ME 5405 Machine Vision

Image Enhancement

KB Lim
Mechanical Engineering Department
National University of Singapore
mpelimkb@nus.edu.sg

1

1. Introduction

The principal objective of image enhancement is to process an image so that the result is more suitable than the original image for a <u>specific</u> application.

Generally, the techniques employed are problem oriented. There are 2 broad categories of approaches:

i. **<u>Spatial Domain Methods</u>**
Spatial Domain refers to the image plane itself, and approaches are based on direct manipulation of pixels in an image. They are easy to implement but can be computational intensive. You get to see the results after processing.

ii. **<u>Frequency Domain Methods</u>**
Frequency Domain processing techniques are based on modifying the Fourier transform of an image. They are effective, but quite imaginative as you don't see the results immediately after processing.

Enhancement techniques based on various combinations of methods from these two categories are not unusual. We shall discuss Spatial Domain techniques in this course.

2

2. Background

Spatial Domain Methods operate directly on the pixels composing an image.
Image processing function

$$g(x, y) = T[\,f(x, y)] \qquad (1)$$

f(*x*,*y*) : input image
g(*x*,*y*) : processed image
T : an operator on *f*(*x*,*y*), defined over some neighbourhood of (*x*,*y*), also known as Transfer Function

T can also operate on a set of input images (e.g. pixel-by-pixel averaging for noise removal).

3

2. Background

Neighborhood about (*x*,*y*) is usually a square or rectangular area centered at (*x*,*y*) (see Fig 1).The centre of this sub-image is moved from pixel to pixel during the operation. The operator is applied at each location to yield *g*(*x*,*y*).

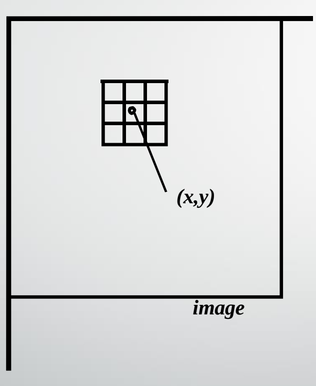


Fig. 1

The simplest form of *T* is when the neighborhood is 1×1.

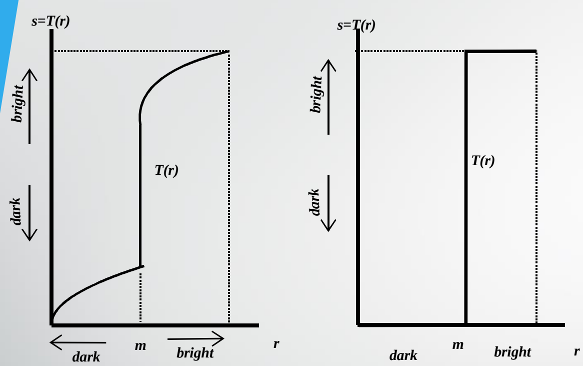
g(*x*,*y*) depends only on the value of *f*(*x*,*y*) and *T* is simply a gray-level transformation.

$$s = T(r) \qquad (2)$$

r and *s* are the gray-levels of *f*(*x*,*y*) and *g*(*x*,*y*) at (*x*,*y*), respectively

4

2. Background



(a) (b)

In Fig. 2(a), *T(r)* produces image of higher contrast:
r < *m* darkening; and
r > *m* brightening
to narrower bands contrast - stretching.

Fig. 2

In Fig. 2(b) *T(r)* produces a binary image.

Techniques in this category are referred to as **Point Processing.**

Larger neighborhoods can be used for various processing functions.

5

2. Background

The general approach is to let the values of $f(x,y)$ in a pre-defined neighbourhood of (x,y) to determine the value of $g(x,y)$.

One of the principal approaches is based on the use of so called masks (also referred to as template, windows, or filters).

A mask is basically a small (say 3 × 3) 2-D array. The values of the coefficients w_i , determine the nature of the process (see Fig. 3).

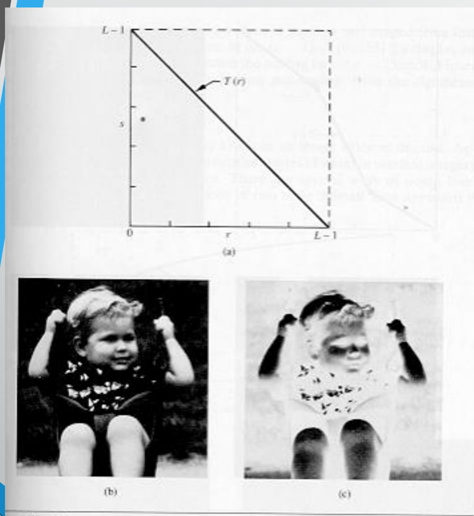
Image Enhancement based on this type of approach is known as Mask Processing or Filtering.

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Fig. 3
A 3 by 3 mask

6

3. Enhancement By Point Processing



<u>3.1 Some Simple Intensity Transformations</u>

<u>Image Negatives</u>

Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing negatives as normal slides.

$s = T(r)$, with $T(r)$ having the shape shown in Fig. 4(a).

Fig 4: Obtaining the negative of an image: (a) gray level transformation function; (b) an image; and (c) its negative. r and s denote the input and output gray levels, respectively.

7

3. Enhancement By Point Processing

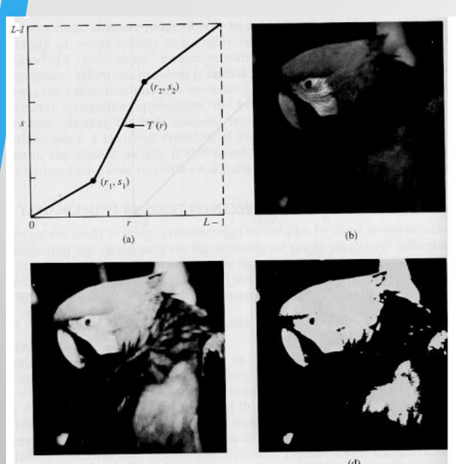


Fig. 5 Contrast Stretching: (a) form of transformation function; (b) a low contrast image; (c) result of contrast stretching; (d) result of thresholding.

<u>3.1 Some Simple Intensity Transformations</u>

<u>Contrast Stretching</u>

Low contrast images can result from: 1. poor illumination; (2) lack of dynamic range in the imaging sensors and (3) wrong aperture setting during image acquisition.

$T(r)$ for contrast stretching is shown in Fig. 5(a), (r_1, s_1) and (r_2, s_2) control the shape of $T(r)$.

if $r_1 = r_2, s_1 = s_2$ $\Rightarrow$ No change in image,
$r_1 = r_2, s_1 = 0, s_2 = L-1$ $\Rightarrow$ Thresholding function

In general, $r_1 \leq r_2$ and $s_1 \leq s_2$ are assumed so that $T(r)$ is single value and monotonically increasingly.

8

3. Enhancement By Point Processing

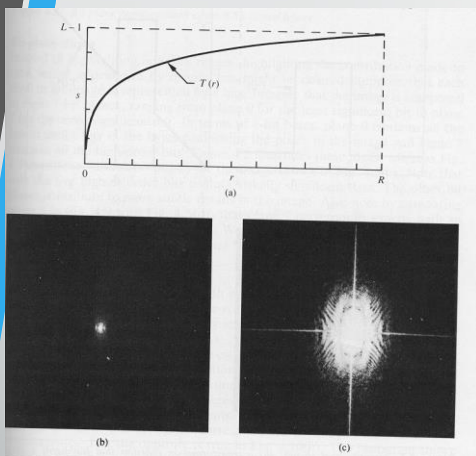


Fig. 6 Compression of dynamic range: (a) logarithm transformation function; (b) image with large dynamic range (pixel value ranging from 0 to 2.5 x 10^6); (c) result after transformation.

3.1 Some Simple Intensity Transformations

Compression of Dynamic Range

When the dynamic range of a processed image far exceeds the capability of the display device, only the brightest parts of the image are visible on the display screen.

To compress the dynamic range of pixel values, we use the following intensity transformation:

$$s = c\log(1+|r|) \qquad (3)$$

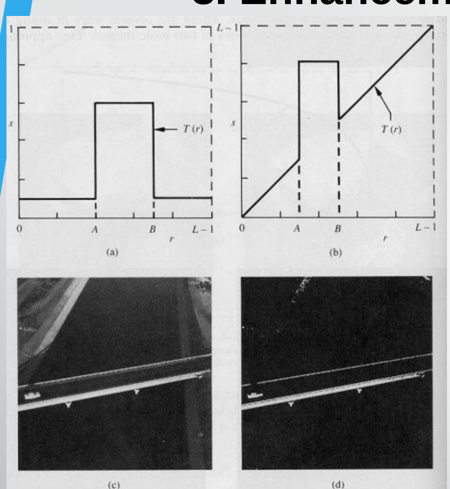
$(1+|r|)$ in equation (3) yield 0 to 6.4.

to scale to 0 to 255(8 bit system), we set c = 255/6.4, where c is a scaling constant.

9

In Fig. 6(b) pixel values are in the range 0 to 2.5 x 10^6,

3. Enhancement By Point Processing



3.1 Some Simple Intensity Transformations

Gray-Level Slicing

It is used to highlight a specific range of gray-levels in an image.

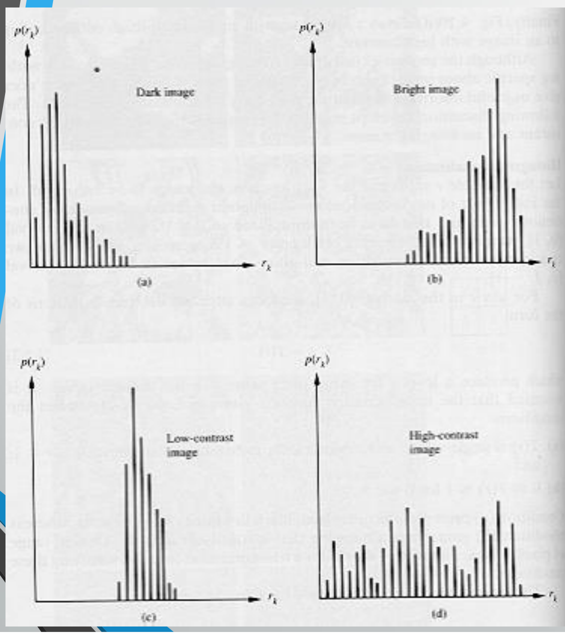
T(r) in Fig. 7(a) produces a binary image. *T(r)* in Fig. 7(b) brightens the gray levels ranging from A to B, but preserves the background and tonalities in the image.

Fig. 7(c) shows an image and 7(d) shows the resulting image after the image has been processed by the *T(r)* shown in 7(a).

Fig. 7 Intensity-level slicing: (a) a transformation that highlights a range *[A,B]* of intensity while diminishing all others to a constant, low level; (b) a transformation that highlights a range *[A,B]* of intensities but preserves all others; (c) an image; (d) result of using the transformation in (a).

10

3.2. Histogram Processing



Histogram for a digital image with gray level in the range [0,*L* - *1*], is a discrete function

$$p(r_k) = \frac{n_k}{n} \qquad (4)$$

where:

r_k is the k^{th} gray level

n_k is the number of pixel with gray-level r_k

n : total number of pixels in the image. k = 0, 1, 2, ..., L -*1*

Fig. 8 shows histograms of four types of images.

Fig. 8 Histograms corresponding to four different image types.

11

3.2 Histogram Equalization

Let r be the gray level in the image to be enhanced. We assume that r has been normalised to lie within the interval [0, 1].

$r = 0 \Rightarrow$ black
$r = 1 \Rightarrow$ bright / white

$\forall r \in [0,1]$, we focus on the transformation with the form

$$s = T(r) \qquad (5)$$

which produces a level s $\forall$ r in the original image.

Equation (5) satisfies the conditions:

(a) $T(r)$ is single-valued and monotonically increasing in the interval $0 \le r \le 1$; and

(b) $0 \le T(r) \le 1$ for $0 \le r \le 1$.

12

3.2 Histogram Equalization

Condition (a) preserves the order from black to white in the gray scale, and condition (b) guarantees a mapping that is consistent with the allowed range of pixel values.

Fig. (9) shows a *T(r)* satisfying these conditions

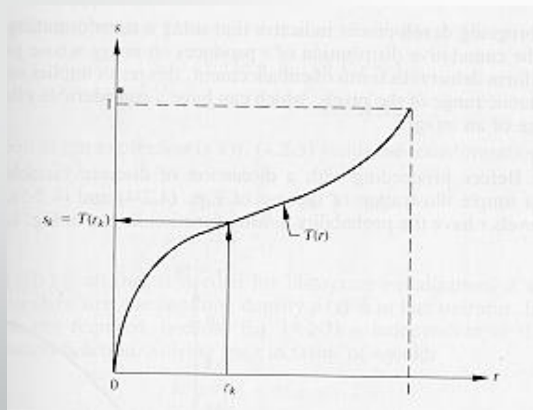


Fig. 9 A gray level transformation function

3.2 Histogram Equalization

The inverse transformation from *s* back to *r* is denoted as:

$$r = T^{-1}(s) \quad , 0 \le s \le 1 \qquad (6)$$

with the assumption that $T^{-1}(s)$ also satisfies conditions (a) & (b) above, with respect to *s.*

The gray levels in an image may be viewed as random variables In the intervals [0, 1]. If they are continuous variables, *r* and *s* can be characterised by their probability density functions, (*p.d.f*), p_r and p_s respectively.

From probability theory, if $p_r(r)$ and $T(r)$ are known, and $T^{-1}(s)$ satisfies condition (a).

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \qquad (7)$$

3.2 Histogram Equalization

Enhancement Technique

It is based on the modification of the appearance of an image by controlling the *p.d.f.* of its gray level via $T(r)$.

Consider:

$$s = T(r) = \int_0^r p_r(w)dw \quad 0 \le r \le 1 \tag{8}$$

$\int_0^r p_r(w)dw$ = cumulative distribution function (CDF) of r.

This CDF satisfies conditions (a) and (b), from (8)

$$\frac{ds}{dr} = p_r(r) \tag{9}$$

15

3.2 Histogram Equalization

Enhancement Technique

Substituting into (7)

$$p_s(s) = \left[p_r(r) \frac{1}{p_r(r)} \right]_{r=T^{-1}(s)} \tag{10}$$

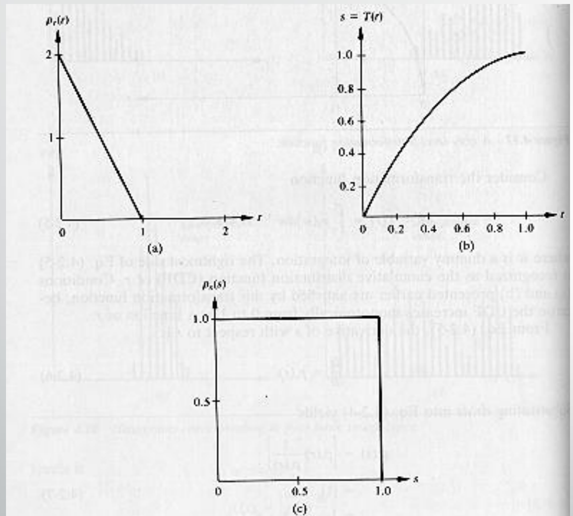
$p_s(s) = 1$ for $0 \le s \le 1$

which is a uniform density in $0 \le s \le 1$.

Note that the result is independent of $T^{-1}(s)$, which is important, because $T^{-1}(s)$ is difficult to obtain.

16

3.2 Histogram Equalization - Example



In this example.

$$p_r(r) = \begin{cases} -2r+2 & 0 \le r \le 1 \\ 0 & otherwise \end{cases}$$

Substituting into (8)

$$s = T(r) = \int_0^r (-2w+2)\,dw$$

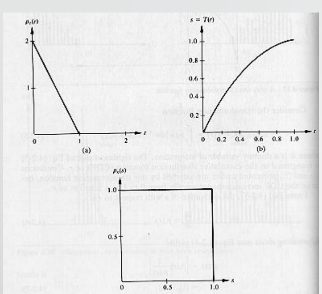
$$T(r) = s = -r^2 + 2r$$

which is shown in Fig. 10

Fig. 10 Illustration of the uniform density transformation method: (a) original probability density function; (b) transformation function; (c) resulting uniform density.

17

3.2 Histogram Equalization - Example



T(r) will perform the required image transformation.

Now, we will show that $p_s(s)$ is indeed uniform.

$$s = -r^2 + 2r$$

$$r = T^{-1}(s) = 1 \pm \sqrt{1-s}$$

Since $0 \le r \le 1$ $\quad \therefore \quad r = T^{-1}(s) = 1 - \sqrt{1-s}$

from (7)

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

$$= \left[(-2r+2) \frac{d}{ds}\left(1-\sqrt{1-s}\right) \right]_{r=1-\sqrt{1-s}}$$

$$= \left(2-\sqrt{1-s}\right) . \frac{1}{2} \frac{1}{\sqrt{1-s}}$$

$$= 1 \quad 0 \le s \le 1$$

Fig. 10(c) shows the result.

18

3.2 Histogram Equalization

Enhancement Technique – Discrete Formulation

For the above concept to be useful for digital image processing, we shall formulate it in discrete form.

For gray-levels that takes on discrete values, we deal with probabilities:

$$p_r(r_k) = \frac{n_k}{n}, \quad 0 \le r \le 1$$
$$k = 0,1,2,\ldots..L-1 \qquad (11)$$

The meaning of various terms are given in Eq. 4 above

A plot of $p_r(r)$ vs r is called a histogram, and the technique used for obtaining a uniform distribution is known as histogram equalization or linearization.

19

3.2 Histogram Equalization

Enhancement Technique – Discrete Formulation

Discrete form of equation (8) is

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \qquad (12)$$

$0 \le r \le 1$. and k = 0, 1, 2, ..., L - 1

and the inverse is $r_k = T^{-1}(s_k) \qquad 0 \le s_k \le 1$

where $T(r_k)$ and $T^{-1}(s_k)$ are assumed to satisfy conditions (a) and (b).

$T(r_k)$ may be computed directly from the image by using equation (12).

$T^{-1}(s_k)$ is not used here.

20

3.2 Histogram Equalization – Discrete Example

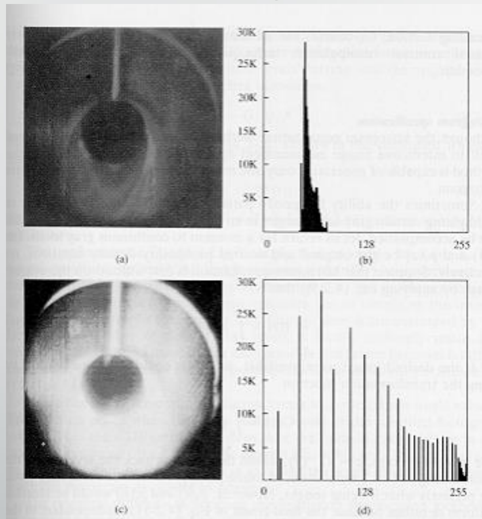


Fig. 11 (a) Original image and (b) its histogram; (c) image subjected to histogram equalization and (d) its histogram.

Example: Fig. 11 shows a 512 x 512, 8 bit image of a welding that is dark and with poor dynamic range (Fig. 11(a)). It has a narrow histogram as shown in Fig. 11(b).

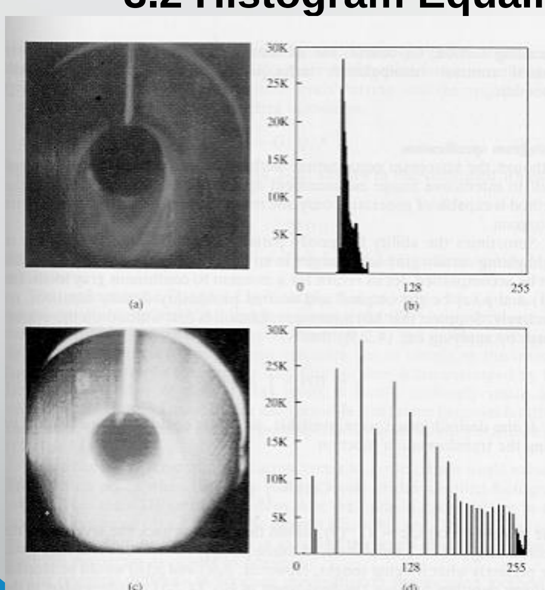
Fig. 11(c) shows the result after performing histogram equalisation enhancement on the original image. Fig. 11(d) shows the corresponding histogram.

Note: The resulting histogram is not flat, as it is not so in the discrete approximation of the continuous results. The histogram are spread out and always reach white.

This process increases the dynamic range of gray levels and, consequently, produces an increase in image contrast.

21

3.2 Histogram Equalization – Discrete Example



Example: Fig. 11 shows a 512 x 512, 8 bit image of a welding that is dark and with poor dynamic range (Fig. 11(a)). It has a narrow histogram as shown in Fig. 11(b).

We will illustrate how the histogram equalization is done in Discrete Formulation. This process is to stretch the contrast of the image.

Fig. 11 (a) Original image and (b) its histogram; (c) image subjected to histogram equalization and (d) its histogram.

22

3.2 Histogram Equalization – Discrete Example

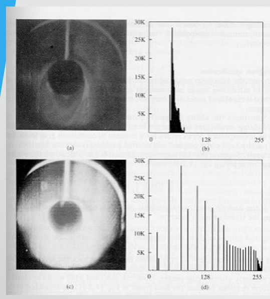


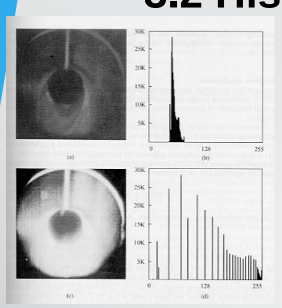
Table 1 shows the distribution of the pixels (Column marked # of pixels).

Gray Level	# of pixels	Pr	s	s*255
0	0	0	0	0
45	10500	0.05043	0.05043	13
46	3600	0.01729	0.06772	17
48	24600	0.11816	0.18588	47
50	28500	0.13689	0.32277	82
51	22800	0.10951	0.43228	110
53	18900	0.09078	0.52305	133
54	17100	0.08213	0.60519	154
55	14400	0.06916	0.67435	172
57	12000	0.05764	0.73199	187
60	10800	0.05187	0.78386	200
62	7200	0.03458	0.81844	209
64	6000	0.02882	0.84726	216
66	6600	0.03170	0.87896	224
68	7200	0.03458	0.91354	233
70	5400	0.02594	0.93948	240
71	3600	0.01729	0.95677	244
73	3000	0.01441	0.97118	248
74	1800	0.00865	0.97983	250
76	1500	0.00720	0.98703	252
78	900	0.00432	0.99135	253
80	1800	0.00865	1.00000	255
255	0	0.00000	1.00000	255
Total	208200	1.00000		

Table 1

23

3.2 Histogram Equalization – Discrete Example



This is a discrete case of Histogram Equalisation, we shall now outline the procedure.

1. Determine *Pr* by dividing each value of # of pixels by the total number of pixels (208200 in this example).
2. Determine the value of s_k using the formula:

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \qquad 0 \le r \le 1 \text{ , and } k = 0, 1, 2, \ldots, L-1$$

For example,

$$s_{53} = \sum_{j=0}^{53} \frac{n_j}{n_{total}} = \sum_{j=0}^{53} \mathrm{Pr}_j = \mathrm{Pr}_0 + \mathrm{Pr}_{45} + \mathrm{Pr}_{46} + \mathrm{Pr}_{48} + \mathrm{Pr}_{50} + \mathrm{Pr}_{51} + \mathrm{Pr}_{53} = 0.52305$$

As the rest of *Pr*'s are equal to zero.

24

3.2 Histogram Equalization – Discrete Example

Gray Level	# of pixels	Pr	s	s*255
45	10500	0.05043	0.05043	13
46	3600	0.01729	0.06772	17
48	24600	0.11816	0.18588	47
71	3600	0.01729	0.95677	244

3. Multiply the values of s_k by the maximum value of the gray level, which is 255 in this example (see column marked *s*255*).

4. We now have the equalized gray level. *r* is now mapped to *s*. For example, *r* = 45 is now mapped to *s* = 13, and *r*=71 is mapped to *s* = 244, and so on.

5. We can now plot the new histogram using *s* and the corresponding *Pr* value. For example, when *s* = 13, use *Pr* = 0.05043 (Value of *Pr* when r = 45), and when *s* = 244, *Pr* = 0.01729 (Value of *Pr* when r = 71).

25

3.2 Histogram Equalization – Discrete Example

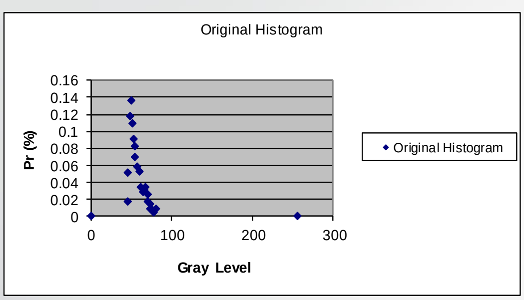
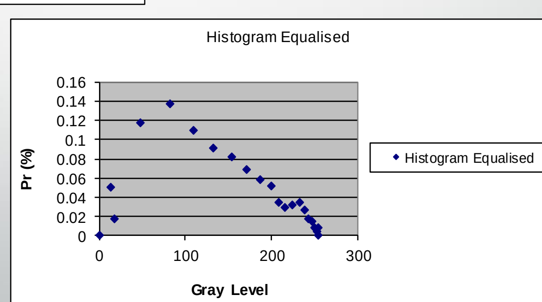


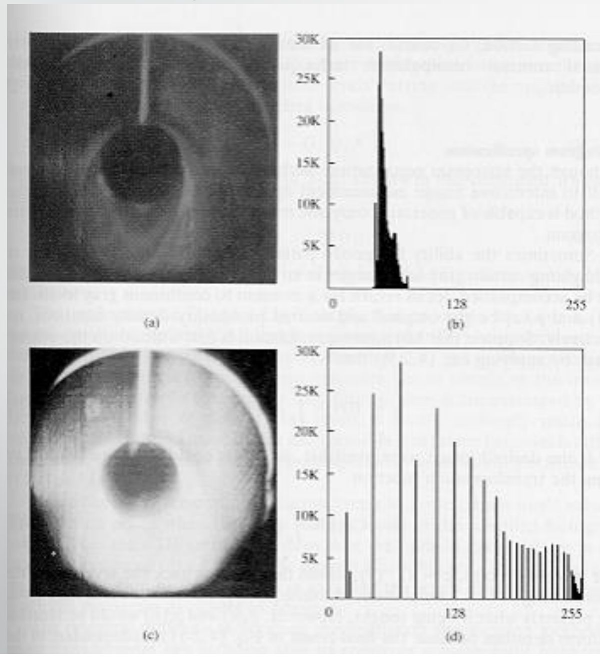
Fig. 12(a)
Original Histogram,

Fig. 12(b)
Equalized Histogram,



26

3.2 Histogram Equalization – Discrete Example



The original image and the associated histogram

Fig. 11 (a) Original image and (b) its histogram; (c) image subjected to histogram equalization and (d) its histogram.

27

3.2 Histogram Specification

Histogram Equalisation is useful, but it does not allow iterative image enhancement applications. This is because it generates only ONE result.

Sometimes, it may be desirable to specify particular histogram shapes capable of highlighting certain gray-level range.

Let us consider images with continuous gray level values.

Let $p_r(r)$ = original *p.d.f.*
$p_z(z)$ = desirable *p.d.f.*

Histogram Equalization is first applied on the original image, i.e.

$$s = T(r) = \int_0^r p_r(w)dw \tag{13}$$

If the desirable image were available, its level can also be equalised as

$$v = G(z) = \int_0^z p_z(w)dw$$

28

3.2 Histogram Specification

Histogram Equalization is first applied on the original image, i.e.

$$s = T(r) = \int_0^r p_r(w)dw \qquad (13)$$

If the desirable image were available, its level can also be equalised as

$$v = G(z) = \int_0^z p_z(w)dw \qquad (14)$$

and

$$z = G^{-1}(v) \qquad (15)$$

will give us the gray level of the desirable image.

In fact, z is what we wish to determine. Now $p_s(s)$ and $p_v(v)$ must be identical uniform densities, because from equation (8), *the end-result is independent of the p.d.f. in the integral*.

29

3.2 Histogram Specification

Replacing v *by* s *in equation (15),*

$$z = G^{-1}(s) \qquad (16)$$

Equation (16) would give the desired *p.d.f.* of the image.

Assuming $G^{-1}(s)$ is single-value, the procedure can be summarized as:

1. Equalize the levels of the original image using

$$s = T(r) = \int_0^r p_r(w)dw; \quad 0 \le r \le 1$$

2. Specify the desired density function and obtain the transformation function $G(z)$ using

$$v = G(z) = \int_0^z p_z(w)dw$$

3. Apply the inverse transformation function $z = G^{-1}(s)$ to the level obtained in step (1).

The procedure yield a new processed image with the news gray level characterised by $p_z(z)$.

30

3.2 Histogram Specification

We can combine both transformations, $T(r)$ followed by $G^{-1}(s)$ into one function. We have

$$z = G^{-1}(s) \qquad (16)$$

From equation (8),

$$z = G^{-1}\left[T(r)\right] \qquad (17)$$

which relates *r to z*. When $G^{-1}\left[T(r)\right] = T(r)$, equation (17) represents simply histogram equalization. In this method, we need to determine $T(r)$ and the inverse transformation G^{-1}.

In continuous case, the difficulty of this method is to determine G^{-1} analytically. This problem is less serious in discrete case. The formulation is similar to that shown in Equation (12).

31

3.2 Histogram Specification

In practice, $z = G^{-1}(s)$ is not single-value. This occurs when there is un-filled gray levels in the specified histogram (which makes CDF remains constant over the unfilled interval).

Any principal difficulty is the construction of a meaningful histogram. We may specify a particular *p.d.f.* (Gaussian, for example) and form the histogram by digitising the function. We may also use graphical input device to specify the required histogram.

32

3.2 Histogram Specification - Example

Example: An image has gray-level *p.d.f.* is represented by Fig. 10(a), if the desired histogram *p.d.f.* is $p_z(z)$ shown in the above figure, determine the required transformation.

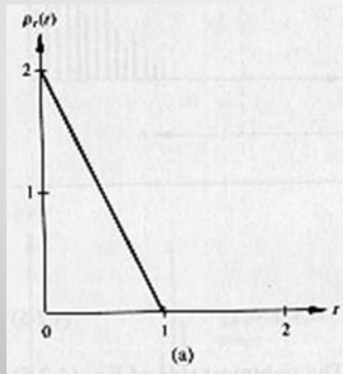


Fig. 10(a)

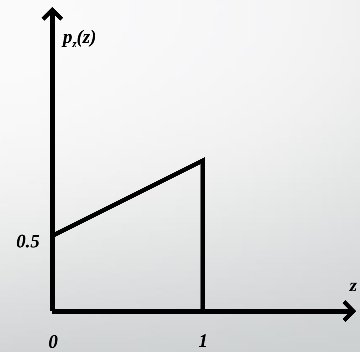
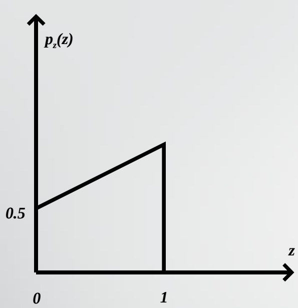


Fig. 13

33

3.2 Histogram Specification - Example



We have (Example on Histogram Equalization)

$$s = T(r) = -r^2 + 2r$$

$$v = G(z) = \int_0^z p_z(w)dw = \int_0^z \left(w + \frac{1}{2}\right)dw$$

$$v = \frac{z^2}{2} + \frac{z}{2} = \frac{z}{2}(z+1)$$

Replacing v by s

$\therefore$

$$s = \frac{1}{2}(z^2 + z), \quad \text{or} \quad z^2 + z - 2s = 0$$

$$z = \frac{1}{2}\left(-1 \pm \sqrt{1+8s}\right)$$

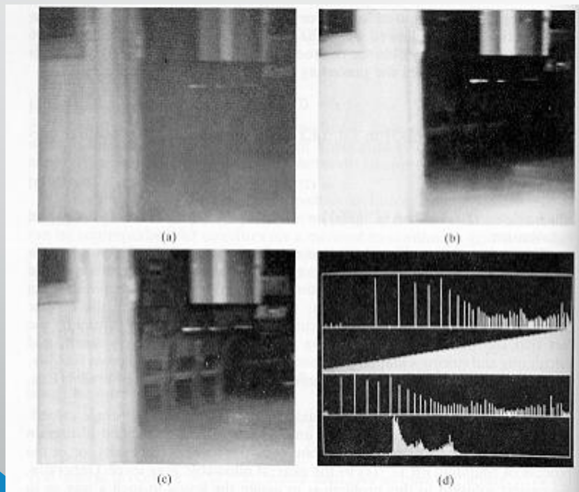
$$z = \frac{1}{2}\left(\sqrt{1+8s} - 1\right) \quad \text{or} \quad \underline{\underline{z = \frac{1}{2}\left(\sqrt{1+16r-8r^2} - 1\right)}}$$

$$0 \le s \le 1$$

34

3.2 Histogram Specification – Pictorial Example

Example: Fig. 14(a) shows a semi-dark room viewed from a door way.



Note that the image after histogram equalization has higher contrast (Fig. 14(c)), whereas that processed by the specified histogram has a more balanced appearance.

Fig. 14 Illustration of the histogram specification method: (a) original image; (b) after histogram equalization; (c) image enhanced by histogram specification; (d) histograms.

35

3.2 Histogram based Processing

<u>Local Enhancement</u>

- The above methods are suitable for global enhancement.
- It is necessary, sometimes, to enhance details over small areas, which contain small number of pixels that may have negligible influence on the computation of a global transformation.
- The solution is to devise transformation functions based on the gray level distribution in the neighbourhood of every pixel in the image.

We can adopt the <u>two histogram processing techniques</u> for the said purpose.

Define a square or rectangular neighbourhood and move the centre of this area from pixel to pixel. At each location, the histogram of the points in the neighbourhood is computed, and either a histogram equalisation or specification is computed.
The function is used to map the gray level of the pixel centered in the neighbourhood.
The centre of the neighbourhood is moved to the adjacent pixel and the procedure is repeated.

36

3.2 Histogram based Processing - Local

Local Enhancement

- Define a square or rectangular neighborhood and move the centre of this area from pixel to pixel (such as a square window).
- At each location, the histogram of the points in the neighborhood is computed, and either a histogram equalization or specification is computed.
- The function is used to map the gray level of the pixel centered in the neighborhood.
- The centre of the neighborhood is moved to the adjacent pixel and the procedure is repeated.

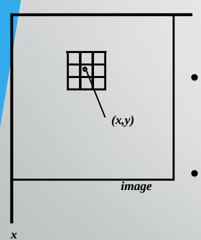


Fig. 1

3.2 Histogram based Processing - Local

Local Enhancement - Example

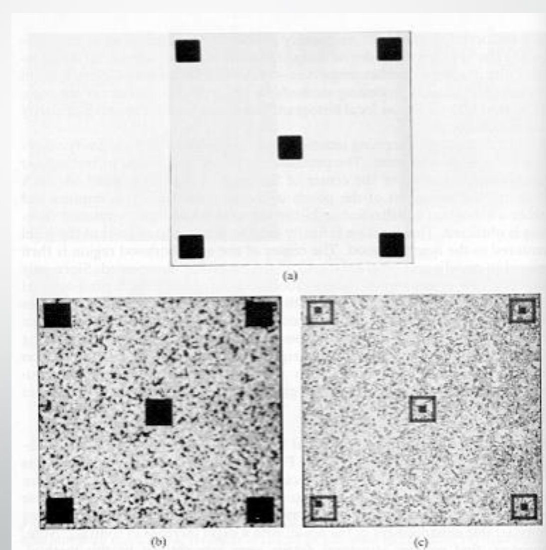


Fig. 15 (a) Original image; (b) result of global histogram equalisation; (c) result of local histogram equalisation using a 7 × 7 neighbourhood about each pixel.

3.2 Other Enhancing Techniques

Instead of using histogram, we can also use other properties of the pixel intensities in a neighborhood.

Two common properties are

m(*x,y*), intensity mean -measure of average brightness.

σ(*x,y*), variance - measure of contrast

The transformation is done with

$$g(x,y)=A(x,y).[\ f(x,y)-m(x,y)\ ]+m(x,y) \qquad (20)$$

where

$$A(x,y)=k\frac{M}{\sigma(x,y)} \quad . \quad 0<k<1 \qquad (19)$$

39

3.2 Other Enhancing Techniques

$$g(x,y)=A(x,y).[\ f(x,y)-m(x,y)\ ]+m(x,y) \qquad A(x,y)=k\frac{M}{\sigma(x,y)} \quad . \quad 0<k<1$$

1. *m*(*x,y*) and σ(*x,y*) are computed in a neighbourhood centered at (*x,y*)
2. *M* is the global mean of *f*(*x,y*)
3. *k* is a constant.

A, *m* and σ depend on a pre-defined neighbourhood of (*x,y*). Application of the gain factor *A* (*x,y*) amplifies the difference between *f*(*x,y*) and the local mean *m*(*x,y*), and hence the local variations.

As $A(x,y) \propto \frac{1}{\sigma(x,y)}$, areas with low contrast receive larger gain.

The mean *m*(*x,y*) is added back in equation (18) to restore the average intensity level of the image in the local region.

40

3.2 Other Enhancing Techniques

In practice, adding back a fraction of $m(x,y)$ and restricting the variation of the gain $A(x,y)$ between (A_{min}, A_{max}) is desirable to balance large excursion of intensity in isolated region.

Fig. 16 shows an application of the technique using a 15 x 15 local region. Note: The enhancement detail at the boundary between two regions of different overall gray levels and the rendition of gray-level details in each region

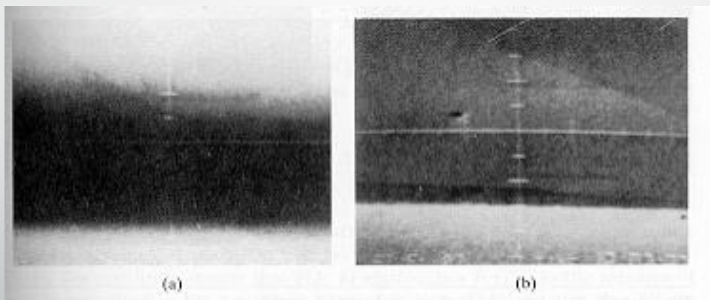


Fig. 16 Image before and after local enhancement.

3.3 Image Subtraction

Difference between two images $f(x,y)$ and $h(x,y)$,

$$g(x,y) = f(x,y) - h(x,y) \tag{20}$$

The difference is taken between all pairs of corresponding pixels from $f(x,y)$ and $h(x,y)$.

A classical application - mask-mode radiography $h(x,y)$ which is the mask; is an x-ray image of a region of a patient's body captured by an intensifier and TV camera. The image $f(x,y)$ is one sample of a series of similar TV images of the same region but acquired after injection of a dye into the blood stream.

3.3 Image Subtraction

The net effect of subtracting the mask from the in-coming stream of the samples is that only the areas that are different between *f*(*x,y*) and *h*(*x,y*) appear in the output image as enhanced details. We will obtain a movie showing how the dye propagates through the various arteries. (Fig. 17).

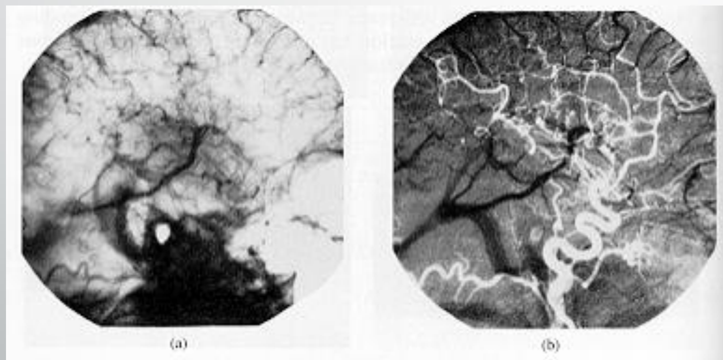


Fig. 17 Enhancement by image subtraction: (a) mask image; (b) image (after injection of dye into the bloodstream) with mask subtracted out.

3.4 Image Averaging

Consider a noisy image *g*(*x,y*) formed by the addition of noise $\eta(x,y)$ to an original image

$$g(x,y) = f(x,y) + \eta(x,y) \tag{21}$$

Assumption: at each (*x,y*), the noise is un-correlated and has zero average mean. If the noise satisfies these constraints, then, if an image $\bar{g}(x.y)$ is formed by averaging *M* different noisy images,

then

$$\bar{g}(x,y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y) \tag{22}$$

$$E\{\bar{g}(x,y)\} = f(x,y) \tag{23}$$

and

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{M}\sigma^2_{\eta(x,y)} \tag{24}$$

where

$E\{\bar{g}(x,y)\}$ is the expected value of $\bar{g}$, and $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{\eta(x,y)}$ are the variances of $\bar{g}$ and η, all at coordinates (*x,y*).

3.4 Image Averaging

The standard deviation at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{\eta(x,y)} \qquad (25)$$

Equations (24) and (25) indicate that as M increases, the variability of the pixel values at each location (x,y) decreases

Because $E\{\bar{g}(x,y)\} = f(x,y)$

$\rightarrow$ $\bar{g}(x,y) \rightarrow f(x,y)$ as M increases.

Fig. 18 shows an example of image averaging. The example clearly shows that as M increases, noise in the processed image decreases.

45

3.4 Image Averaging

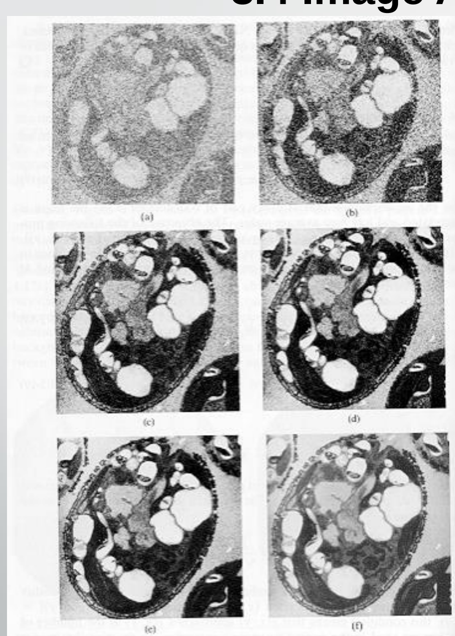


Fig. 18 Example of noise reduction by averaging (a) a typical noisy image; (b)-(f), results of averaging 2, 8, 16, 32, and 128 noisy images.

46

4.1 Spatial Filtering - Background

4.1 Background

Spatial filtering refers to the use of spatial masks for image processing. The masks are called spatial filters.

Low-pass filters

They attenuate or eliminate high frequency components in the Fourier Domain while leaving low frequencies un-touched.

High-frequency components characterize edges and other sharp details in an image. Hence, low-pass-filters will cause image blurring.

High-pass filters

They attenuate or eliminate low frequency components. The latters are responsible for the slowly varying characteristics, such as overall contrast and average intensity. This kind of filters has a net effect of reducing these features and a corresponding apparent sharpening of edges and other sharp details

47

4.0 Spatial Filtering

Band-pass filters

They are used for image restoration, and are seldom used in image enhancement.

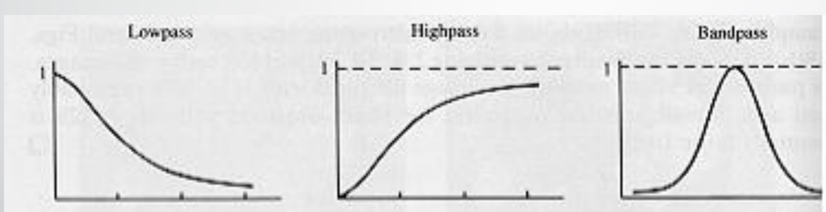


Fig. 19 Spatial Filers

48

4.0 Spatial Filtering

Application of Filters

Regardless of the types of filters used, the basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image. Fig. 20 shows a 3 x 3 mask. Denoting the gray levels of pixels under the mask at any location by $z_1, z_2, \ldots, z_9$, the response of a linear marks is:

Fig. 20 A 3 by 3 mask with arbitrary coefficients

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 \qquad (26)$$

$$R = \sum_{i=1}^{9} w_i z_i$$

The centre of the mask is placed at (x,y), the gray level of the pixel is then replaced by R.

The mask is then moved to the next pixel position and the process is repeated, until all the pixel locations in the image have been covered.

4.0 Spatial Filtering

Application of Filters

R is computed using partial neighbourhood for pixels located at the border. A new image is usually created to store the new values of the pixels.

Non-linear spatial filters also operate on neighbourhoods. They usually operate directly on the values of the pixels in the neighbourhood under consideration.

They **do not explicitly use** coefficients in the manner described in equation (26).

Examples of non-linear filters are

median filtering for noise reduction;

max filter: $R = \max\{z_k \mid k = 1, 2, \ldots, 9\}$ to find the brightest spot.

min filter, $R = \min\{z_k \mid k = 1, 2, \ldots, 9\}$ to find the dimmest spot.

4.2 Spatial Filtering – Smoothing Filters

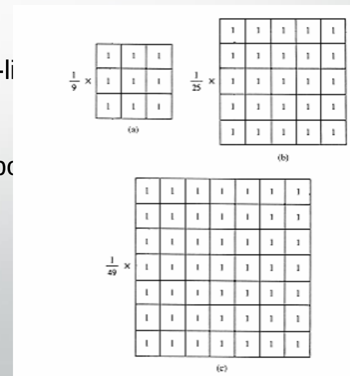
4.2 Smoothing Filters

Smoothing filters are used for blurring and for noise reduction. Blurring is used in pre-processing steps, such as,

- removal of small details prior to large object extraction, and
- bridging gaps in lines or curves.

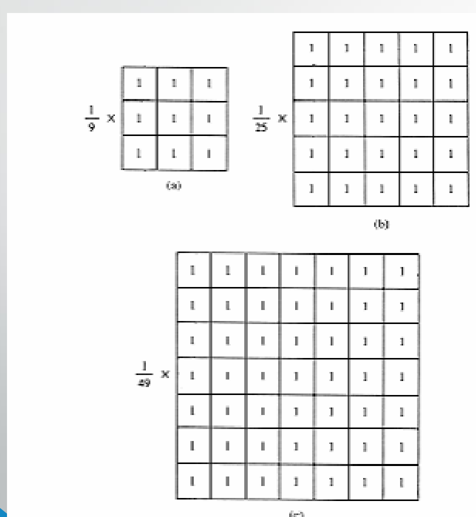
Noise reduction can be achieved with a linear on non-li

Low-pass spatial filtering
Fig. 21(a) shows that this type of filter must have all po



4.2 Spatial Filtering – Smoothing Filters

Low-pass spatial filtering
Fig. 21(a) shows that this type of filter must have all positive coefficients.



Three such filters are shown in Fig. 21. The division is necessary in each case to prevent *R* from exceeding the valid gray level range. Note that *R* is simply the average of all the pixels in the area of the mask. Hence, the use of masks with this form is known as neighbourhood averaging.

Fig. 21 Spatial low pass filters of various sizes.

4.2 Spatial Filtering – Smoothing Filters

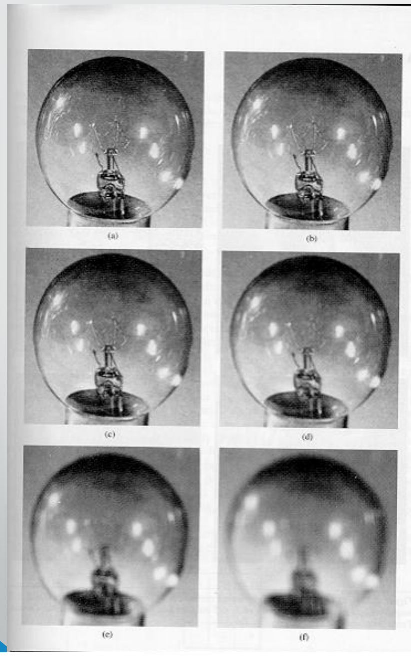


Fig. 22 (a) Original images; (b)-(f) results of spatial lowpass filtering with a mask of size $n \times n$, n = 3, 5, 7, 15, 25.

4.2 Spatial Filtering – Smoothing Filters

Median Filtering - Non-linear filter.

The difficulty with low-pass filtering is that it blurs edges and other sharp details. We can use median filtering to just remove noise.

In median filtering, the gray-level of each pixel is replaced by the median of the gray levels in a neighbourhood of that pixel. This method is effective in removing noise pattern consisting of strong and spike-like components while preserving the sharp edges.

The median *m* of a set of values is such that half the values in the set are less than *m*, and half are greater than *m*.

4.2 Spatial Filtering – Smoothing Filters

Steps:

1. Sort the values of the pixel and its neighbours.
2. Determine the median.
3. Assign this value to the pixel.

Eg. In. a 3 x 3 mask, the median is the 5th largest value, while in a 5×5 mask, it would be the 13th.

(10, 20, 20, 20, 15, 20, 20, 25, 100),
after sorting (10, 15, 20, 20, 20, 20, 20, 25, 100).

The median is then 20. The pixel occupying the 5th position (15 in this example) is then replaced by 20. In this manner, pixels are forced to have values closer to their neighbours.

55

4.2 Spatial Filtering – Smoothing Filters

Fig. 23 shows an example of an image enhanced by neighbourhood averaging and by median filtering. The latter seems to have performed better.

Fig. 23 (a) Original image; (b) image corrupted by impulse noise; (c) result of 5×5 neighbourhoold averaging; (d) result of 5×5 median filtering.

56

4.3 Spatial Filtering – Sharpening Filters

Objective: To highlight fine detail in an image or to enhance detail that has been blurred, either in error, or as a natural effect of a particular method of image acquisition.

Applications: Electronic printing, Medical imaging, industrial application/inspection, and automatic target detection in smart weapons.

4.3 Spatial Filtering – Sharpening Filters

4.3.1. Basic high-pass spatial filtering

To implement high-pass spatial filtering, the filters should have positive coefficients near its centre, and negative coefficients in the outer periphery.

For a 3 x 3 mask, a positive coefficient at the central location and negative ones in the rest of the mask meet the said condition.

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Fig. 24 A basis 3×3 high pass spatial filer

Fig. 24 shows a high pass spatial filter. Note that the sum of the coefficients is equal to zero. When the mask shown in Fig. 24 is over an area of constant or slow varying gray level, the output is zero or very small.

i.e.

$$R = \frac{1}{9}\sum_{i=l}^{9} w_i z_i \approx 0 \qquad (26)$$

4.3 Spatial Filtering – Sharpening Filters

Applying this filter may yield some negative gray-level values. As we deal only with positive levels, the results of high pass filtering involve some form of scaling and/or clipping so that the results will lie in the range [0, L - 1].

Absolute values of the results should not be taken because large negative gray-level values will give very bright spots in the image.

Fig. 25 (a) Image of a human retina; (b) highpass filter using the mask in Fig. 24.

Application of high pass spatial filters will reduce the average intensity in the image to zero, reducing significantly the contrast. Fig. 25(b) shows the result of applying the 3×3 mask on the original image shown in Fig. 25(a). We see enhanced edges over a rather dark background.

59

4.3 Spatial Filtering – Sharpening Filters

<u>High Boost Filtering</u>

High-pass filtered image may be expressed as

$$\text{High-pass} = \text{Original} - \text{Low-pass} \qquad (27)$$

This may be verified by using equation (26), Fig. 21(a) and Fig. 24 (Try it !).

High-boost or High-frequency emphasis filter is given by

$$\begin{aligned} \text{High boost} &= (A)\,(\text{Original}) - \text{Low-pass} \\ &= (A - 1)\,(\text{Original}) + \text{Original} - \text{Low-pass} \\ &= (A - 1)\,(\text{Original}) + \text{High-pass} \qquad (28) \end{aligned}$$

where A is an amplification factor ≥ 1.
when $A = 1$, the standard high-pass results.
when $A > 1$, part of the original is added back to the high-pass result.

60

4.3 Spatial Filtering – Sharpening Filters

The Amplication Factor, *A*, has the effect of restoring partially the low-frequency components lost in the high-pass filtering operation.

Hence, the end result is an image which looks more like the original image, with a relative degree of edge enhancement that depends on the value of *A*.

61

4.3 Spatial Filtering – Sharpening Filters

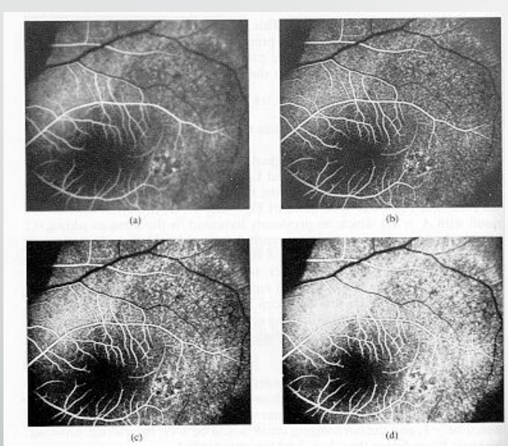


Fig. 26 (a) Original images; (b)-(d) result of high-boost filtering using the mass in Fig. 27, with A – 1.1, 1.15, and 1.2, respectively. Compare these results with those shown in Fig. 25.

Fig. 26 shows the application of high-boost filter. Note that noise is also enhanced because high-pass filters enhance noise and high boost filters increase that effect.

Fig. 26(b) shows significant improvement over pure high passhigh filtering. (Brighter background). But, at *A* = 1.2, the image is at the verge of being unacceptable.

62

4.3 Spatial Filtering – Sharpening Filters

In term of implementation, the high-boost filter can be achieved by letting the central-pixel of the mask shown in Fig. 27 to be

$$w = 9A - 1 \quad (29)$$

with $A \geq 1$. A determines the nature of the filter

	-1	-1	-1
1/9 ×	-1	W	-1
	-1	-1	-1

Fig. 27 High-boost filter mask with w=9A-1 with $A\geq 1$

63

4.3 Spatial Filtering – Sharpening Filters

Derivative Filters

Image averaging tends to blur details. Since averaging is analogous to integration, differentiation can be expected to have the opposite effect and thus sharpen an image.

The most common method of differentiation in image processing application is gradient.

For a function $f(x,y)$, the gradient of coordinates (x,y) is defined as the vector

$$\nabla \vec{f} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (30)$$

The magnitude

$$\nabla f = mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (31)$$

64

4.3 Spatial Filtering – Sharpening Filters

Derivative Filters

$$\nabla \vec{f} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (30)$$

The magnitude

$$\nabla f = mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \qquad (31)$$

65

4.3 Spatial Filtering – Sharpening Filters

Derivative Filters

Equation (31) can be applied at z_5 (centre pixel of a 3×3 mask) in a number of ways.

The simplest is

$$\nabla f = \left[(z_5 - z_8)^2 + (z_5 - z_6)^2 \right]^{1/2} \qquad (32)$$

or by absolute values

$$\nabla f = |z_5 - z_8| + |z_5 - z_6| \qquad (33)$$

Using cross differences:

$$\nabla f \approx \left[(z_5 - z_9)^2 + (z_6 - z_8)^2 \right]^{1/2} \qquad (34)$$

or

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8| \qquad (35)$$

66

4.3 Spatial Filtering – Sharpening Filters

Equations (32) - (35) can be implemented by using masks of size 2 x 2.

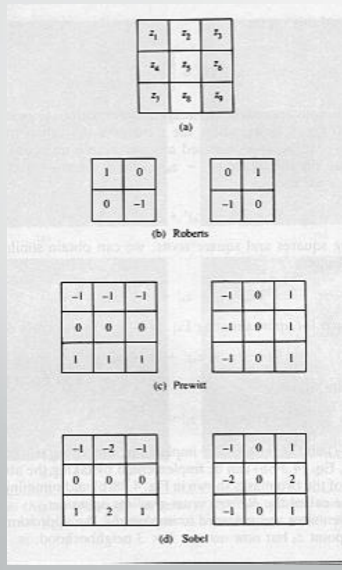


Fig. 28 A 3×3 region of an image (the *z*'s are gray-level values) and various masks used to compute the derivative at point label z_5 . Note that all mask coefficients sum to 0, indicating a response of 0 in constant areas, as expected of a derivative operator.

67

4.3 Spatial Filtering – Sharpening Filters

Equation (31) can be implemented by taking the absolute value of the response of the two masks shown in Fig. 28(b) and summing the results. These masks are called Roberts cross-gradient operators.

Note: Masks of even size are awkward to implement.

Equation (31) can also be implemented using a 3 × 3 neighbourhood centered at z_5.

$$\nabla f = \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right| \quad (36)$$

Prewitt Operators shown in Fig. 28(c) can be used to implement this operator (Equation 36). Sobel operators shown in Fig. 28(d) are also an operator.

68

4.3 Spatial Filtering – Sharpening Filters

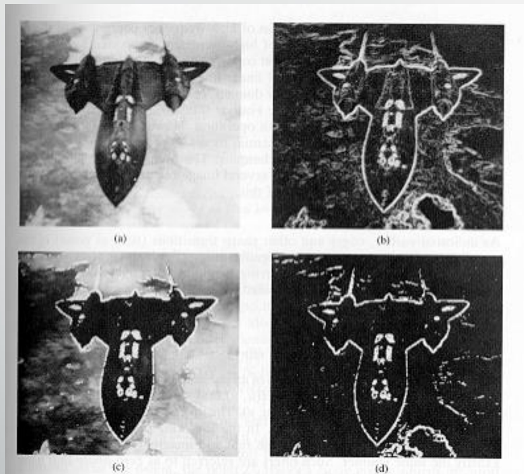


Fig. 29(c).

When $\nabla f > 25$, $f(x,y) = 255$
$\nabla f < 25$, $f(x,y) =$ original value.

Fig. 29(d).

When $\nabla f > 25$, $f(x,y) = 255$
$\nabla f < 25$, $f(x,y) = 0$

Fig. 29 Edge enhancement by gradient technique. (a) Original image; (b) results of using Prewitt operators.

69