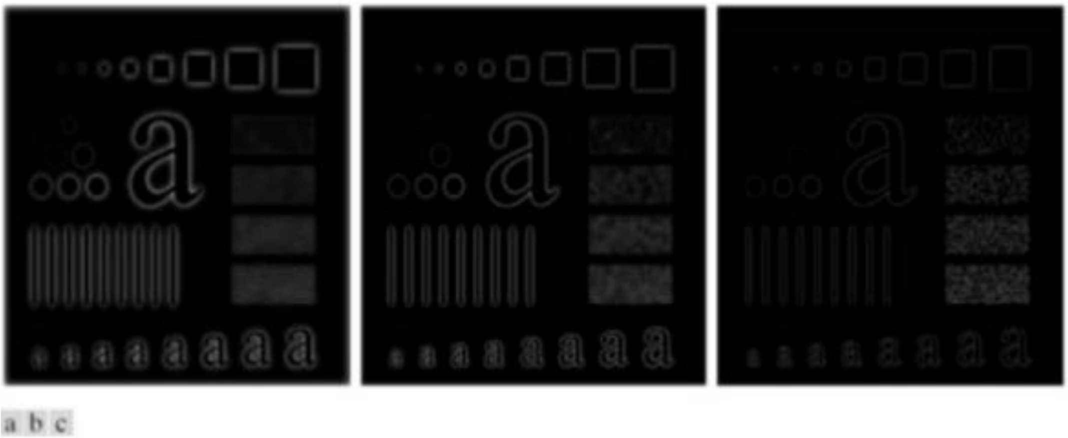


Fig. Results of high-pass filtering the image in fig.(a) using a GHPF with D0 = 30, 60 and 160, corresponding to the circles in Fig.(b).

IMAGE TRANSFORMS:

2-D FFT:

2D Discrete Fourier Transform

The independent variable (t,x,y) is discrete

$$F_r = \sum_{k=0}^{N_0-1} f[k]e^{-jr\Omega_0 k}$$

$$f_{N_0}[k] = \frac{1}{N_0}\sum_{r=0}^{N_0-1} F_r e^{jr\Omega_0 k}$$

$$\Omega_0 = \frac{2\pi}{N_0}$$

$$F[u,v] = \sum_{i=0}^{N_0-1}\sum_{k=0}^{N_0-1} f[i,k]e^{-j\Omega_0(ui+vk)}$$

$$f_{N_0}[i,k] = \frac{1}{N_0^{\,2}}\sum_{u=0}^{N_0-1}\sum_{v=0}^{N_0-1} F[u,v]e^{j\Omega_0(ui+vk)}$$

$$\Omega_0 = \frac{2\pi}{N_0}$$

Properties

- Linearity $\quad af(x,y)+bg(x,y) \Leftrightarrow aF(u,v)+bG(u,v)$
- Shifting $\quad f(x-x_0, y-x_0) \Leftrightarrow e^{-j2\pi(ux_0+vy_0)}F(u,v)$
- Modulation $\quad e^{j2\pi(u_0x+v_0y)}f(x,y) \Leftrightarrow F(u-u_0, v-v_0)$
- Convolution $\quad f(x,y)*g(x,y) \Leftrightarrow F(u,v)G(u,v)$
- Multiplication $\quad f(x,y)g(x,y) \Leftrightarrow F(u,v)*G(u,v)$
- Separability $\quad f(x,y)=f(x)f(y) \Leftrightarrow F(u,v)=F(u)F(v)$

Separability

1. Separability of the 2D Fourier transform
   – 2D Fourier Transforms can be implemented as a sequence of 1D Fourier Transform operations performed *independently* along the two axis

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy=$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}e^{-j2\pi vy}dxdy=\int_{-\infty}^{\infty} e^{-j2\pi vy}dy\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}dx=$$

$$=\int_{-\infty}^{\infty} F(u,y)e^{-j2\pi vy}dy=F(u,v)$$

2D DFT ⇒ 1D DFT along the rows → 1D DFT along the cols

Separability

- Separable functions can be written as $f(x,y)=f(x)g(y)$

2. The FT of a separable function is the product of the FTs of the two functions

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy=$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x)g(y)e^{-j2\pi ux}e^{-j2\pi vy}dxdy=\int_{-\infty}^{\infty} g(y)e^{-j2\pi vy}dy\int_{-\infty}^{\infty} h(x)e^{-j2\pi ux}dx=$$

$$=H(u)G(v)$$

$$f(x,y)=h(x)g(y)\Rightarrow F(u,v)=H(u)G(v)$$