3 Filtering and Fourier Methods

- ✓ Smoothing
- ➡ Edge Sharpening
- Fourier Filtering
- Pyramid Filtering

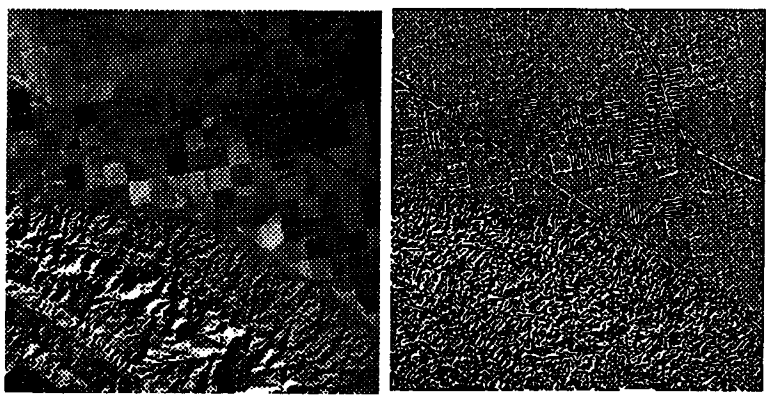
Picture degradation due to image blurring is a common problem, often caused by poor optical systems or by camera/object motion during image acquisition. Blurring is the result of an averaging operation over a local neighborhood and is a type of integration process (in the mathematical sense). To compensate for this integration process, one might expect that a differentiation process could be used to sharpen edges in the image. In this section, one method (out of many) for compensating for image blur by sharpening or crispening edges is presented.

Alternatively, since blurring attenuates high frequencies, another method for sharpening an image might be to accentuate high frequencies. This can be accomplished rather easily in the Fourier domain, where it is known as high-emphasis filtering.

3 Edge Sharpening

The LAPLACIAN* is a linear operator which computes an approximation to the second derivative of the image function.

0	1	0
1	-4	1
0	1	0



* for more information, see the section on Edge Detection

The Laplacian mask approximates the second derivative of the image function. By convolving this mask over the blurred image, the result is an image that represents the rate of change of edges in the original image. Thus, the magnitude of the Laplacian is high where the image function is changing most rapidly. It is zero in absolutely uniform areas, as an inspection of the mask will confirm.

3 The Laplacian and Edge Sharpening

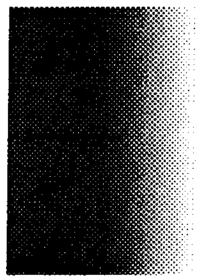
- The Laplacian of an image is often written as $\nabla^2 I$.
- The operator $S[I(x,y)] = I(x,y) - \nabla^2 I(x,y)$ will sharpen edges in an image.
- A detailed explanation of why this occurs can be found in the Edge Detection section.

The Laplacian operator plays an important role in detecting edges in images and we will return to it in the edge detection section. For the purposes of this section, one use for the Laplacian is as an edge sharpening operator. Subtracting the Laplacian image from the original image has the effect of enhancing edges, thereby sharpening the visual appearance of the image.

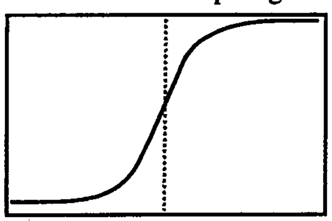
One disadvantage of this operator is that, since it is a second derivative operator, it has the effect of enhancing noise as well as structure. Therefore, it should be applied only in areas which have low noise, or areas which have been subjected to a noise reduction operator.

3 An Intuitive Rationale

Ramp Edge Image



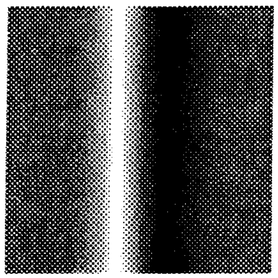
Profile of Ramp Edge



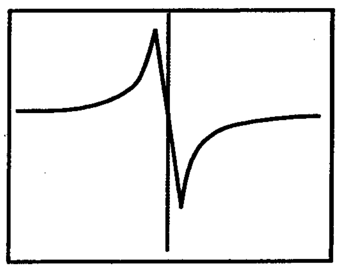
An intuitive understanding of why subtraction of a Laplacian image from the original results in edge sharpening can be seen by examining this slide and the next two. The first slide contains an image of a ramp edge; immediately below it is a cross sectional profile of the edge.

3 An Intuitive Rationale continued

Laplacian Applied to Ramp Edge



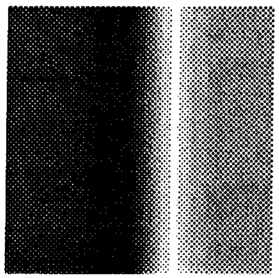
Profile of Laplacian Result



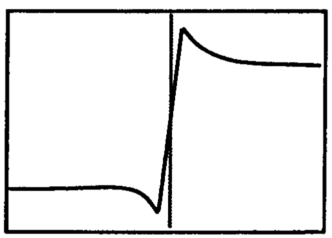
The second image is the result of convolving the 3x3 Laplacian mask over the rampedge image. This image approximates the second derivative of the step edge image. The Laplacian image is composed of positive and negative pixel values. By comparing the two profiles, it can be seen that the Laplacian is zero where the image is constant, and has its maximum value (both positive and negative) where the rate of change of the intensity profile is greatest. Also note that the Laplacian is zero at the inflection point of the rampedge. We will use this fact in the section on edge detection to develop an edge detector based on this 'zero-crossing'.

3 An Intuitive Rationale continued

Result after Subtracting
Laplacian from Original Image



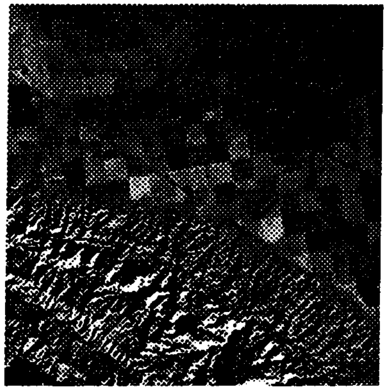
Profile of Enhanced Edge



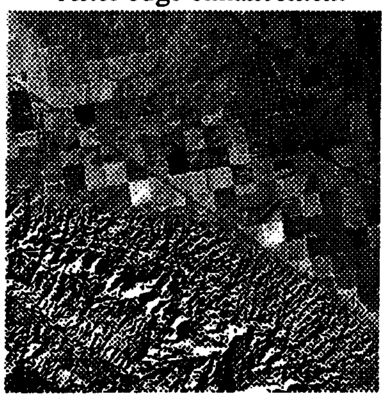
This last image shows the results of subtracting the Laplacian image from the original rampedge image. By examining the profile of the enhanced edge it is evident that the edge now has over- and under-shoots at its beginning and end, while the gray levels to either side of the edge are the same as in the original image. This has the effect of increasing the step size of the edge and localizing it more in space. The visual result is qualitatively a more 'crisp' image, as the next example shows.

3 Edge Enhanced Result

Before edge enhancement



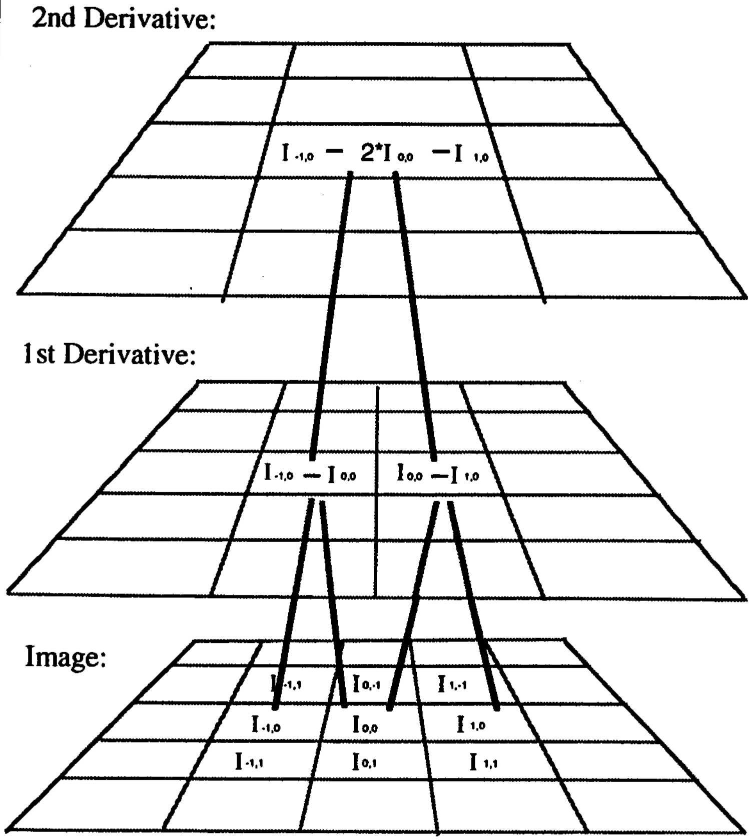
After edge enhancement



The slide shows an example of an image before and after edge enhancement using the 3x3 Laplacian. The lower image is noticeably 'crisper', particularly in the area of the mountains in the lower part of the image. In this case, the enhancement technique worked reasonably well. However, performing edge enhancement in this way may not be appropriate for all images. How well it is judged to have worked is highly dependent on the goal(s) of the user.

As stated earlier, edge enhancement emphasizes all high frequency events in an image. As noise is typically high frequency, edge enhancement could also result in a less coherent image. For this reason, images are often subjected to noise reduction processes prior to enhancement.

3 Basis for Laplacian Mask



This figure shows how the Laplacian mask computes the approximation to the second derivative of an image. Here, only one dimension (x direction) is considered.

3 Laplacian Masks

One-Dimensional Laplacian Mask:

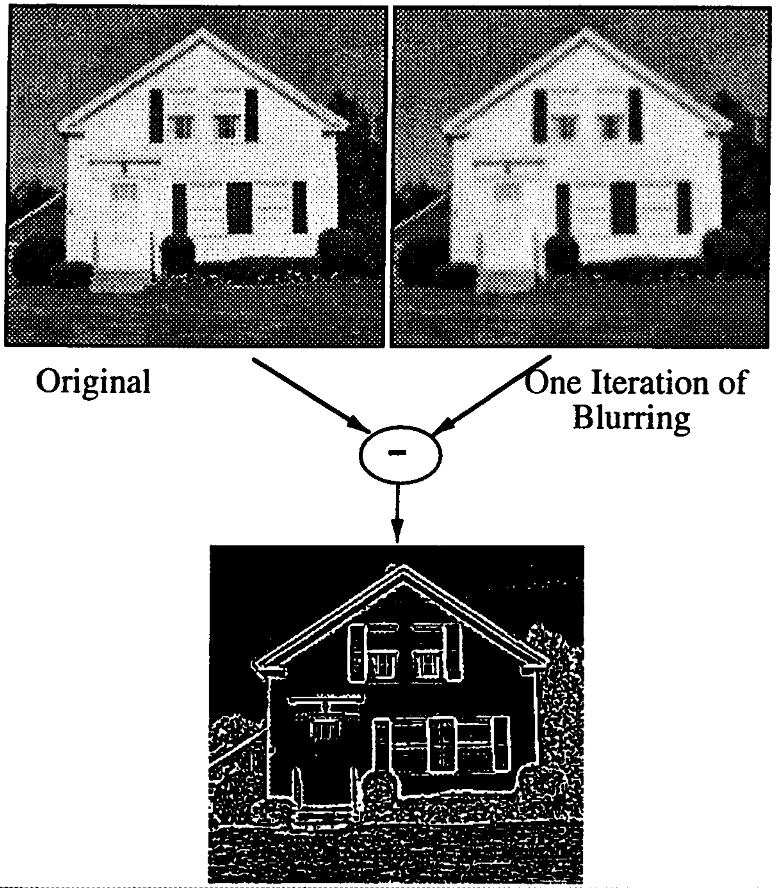
1	-2	1

Two-Dimensional Laplacian Mask:

0	1	0
1	-4	1
0	1	0

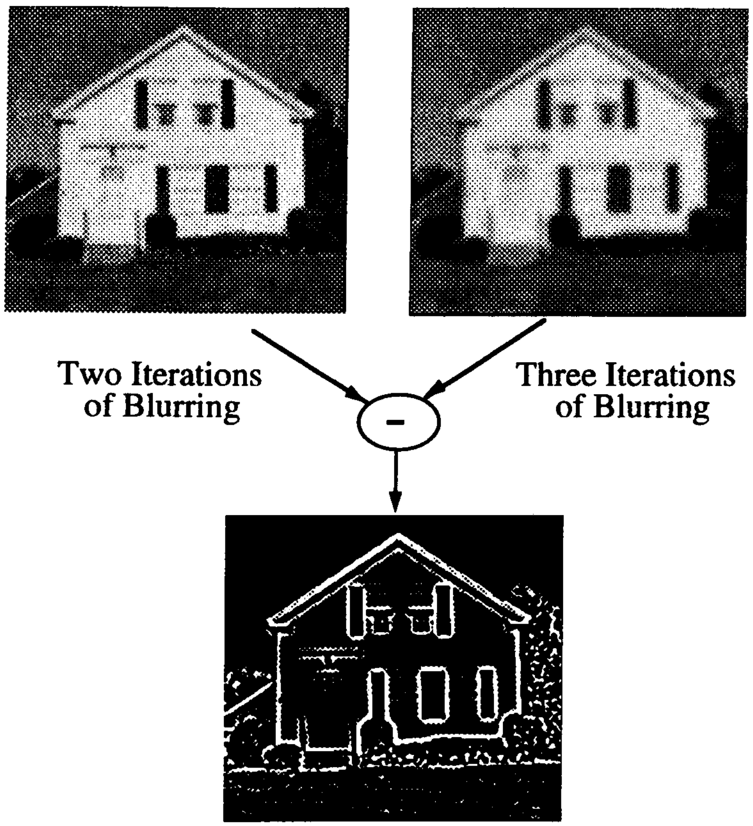
The two-dimensional Laplacian mask can be derived in the same manner as the one-dimensional mask, as shown on the previous slide. A family of Laplacian masks can be defined as a function of the size of the mask and various pixel weighting schemes; we will return to this in the section on edge detection.

3 A Potential Use for Averaging/Blurring



The figure shows the results of performing a pixel by pixel difference between the original image and the image obtained after one iteration of Gaussian averaging. The difference image looks very much like an edge image - that is, it is bright where there are edges in the original image and zero elsewhere. In Gaussian averaging, the weights in the masks approximate a Gaussian distribution (we will return to Gaussian filtering later on).

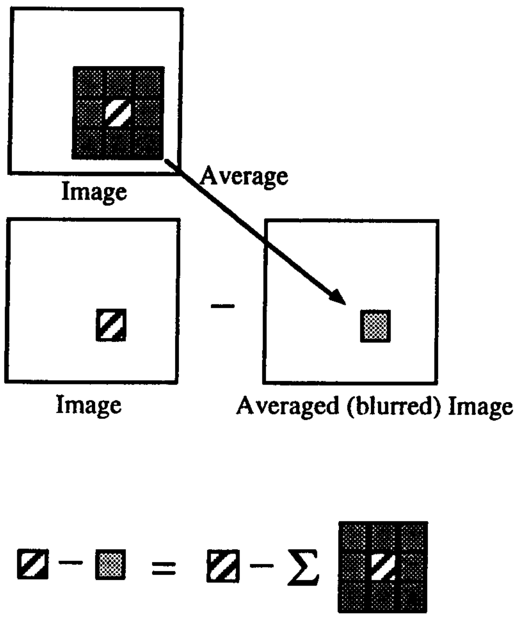
3 Blurring, continued



3- 58 VisionTutor Lecture Guide -- Version 1.0 (CSU Version, Spring 1994) Image Enhancement

This figure is similar to the previous one, except the difference is now taken over two images formed by two and three iterations of Gaussian averaging, respectively. Examination of the resulting image shows that many of the edges corresponding to fine detail in the original image are now gone (as one would expect, given the averaging) and what remains are the 'stronger' or more obvious edges in the original image.

3 Why it Works



≈ Laplacian

Subtracting a blurred image from the original is equivalent to computing the difference between a pixel in the original image and the average of its surrounding pixels (including the central pixel). This in turn is equivalent to computing the Laplacian of the original image up to a scaling factor on the central pixel. Subtracting a less blurred version of an image from a more highly blurred version is equivalent to computing a Laplacian which has more 'support' - that is, the central positive area is the result of averaging over some set of pixels and the negative surround is also the result of averaging over some larger set of pixels. As we shall see in subsequent sections, this technique can be used as a multi-resolution edge detector. The size of the blurring operator, if chosen judiciously, determines the 'scale' of the resulting edges.

3 Filtering and Fourier Methods

- ✓ • Smoothing
- ✓ • Edge Sharpening
- → • Fourier Filtering
- • Pyramid Filtering

Thus far we have used the term frequency somewhat intuitively, relying on a loose correlation with the rate at which something changes in order to classify image events into high frequency and low frequency events. In this section we make a brief foray into the world of spatial frequencies and Fourier transforms, still on a very intuitive level. More detail can be found in the section on Fourier techniques.

3 Fourier Transform: General Idea

- Image = spatially varying continuous function
- Decompose into a set of orthogonal functions, called basis functions.
- Fourier basis functions: sinusoids
- Changes in image intensity ➡ spatial frequency
- At any image point (x,y),

$$I(x,y) = \Sigma \text{ weighted basis functions}$$

$$FT[I(x,y)] = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x,y)\, e^{[-j2\pi(ux+vy)]}\, dx\, dy$$

The basic concepts underlying Fourier transforms are quite simple, although the mathematics can be tedious. Consider the case of continuous images first. One way to view a continuous image is as a spatially varying function I, which can be evaluated at a point (x,y) to obtain its value (in the discrete case we've called this value the gray level). It turns out that any function which satisfies a set of reasonable constraints can be written as a weighted sum (possibly infinite) of orthogonal basis functions.

We already have an example of this from physics: we know that light of any color can be obtained by a weighted sum of three colors: red, green, and blue. The red, green, and blue lights are the basis (lights) vectors for color space. Just as there are many different choices for the color of the lights comprising the basis lights, there are many possible choices for the basis functions. When sines and cosines are chosen, the decomposition of a function into a weighted sum of these functions is called the Fourier transform of the function. The exponential function inside the integral represents spatially periodic functions, and consequently one can relate the period to a frequency.

3 Fourier Transform: General Idea continued

- Exponential part is really a sum of sine and cosines:

$$e^{-j\pi(ux+vy)} = \cos 2\pi(ux+vy) + j \sin 2\pi(ux+vy)$$

- The inverse Fourier Transform exists:

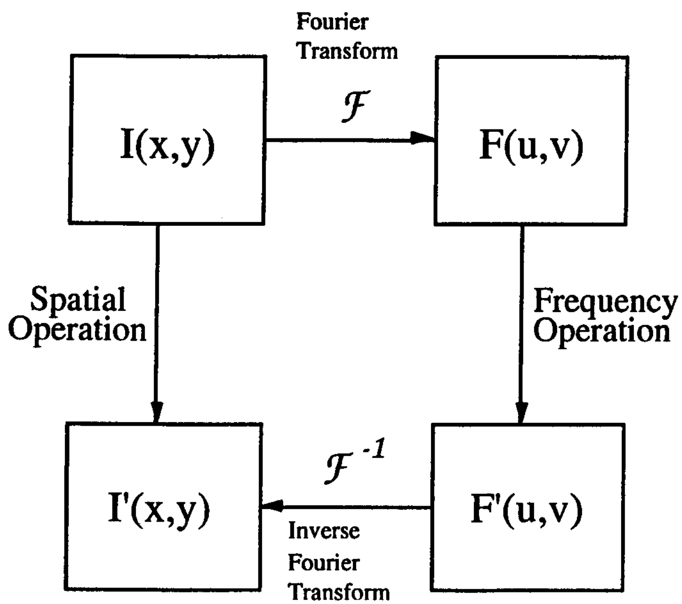
$$I(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{F(u,v)}_{\text{Weights}} [\cos 2\pi(ux+vy) + j \underbrace{\sin 2\pi(ux+vy)]}_{\text{Basis Functions}}\, du\, dv$$

- Fast algorithms for computing the discrete Fourier transform.

To see the relationship to frequency more clearly, note that the exponential part is just the sum of a sine component and a cosine component. By changing u and v, sines and cosines of varying periods can be obtained. Thus, the Fourier transform of a function I is simply its representation as a weighted sum of complex sinusoids of varying frequency. Given the Fourier transform of a function, the original function can be obtained by applying the inverse Fourier transform. In the formulation for the inverse transform, it is evident that H(u,v) is simply the weighting function which describes how much (i.e. the relative fraction) of a complex sinusoid is required to reconstruct the original signal.

Fast algorithms exist for computing the two-dimensional Fourier transform of discrete images. We now have a mechanism for discussing frequency with a little more rigor. If we examine the Fourier transform of a signal, high frequencies are exactly those which are represented with the high frequency sines and cosines.

3 Processing in the Frequency Domain



The Fourier transform provides an alternate way of performing filtering operations on an image. First compute its Fourier transform, modify its frequency representation by adding, removing, attenuating, or shaping certain frequencies or frequency ranges, and then compute the inverse Fourier transform. Removing high frequencies has the effect of blurring the image, for example.

3 Introduction to Fourier Methods

The Fourier Transform converts spatial image data I(x,y) into a frequency representation H(u,v).

- Both representations contain the same underlying information.
- Each repreentation has a set of strengths and weaknesses.

Spatial Domain	Frequency Domain
Intuitive representation of image data.	Non-intuitive representation of image data.
Filtering with large masks may result in long processing times.	Filtering with large masks can be performed very quickly.
Masks are applied directly to the spatial data.	Image and mask must first be converted to frequency domain, modified, then reconverted.

The Fourier transform H(u,v) is yet another way of representing the information contained in an image. H(u,v) is a complex function and is therefore composed of two parts: a real part and an imaginary part. Since the original image can be reconstructed exactly from the Fourier description, it should be evident that it contains exactly the same information as the original image, albeit in a different form. This form is not particularly intuitive and from just looking at the Fourier transform H(u,v) it is impossible to imagine what the original image looked liked.

There are several advantages gained from the Fourier representation. For example, filtering an image with a large mask can be performed very quickly in the Fourier domain (it is just a pointwise multiplication of coefficients). However, this must be weighed against the cost of the transformation of both the image and the mask to/from the frequency domain. We shall encounter other uses for the Fourier transform in a later section of the course.

3 Fourier Filtering Methods

Basic procedure for Fourier Filtering

1) Convert the spatial image to a frequency image via the Fourier Transform.

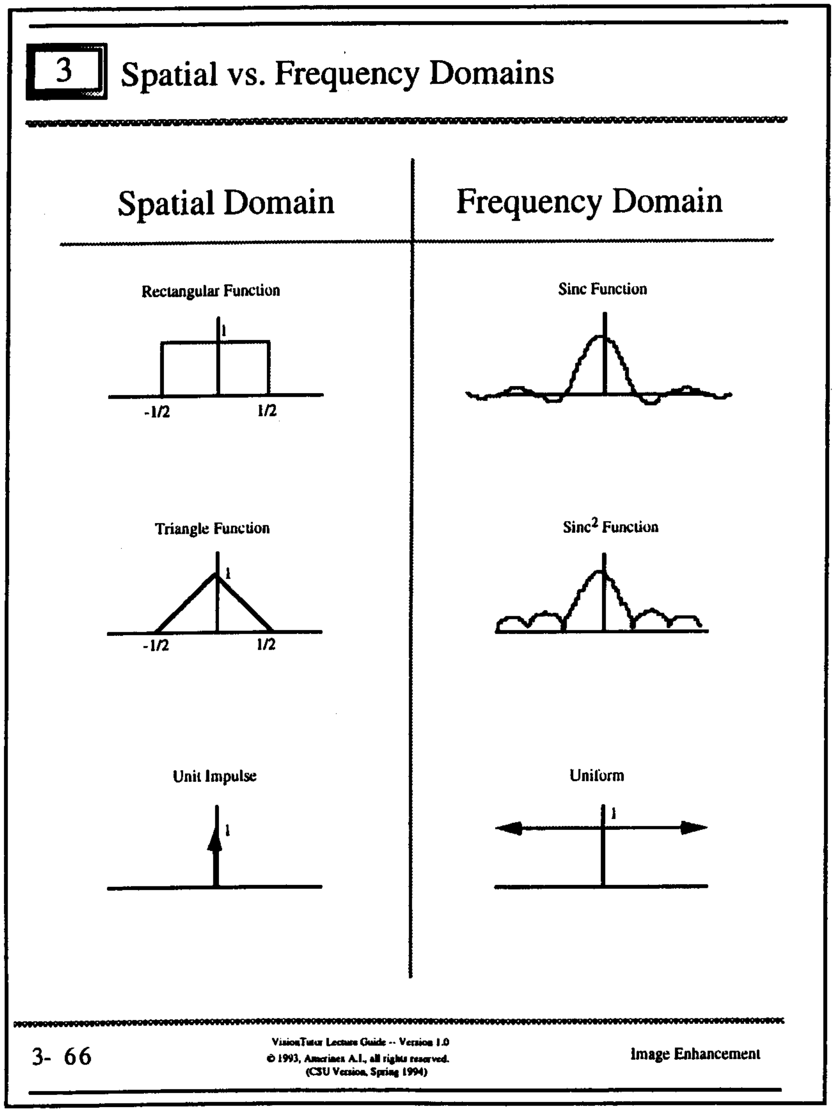
2) Modify the frequencies as desired using a filter.

3) Convert the frequency image back to a spatial image via the inverse Fourier Transform.

The basic technique for filtering in the Fourier domain is quite simple. First, the image is transformed into the frequency domain using the forward Fourier transform. There, the coefficients of the real and imaginary parts are modified in some way. For example, by setting all the coefficients for which

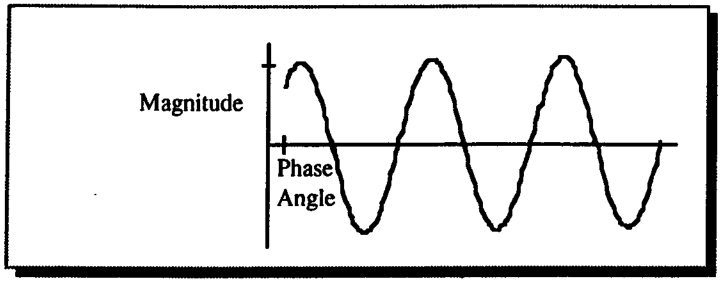
$$u^2+v^2>k$$

to zero (that is, all the coefficients outside a circle of radius k), all of the frequency components lying outside the circle (that is, the high frequency components) are removed from the image. As you might expect, this has the effect of blurring the image (when it is reconstructed); the degree of blurring depends on the value of k. After the coefficients are modified, the original image is reconstructed by applying the inverse Fourier transform.

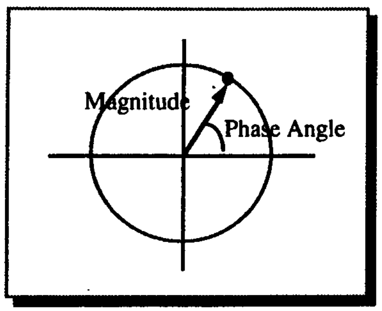


The table shows the Fourier transform of some common one-dimensional signals. Notice that the impulse function (the last function on the left) has a Fourier transform which has *all* sinusoids equally contributing.

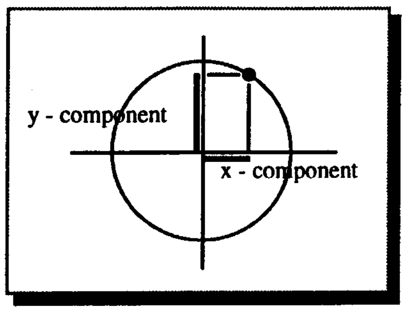
3 Frequency Data Representations



Polar Coordinates



Cartesian Coordinates



There are several different ways of representing a sinusoid. The first is the common representation as a sine (or cosine) wave showing the amplitude and the phase. Note that the phase corresponds to the angle at which the sinusoid is considered to 'start'. The second is as a vector in polar coordinate space; in this representation, the magnitude of the sinusoid is given by the length of the vector and the phase angle is the angle between the vector and the horizontal axis. It is this form of representation which gives rise to the complex exponential form of the Fourier transform. Regardless of the representational form chosen, the Fourier transform of a two-dimensional signal also has two dimensions (u,v).

3 Frequency Image

- The Fourier coefficients are generally a complex number and are expressed as two components:

    - the real component
    - the imaginary component

- H(u,v) = Real(u,v) + j Imaginary(u,v)

- Typically these are converted into an equivalent representation consisting of a vector magnitude and phase angle:

    - the magnitude

$$M(u,v) = \sqrt{Real(u,v)^2 + Imaginary(u,v)^2}$$

    - the phase angle

$$\theta(u,v) = \tan^{-1}\frac{Imaginary(u,v)}{Real(u,v)}$$

The value of the transform H at point (u,v) is a complex number having a real part and an imaginary part:

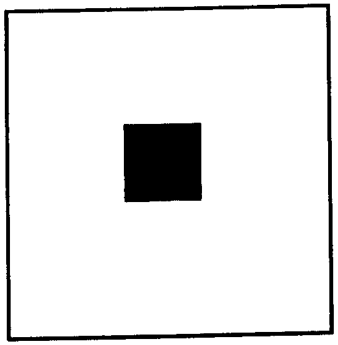
$$H(u,v) = R(u,v) + j\,I(u,v)$$

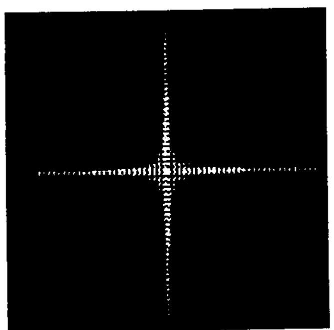
In order to visualize the Fourier tranform, it is convenient to convert this representation into an equivalent representation as a vector mangitude and a phase angle. In this representation, the magnitude of the vector corresponds to the 'amount' of the frequency determined by u and v in the original image.

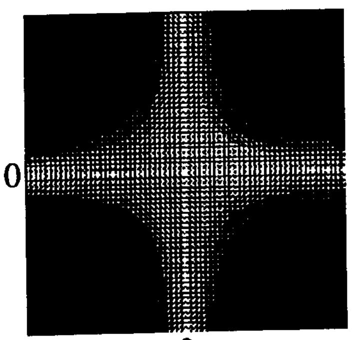
In this representation, the Fourier transform of an image can be visualized in (u,v) space by plotting the magnitude of the transform in (u,v) space. A second 'image' can be formed by plotting θ in (u,v) space. In the 'magnitude' image, each pixel uniquely describes a two-dimensional sinusoid in the original image coordinate space (something like an egg-carton) oriented at angle θ to the image coordinate axes.

3 The Fourier Transform

Image







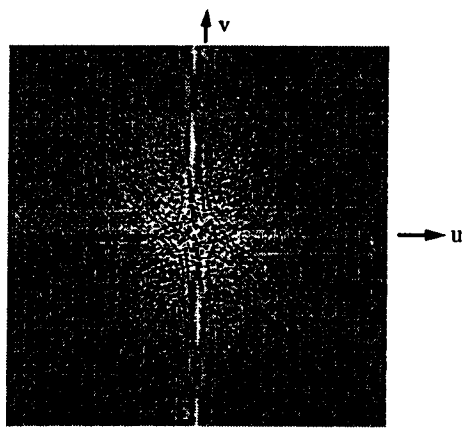
0 Magnitude of the FT

0 Magnitude of the FT (scaled differently)

In this slide, the magnitude of the Fourier transform of a simple image consisting of a black square on a white background is shown in (u,v) space under two different scalings of the magnitude information. The left image shows the magnitude scaled so as to make the differences in the magnitudes along the axes apparent. The second image shows the same magnitude image scaled to make the smaller coefficients off the axes visible. In both of these images, u=v=0 is in the center of the image.

3 Filtering in the Frequency Domain

Original Image and its Fourier Transform



We have already noted that the Fourier transform is a complex function of two variables, that is:

$$H(u,v) = R(u,v) + j\, I(u,v)$$

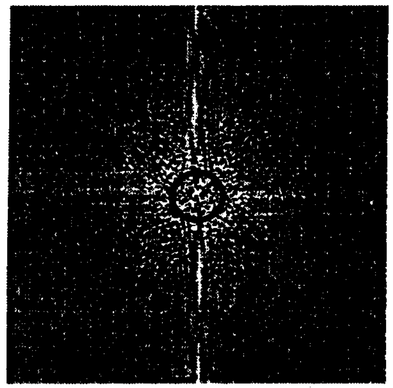
where R and I are the real and imaginary parts. When viewing a two-dimensional Fourier transform, it is convient to plot the magnitude of H(u,v) against (u,v). Recall that the magnitude of a complex vector is given by:

$$|H(u,v)| = \{R(u,v)^2 + I(u,v)^2\}^{1/2}$$

When filtering in the frequency domain, the filter is used to 'shape' or alter the magnitude information by removing or attentuating certain frequencies or frequency ranges. When the inverse Fourier transform is computed, it is not computed on the original transform data, but rather on the transformed data.

3 Low-Pass Filtering

Only frequencies INSIDE of the circle are used for the inverse Fourier Transform.



Inverse FFT

3- 71 VisionTutor Lecture Guide -- Version 1.0

(CSU Version, Spring 1994) Image Enhancement

Low-pass filters allow the low frequencies to pass 'through' the filter while all others (i.e. the high frequencies) are attenuated or completely removed. In the plot, recall that the lower frequencies are represented toward the center of the image. Typically, filters will be symmetric about the center (0,0) point in (u,v) space. The filter used in this example is circular in shape. Inside the circle, the pixel values of the filter are set to 1 in the magnitude image and 0 in the phase image (k+0i is effectively the representation of a real scalar in complex notation - here, k=1). Outside the circle, the pixels in the magnitude image are set to zero and the pixels in the phase image are set to 0. When this filter is multiplied (using complex multiplication) pixel-by-pixel with the real and imaginary component of the frequency image, the transformed frequency spectrum will contain only the frequency information interior to the circle - only these frequencies will 'pass' through the filter.

When a complex multiply is performed, remember:

$$a+bi \quad * \quad c+di \quad = (ac - bd) + (ad + cb)\, i$$

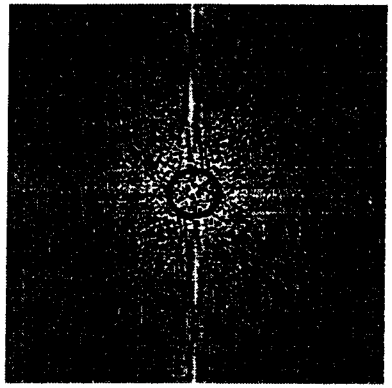
In our case, *c* represents the magnitude component of the circular filter, and *d* represents the phase component (which contains all zeroes in this example). Therefore the new complex image result will be:

$$(ac) + (bc)i$$

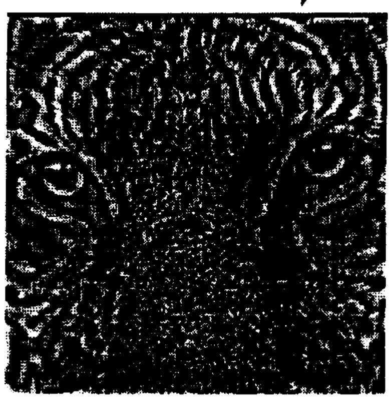
Only low frequencies will be passed, and all others will be zeroed. The resulting inverse transformed image is a blurred version of the original.

3 High-Pass Filtering

Only frequencies OUTSIDE of the circle are used for the inverse Fourier Transform.

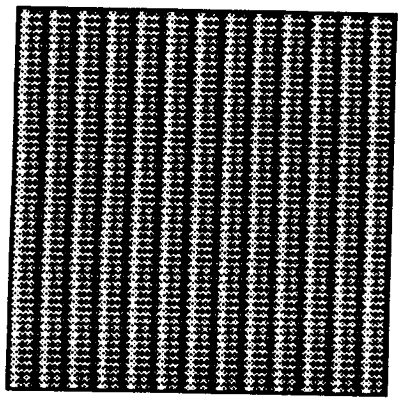


Inverse FFT

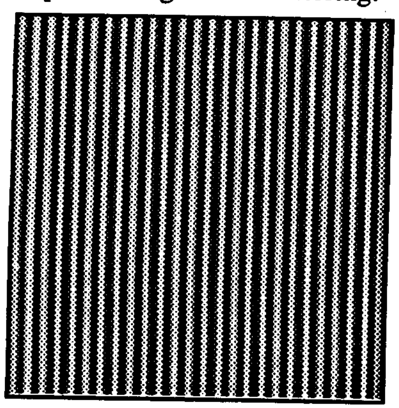


The procedure for high-pass filtering is identical to that of low-pass filtering except that a different filter is used. In this case, the magnitude image of the filter contains zeroes within the circle and ones otherwise; again, the corresponding phase component is zero everywhere. Only pixel values that represent high frequencies will 'pass' during the complex multiply and only these frequencies will be used to reconstruct the image when taking the inverse transform. The result is an image will looks very much like an 'edge' image.

3 Band-Pass Filtering



Spatial image before filtering.



Spatial image after filtering.

Like high-pass and low-pass filtering, band-pass filtering requires that a filter be applied to the transformed frequency information. Band-pass filters usually contain two concentric circles, where all pixel values within the smaller circle and outside of the larger circle are set to zero. All pixel values within the 'band' between the two circles are set to one (the phase plane is again all zero). Band-pass filters will select a range of frequencies to be used when taking the inverse transform (these are the frequencies that pass through the filter). If the band were inverted (ones outside and zero inside), then the band-pass filter would selectively remove frequencies. This can be especially helpful in the case where imaging equipment inserts noise that is composed of a small set of frequencies into the image during a particular process, such as digitizing, transmission, etc. Band-pass filters can help remove this type of noise.

The top image is composed of three sine waves of different frequencies, two in the horizontal direction and one in the vertical direction. The bandpass frequencies were chosen to select only one of the horizontal sine waves. Reconstructing the image using the inverse Fourier transform results in the image at the bottom of the page. It should be visually clear that this image is composed of only one sine wave in the horizontal direction.

3 Frequency Filtering using a Mask

<u>Frequency domain filtering with a mask</u>

1) Convert the spatial image to a frequency image via the Fourier Transform.

2) Convert the mask into an image. The image must be of the same dimensions at the input image.

3) Convert the mask image into a frequency image via the Fourier Transform.

4) Multiply the two frequency images together using the complex multiply operation.

5) Convert the resulting image from a frequency image back to a spatial image using the inverse Fourier Transform.

The resulting image is identical to the image that would result if the input mask were convolved over the input image.

It is also possible to compute the effect of applying a specific mask M to an image I in the frequency domain. In order to do this, the image is converted into the frequency domain using the Fourier transform. Next, the mask is embedded in an image whose size is the same as the original image. Where this mask is placed in the new image makes no difference. The mask image is also converted to its frequency representation. The resulting complex frequency images are multiplied in the frequency domain using complex multiplication, yielding a new complex image. Using the inverse Fourier transform, this frequency image is converted into a spatial image. The resulting image is identical to the one which would have been obtained had the mask M been applied to the spatial image I using convolution.

Whether or not it is more efficient to perform the mask application in the frequency domain depends upon the size (e.g., complexity) of the spatial mask. For large masks, it is very often the case that the overhead of the Fourier transforms is justifiable.