2.5 A CCD camera chip of dimensions 7 × 7 mm, and having 1024 × 1024 elements, is focused on a square, flat area, located 0.5 m away. How many line pairs per mm will this camera be able to resolve? The camera is equipped with a 35-mm lens. (*Hint:* Model the imaging process as in Fig. 2.3, with the focal length of the camera lens substituting for the focal length of the eye.)

Answer:

Let,

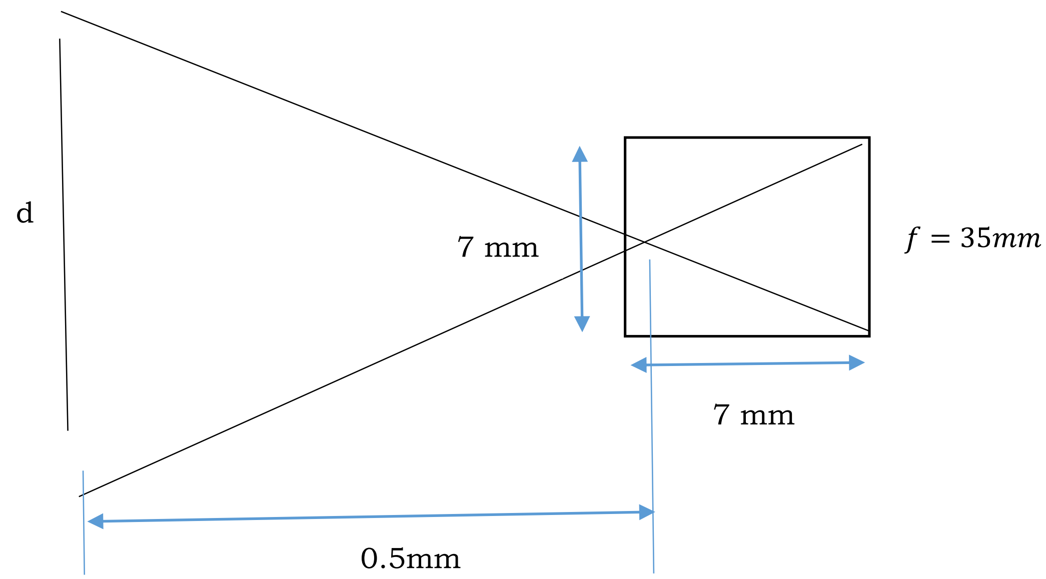
d : be the object size (target size);

h : Height of a CCD camera = $7mm$

f : focal length of the camera = $7mm$

s : distance from the camera to the object = $500\ mm$

By using eye Geometry model,



We have,

$$\frac{h}{d} = \frac{f}{s}$$

$$\text{or,} \quad \frac{7mm}{d} = \frac{35mm}{500mm}$$

$$\text{So, target size, d } = \frac{7*500}{500mm} = 100\text{mm}$$

Given, Number of elements per line= 1024 mm elements

Resolution per one line $= \frac{1024}{100} \cong 10\ elements/mm$

Therefore, for line pair we get $= \frac{1}{2} \times 10 = 5\ line\ pairs/mm$

2.7 Suppose that a flat area with center at (x_0, y_0) is illuminated by a light source with intensity distribution

$$i(x, y) = Ke^{-[(x-x_0)^2+(y-y_0)^2]}.$$

Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let $K = 255$. If the resulting image is digitized with k bits of intensity resolution, and the eye can detect an abrupt change of eight shades of intensity between adjacent pixels, what value of k will cause visible false contouring?

Answer:

Image is represented by two dimension function $f(x,y)$ given by:

$$f(x,y) = i(x,y)\ \ r(x,y)$$

Where, (x,y) : a light source with intensity distribution

$r(x,y)$: reflected light intensity (Reflectance)

According to the question, it is given by,

Given number of bits of quantization $= K$

Given that, Type equation here.

A flat area with center at $(x_0 y_0)$ is illuminated by a light source with intensity distribution, $I(x,y) = Ke^{-[(x-x_0^2)+(y-y_0^2)]}$

$k = 255$

And Reflectance, $r(x,y) = 1$

ΔI will be intensity detectable by human eye $= \frac{256}{2^k}$

Since, detectable shades of intensity = 8

or, $\Delta\text{I} = 8 = \frac{256}{2^k}$

=> $2^k = 32$

=> $k = 5$

∴ 32 gray levels (or lens) will lead to false contouring.

3.5 Explain why the discrete histogram equalization technique does not, in general, yield a flat histogram.

Answer:

What histogram equalization does is a transformation from original image intensity distribution to a uniform distribution. However, the obtained distribution is not actually flat.

For an image of M levels of discrete intensity and K number of pixels, to have a flat uniform distribution requires all $\frac{\mathrm{k}}{\mathrm{M}}$ pixels to have the same intensity which cannot be achieved by histogram equalization.

3.7 In some applications it is useful to model the histogram of input images as Gaussian probability density functions of the form

$$p_r(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r-m)^2}{2\sigma^2}}$$

where m and σ are the mean and standard deviation of the Gaussian PDF. The approach is to let m and σ be measures of average gray level and contrast of a given image. What is the transformation function you would use for histogram equalization?

Answer:

The general histogram equalization transformation function is

$$S = T(r) = \int_0^r P_r\,(W)dW \tag{1}$$

In order to use Gaussian distribution we are assuming the following equation

$$P = T(t) = \frac{1}{\sqrt{2\pi}\sigma} \int_0^t e^{-\frac{(w-m)^2}{2\sigma^2}} \tag{2}$$

In order to use equation (2) as a transformation function we must make a couple of assumption:

- First, equation (1) does not take negative values while Gaussian equation (2) can take both negative and positive numbers. We can mitigate this problem by making standard deviation very small to make area under the curve of $P_r(r)$ for negative numbers very negligible.
- The other problem is that equation (2) is the cumulative distribution function of the Gaussian density, which is either integrated numerically, or its values are looked up in a table. To solve this problem, we can scale down by dividing with a large number so that numbers at positive and negative tails become very negligible (this scaling reduces the standard deviation).
- Another good transformation that can be done is using a histogram by sampling a continuous Gaussian function so that transformation will be the sum. This will take care of the above two limitations.

3.14 Image subtraction is used often in industrial applications for detecting missing components in product assembly. The approach is to store a "golden" image that corresponds to a correct assembly; this image is then subtracted from incoming images of the same product. Ideally, the differences would be zero if the new products are assembled correctly. Difference images for products with missing components would be nonzero in the area where they differ from the golden image. What conditions do you think have to be met in practice for this method to work?

Answer:

Let $g(x, y)$ denote the golden image, and $f(x, y)$denote any input image acquired during routine operation of the system. Change detection via subtraction is based on computing the simple difference $d(x, y) = g(x, y) - f(x, y)$. The resulting image, $d(x, y)$ can be used in two fundamental ways for change detection. One way is use pixel-by-pixel analysis. In this case we say that $f(x, y)$ is "close enough" to the golden image if all the pixels in $d(x, y)$ fall within a specified threshold band $[T_{min}, T_{max}]$ where, T_{min} is negative and T_{max} is positive. Usually, the same value of threshold is used for both negative and positive differences, so that we have a band $[-T, T]$ in which all pixels of $d(x, y)$ must fall in order for $f(x, y)$ to be declared acceptable. The second major approach is simply to sum all the pixels in $d(x, y)$ and compare the sum against a threshold Q.

As mentioned in the problem, Image subtraction is done by simple subtraction of a given image from our preset target image and performance depends on how small this difference is. Ideally, we get the zero difference if two images are similar but in real life, we set the threshold for the desired difference range. To achieve a good image subtraction, we can control the following conditions:

- First, we need to make sure that the images are very well aligned. Even if the image may be of the same person or subject, subtraction may not produce good results if they were taken for example by different cameras or in different conditions. So, a proper image registration ahead of computing image subtraction is very important.

- Another important factor that may affect the accuracy of image subtraction is environment conditions such as illumination. So, illumination should be controlled and kept constant as much as possible. Taking images under varying light intensities both visible and invisible may affect the actual image difference.

- The other way to achieve a desirable image subtraction result is to control signal to noise ratio in of an image resulting from the subtraction operation. We need to reduce the noise in image difference low to avoid wrong comparison between our fixed image and input image.

3.20 **(a)** Develop a procedure for computing the median of an $n \times n$ neighborhood.

(b) Propose a technique for updating the median as the center of the neighborhood is moved from pixel to pixel.

Answer:

For part (a):

For an $n \times n$ neighborhood, we have n^2 numbers, so to find the median we first sort the elements in increasing order and the $\frac{n^2+1}{2}th$ largest number will be the median. For example in a 3×3 neighborhood, the 5th largest element will the median, for a 5×5 neighborhood, the 13th largest number will be the median.

For part (b):

When the center point is moved from pixel to pixel, like the above procedure we sort the element and form a list of sorted values (get the first median). Then we will remove the values on the rear edge of the neighborhood from our sorted list. After removing these values, we then add to the list, the values from the leading edge of the neighborhood. After finding the median we take numbers before the median and place them on the foremost edge of the list.

4.4 A Gaussian lowpass filter in the frequency domain has the transfer function

$$H(u, v) = Ae^{-(u^2+v^2)/2\sigma^2}.$$

Show that the corresponding filter in the *spatial* domain has the form

$$h(x, y) = A2\pi\sigma^2 e^{-2\pi^2\sigma^2(x^2+y^2)}.$$

(*Hint:* Treat the variables as continuous to simplify manipulation.)

Answer:

Given that, $\mathrm{H(u,v)} = \mathrm{Ae}^{\frac{-(u^2+v^2)}{2\sigma^2}}$ (1)

$\mathrm{h(x,y)} = \mathrm{A2\pi\sigma^2 e^{-2\pi^2\sigma^2(x^2+y^2)}}$ (2)

In order to find the$h(x,y)$, we should find the Inverse Fourier Transform (IFT) of $H(u,v)$.

From the equation (1):

$$H(u,v) = Ae^{\frac{-(u^2+v^2)}{2\sigma^2}}$$

Let, $\mathrm{t} = \mathrm{u}^2 + \mathrm{v}^2$, so, $\mathrm{H(u,v)} = \mathrm{Ae}^{\frac{-t^2}{2\sigma^2}}$

From IFT formula, $h(z) = \int_{-\infty}^{+\infty} H(u,v)e^{2\pi jtz}\,dt$

$$=> h(z) = \int_{-\infty}^{+\infty} e^{\frac{-t^2}{2\sigma^2}}\ e^{2\pi jtz}\,dt$$

$$= \int_{-\infty}^{+\infty} e^{\frac{-1}{2\sigma^2}}\ [t^2 - j4\pi\sigma^2 tz]dt \qquad (3)$$

From identity, $\mathrm{e}^{-\frac{(2\pi)^2 z^2\sigma^2}{2}} \times \mathrm{e}^{\frac{(2\pi)^2 z^2\sigma^2}{2}} = 1$ (4)

Then, we replace equation (4) into (3), we get,

$$\mathrm{h(z)} = \mathrm{e}^{-\frac{(2\pi)^2 z^2\sigma^2}{2}} \int_{-\infty}^{+\infty} \mathrm{e}^{\frac{-1}{2\sigma^2}}\ [t^2 - j4\pi\sigma^2 tz - (2\pi)^2\sigma^4 z^2]dt$$

$$= \mathrm{e}^{-\frac{(2\pi)^2 z^2\sigma^2}{2}} \int_{-\infty}^{+\infty} \mathrm{e}^{\frac{-1}{2\sigma^2}}\ [t^2 - j4\pi\sigma^2 tz - (2\pi)^2\sigma^4 z^2]dt$$

$$= \mathrm{e}^{-\frac{(2\pi)^2 z^2\sigma^2}{2}} \int_{-\infty}^{+\infty} \mathrm{e}^{\frac{-1}{2\sigma^2}}\ [t - j2\pi\sigma^2 z]^2 dt$$

By change of variables, $r = \mathrm{t} - \mathrm{j}2\pi\sigma^2 z$, then $dr = dt$, we get,

$$h(z) = \mathrm{e}^{-\frac{(2\pi)^2 z^2\sigma^2}{2}} \int_{-\infty}^{+\infty} \mathrm{e}^{\frac{-r^2}{2\sigma^2}}\ \mathrm{dr} \qquad (5)$$

Finally, we multiply and divide the right side of this equation by $\sqrt{2\pi}\sigma$ and obtain,

$$h(z) = \sqrt{2\pi}\sigma\, \mathrm{e}^{-\frac{(2\pi)^2 z^2 \sigma^2}{2}} \left[\sqrt{2\pi}\sigma \int_{-\infty}^{+\infty} \mathrm{e}^{\frac{-r^2}{2\sigma^2}}\, \mathrm{dr}\right]$$

The above expression inside the brackets is recognized as the Gaussian probability density function whose value from $-\infty$ to ∞ is 1. Therefore,

$$h(z) = \sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 z^2 \sigma^2}$$

We can apply it for two variables,

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Ae^{-\frac{u^2+v^2}{2\sigma^2}} e^{j2\pi(ux+vy)} dudv$$

$$= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} Ae^{-\frac{u^2}{2\sigma^2}+j2\pi ux} du\right] Ae^{-\frac{v^2}{2\sigma^2}+j2\pi vy} dv$$

The integral inside the brackets is recognized from the previous discussion to be equal to $A\sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 x^2 \sigma^2}$. Then, the preceding integral becomes

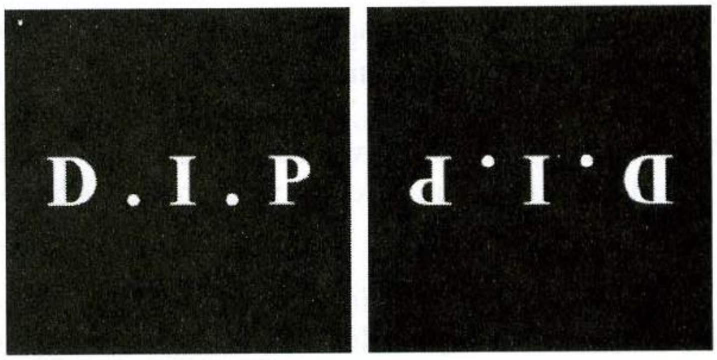
$$h(x,y) = A\sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 x^2 \sigma^2} \int_{-\infty}^{\infty} Ae^{-\frac{v^2}{2\sigma^2}+j2\pi vy} dv$$

We now recognize the remaining integral to be equal to $\sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 y^2 \sigma^2}$.

So, we can say that, $h(x,y) = (A\sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 x^2 \sigma^2})\ (\sqrt{2\pi}\sigma \mathrm{e}^{-2\pi^2 y^2 \sigma^2})$

$$= A2\pi\sigma^2 e^{-2\pi^2\sigma^2(x^2+y^2)}$$

4.9 Consider the images shown. The image on the right was obtained by (a) multiplying the image on the left by $(-1)^{x+y}$; (b) computing the DFT; (c) taking the complex conjugate of the transform; (d) computing the inverse DFT; and (e) multiplying the real part of the result by $(-1)^{x+y}$. Explain (mathematically) why the image on the right appears as it does.



Answer:

Here, in the question part (c) Complex Conjugate was taken before inverse Fourier Transform.

So, the complex conjugate was change the value of j to $-j$. So, we find inverse Fourier Transform (FT) of this complex conjugate.

$$IFT[F^*(u,v)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)e^{-j2\pi\left(\frac{u(-x)}{M}+\frac{v(-y)}{N}\right)}$$

$$= f(-x,-y)$$

$f(-x,-y)$ is similar to reflection of original image about the origin.

4.15 The basic approach used to approximate a discrete derivative (Section 3.7) involves taking differences of the form $f(x+1, y) - f(x, y)$.

(a) Obtain the filter transfer function, $H(u, v)$, for performing the equivalent process in the frequency domain.

(b) Show that $H(u, v)$ is a highpass filter.

Answer:

For part (a):

Given, a discrete derivative involve taking derivatives in the form $f(x+1,y) - f(x,y)$.

Let's find the filter function in frequency domain and spatial domain.

Filter function in spatial domain, $g(x,y) = f(x+1,y) - f(x,y) + f(x,y+1) - f(x,y)$

The translation property, $f(x,y)e^{j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}$ <=> $F(u-u_0, v-v_0)$

In frequency domain, we get,

$$G(u,v) = F(u,v)e^{\frac{j2\pi u}{M}} - F(u,v) + F(u,v)e^{\frac{j2\pi v}{N}} - F(u,v)$$

[Shifting Spatial Domain give exponential in Frequency Domain]

$$= \left[e^{\frac{j2\pi u}{M}} - 1\right]F(u,v) + \left[e^{\frac{j2\pi v}{N}} - 1\right]F(u,v)$$

$$= H(u,v)\,F(u,v)$$

Where, $H(u,v)$ is the filter function:

$$H(u,v) = \left[\left(e^{\frac{j2\pi u}{M}} - 1\right) + \left(e^{\frac{j2\pi v}{N}} - 1\right)\right]$$

$$= 2j\left[\sin\left(\frac{\pi u}{M}\right)e^{\frac{j\pi u}{M}} + \sin\left(\frac{\pi v}{n}\right)e^{\frac{j\pi v}{N}}\right]$$

For part (b):

To see that this is a High pass filter, it helps to express the filter function in the form of our familiar centered functions:

$$H(u,v) = 2j\left[\sin\left(\pi\left[u-\frac{M}{2}\right]/M\right)e^{\frac{j\pi u}{M}} + \sin\left(\pi\left[u-\frac{N}{2}\right]/N\right)e^{\frac{j\pi v}{N}}\right]$$

The function is 0 at the center of the filter $u = M/2$. As u and v increase, the value of the filter decreases, reaching its limiting value of close to $-4j$ when $u = M-1$ and $v = M-1$. The negative limiting value is due to the order in which the derivatives are taken. If, instead we had taken differences of the form $f(x,y) - f(x+1,y)$ and $f(x,y) - f(x,y+1)$, the filter would have tended toward a positive limiting value. The important point here is that the *dc* term is eliminated and higher frequencies are passed, which is the characteristic of a high pass filter.

4.18 Can you think of a way to use the Fourier transform to compute (or partially compute) the magnitude of the gradient for use in image differentiation (see Section 3.7.3)? If your answer is yes, give a method to do it. If your answers is no, explain why.

Answer:

The answer is "NO". The magnitude of the gradient is given by the equation:

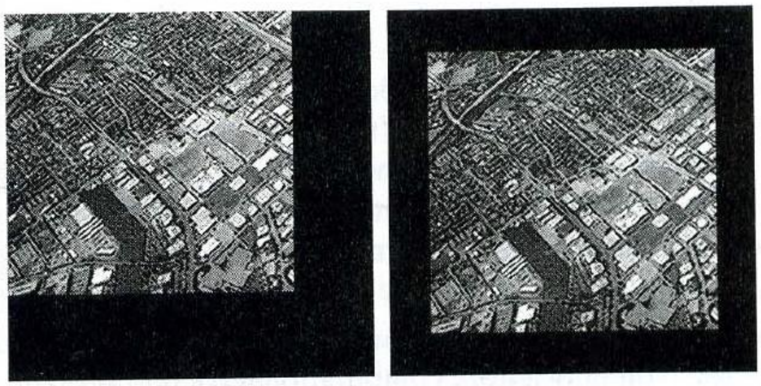
$$\nabla f = mag(\nabla f)$$

$$= \left[G_x^2 + G_y^2\right]^{\frac{1}{2}}$$

$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}}$$

From the above equation, we can see that magnitude of the gradient is not a linear function because of its square and square root terms. So, we cannot use Fourier transform to compute magnitude of gradient because FT is a linear function while the magnitude of the gradient is not.

4.21 The need for image padding when filtering in the frequency domain was discussed in some detail in Section 4.6.3. We showed in that section that images needed to be padded by appending zeros to the ends of rows and columns in the image (see the following image on the left). Do you think it would make a difference if we centered the image and surrounded it by a border of zeros instead (see image on the right), but without changing the total number of zeros used? Explain.



Original image courtesy of NASA.

Answer:

No, there will be no difference because padding an image with zero increases the size for DFT to be calculated but does not change the pixel values of an image. So, distributing zeroes around the image in any direction will not create any difference.

The purpose of padding is to create a "buffer" between the periods implied by the DFT. Consider duplicating the image on the left an infinite number of times to cover the x y plane. The outcome would be a checkerboard, with each square representing a different image (in the the black extensions). Imagine performing the same thing to the right-hand image. The outcomes would be the same. As a result, either type of padding achieves the appropriate gap between images.