Image Enhancement

Digital Image Processing, Pratt

Chapter 10 (pages 243-261)

Part 1: pixel-based operations

Image Processing Algorithms

Spatial domain

- Operations are performed in the image domain
- Image ⇔ matrix of numbers
- Examples
  - luminance adaptation
  - chromatic adaptation
  - contrast enhancement
  - spatial filtering
  - edge detection
  - noise reduction

Transform domain

- Some operators are used to project the image in another space
- Operations are performed in the transformed domain
  - Fourier (DCT, FFT)
  - Wavelet (DWT,CWT)
- Examples
  - coding
  - denoising
  - image analysis

Most of the tasks can be implemented both in the image and in the transformed domain. The choice depends on the context and the specific application.

Spatial domain processing

Pixel-wise

- Operations involve the single pixel
- Operations:
  - histogram equalization
  - change of the colorspace
  - addition/subtraction of images
  - get negative of an image
- Applications:
  - luminance adaptation
  - contrast enhancement
  - *chromatic adaptation*

Local-wise

- The neighbourhood of the considered pixel is involved
  - Any operation involving digital filters is local-wise
- Operations:
  - correlation
  - convolution
  - filtering
  - transformation
- Applications
  - smoothing
  - sharpening
  - noise reduction
  - edge detection

Image enhancement

- Image enhancement processes consist of a collection of techniques that seek to improve the visual appearance of an image or to convert the image to a form better suited for analysis by a human or a machine.
- There is no general unifying theory of image enhancement at present because there is no general standard of image quality that can serve as a design criterion for an image enhancement processor.
  - Consideration is given here to a variety of techniques that have proved useful for human observation improvement and image analysis.
- [Pratt, Chapter 10]

Pixel-wise operations
Pratt Ch. 10

- Contrast enhancement
  - Amplitude scaling
  - Histogram straching/shrinking, sliding, equalization
- Contrast can often be improved by amplitude rescaling of each pixel

Contrast manipulation: Amplitude scaling

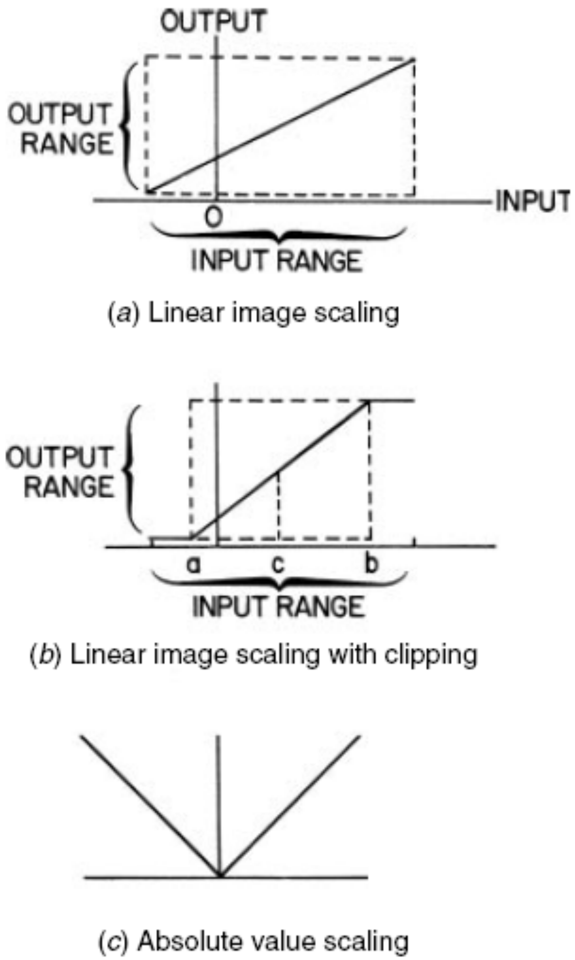


FIGURE 10.1-2. Image scaling methods.

In the case (a) the processed image is linearly mapped over its entire range, while by the second technique, (b), the extreme amplitude values of the processed image are clipped to maximum and minimum limits.

Window-level transformation. The window value is the width of the linear slope; the level is located at the midpoint *c* of the slope line. Very common in medical imaging.

The third technique of amplitude scaling, shown in Figure 10.1-2**c** , utilizes an absolute value transformation for visualizing an image with negatively valued pixels.

Amplitude scaling

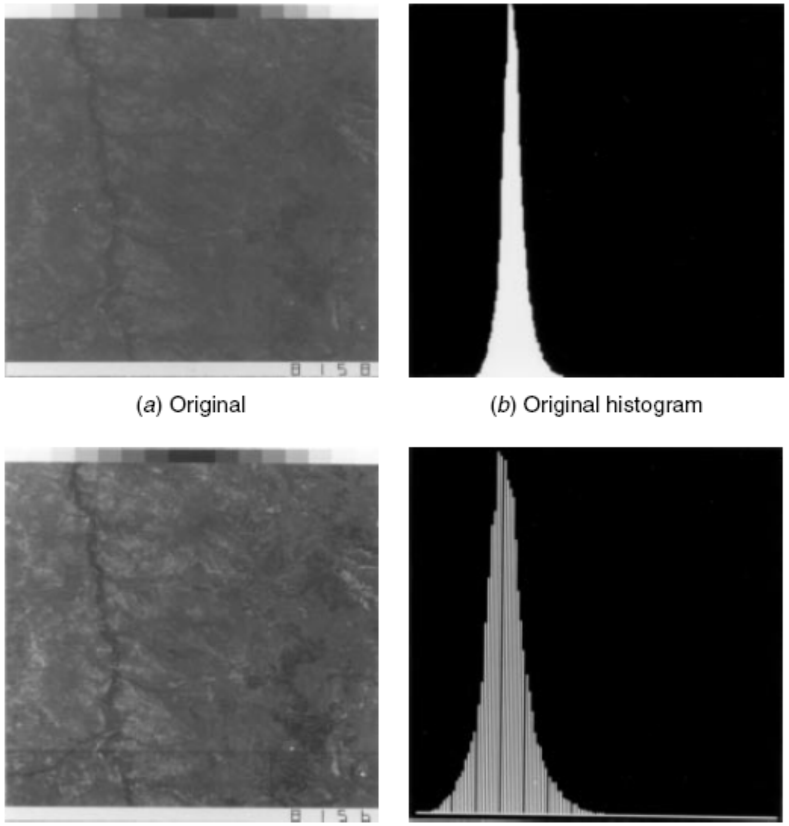
Q component of a YIQ image representation.

(*a*) Linear, full range, – 0.147 to 0.169

(*b*) Clipping, 0.000 to 0.169

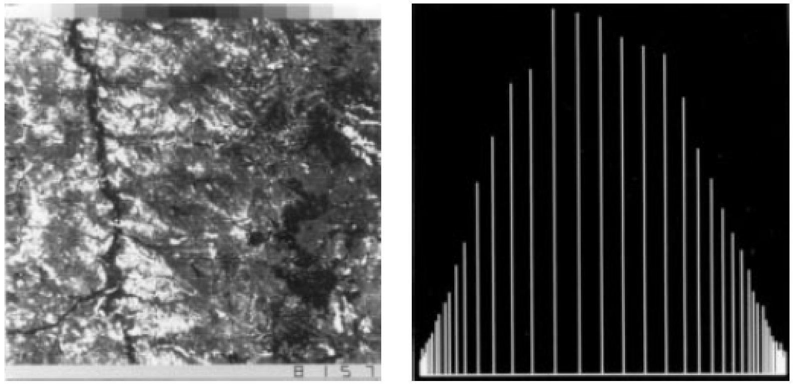
(*c*) Absolute value, 0.000 to 0.169

Window level transformation: ex.



(*a*) Original (*b*) Original histogram

(*c*) Min. clip = 0.17, max. clip = 0.64 (*d*) Enhancement histogram



(*e*) Min. clip = 0.24, max. clip = 0.35 (*f*) Enhancement histogram

FIGURE 10.1-4. Window-level contrast stretching of an earth satellite image.

In Figure 10.1-4**c** , the clip levels are set at the histogram limits of the original, while in Figure 10.1-4**e** , the clip levels truncate 5% of the original image upper and lower level amplitudes. It is readily apparent from the histogram of Figure 10.1-4**f**
that the contrast-stretched image of Figure 10.1-4**e** has many unoccupied amplitude levels.

Contrast enhancement via graylevel transf.

- Point transformations that modify the contrast of an image within a display's dynamic range
- Often nonlinear point transformations
- Power law point transformations

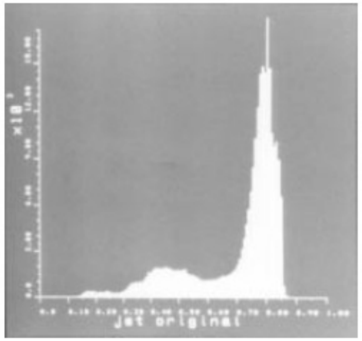
$$G[j,k] = (F[j,k])^p$$

$$0 \le F[j,k] \le 1$$

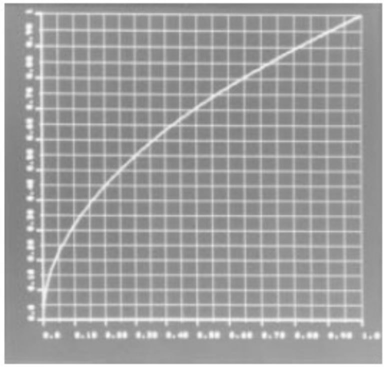
$$p: \text{ power law varaible}$$

example

original

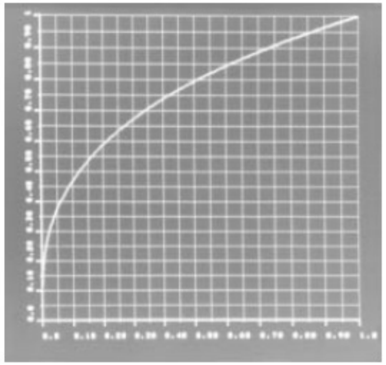


(*b*) Original histogram



(*a*) Square root function

(*b*) Square root output

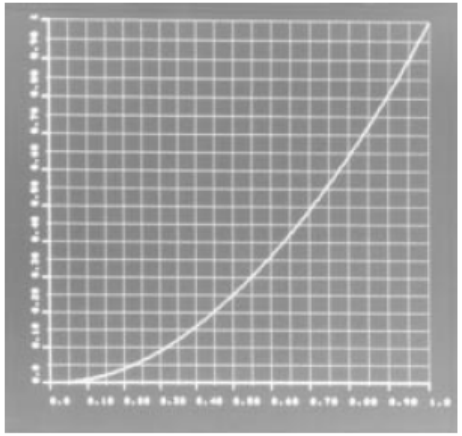


(*c*) Cube root function

(*d*) Cube root output

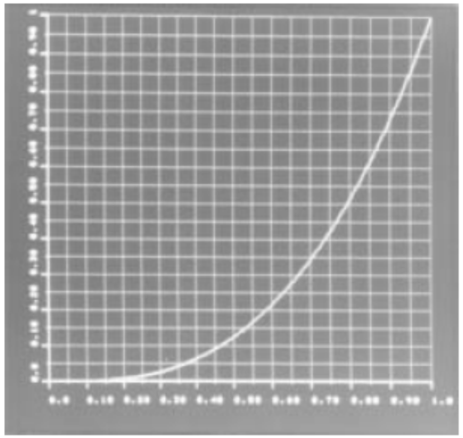
Figure 10.1-5a contains an original image of a jet aircraft that has been digitized to 256 gray levels and numerically scaled over the range of 0.0 (black) to 1.0 (white). Examination of the histogram of the image reveals that the image contains relatively few low- or highamplitude pixels. Consequently, applying the window-level contrast stretching function of Figure 10.1-5c results in the image of Figure 10.1-5d , which possesses better visual contrast but does not exhibit noticeable visual clipping.

Contrast enhancement



(*a*) Square function

(*b*) Square output



(*c*) Cube function

(*d*) Cube output

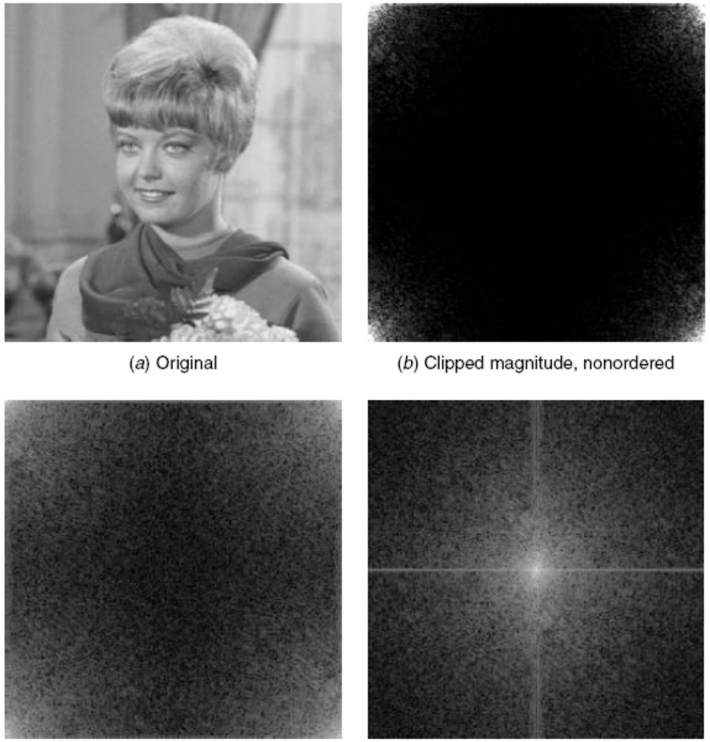
FIGURE 10.1-6. Square and cube contrast modification of the `jet_mon` image.

log amplitude scaling

- The logarithm function is useful for scaling image arrays with a very wide dynamic range.

$$G(j,k) = \frac{\log_e\{1.0 + aF(j,k)\}}{\log_e\{2.0\}}$$

a>0



(a) Original (b) Clipped magnitude, nonordered (c) Log magnitude, nonordered (d) Log magnitude, ordered

Reverse and Inverse functions

- Reverse function

$$G[i,k] = (1 - F[i,k])$$
$$0 \le F[i,k] \le 1$$

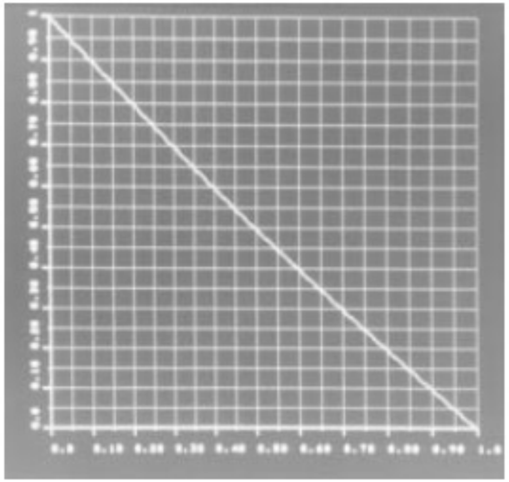
- Contrast reverse and contrast inverse transfer functions, as illustrated in Figure 10.1-9, are often helpful in visualizing detail in dark areas of an image.

- Inverse function

$$G(j,k) = \begin{cases} 1.0 & \text{for } 0.0 \le F(j,k) < 0.1 \\ \dfrac{0.1}{F(j,k)} & \text{for } 0.1 \le F(j,k) \le 1.0 \end{cases}$$

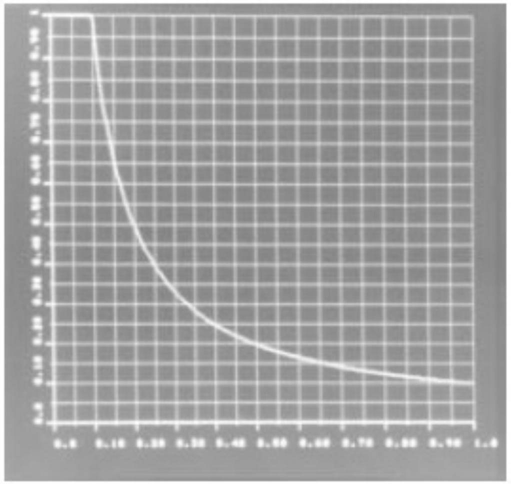
clipped below 0.1 to maintain the range (max value=1)

example



(*a*) Reverse function

(*b*) Reverse function output

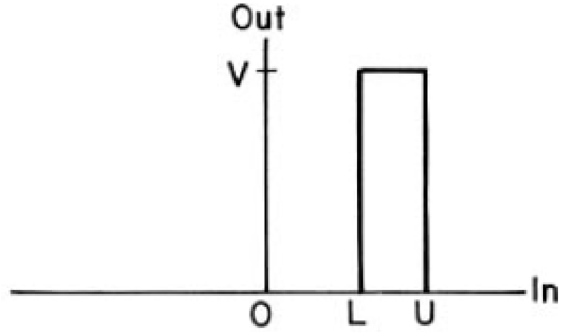


(*c*) Inverse function

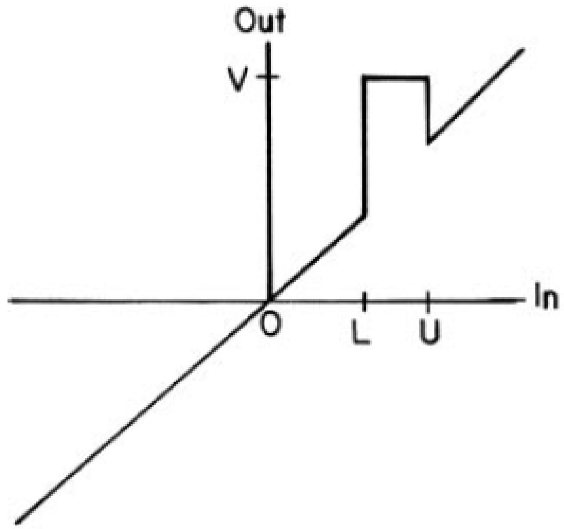
(*d*) Inverse function output

Level slicing

- Amplitude-level slicing, as illustrated ir Figure 10.1-10, is a useful interactive tool for visually analyzing the spatia distribution of pixels of certair amplitude within an image.
- With the function of Figure 10.1-10**a** all pixels within the amplitude passband are rendered maximum white in the output, and pixels outside the passband are rendered black.
- Pixels outside the amplitude passbanc are displayed in their original state witt the function of Figure 10.1-10**b** .
-



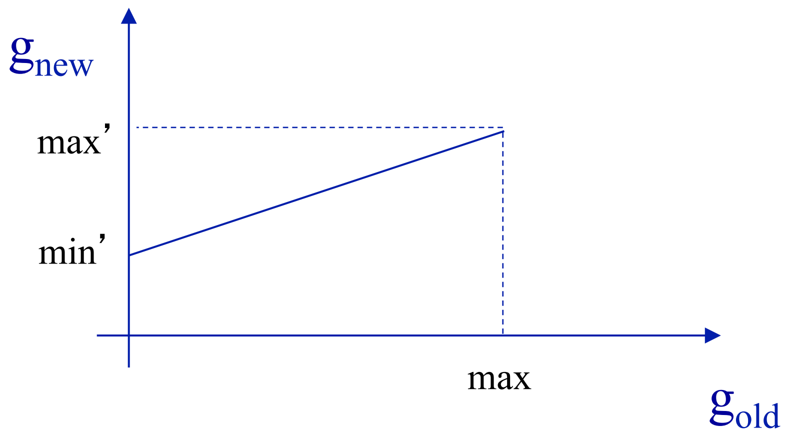
(a) Zero background scaling transformation

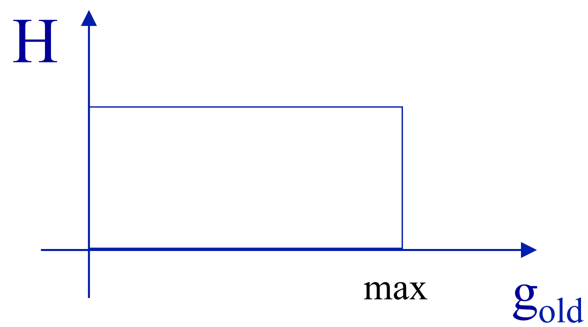


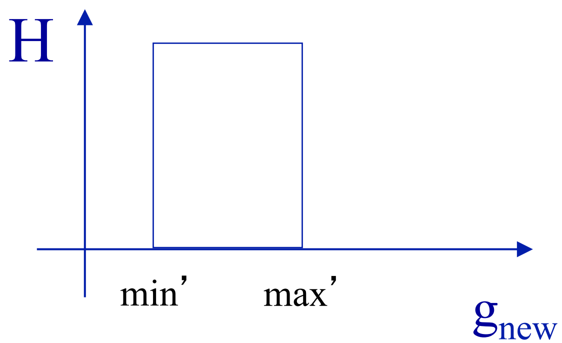
(b) Image background scaling transformation

Histogram changes

- Graylevel transformations induce histogram changes



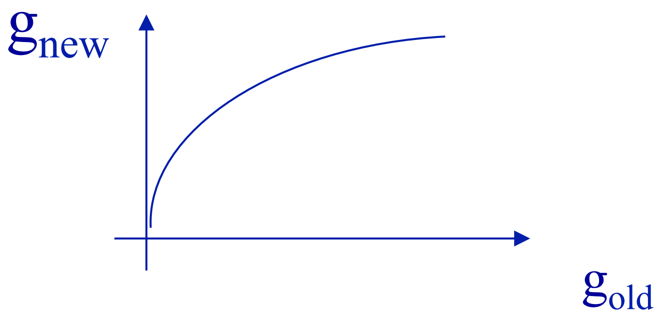


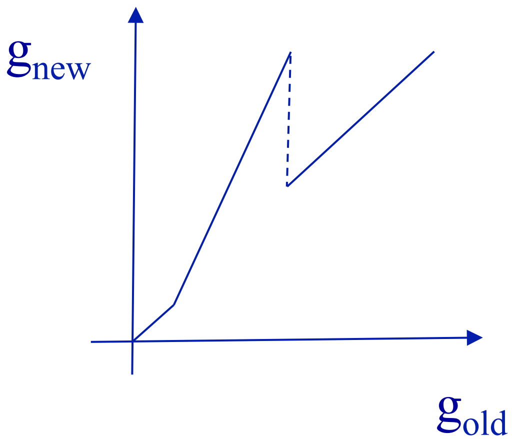


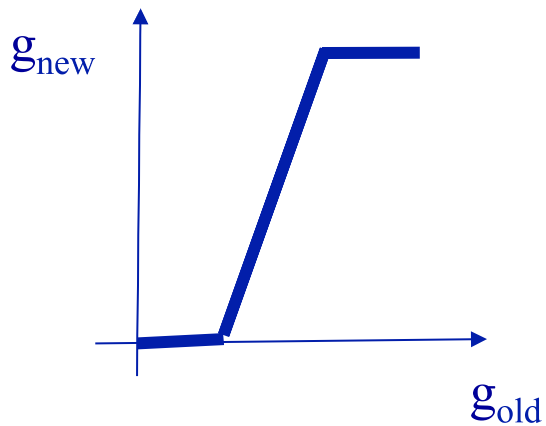
Other non-linear transformations

- Used to emphasize mid-range levels

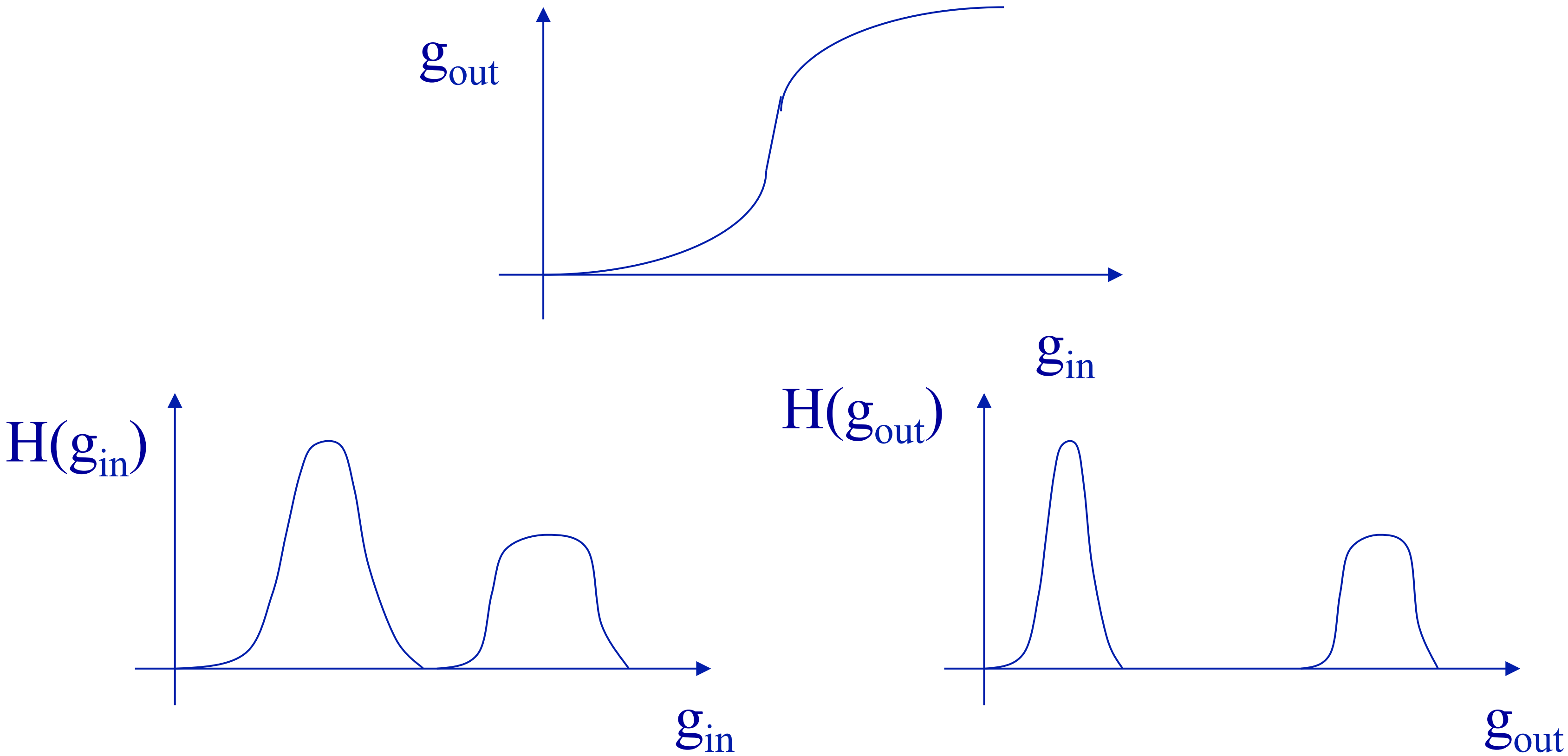
$g_{new} = g_{old} + g_{old}\, C\, (g_{old,max} - g_{old})$





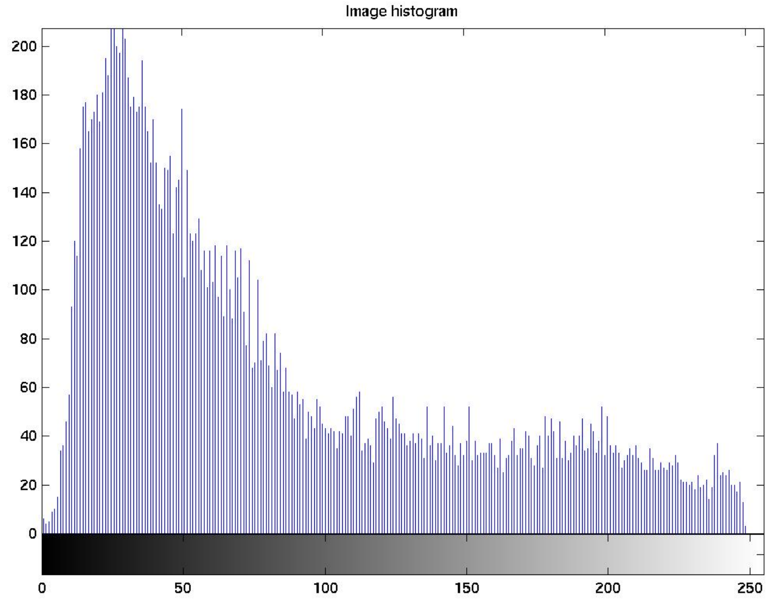


Sigmoid transformation (*soft thresholding*)



Pixel-wise: Histogram equalization

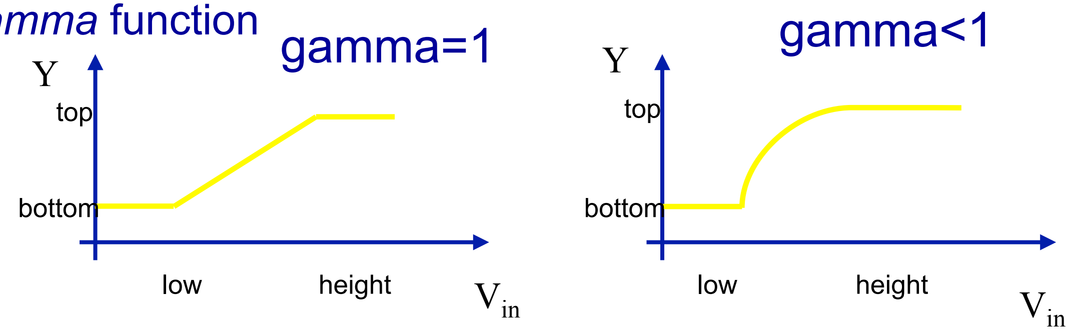
- Pixel features: luminance, color,
- Histogram equalization: shapes the intensity histogram to approximate a specified distribution
  - It is often used for enhancing contrast by shaping the image histogram to a uniform distribution over a given number of grey levels. The grey values are redistributed over the dynamic range to have a constant number of samples in each interval (i.e. histogram bin).
  - Can also be applied to colormaps of color images.

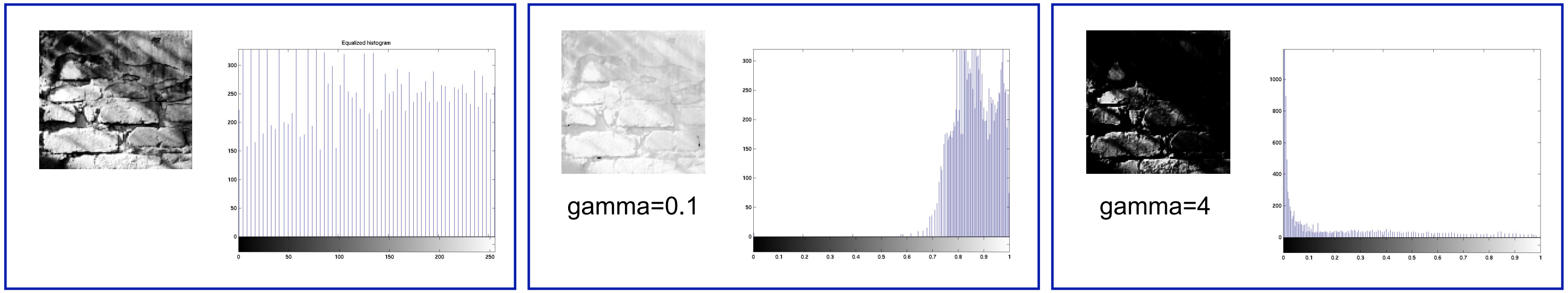


Histogram equalization

Can be used to compensate the distortions in the gray level distribution due to the non-linearity of a system component

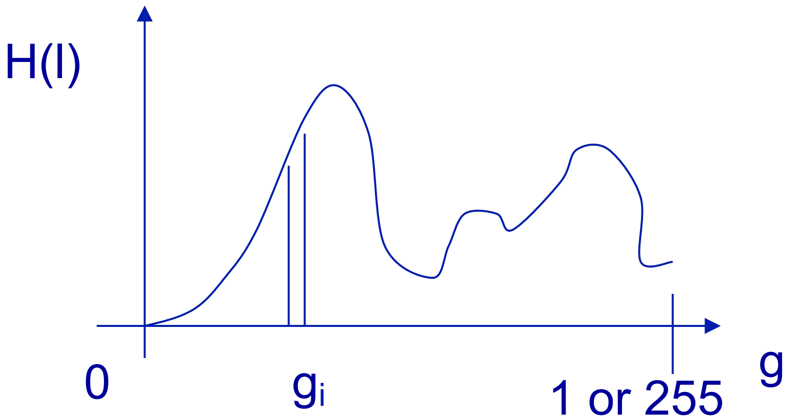
Gamma function





Histogram

- Function H=H(g) indicating the number of pixels having gray-value equal to g
  - Non-normalized images: 0≤g ≤255 → bin-size≥1, can be integer
  - Normalized images: 0≤g ≤1 → bin-size<1



$$A = \int_0^{max} H(g)dg$$

area under the curve=number of pixels

$$A = \sum_{i=1}^{N_g} H[g_i]$$

In the continuous case

$$H(g) = -\frac{dA(g)}{dg} = \lim_{\Delta g \to 0} \frac{A(g) - A(g + \Delta g)}{\Delta g}$$

Histogram transformation

$$g_{out} = f(g_{in}) \Rightarrow g_{in} = f^{-1}(g_{out}), \quad f \quad \text{non-decreasing function}$$

$$H(g_{in}) \Rightarrow H(g_{out}), \quad \text{namely}$$

$$H(g_{out}) = \frac{H[f^{-1}(g_{out})]}{f'[f^{-1}(g_{out})]}, \quad f' = \frac{\partial f}{\partial g}$$

More formally

- The histogram modification process can be considered to be a monotonic point transformation $g_d = T\{f_c\}$ for which the input amplitude variable $f_1 \leq f_c \leq f_C$ is mapped into an output variable $g_1 \leq g_d \leq g_D$ such that the output probability distribution $Pr\{g_d = b_d\}$ follows some desired form for a given input probability distribution $Pr\{f_c = a_c\}$ where a_c and b_d are reconstruction values of the c^{th} and d^{th} levels.
  - Clearly, the input and output probability distributions must each sum to unity.

$$\sum_{c=1}^{C} P_R\{f_c = a_c\} = 1$$

$$\sum_{d=1}^{D} P_R\{g_d = b_d\} = 1$$

NB: C and D are caps!

Histogram equalization

– Furthermore, the cumulative distributions must equate for any input index *c*.

- the probability that pixels in the input image have an amplitude less than or equal to a_c must be equal to the probability that pixels in the output image have amplitude less than or equal to b_d, where $b_d=T\{a_c\}$ because the transformation is monotonic. Hence

$$\sum_{n=1}^{d} P_R\{g_n = b_n\} = \sum_{m=1}^{c} P_R\{f_m = a_m\} \qquad \text{(a)}$$

cumulative probability distribution of the output

cumulative probability distribution of the input

$$P_f(f) \approx \sum_{m=1}^{c} H_F(m)$$

histogram

- in the continuous domain (easier for calculations)

$$\int_{g_{\min}}^{g} p_g(g)\,dg = \int_{f_{\min}}^{f} pf(f)\,df$$

$p_f(f)$ and $p_g(g)$ are the probability densities of f and g

NB: c and d are lowercase!

Histogram equalization

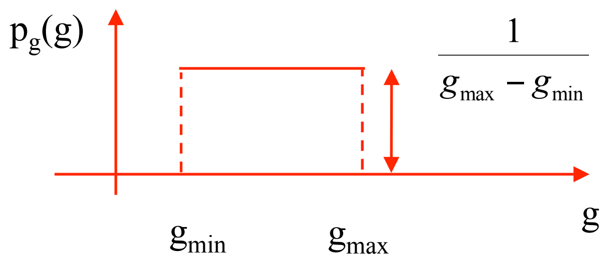
cumulative probability distribution of the output

(a) $$\sum_{n=1}^{d} P_R\{g_n = b_n\} = \sum_{m=1}^{c} H_F(m)$$ $$\int_{g_{min}}^{g} p_g(g)\, dg = P_f(f)$$ (b)

cumulative histogram

When the output density is forced to be the uniform density

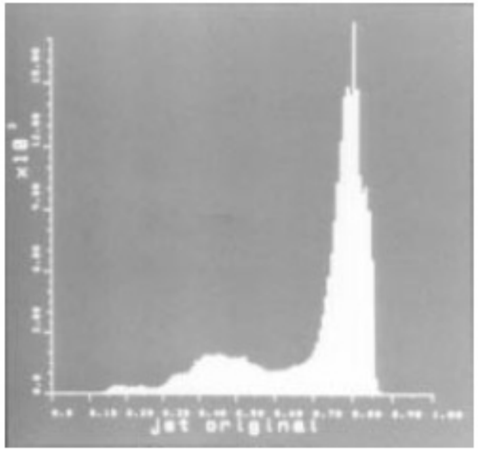
$$p_g(g) = \frac{1}{g_{max} - g_{min}} \quad \text{(area=1)}$$



Solving (b) for g we get the histogram equalization transfer function:

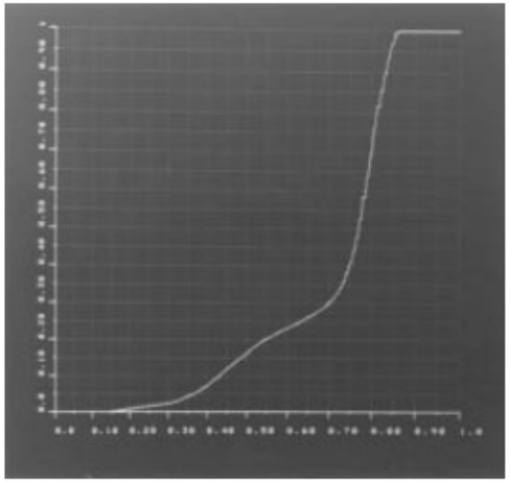
$$g = (g_{max} - g_{min}) P_f(f) + g_{min}$$

example



(*b*) Original histogram

(*a*) Original



(*b*) Transfer function

(*c*) Histogram equalized

Some mappings

TABLE 10.2-1. Histogram Modification Transfer Functions

Output Probability Density Model		Transfer Function[a]
Uniform	$p_g(g) = \dfrac{1}{g_{max} - g_{min}} \quad g_{min} \le g \le g_{max}$	$g = (g_{max} - g_{min}) P_f(f) + g_{min}$
Exponential	$p_g(g) = \alpha \exp\{-\alpha(g - g_{min})\} \quad g \le g_{min}$	$g = g_{min} - \dfrac{1}{\alpha} \ln\{1 - P_f(f)\}$
Rayleigh	$p_g(g) = \dfrac{g - g_{min}}{\alpha^2} \exp\left\{-\dfrac{(g - g_{min})^2}{2\alpha^2}\right\} \quad g \ge g_{min}$	$g = g_{min} + \left[2\alpha^2 \ln\left\{\dfrac{1}{1 - P_f(f)}\right\}\right]^{1/2}$
Hyperbolic (Cube root)	$p_g(g) = \dfrac{1}{3} \dfrac{g^{-2/3}}{g_{max}^{1/3} - g_{min}^{1/3}}$	$g = \left[g_{max}^{1/3} - g_{min}^{1/3} [P_f(f)] + g_{max}^{1/3}\right]^3$
Hyperbolic (Logarithmic)	$p_g(g) = \dfrac{1}{g[\ln\{g_{max}\} - \ln\{g_{min}\}]}$	$g = g_{min} \left(\dfrac{g_{max}}{g_{min}}\right)^{P_f(f)}$

[a] The cumulative probability distribution $P_f(f)$, of the input image is approximated by its cumulative histogram:

$$p_f(f) \approx \sum_{m=0}^{j} H_F(m)$$

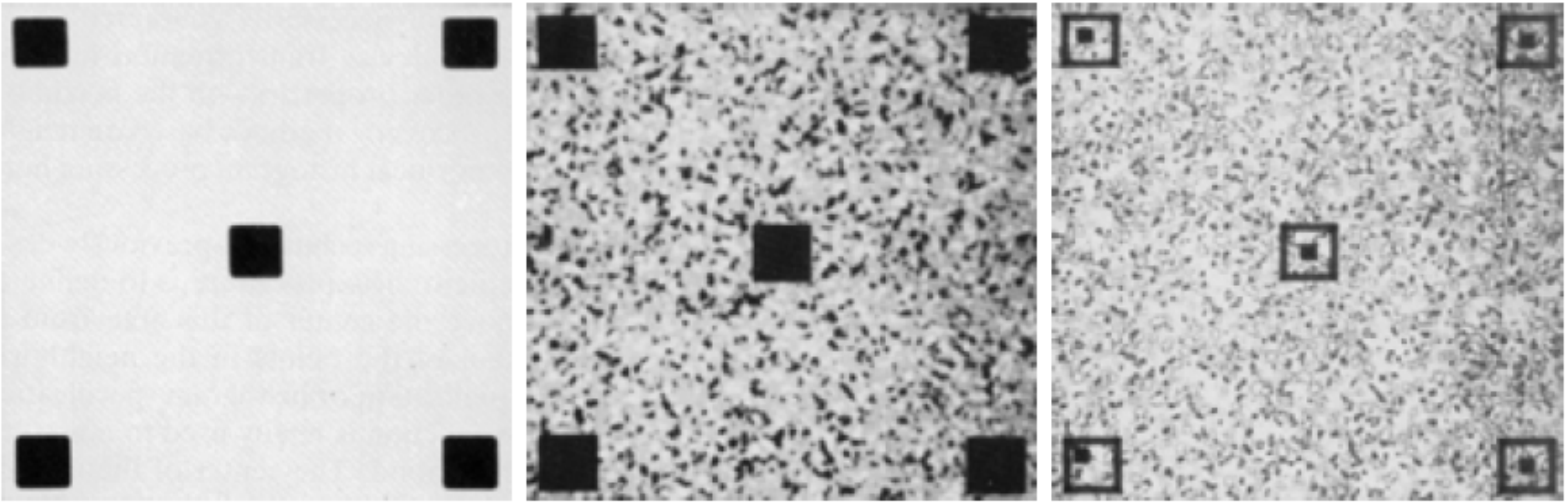
Adaptive hist. equalization (GZ Ch3)

- The histogram processing methods discussed in the previous two sections are global , in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image. Although this global approach is suitable for overall enhancement, there are cases in which it is necessary to enhance details over small areas in an image.
- The number of pixels in these areas may have negligible influence on the computation of a global transformation whose shape does not necessarily guarantee the desired local enhancement.
- The solution is to devise transformation functions based on the gray-level distribution- or other properties— in the neighborhood of every pixel in the image.

Algorithm

- The procedure is to define a square or rectangular neighborhood and move the center of this area from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained.
- This function is finally used to map the gray level of the pixel centered in the neighborhood.
- The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.

Example



a b c

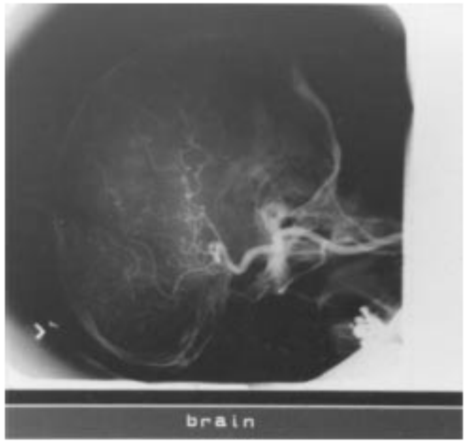
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7 × 7 neighborhood about each pixel.

Note that no new structural details were brought out by this method. However, local histogram equalization using a 7* 7 neighborhood revealed the presence of small squares inside the larger dark squares.The small squares were too close in gray level to the larger ones, and their sizes were too small to influence global histogram equalization significantly.

Adaptive hist. equalization (GZ Ch3)

- The mapping function can be made *spatially adaptive* by applying histogram modification to each pixel based on the histogram of pixels *within a moving window neighborhood*.
  - This technique is obviously computationally intensive, as it requires histogram generation, mapping function computation, and mapping function application at each pixel.
  - Some interpolation-based solutions can be envisioned to improve computational efficiency

example

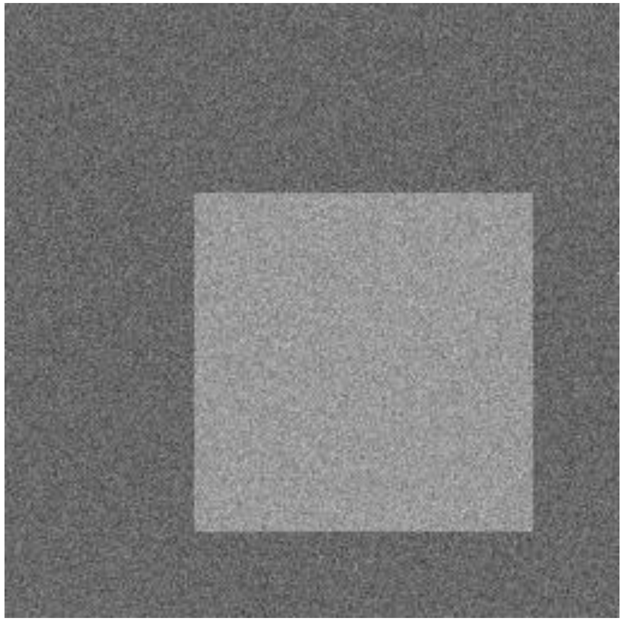


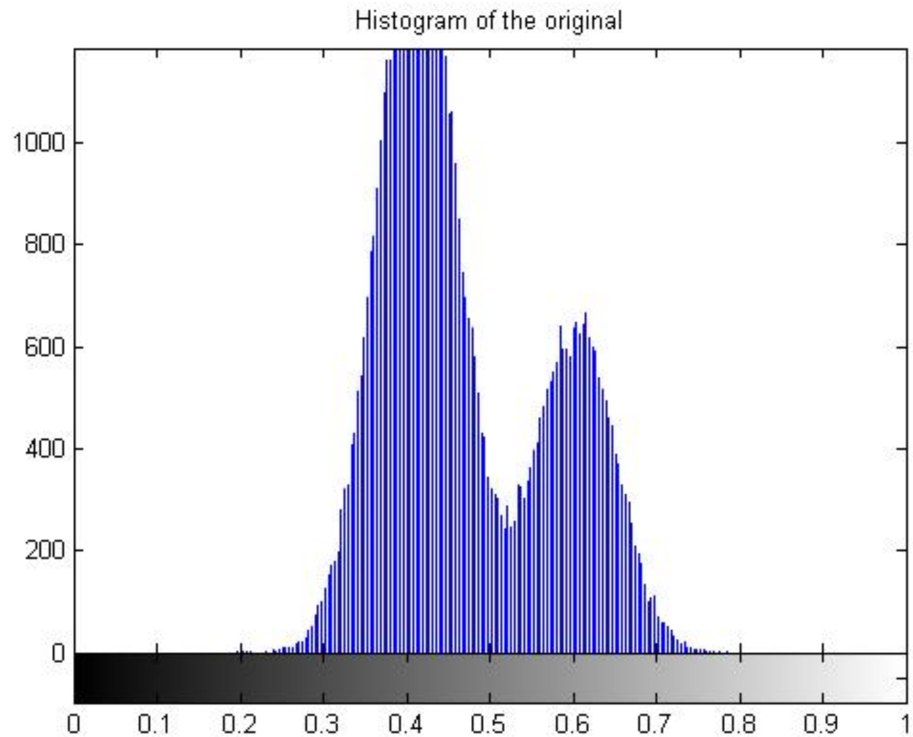
(*a*) Original

(*b*) Nonadaptive

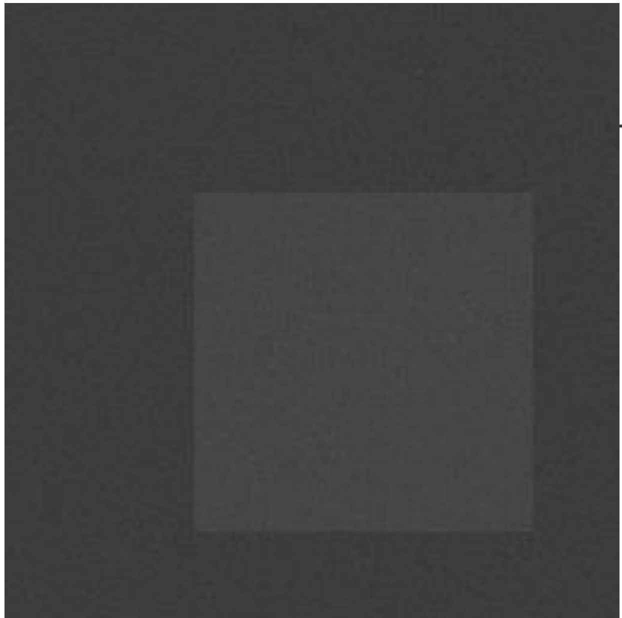
(*c*) Adaptive

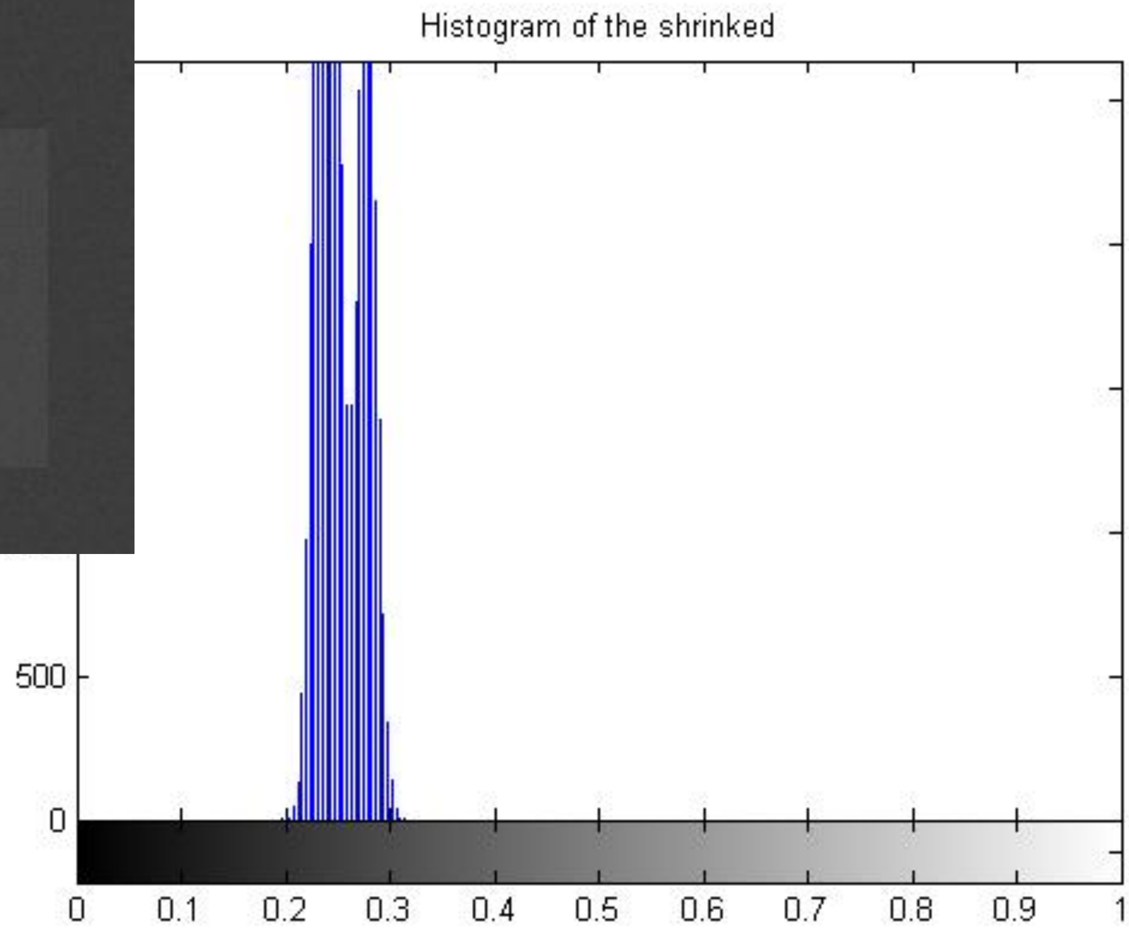
H. original



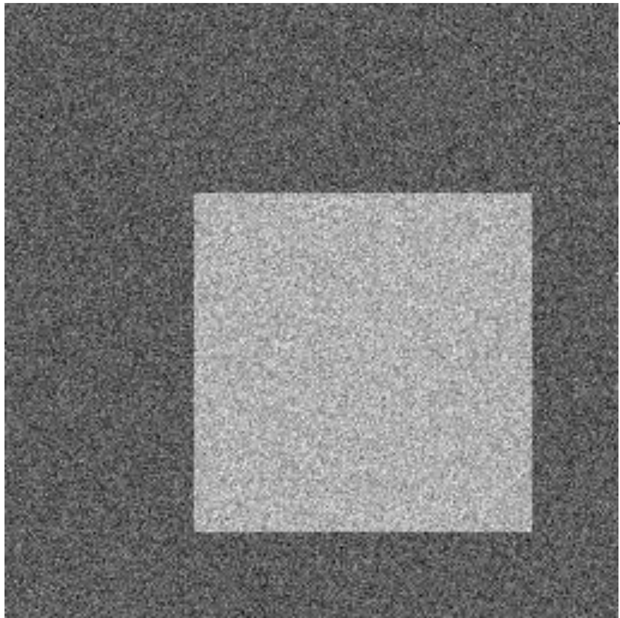

Histogram of the original
1000
800
600
400
200
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

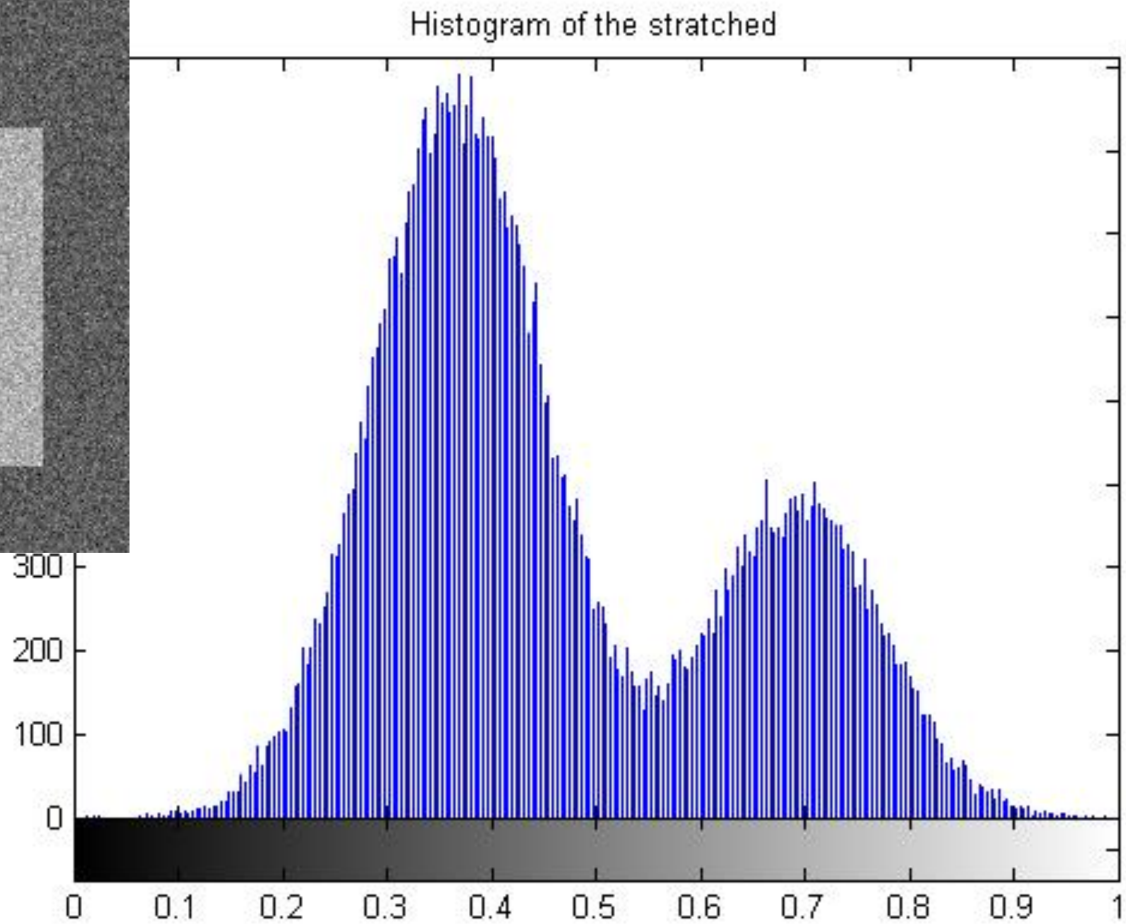
H. shrinked



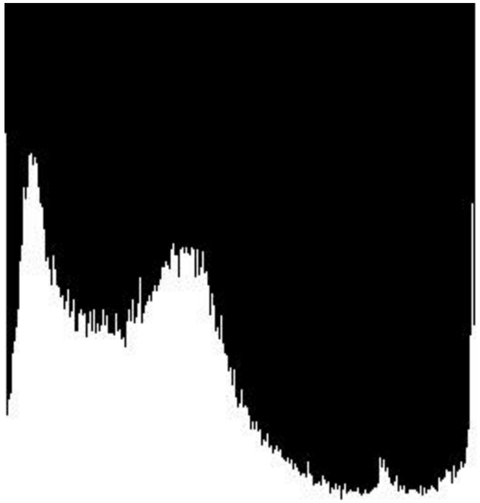


H. stratched



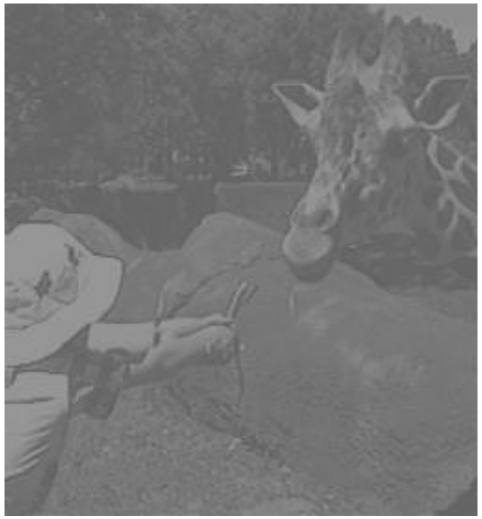

Histogram of the stratched
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

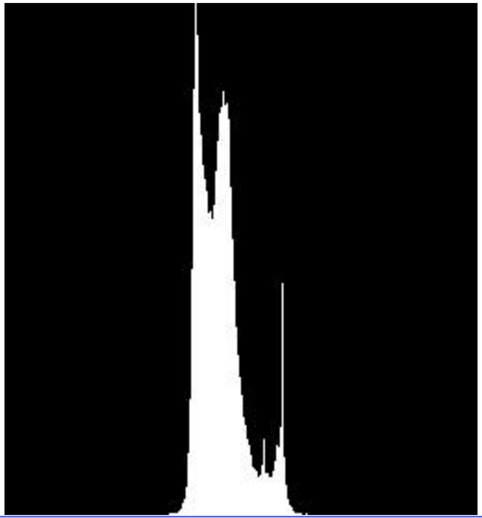
H. stratching/shrinking



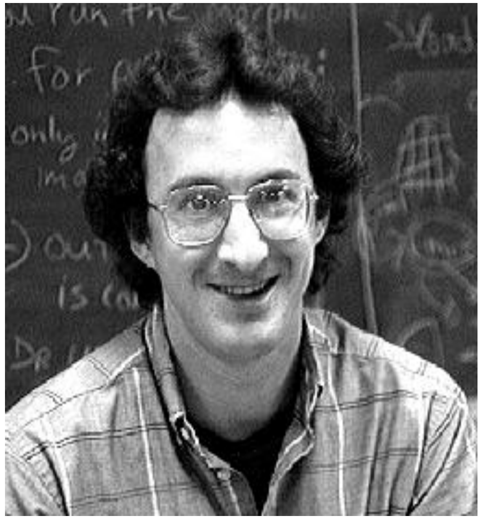
stratching

shrinking





H. stratching/shrinking



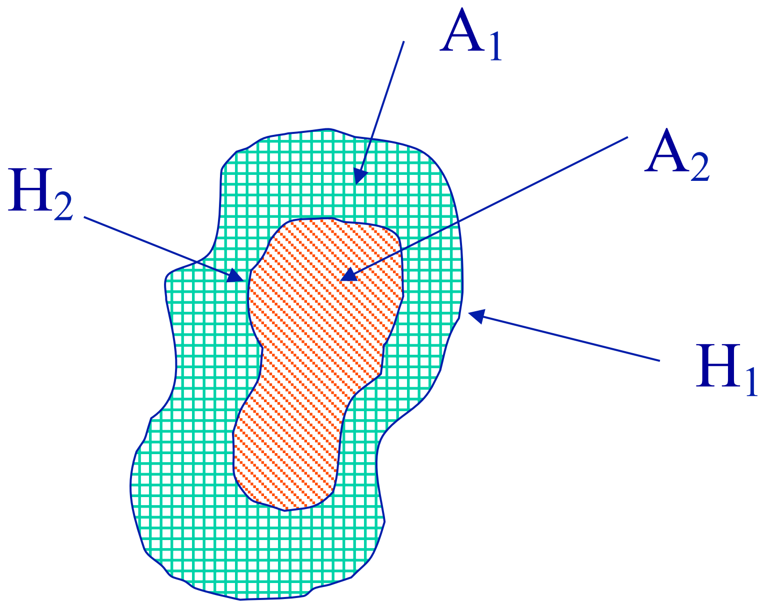






Example: region-based segmentation

- If the two regions have different graylevel distributions (histograms) then it is possible to split them by exploiting such an information



Example: region-based segmentation