INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE CENTRATOM TRIESTE

H4.SMR/383 - 12

WORKSHOP ON REMOTE SENSING TECHNIQUES WITH APPLICATIONS TO AGRICULTURE, WATER AND WEATHER RESOURCES

(27 February - 21 March 1989)

STATISTICS BASED DIGITAL IMAGE ENHANCEMENT TECHNIQUES

STEFANO BARONTI
IROE - CNR
Via Panciatichi, 64
50127 Firenze
Italy

Statistics Based Digital Image Enhancement Techniques

Stefano Baronti
IROE - CNR
Via Panciatichi 64
50127 Florence, Italy

Introduction

Enhancement techniques are aimed to improve the quality of images and to make the successive analysis easier. Depending on the user needs, these techniques are used to eliminate unwanted effects (for example noise) or to point out particular features (for example contours) of an image. The processing is carried out without considering into depth the distortion causes, but only examining the effects.

In the following, we shall consider some techniques which are based on the grey level histogram; we shall start with de-striping up to consider histogram transformations and histogram based filters.

De-striping methods

De-striping consists of the elimination of line banding, present in some remotely sensed images, due to the imbalance of the detectors used in the scanning process. In fact, as an example, Landsat MSS uses six detectors while Landsat TM sixteen. It happens that the response of each detector is slightly different from the others. This effect appears in the acquired image as line banding, with same lines presenting an average grey level different from the others, also in the presence of a uniform area on the image. This effect is particularly evident on the sea areas by issuing a pseudo color representation aimed to enhance contrasts.

A way to correct this effect is to use the grey level histogram. The idea is that each detector "sees" a similar distribution of all the land cover categories present in the imaged area. So, the mean and the standard deviation of the data acquired from each detector should be the same. We can impose this constraint and the resulting algorithm is the following.

- Data are divided into N categories depending on the sensor by which they have been acquired (6 categories for MSS and 16 for TM)
- for each category the mean $x_m(k)$, the variance $\sigma(k)$ and the standard deviation $s(k)$ of the pixels belonging to that

category are computed.

$$x_m(k) = \frac{1}{n(k)} \sum_{i=1}^{n(k)} x_i(k) \qquad (1)$$

$$\sigma(k) = \sum_{i=1}^{n(k)} x_i^2(k)/n(k) - x_m^2(k) \qquad (2)$$

$$s(k) = (\sigma(k))^{1/2} \qquad (3)$$

where $n(k)$ is the number of pixels of the k^{th} category.

- the global variance V, the global mean M and the global standard deviation S, are identified as targets and computed with formulas analogous to (1), (2), (3) but over the all image.
- a gain factor and a bias factor are computed for each band:

$$g(k) = S/s(k) \qquad (4)$$

$$b(k) = M - g(k)x_m(k) \qquad (5)$$

- each pixel of the category k is corrected with the formula:

$$y_i(k) = g(k)x_i(k) + b(k) \qquad (6)$$

Histogram transformation

The objective is to change the grey level of the pixels in the image in order to modify the histogram in such a way it assumes a given form. In fact, the grey level histogram is very useful to characterize an image or in general the acquisition process. If, for example, the histogram is concentrated towards low grey levels, there is a good probability that the gain of the sensor is low or the radiance coming from the imaged area is scarce; if, on the contrary, the histogram is strongly concentrated towards high grey levels, it may be that some component of the acquisition system is working near saturation. In both cases the acquired signal is not well distributed in the grey level range.

We shall consider here three types of histogram transformations: histogram stretching, histogram equalization, histogram modification.

Histogram stretching

Histogram stretching is the more immediate way to enhance image contrast; the grey levels of an image are redistributed in the full range of the output device. The result is particularly evident if the original histogram is strongly grouped around the mean value.

Let min and max the minimum and the maximum grey level of the original image. Let $[O_{min}\ O_{max}]$ the range of grey levels we want in output (usually these levels are 0 and 255). The transformation defining the stretched image is:

$$g(m,n) = (f(m,n) - min)G + O_{min} \qquad (7)$$

where f(m,n), g(m,n) are respectively the input and the output images, and G is the gain factor:

$$G = (O_{max} - O_{min})/(max - min) \qquad (8)$$

The operation so defined is linear and reversible. There are however some variants to this basic scheme which have to be reported.

The first is "clipping"; clipping consists in applying equations (7) and (8) introducing two clipping factors $C_{min} \geq min$ and $C_{max} \leq max$ in the place of min and max respectively; values of the original image less than C_{min} are forced to O_{min} in the output image, while values greater than C_{max} are forced to O_{max}. In this way the gain factor G is greater and the contrast effect results more evident. This technique is very useful for sparse histograms, specially if the lateral points are due to noise. Clipping introduce a non-linearity in the transformation which also becomes not reversible.

A second variant is to divide the original grey level range into a given number of sub-intervals and apply eqs. (7), (8) for each sub-interval. In this way it is possible to increase contrast for those grey levels which define the objects in which we are interested.

The last variant considers a transformation on the original histogram that is not linear, and that can be applied also with clipping or by considering sub-intervals. All kind of transformations are applicable. As an example we consider here an exponential transformation:

$$g(m,n) = C(f(m,n) - min)exp((f(m,n)-min)/(max-min)) + O_{min} \qquad (9)$$

The exponential function varies from 1 to e when f(m,n) varies between min and max. C is a gain factor which has to be choose in such a way that for f(m,n) = max, g(m,n) = O_{max}. This means that

$$O_{max} = C(max - min)e + O_{min} \qquad (10)$$

and

$$O_{max} - O_{min} = C(max - min)e$$

from which

$$C = (O_{max} - O_{min})/(e(max - min)) \qquad (11)$$

Histogram equalization

The best way to quantize a signal is to attribute finer levels of quantization where the signal is more likely. This is not usually performed at the acquisition time, since the input signal statistics is unknown. A posteriori redistribution of grey levels can be applied to enhance the visual aspect of images. The redistribution is done in such a way that the grey level histogram of the output image is as flat as possible. To facilitate the redistribution process the number of grey levels of the output image should be less than those of the input one. The algorithm is:

1 choose the number of grey levels L_o

2 compute how many pixels N_o have to be attributed to each output level:

$$N_o = N/L_o \qquad (11)$$

where N is the total number of pixels

3 put k=0, j=0

4 scan the input histogram from level j and find l such that: $\sum_{i=j}^{l} n(i)$ approximates N_o, where n(i) indicates the number of pixels at level i

5 map the grey levels in the interval [j l] in the k^{th} level of the output histogram

6 if all the grey levels in the input histogram have been processed, stop; otherwise continue the scan of the input histogram by putting j = l+1 and k = k+1.

As pointed out at step 4, the redistribution is not perfectly realized since we are working with discrete quantities; a tradeoff between the flatness of the histogram and the quality of the output image is determined by the choice of the number of output levels L_o. The equalized histogram can be stretched to report the range of grey levels in the range of the output display.

Perfectly flat histograms are obtained by choosing some pixels and changing their values in such a way that the equality condition at step 4 is satisfied. This is not quite correct indeed, even if the resulting error is at most one category of the output histogram. The choice of the pixels to be modified can obey to some criterium or be random. In the former case information about neighbouring pixels is used to take decisions.

Histogram modification

If we have to put into correspondence two images taken in different times and situations, it may be that some differences exist due to some causes in which we are not interested. Let us explain with an example; if we are considering a remotely sensed image in the visible band taken in the morning and an image of the same area taken in the afternoon, we can find that some differences are due to the illumination angle. It may be, for example, that the first is darker than the second or viceversa.

A second example regards sensor tuning; this can cause spurious differences on images of the same area recorded in different times in the same local conditions.

One way to face these problems is to operate on the two images in order to make their histograms as similar as possible, thus correcting variable gain effects or bias factors due to unwanted causes. This is obtained by defining a transformation between the two histograms.

The problem can be formulated in the following way: given the histogram of the first image identified as $\{p_i\}$ and the target histogram of the second image identified as $\{q_i\}$, we want to find a transformation f: $\{p_i\}$ ===> $\{q_i\}$. In general the two histograms may have also a different number of levels.

The algorithm we propose is a generalization of the algorithm

presented for histogram equalization:

1 choose a target histogram $\{q_i\}$

2 put k = 0, j = 0

3 scan the histogram $\{p_i\}$ from level j and find l such that $\sum_{j}^{l} p_i$ approximates q_k

4 map the grey levels in the intervals [j l] in the k_{th} level of $\{q_k\}$

5 if $\{p_i\}$ has been scanned completely stop; otherwise j=l+1, k=k+1 and go to step 3.

Some variations are also in this case possible and regards essentially step 3. A best way to approximate $\{q_i\}$ is to request that for level k of $\{q_i\}$

$$\sum_{0}^{j_k} p_i \quad \text{approximates} \quad \sum_{0}^{k} q_i$$

where j_k indicates the index for which the best approximation is reached. In this way the errors at each step tend to compensate each other. The levels which are mapped to q_k are the levels which are in the interval $[j_{k-1}+1 \;\; j_k]$.

In the case we decide to force the equality at step 3 by changing the grey level of some pixels of the input image, the same considerations of histogram equalization apply.

Histogram based filters

The techniques examined up to now are considered as "point" techniques. We mean that, once the transformation has been defined, the grey level of a point is changed independently of its neighbouring pixels. Only in the particular case of the forced solution described for histogram equalization some local interferences are possible , but the intrinsic nature of the proposed techniques does not change . By the way, we can note that de-striping is only a particular case of histogram stretching; histogram stretching and histogram equalization are only a particular case of histogram modification.

We are going now to consider "local" techniques. The attribute of local refers to the fact that modifications applied to a pixel are decided on the basis of its neighbouring pixels, so utilizing the "local" information.

Local techniques are greatly diffuse in image processing and are furtherly distinguished in linear and non-linear. Linear local techniques assumes the form of a convolution between an input image and a PSF (the filter mask); the convolution theorem is valid, and is used as a valid alternative to direct spatial convolution.

Local non-linear techniques are strongly etherogeneous and not easily definable; among them some algorithms are based on grey level histogram. One of the most famous is median filtering which is described in next section.

Median filter

The filter is applied by defining a window (odd dimensions for simplicity) which is progressively overlapped to every pixel of the input image. As we shall see later, the dimensions of the window define the properties of the filter.

For the pixel falling in the window, the "local" grey level histogram is computed. In its simplest form the algorithm consists in substituting the grey level of the central pixel of the window, with the grey level of the pixel (the median) which is in the center of the local histogram.

The result of the filter is a smoothing of the input image; this smoothing is quite effective in the presence of random impulse noise. However, like all low-pass filters, also some unwanted effects appear, since the filter cannot obviously make distinctions between the informative signal and noise. The variations to the basic scheme described above are in fact aimed to make this filter more selective.

The first variant is to consider a threshold: the grey level substitution is performed only if the difference between the central pixel of the window and the median exceeds the threshold. In this way the image remains unchanged where it is sufficiently uniform. The threshold has to be tuned on the statistics of the signal and on the application needs.

A second variant is to give importance to the central pixel of the window by "weighting" its effect; one, two, three,....., k pixels of its value are included into the local histogram; in this way the probability that it may be the median increases.

Weighting and threshold are often combined together giving origin to the so called weighted median filter with threshold which is the more general form of this filter.

A third variation, known as K-neighbours, consists of fixing a parameter k which determines a number of pixels to be eliminated in the local window. If, for example, k=2, two pixels are eliminated from the sliding window and so from the local histogram. If the elimination on the histogram is symmetric,the result is equivalent to median filter; if not, the elimination of the two pixel is from the same side of the histogram, thus possibly changing the median. This is important for example to try to distinguish an edge from noise. For noise, it is likely that noisy pixels are locally similar; in this way it may happen that they are eliminated from the same side of the histogram. For an edge, it is likely that points defining the edge are on opposite sides, thus maintaining the original median. So, this filter has the potential advantage to preserve edge with respect to classic median filter.

A final consideration regards the pattern and the size of the sliding window. Usually a square or rectangular window is used; other patterns are also applicable depending on the noise characteristics. The size of the window is really an important parameter of the filter. To realize this, it is sufficient to consider an isolated pixel of noise on a uniform background; it is always eliminated when a 3x3 window filter is applied. If the spot of noise is of dimension 2x2, it is evident that a window 3x3 cannot remove this noise. This can be done by a 5x5 window.

In general, details are preserved when their dimensions are greater than the dimensions of the window divided by two (integer division is here considered).

Median filter is largely used in digital image processing, specially in the presence of random impulsive noise; in fact, median filtered images are less blurred with respect to others obtained by linear filtering; contours are more preserved. This is due to the fact that linear filters redistribute the energy in the local window thus changing also pixels which are near noise.

From an implementation point of view, median filter complexity is greater than that of convolution based filters, since for each pixel of the input image, the local histogram has to be computed. However a fast version exists in which the local histogram is updated by considering pixels entering and leaving the sliding window. This version is statistically faster but the number of needed operations cannot be forecasted a-priori.

Bibliography

A. V. Oppenheim, R. W. Schafer, Digital Signal Processing (Prentice Hall, Englewood Cliffs, 1975)

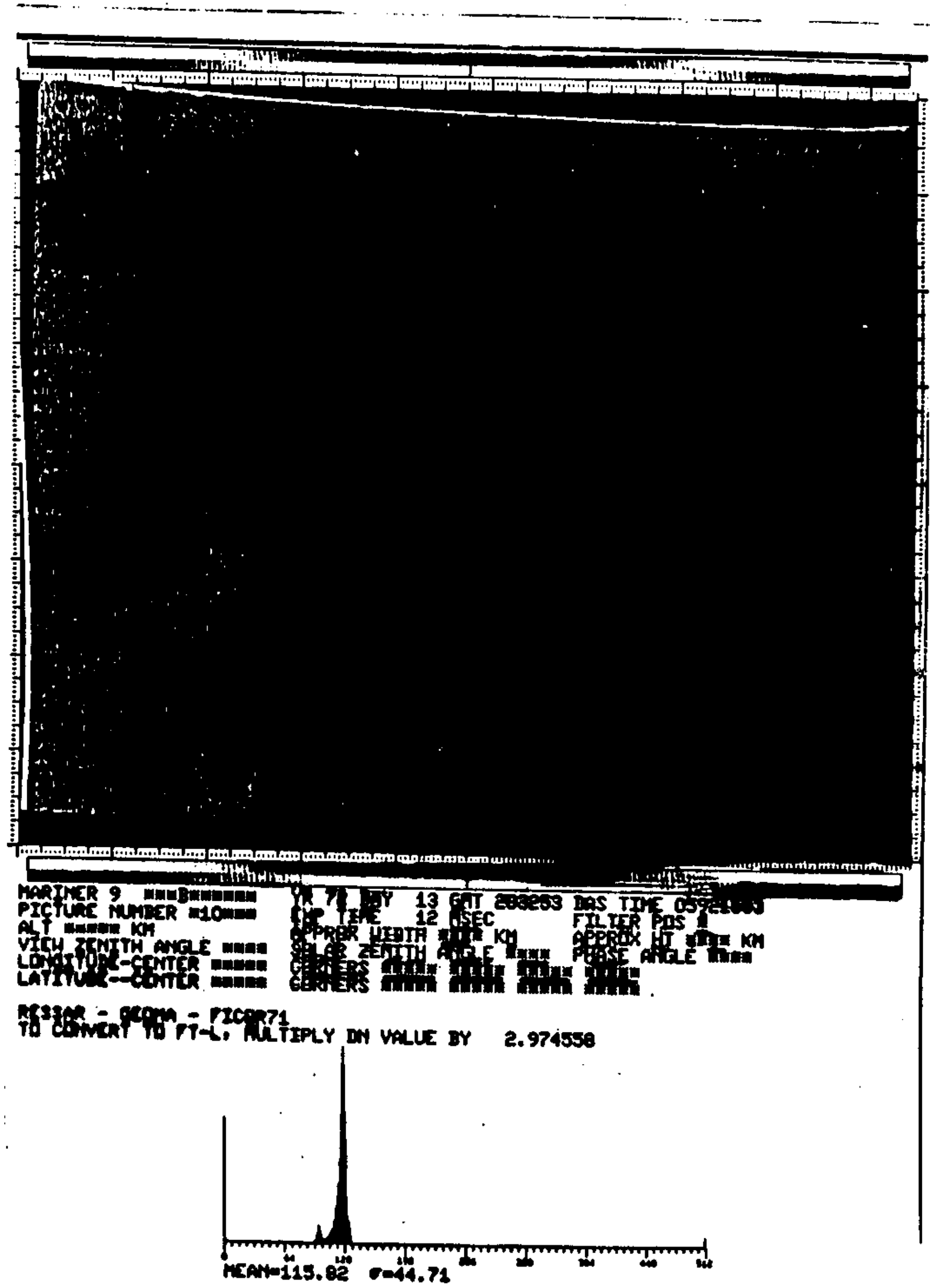
W. K. Pratt, Digital Image Processing (Wiley, New York, 1978)

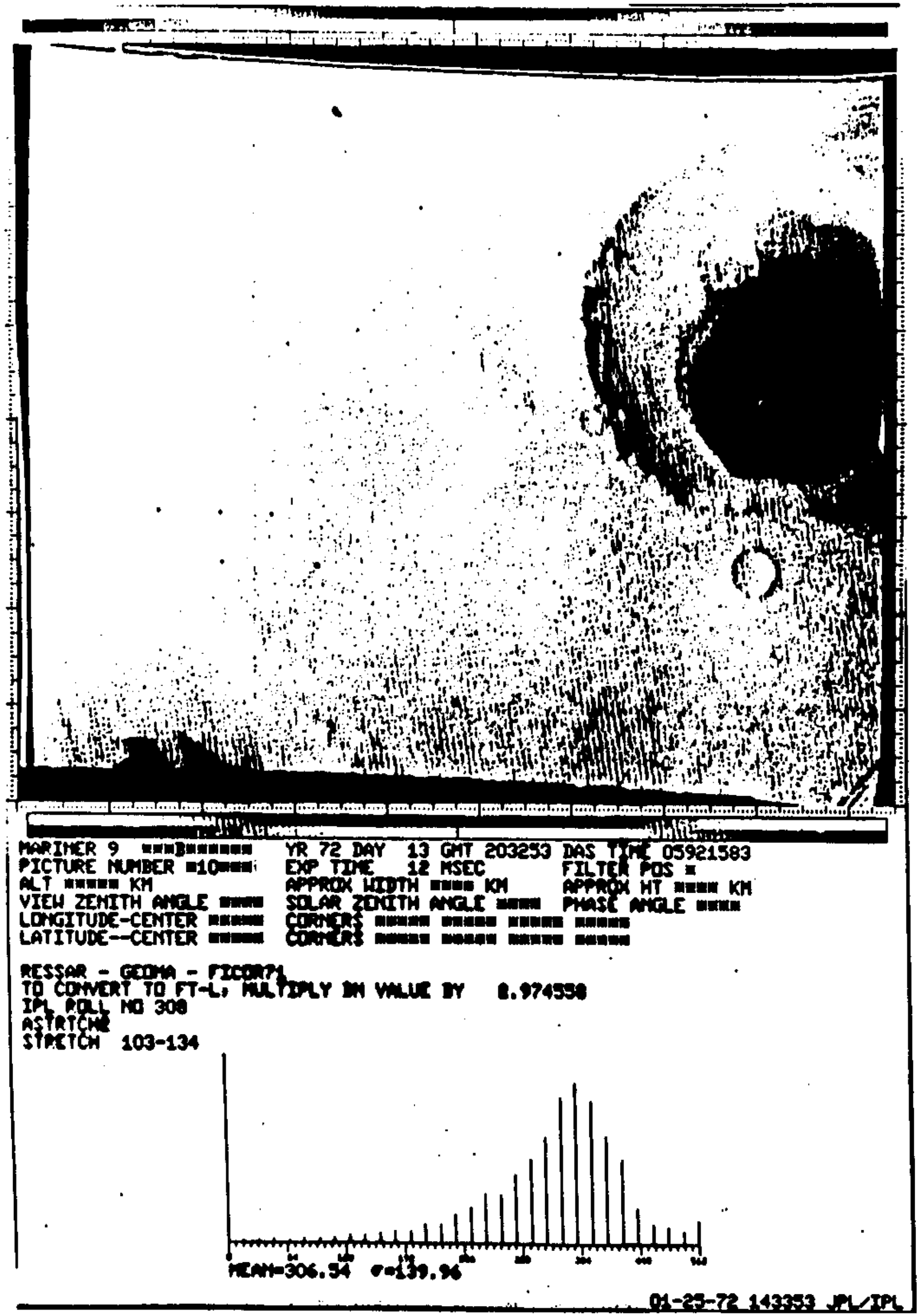
J. A. Richards, Remote Sensing Digital Image Analysis (Springer-Verlag, 1986)

P. M. Mather, Computer Processing of Remotely Sensed Images (Wiley & Sons, 1987)

R. C. Gonzalez, P. W. Wintz, Digital Image Processing (Addison-Wesley 1982)

A. Rosenfeld, A. Kak, Digital Picture Processing,(Academic Press, 1982)





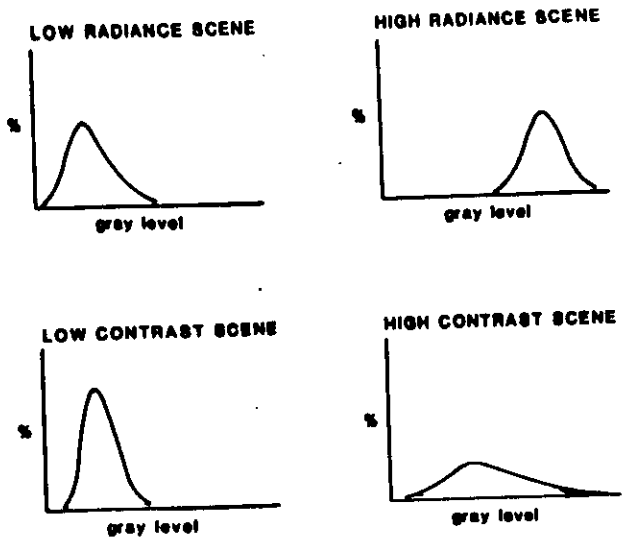


FIGURE 2-1. Histogram characteristics for different types of scenes.

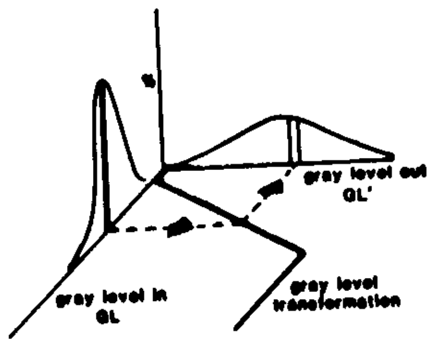
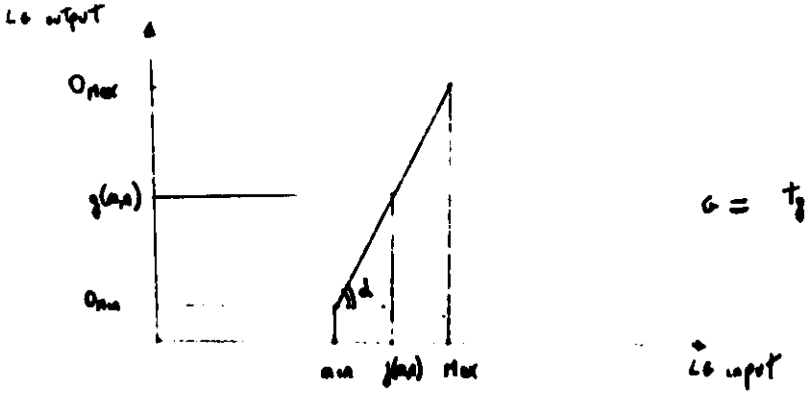


FIGURE 2-2. Gray level transformation.

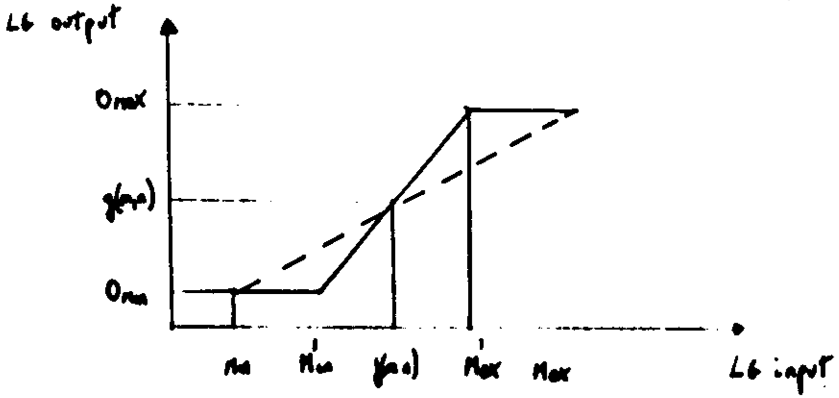
Contrast enhancement



$$G = tg\,\alpha = \frac{O_{max} - O_{min}}{Max - Min}$$

$$g(m,n) = \left(f(m,n) - min\right) tg\,\alpha + O_{min}$$

Contrast enhancement
+ clipping



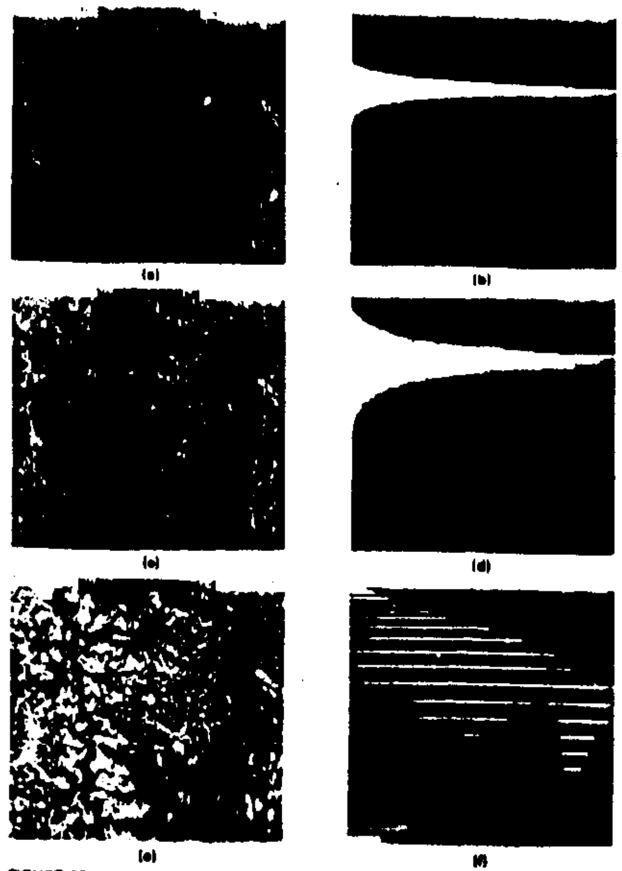


FIGURE 10.1-3. Example of contrast enhancement. (a) Original. (b) Original histogram. (c) Contrast enhancement; min clip = 43, max clip = 163. (d) Enhancement histogram. (e) Contrast enhancement; min clip = 63, max clip = 83. (f) Enhancement histogram.

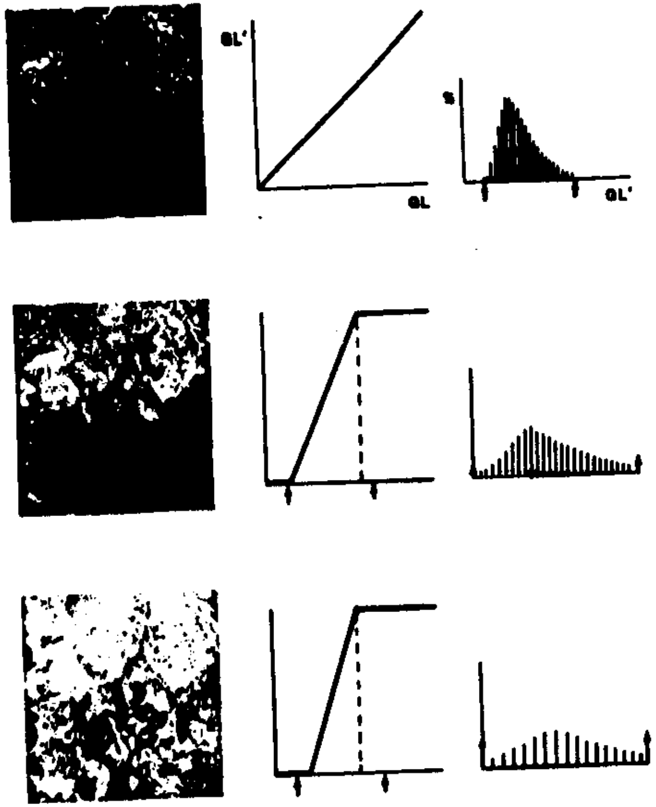


FIGURE 3-2. Linear contrast enhancement with variable saturation.

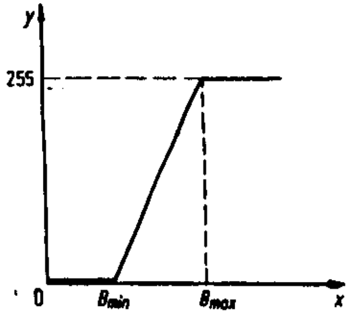


Fig. 4.7. Saturating linear contrast mapping

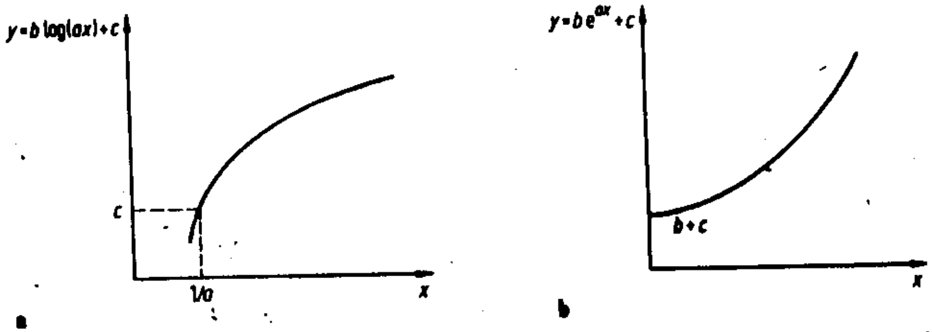


Fig. 4.8. Logarithmic a and exponential b brightness mapping functions. The parameters a, b and c are usually included to adjust the overall brightness and contrast of the output product

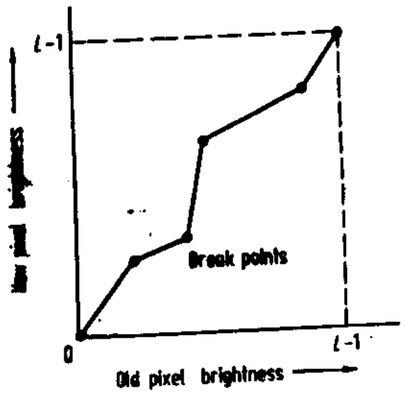


Fig. 4.9. Piecewise linear contrast modification function, characterised by the break points shown. These are user specified (as new, old pairs). It is clearly important that the function commence at 0,0 and finish at $L-1, L-1$ as shown

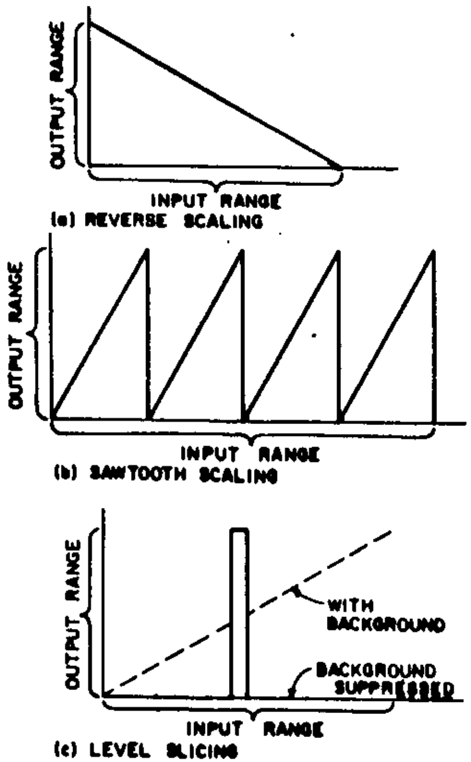


FIGURE 12.1-4. Reverse-scaling, sawtooth scaling, and level slicing.

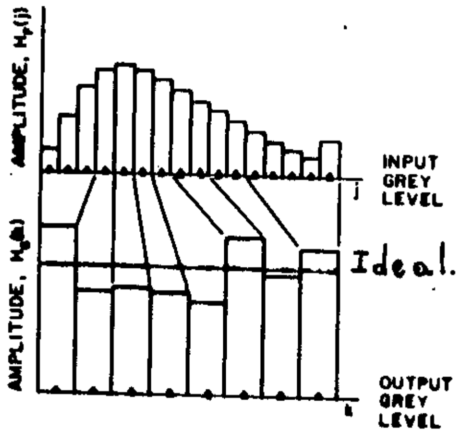
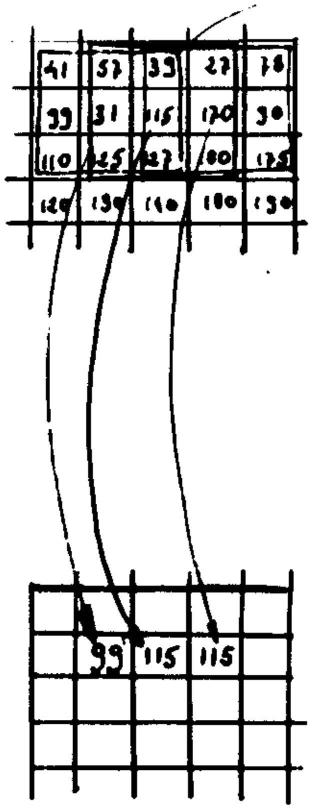
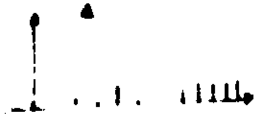


FIGURE 13.2-1. Example of approximate grey level histogram equalization with unequal number of quantization levels.

Example





31 39 41 57 (99) 110 115 125 127

27 31 39 57 (115) 125 127 170 180

27 39 78 90 (115) 127 170 175 180

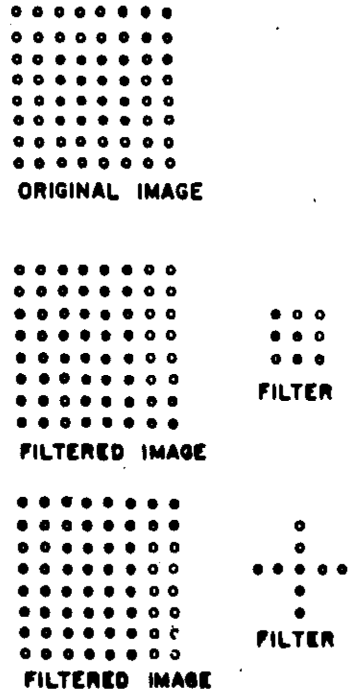


FIGURE 12.6-2. Example of two-dimensional median filtering.

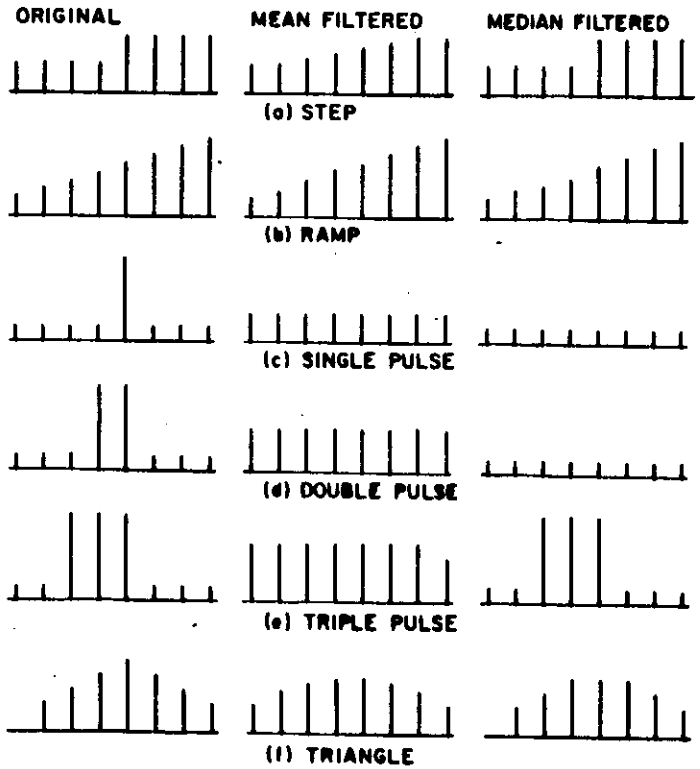


FIGURE 12.6-1. Examples of median filtering on primitive signals, $L = 5$.

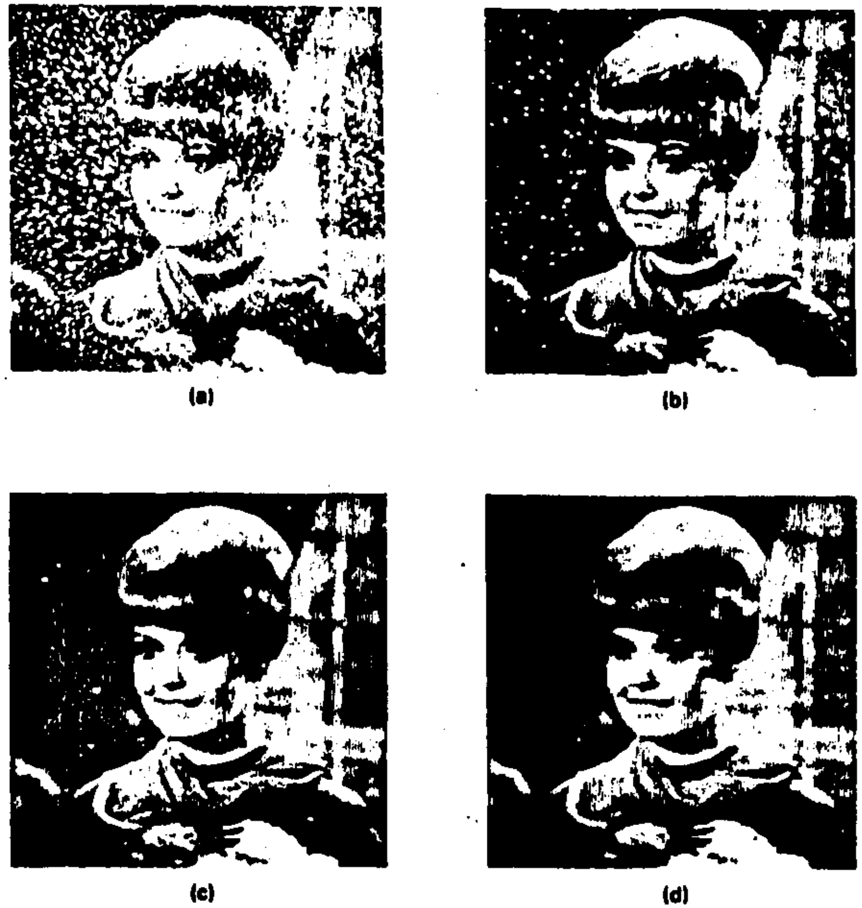


FIGURE 12.6-3. Examples of one-dimensional median filtering for images corrupted by impulse noise. (a) Image with impulse noise, 15 errors per line. (b) Median filtering of (a) with $L = 3$. (c) Median filtering of (a) with $L = 5$. (d) Median filtering of (a) with $L = 7$.

Range filters

Also for these filters the local histogram is computed for each pixel.

Instead of considering the median, we fix a different position in the local histogram and we pick up this value to be substituted to the central pixel of the sliding window.

Depending on the position we choose the filter properties vary.

Ex.1: Minimum of the histogram : EROSION

Ex. 2: Maximum of the histogram : DILATION