Image Enhancement

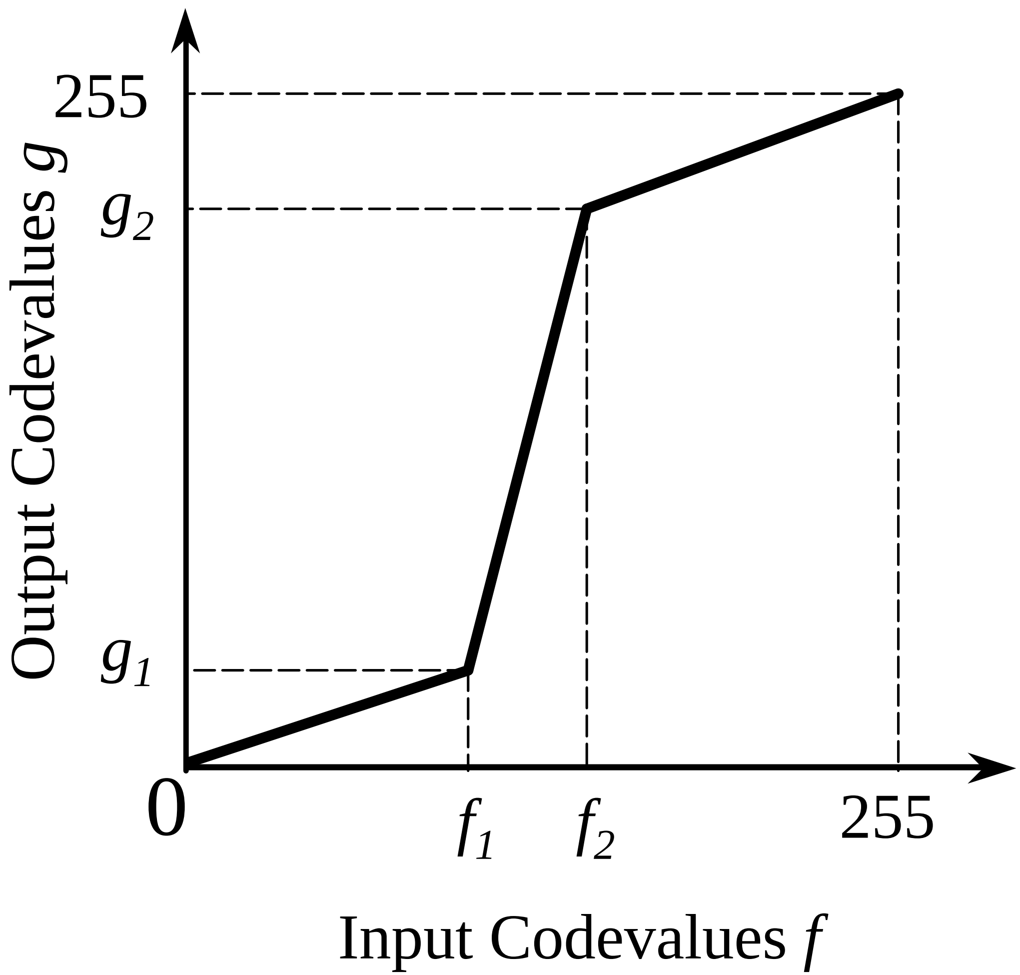
- The goal of image enhancement is to improve the usefulness of an image for a given task, such as providing a more subjectively pleasing image for human viewing.
- In image enhancement, little or no attempt is made to estimate the actual image degradation process, and the techniques are often *ad hoc*. Image enhancement usually involves:
  - Contrast manipulation
  - Sharpening
  - Noise reduction

Contrast Manipulation

Contrast Manipulation

- One of the most common defects of photographic or electronic images is poor contrast resulting from either
  - Poor lighting conditions (e.g., an underexposed or an overexposed scene), or
  - Sensor (capture medium) nonlinearity or small dynamic range compared to the dynamic range of the captured scene (e.g., back-lit scene).
- Image contrast can often be improved by a look-up table (LUT) operation that rescales the amplitude of each pixel. For example, in an underexposed image with the histogram centered around small codevalues, a LUT that stretches that range can result in the desired image enhancement.

Tone-Scale Look-Up Tables (LUT)



Input f	Output g
0	0
1	0
2	1
3	1
4	2
…	…
…	…
80	40
81	44
82	48
83	52
…	…
…	…
252	254
253	254
254	255
255	255

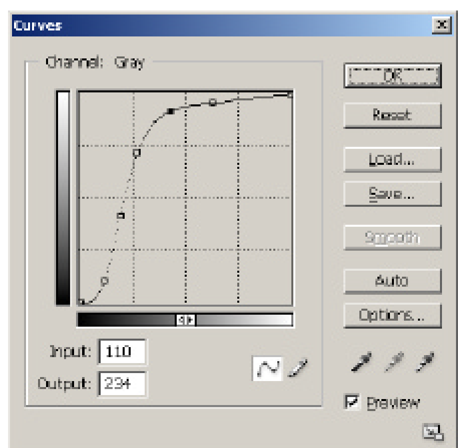

Original 512x512 8-bit

Photoshop “Curves”

Example:

Applying a LUT using Photoshop “Curves”

Input



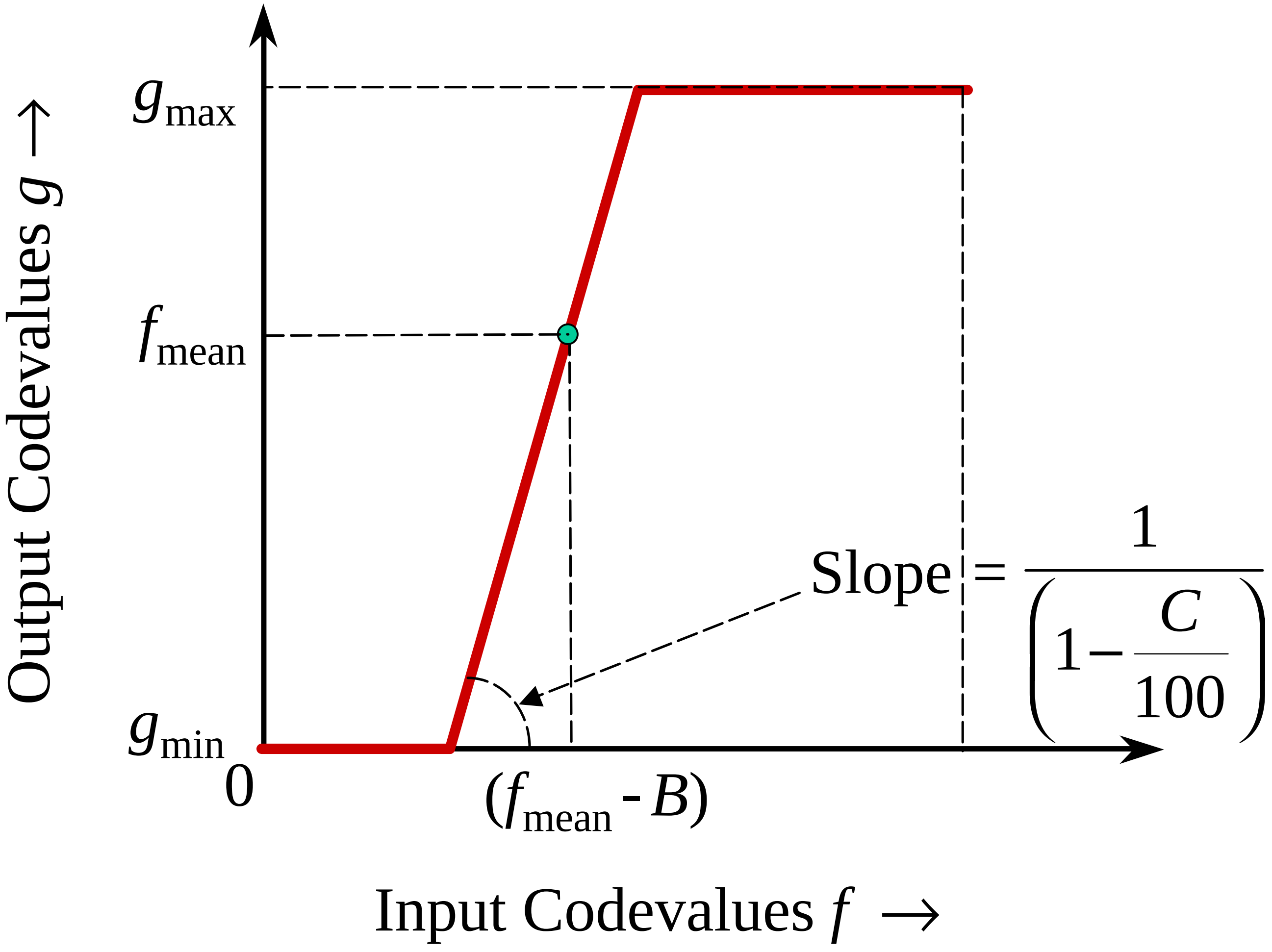
Output

Photoshop Brightness/Contrast Adjustment

- First, the average brightness of the input image, f_{mean}, is calculated from the histogram.
- For a brightness parameter B and a contrast parameter C, a linear transformation is constructed that maps the input codevalue $f = (f_{\text{mean}} - B)$ to the output codevalue $g = f_{\text{mean}}$ with an inverse slope of $(1 - C/100)$.
- The output is converted to an integer and its range is clipped to $(g_{\text{min}}, g_{\text{max}})$, e.g., 0 - 255 for an 8-bit image.

$$g = \frac{\left(f - f_{\text{mean}} + B\right)}{1 - \left(\dfrac{C}{100}\right)} + f_{\text{mean}}$$

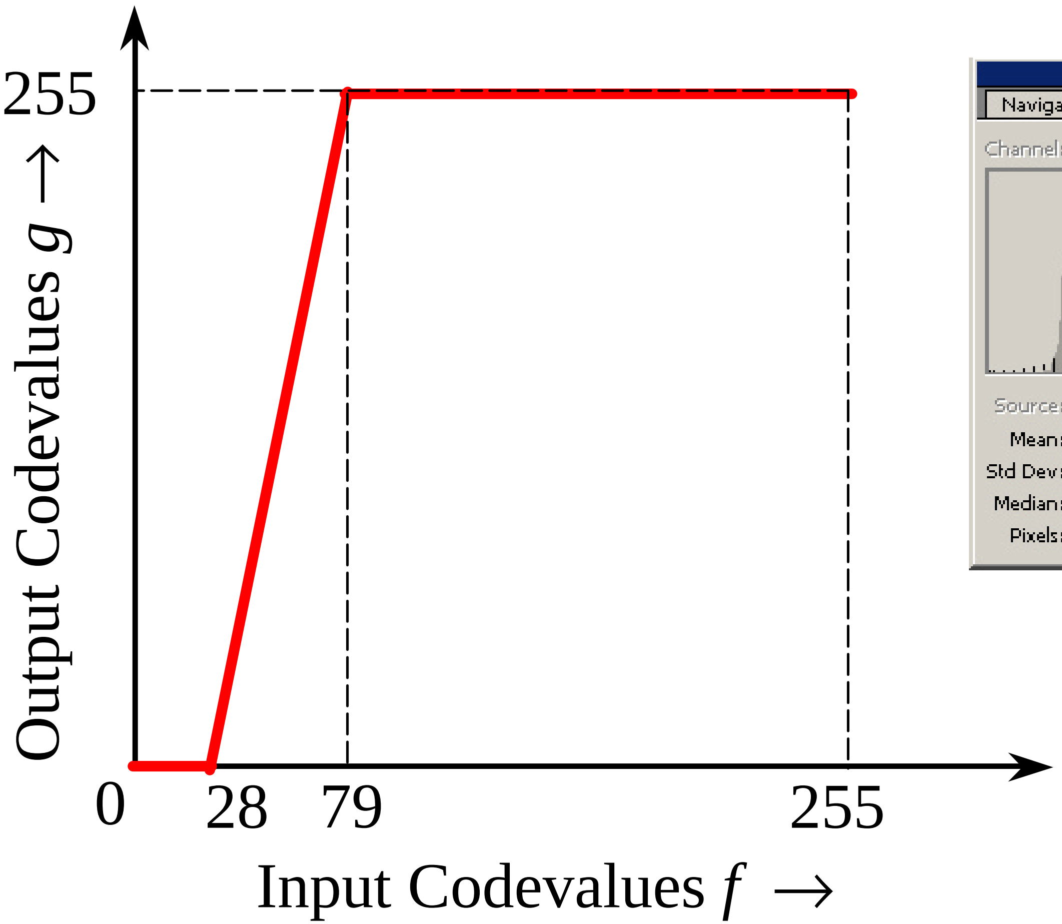
Photoshop Brightness/Contrast Adjustment

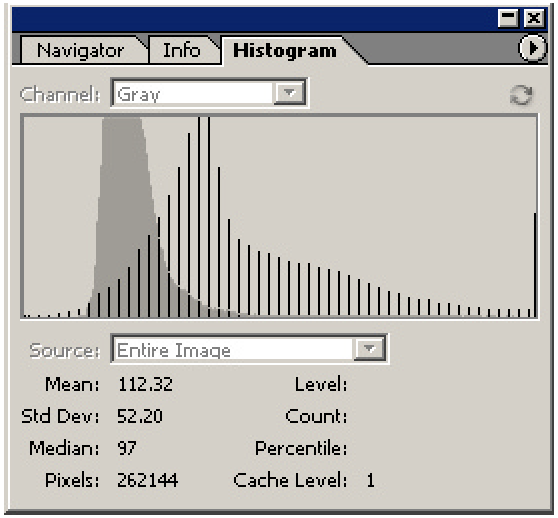



Original 512x512 8-bit

Photoshop Brightness/Contrast, B=13, C=80

“Brightness/Contrast” LUT for Aerial Image





<u>Example:</u>

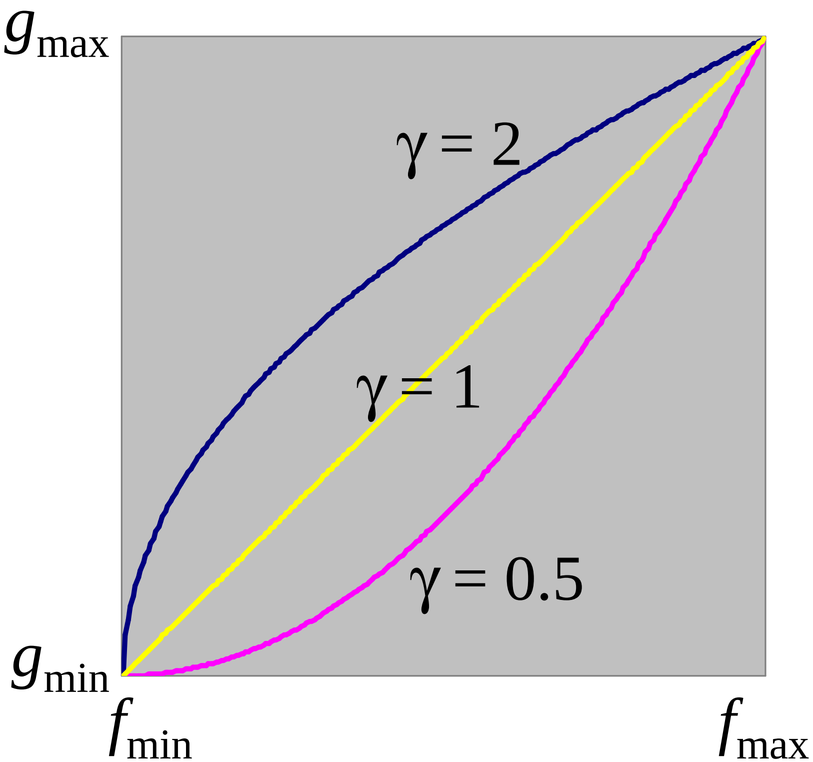
B = 13

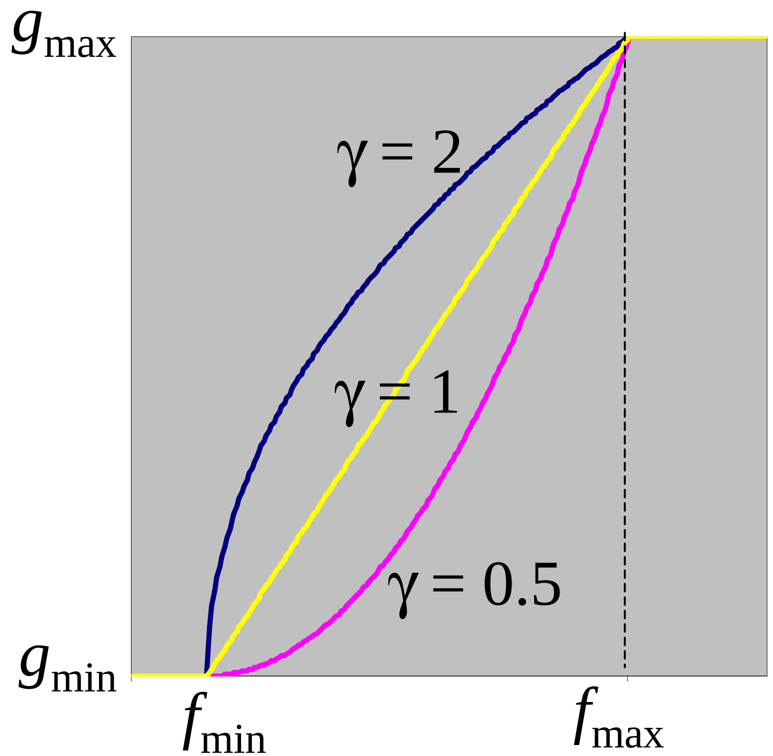
C = 80

$f_{\text{mean}} = 52$

"Levels" in Photoshop

- The input range of ($f_{\min}$, $f_{\max}$) is mapped to the output range of ($g_{\min}$, $g_{\max}$) with a contrast value of γ. All parameters are user-defined.





“Levels” in Photoshop

$$g = g_{\min} + \alpha\left\{\left(f - f_{\min}\right)\left(\frac{g_{\max} - g_{\min}}{f_{\max} - f_{\min}}\right)\right\}^{1/\gamma}$$

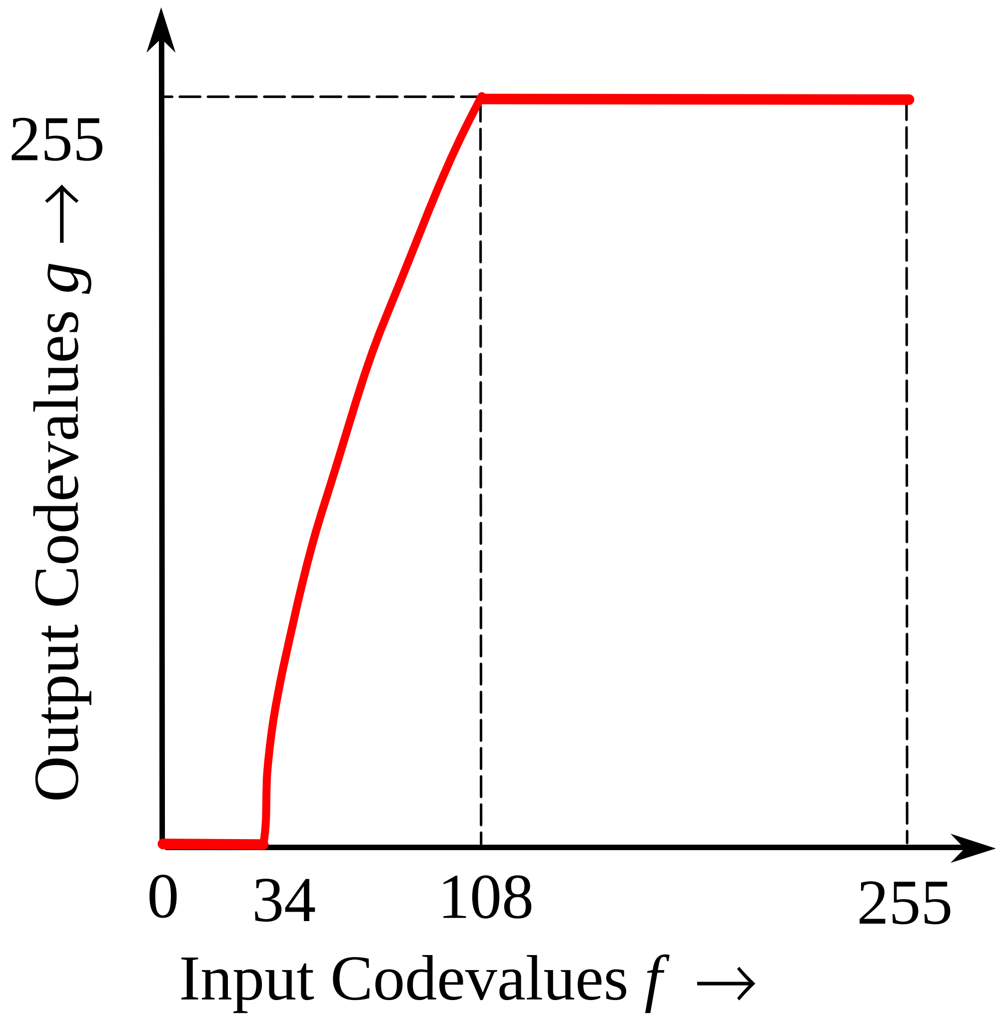
where $\alpha = \left(g_{\max} - g_{\min}\right)^{1 - 1/\gamma}$

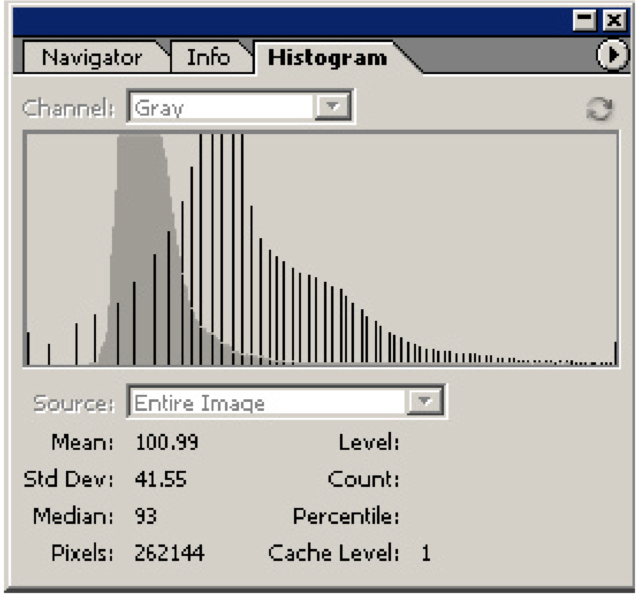
- All values of $f < f_{\min}$ are clipped to $g_{\min}$ in the output, while all values of $f > f_{\max}$ are clipped to $g_{\max}$.
- Values of $\gamma > 1$ stretch the contrast of the shadow regions (codevalues less than the midtone of 128), while values of $\gamma < 1$ stretch the contrast of the highlight regions (values larger than128). For $\gamma = 1$, the mapping is linear.


Original 512x512 8-bit

Photoshop Levels, $f_{\min}=34$, $f_{\max}=108$, $\gamma=1.65$

"Levels" Example LUT For Aerial Image





Example:

$f_{min} = 34$

$f_{max} = 108$

$\gamma = 1.65$

Automatic Tone-Scale Modification

- In order to improve the system throughput and reduce costs, it is desirable to perform the tone-scale modification automatically.

- In general, the outcome of an automatic tone-scale operation will not be as robust as that of a manual one.

- Most automatic tone-scale operations are histogram based. Examples are:

    - **Auto Levels** as used in Photoshop.

    - **Histogram Equalization** as available in most image editing software packages.

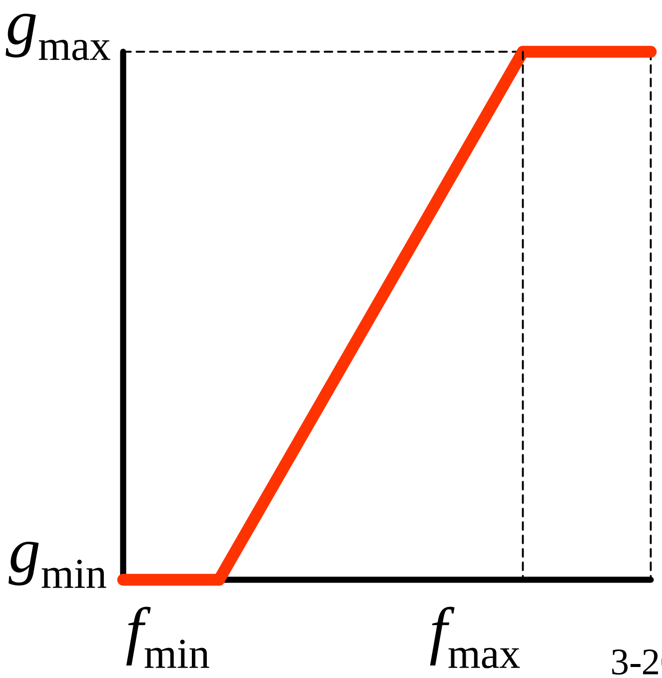
“Auto Levels” in Photoshop

- The two codevalues $f_{\min}$ and $f_{\max}$ in the original image are determined such that x% of the pixels have a codevalue less than or equal to $f_{\min}$ while y% of the pixels have a codevalue larger than or equal to $f_{\max}$.
- The values of x and y are defined by the user and correspond, respectively, to the clip points for the dark regions (shadows) and the bright regions (highlights).
- For example, for Photoshop 7.0 the default values are $x = y = 0.50$, hence the range of ($f_{\min}$, $f_{\max}$) contains 99% of the input pixels. In Photoshop CS, the default values are $x = y = 0.10$. The default values can be changed by the user.

"Auto Levels" in Photoshop

- The input range of (f_{min}, f_{max}) is **linearly** mapped to the output range of (g_{min}, g_{max}), where g_{min} and g_{max} are the minimum and maximum values of the output g (usually 0 to 255 for an 8-bit image), i.e., for each input pixel codevalue f, the output codevalue g is given by:

$$g = g_{\min} + \frac{(f - f_{\min})(g_{\max} - g_{\min})}{(f_{\max} - f_{\min})}$$

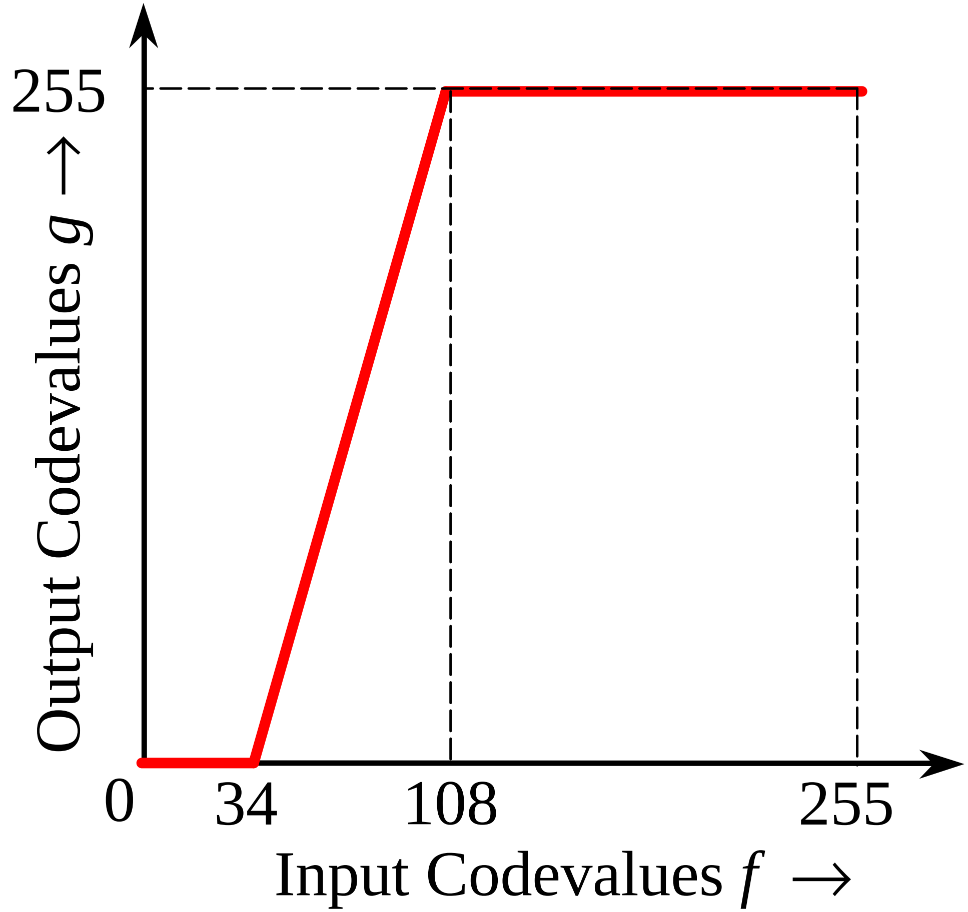


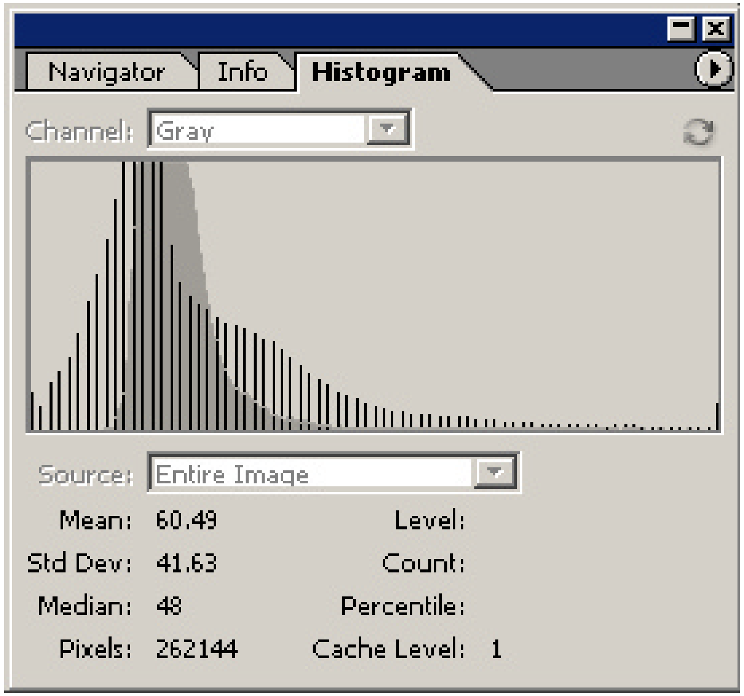
Shadows and highlights histogram clip points can be set by the user ("Image" → "Adjustments" → "Levels" → "Options".)


Original 512x512 8-bit

Photoshop Auto Levels

"Auto Levels" LUT For Aerial Image



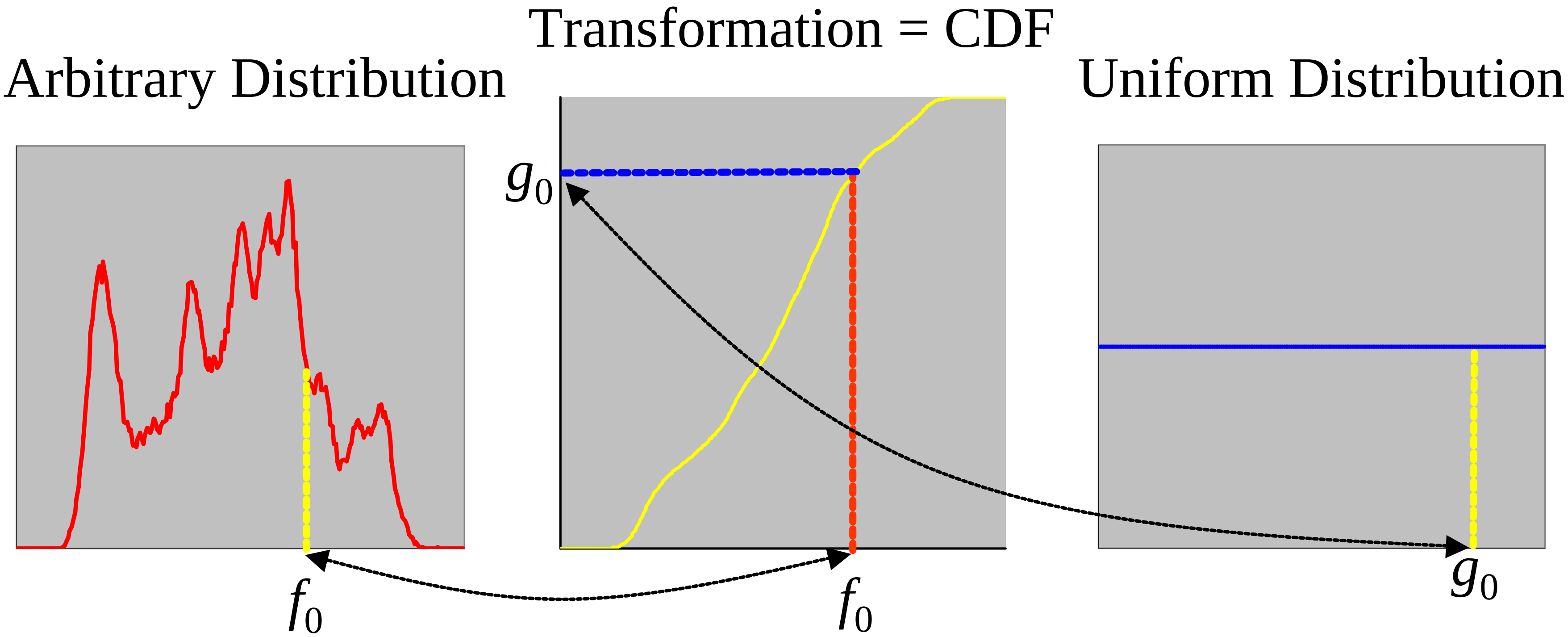


Histogram Equalization

- **Histogram Equalization** (HE) is a look-up-table (LUT) operation that attempts to improve the contrast by redistributing the histogram in a roughly uniform fashion.
- It is done automatically and the resulting **nonlinear** LUT is derived from the histogram of the image. The slope of the LUT for a given input codevalue is proportional to the value of the histogram at that codevalue.
- In **nonadaptive** or **global HE**, the same LUT is utilized to process the entire image. This is useful for images that have an overall low contrast (e.g., underexposed or overexposed images). As in any LUT operation, HE often reduces the total number of output levels, so it may not be suitable for archiving applications.

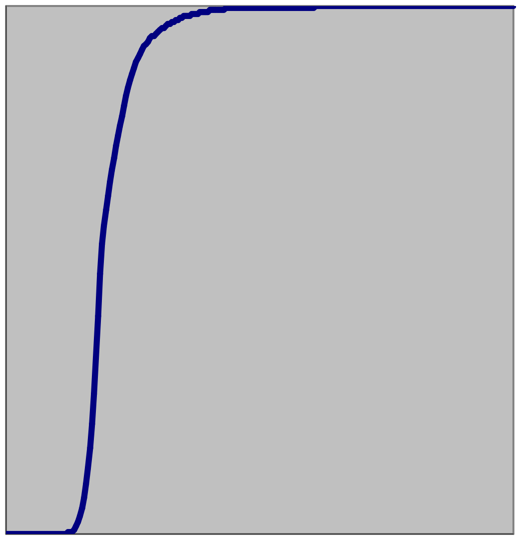
Histogram Equalization: Continuous Case

- The transformation that converts a random variable with a specific probability distribution into a random variable with a uniform distribution is given by the CDF of that distribution.

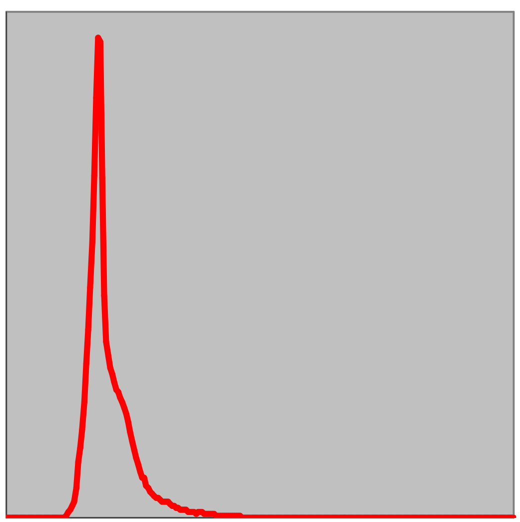


Examples of HE Look Up Tables

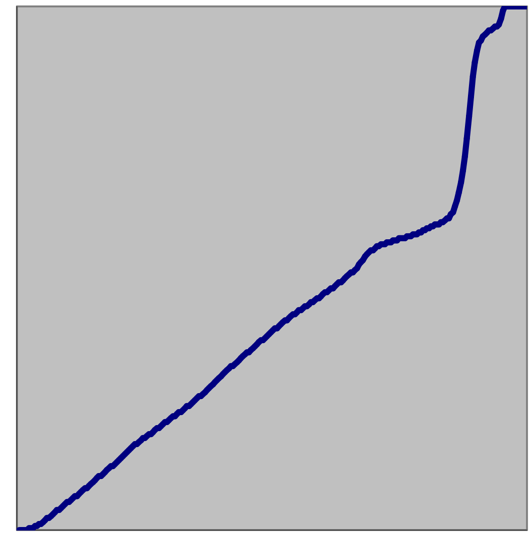
CDF
HE LUT



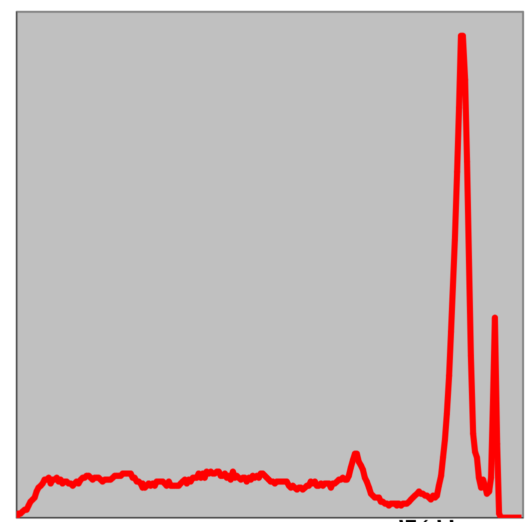
Histogram



CDF
HE LUT



Histogram



Histogram Equalization: Discrete Case

- For the discrete HE case, for each input codevalue *f*, the output codevalue *g* is given by:

$$g = g_{\min} + \left(g_{\max} - g_{\min}\right) \times \mathrm{CDF}(f)$$

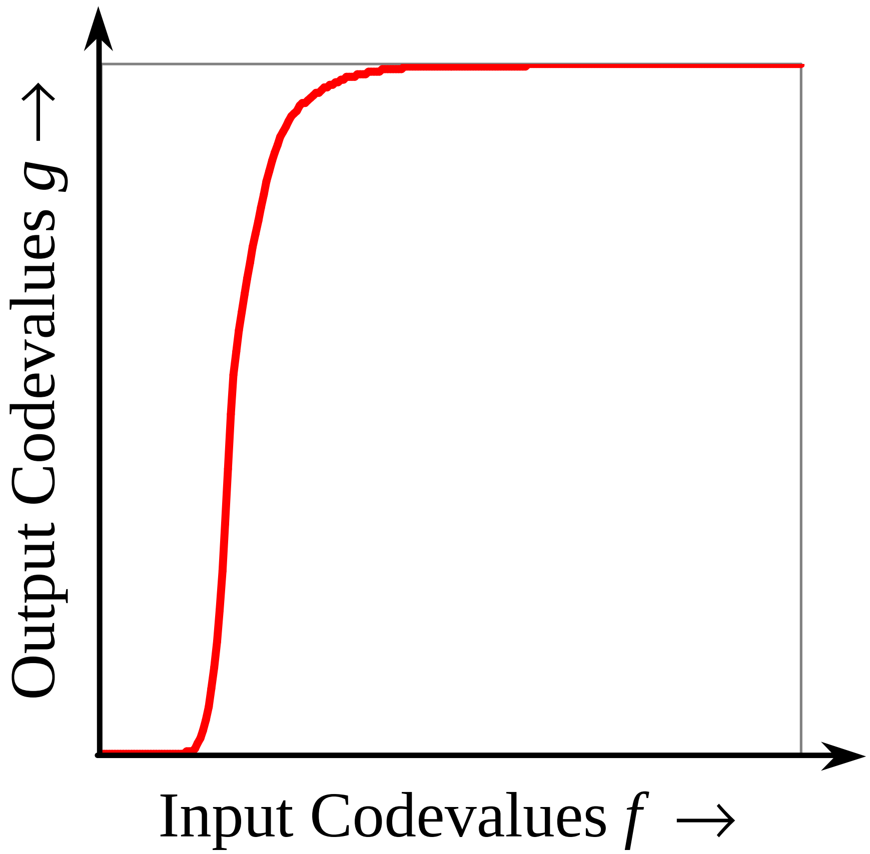
$$g = g_{\min} + \mathrm{NINT}\left[\left(g_{\max} - g_{\min}\right) \times \frac{\sum_{l=0}^{f} h_f(l)}{N_{total}}\right]$$

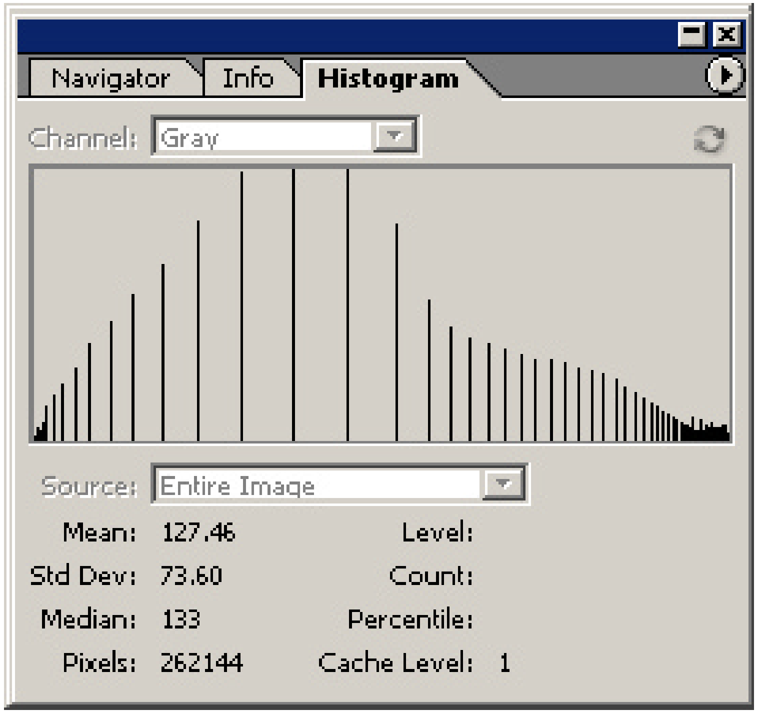
- N_{total} is the total number of pixels in the image, $h_f(l)$ is the input image histogram at codevalue *l*, and NINT denotes the operation of rounding to the nearest integer.


Original 512x512 8-bit

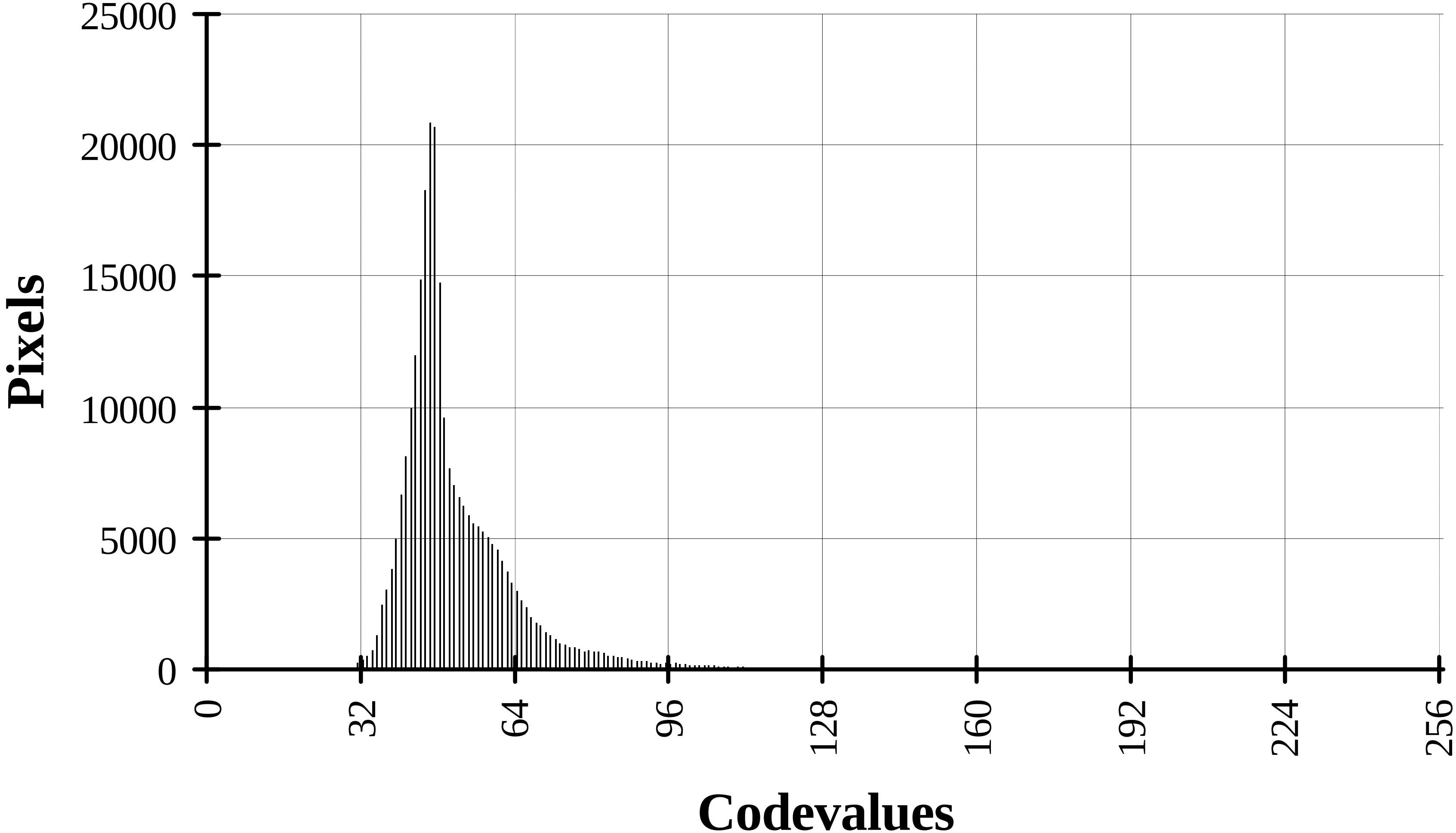

Histogram Equalized (HE)

HE Look-Up Table For Aerial Image

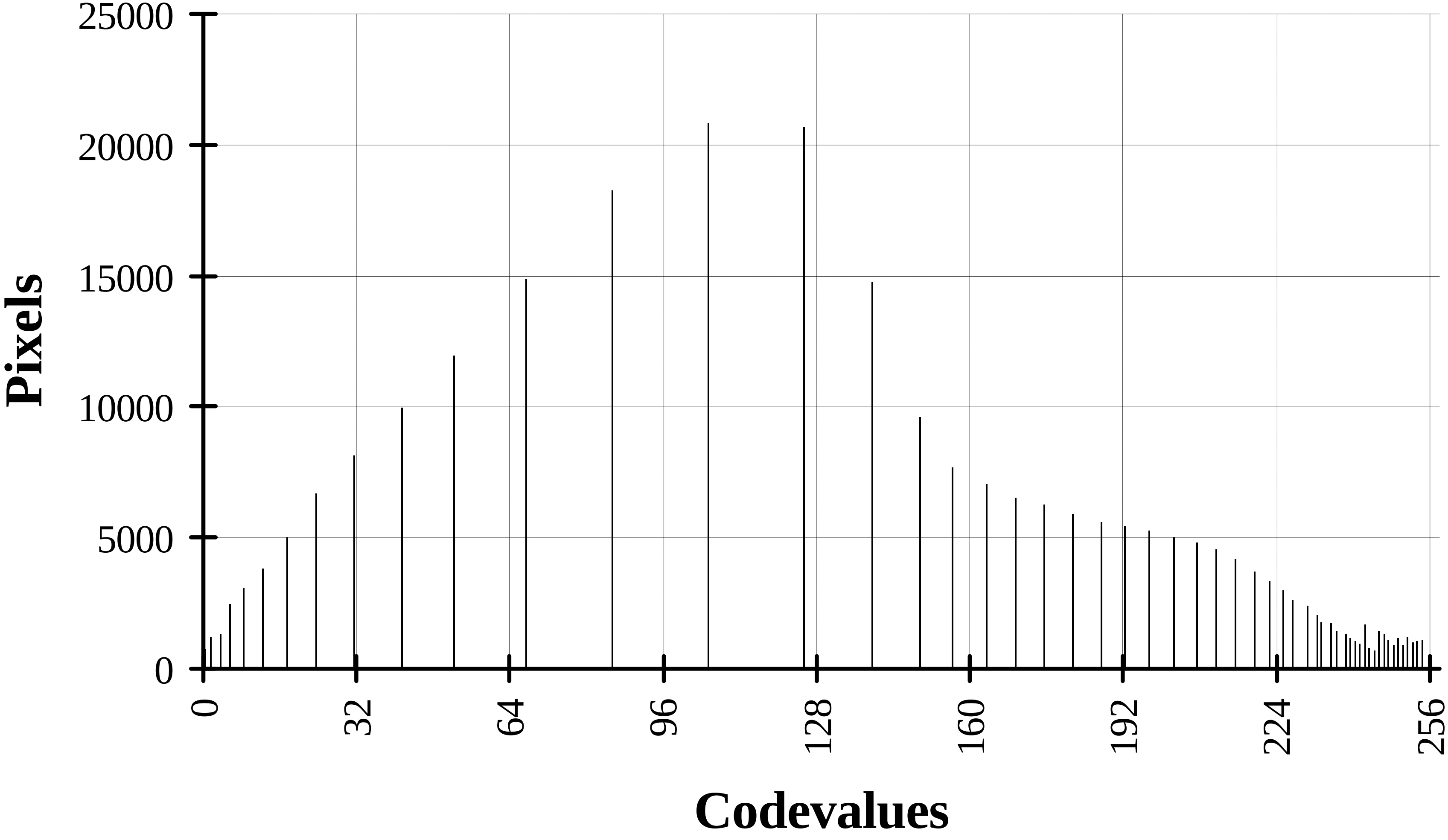




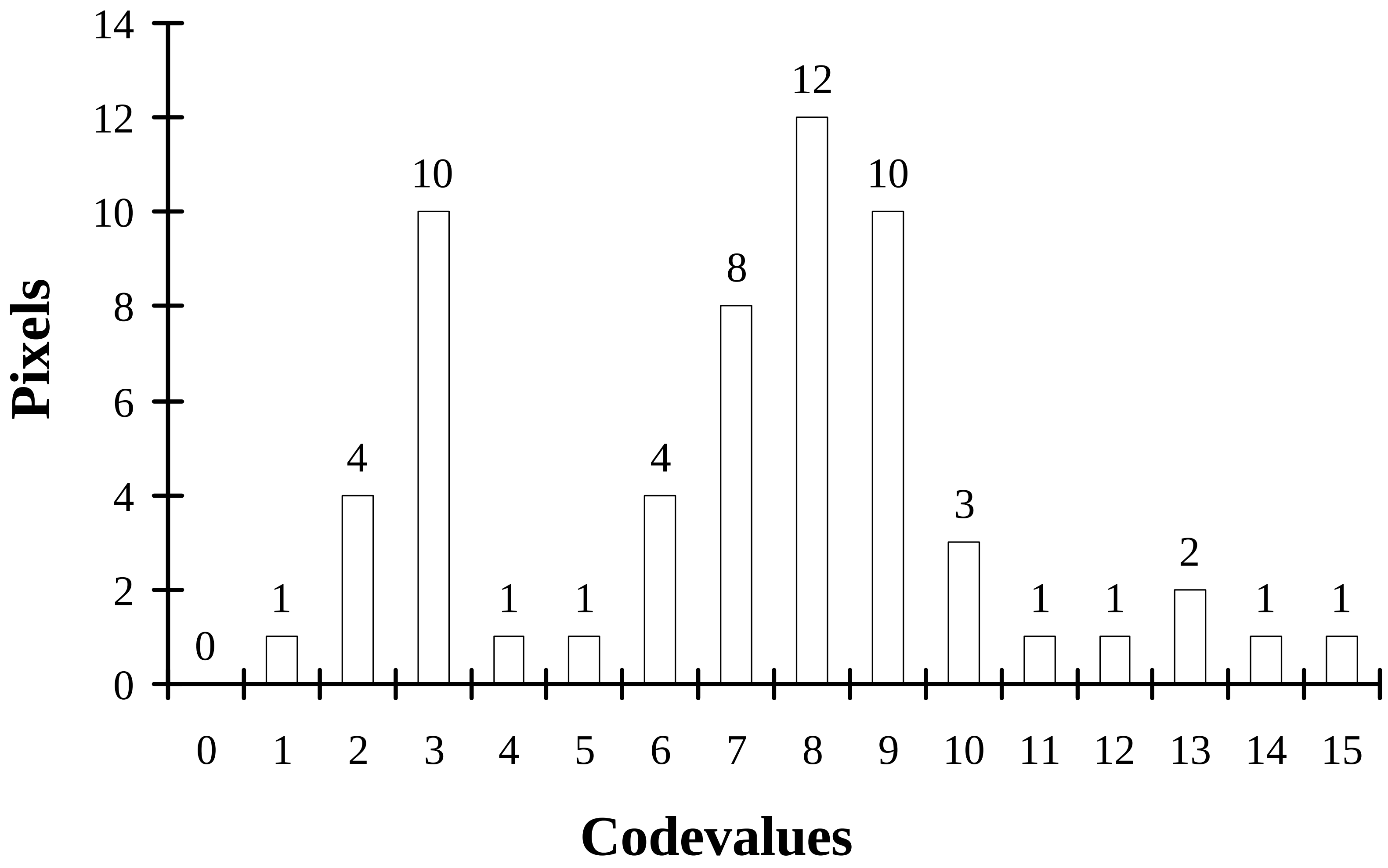
Histogram of the Aerial Image



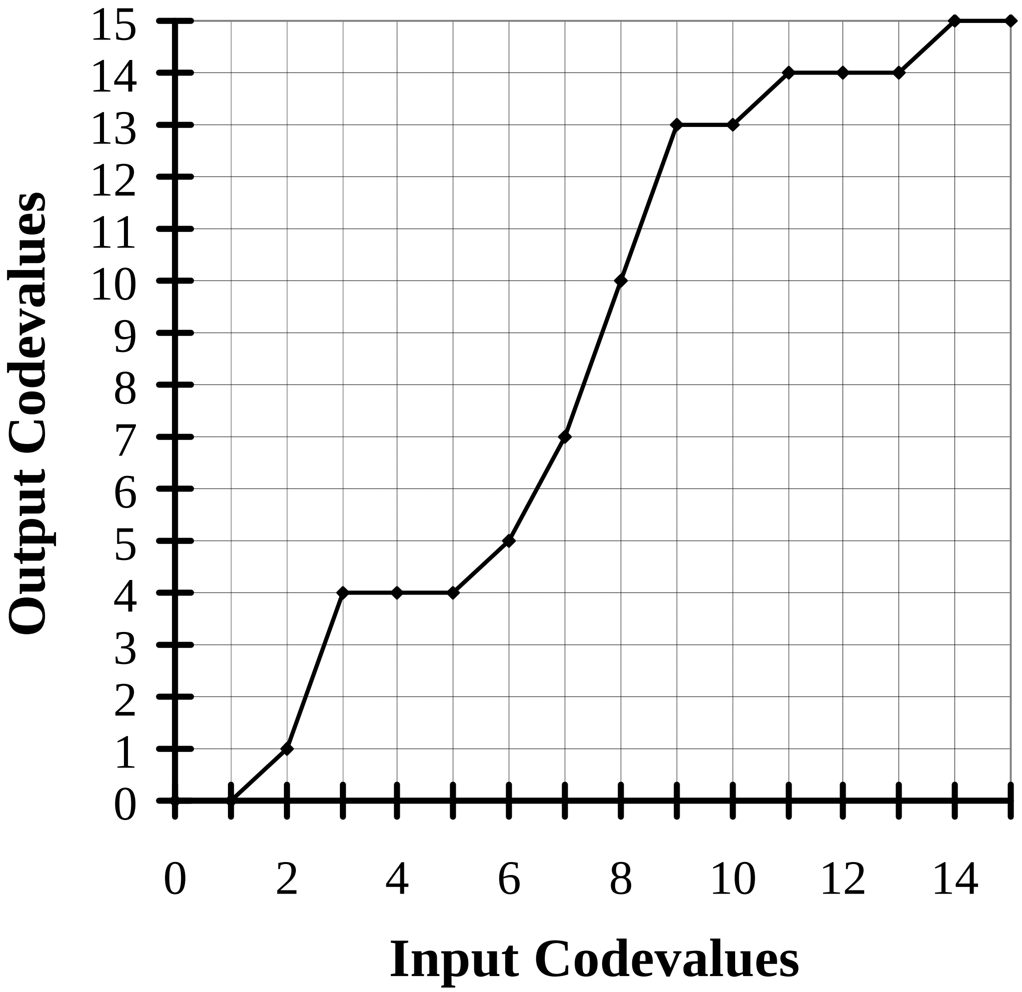
Histogram Equalized Image



Example: Original Histogram

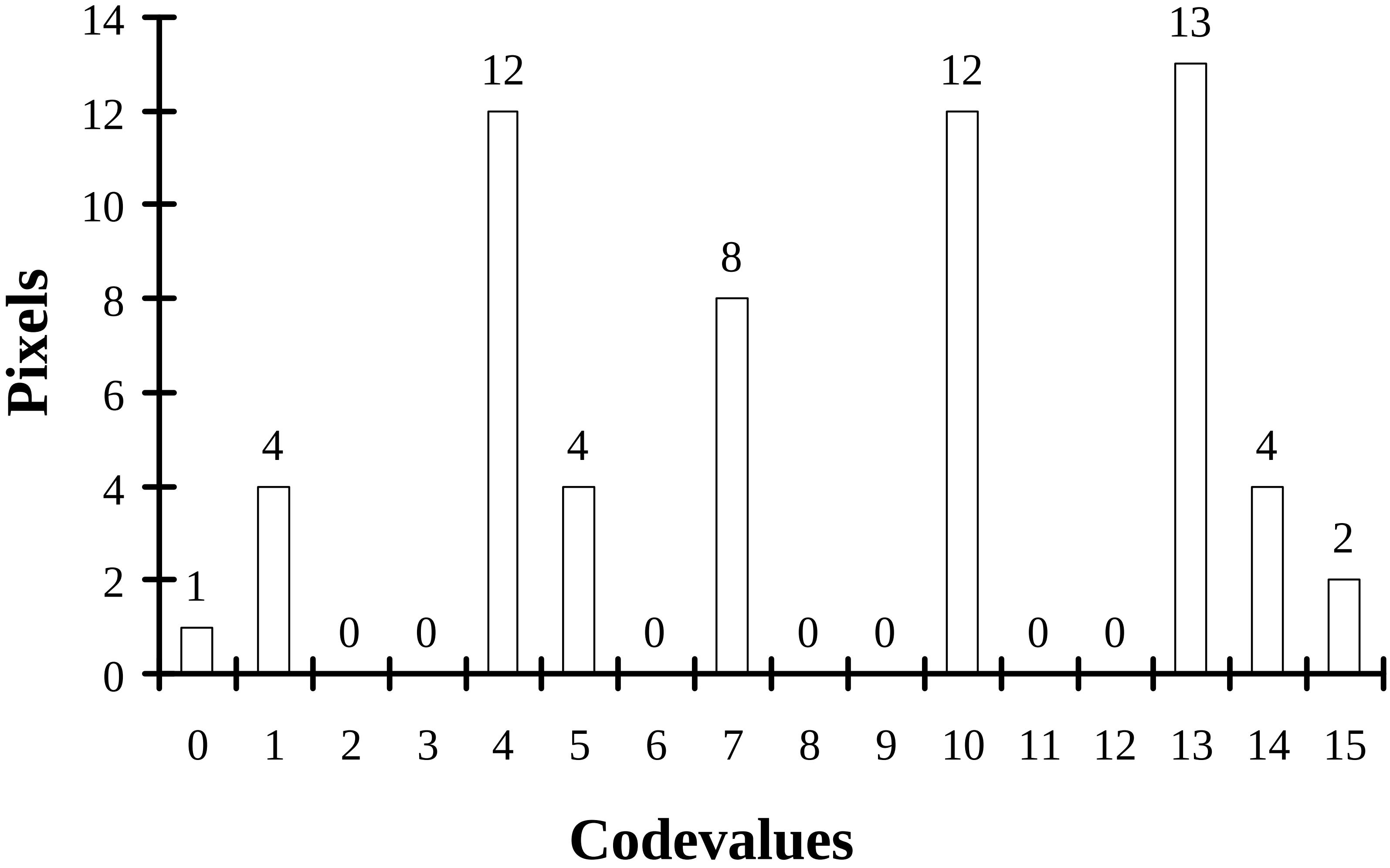


Histogram Equalization Transformation



Input *f*	**Output *g***
0	0
1	0
2	1
3	4
4	4
5	4
6	5
7	7
8	10
9	13
10	13
11	14
12	14
13	14
14	15
15	15

Equalized Histogram




Original 512x512 8-bit

Photoshop “Curves”

Photoshop Brightness/Contrast, B=13, C=80

Photoshop Levels, $f_{\min}=34$, $f_{\max}=108$, $\gamma=1.65$

Photoshop Auto Levels


Histogram Equalized (HE)

Original

“Auto Levels”

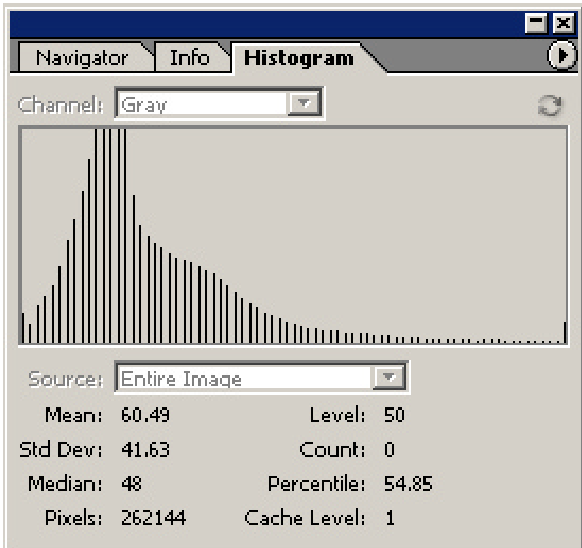
Histogram Equalized

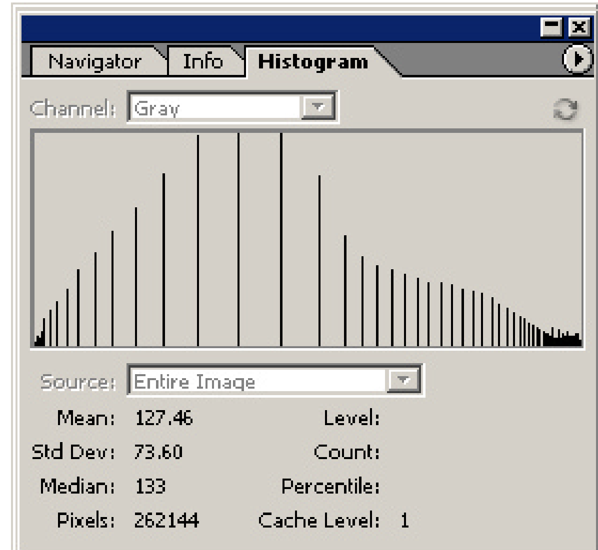
“Curves”

“Levels”, $\gamma = 1.65$

“Brightness/Contrast”

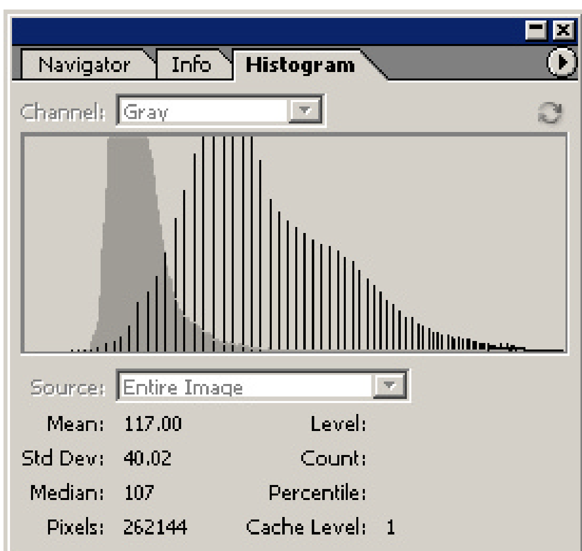
“Auto Levels”

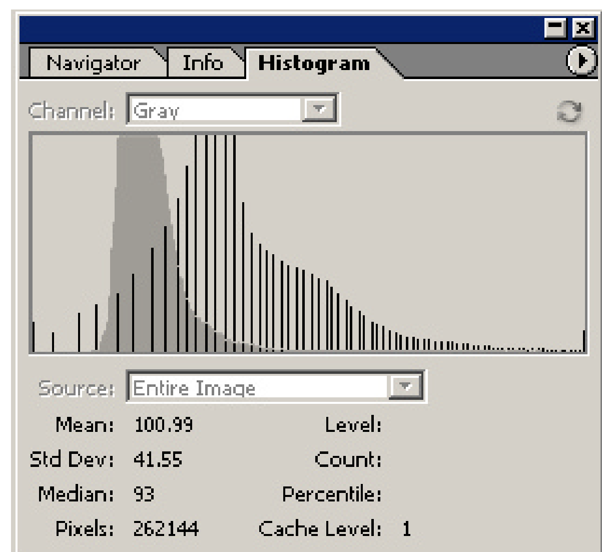




Histogram Equalization

“Curves”





“Levels” γ=1.65

Histogram Equalization Considerations

In the process of HE, gray levels can only be merged but not broken up, so the histogram equalized image will have at most the same number of levels as the original image. In fact, it will often have less levels.

- It is advantageous to capture the original image with more bits/pixel.
- Sometimes large ranges of gray levels at the ends of the gray scale are mapped into a single level. It may thus be preferable to transform into a histogram that somewhat rolls off at the ends of the scale.
- To preserve the maximum information for image archiving applications, it may be preferable not to perform any type of LUT operation.

Continuous Histogram Modification

- Assume that a transformation $g=T(f)$ is sought that transforms a given continuous histogram h_f into a desired continuous histogram h_g. This can be performed in two steps:
  - First, transform f into z using the CDF of f, P_f, where z will have a uniform histogram.
  - Next, transform z into g using the inverse CDF of g, P_g^{-1}.
- Thus, the desired transformation is given by:

$$g = T(f) = P_g^{-1}(z) = P_g^{-1}\left[P_f(f)\right]$$

Discrete Histogram Modification

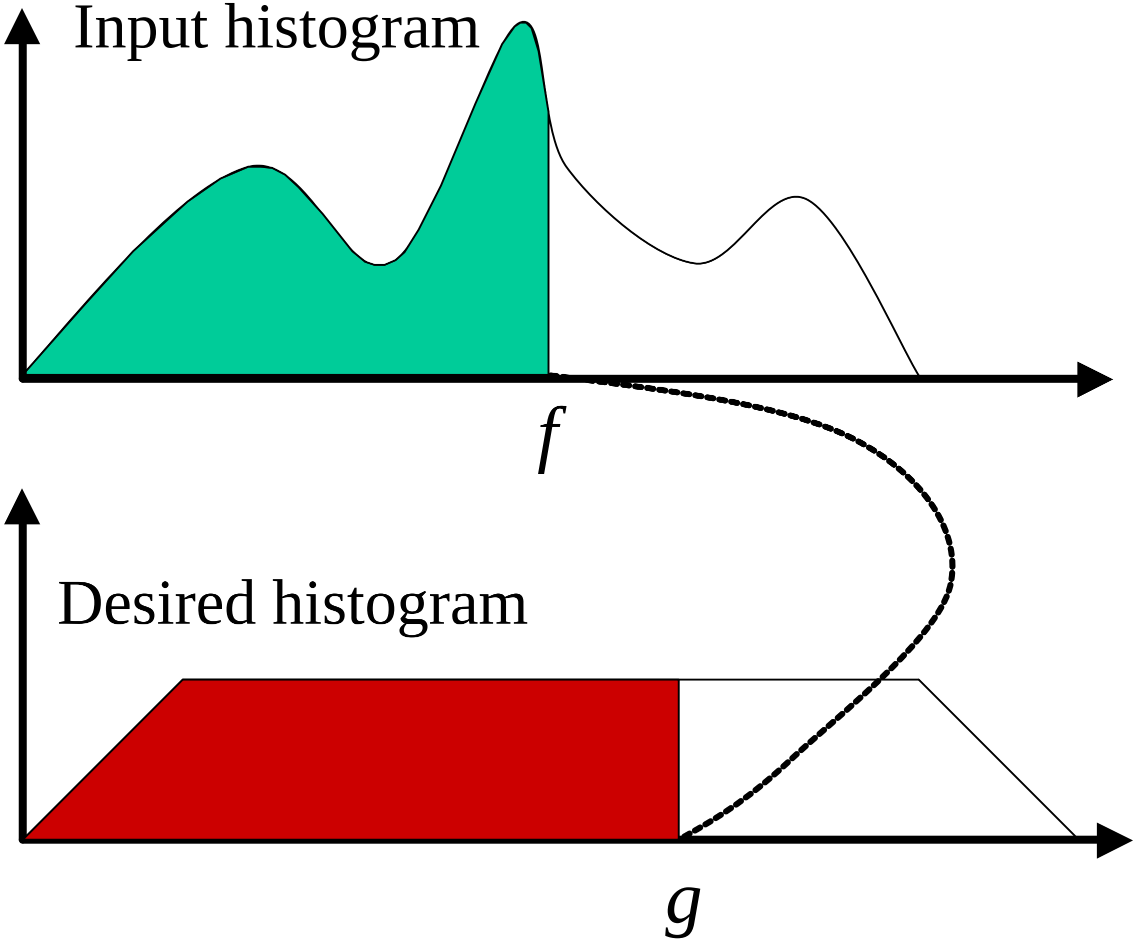
For a discrete input histogram, $h_f(f)$, the cascaded application of the two CDF LUTs can cause too much distortion. Instead, the following steps are performed in order to find the discrete histogram modification LUT:

- For every input codevalue f, find: $P_f(f) = \sum_{j=0}^{f} h_f(j)$
- The output codevalue g is determined such that:

$$\left| P_g(g) - P_f(f) \right| \le \left| P_g(l) - P_f(f) \right|, \quad \text{for all } l$$

$$\text{where} \quad P_g(g) = \sum_{j=0}^{g} h_g(j)$$

Histogram Modification

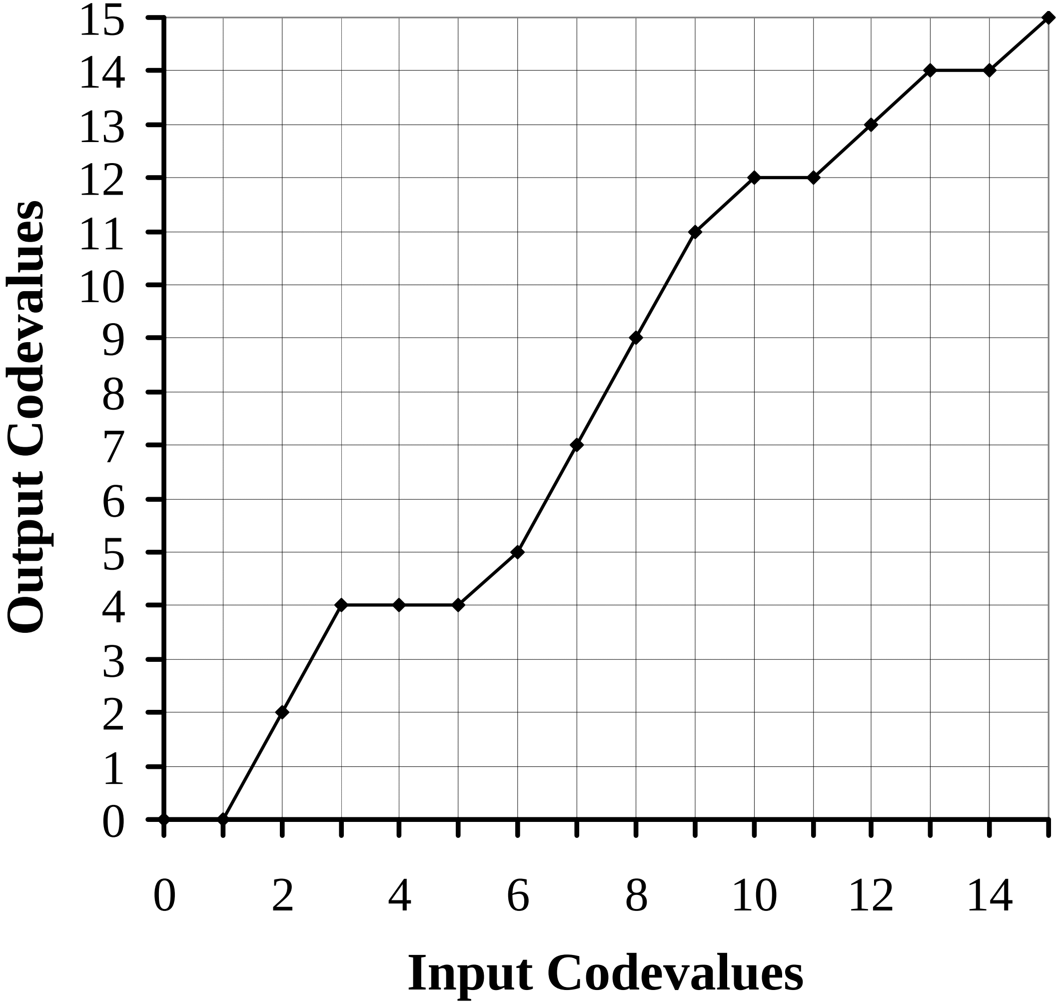


For every input codevalue *f*, the output codevalue *g* is determined so that the area of the two shaded regions are as close as possible.

Histogram Modification Example

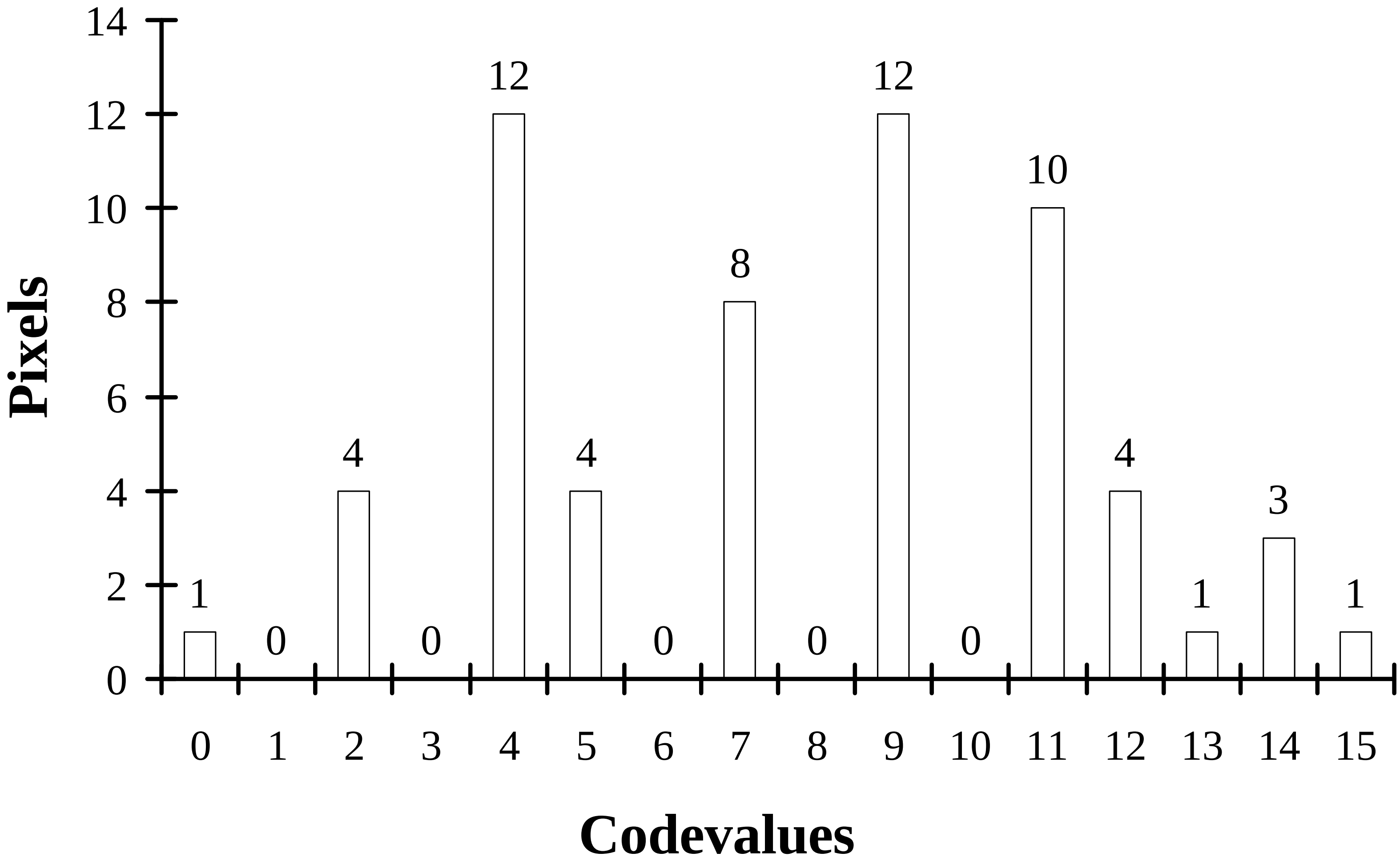
Codevalue	h_f	h_g	P_f	P_g
0	0	1	0	1
1	1	2	1	3
2	4	3	5	6
3	10	4	15	10
4	1	5	16	15
5	1	5	17	20
6	4	5	21	25
7	8	5	29	30
8	12	5	41	35
9	10	5	51	40
10	3	5	54	45
11	1	5	55	50
12	1	4	56	54
13	2	3	58	57
14	1	2	59	59
15	1	1	60	60

Histogram Modification Transformation



Input *f*	**Output *g***
0	0
1	0
2	2
3	4
4	4
5	4
6	5
7	7
8	9
9	11
10	12
11	12
12	13
13	14
14	14
15	15

Modified Histogram



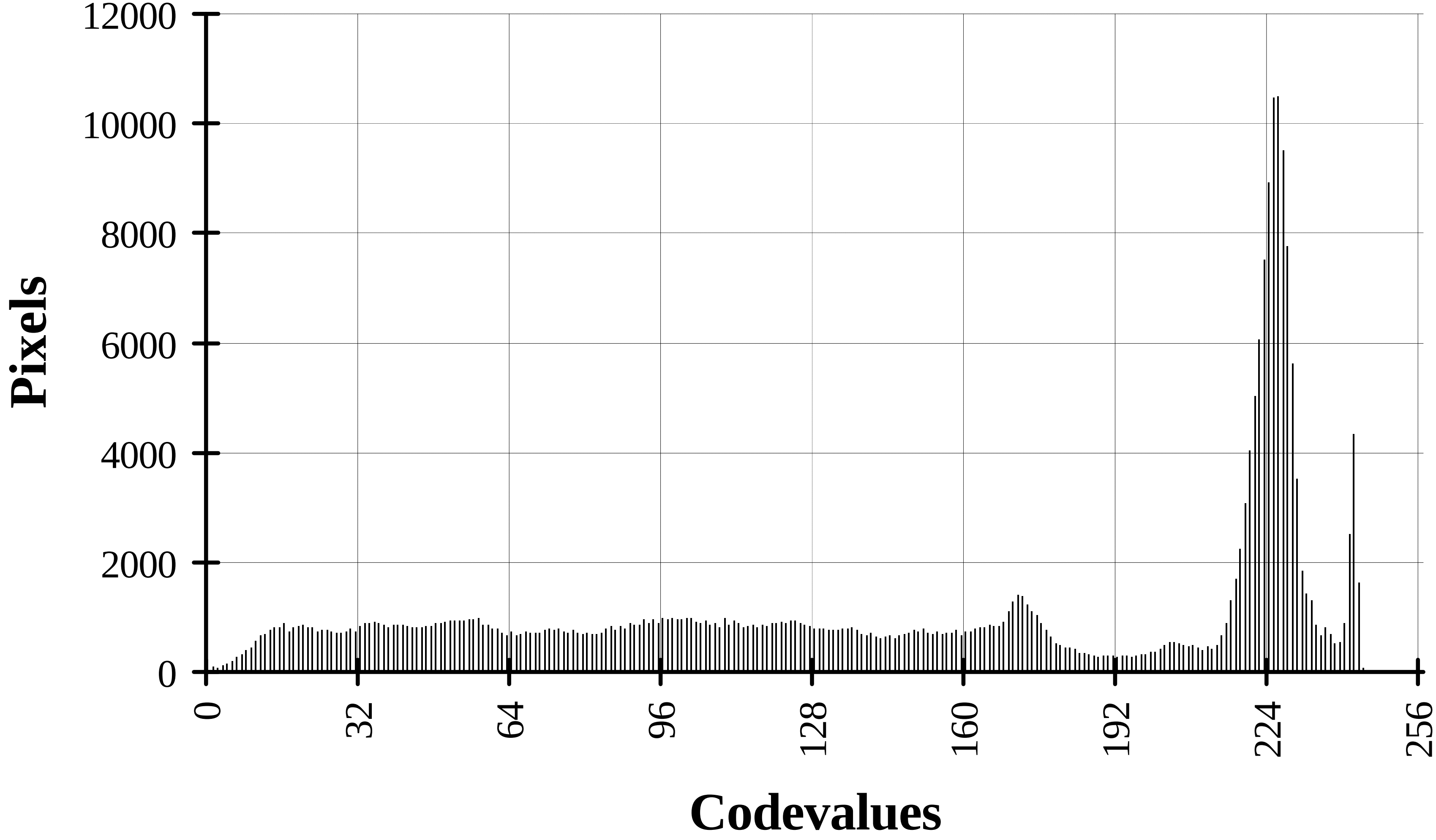
Adaptive Histogram Equalization

- Sometimes the overall histogram of an image may have a wide distribution while the histogram of its local regions are highly skewed. In such cases, it is often desirable to enhance the contrast of these local regions. This is performed by **adaptive histogram equalization (AHE)**.
- The term **adaptive** implies that different regions of the image are processed differently (e.g., with different look-up tables). AHE is applied in an automatic fashion, however, because of its additional computational complexity, it may not be implementable in real-time.
- AHE can be used to create an effect similar to the dodging and burning performed in conventional darkrooms.

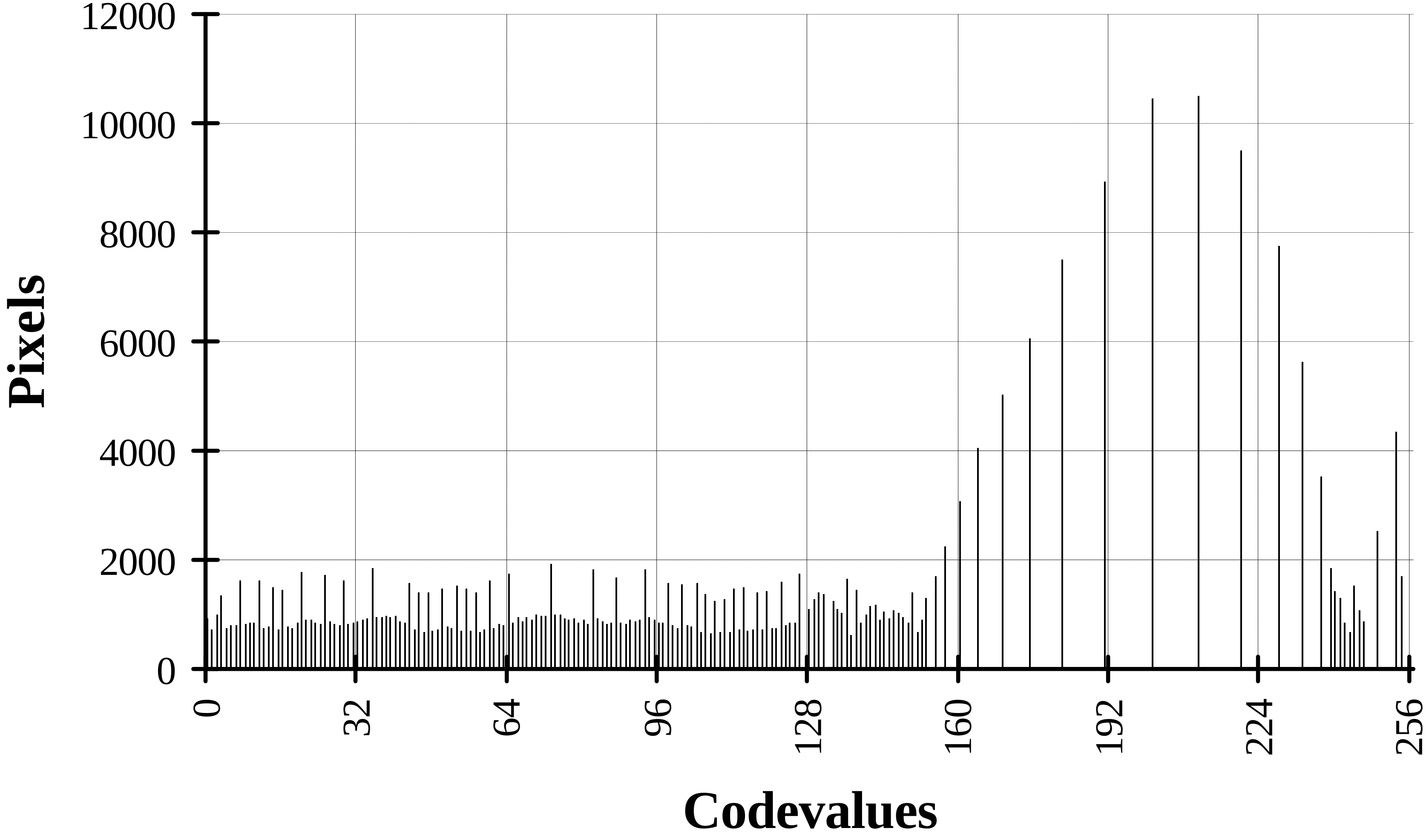

Original 512x512 8-bit


Histogram Equalized (HE)

Histogram of Building Image



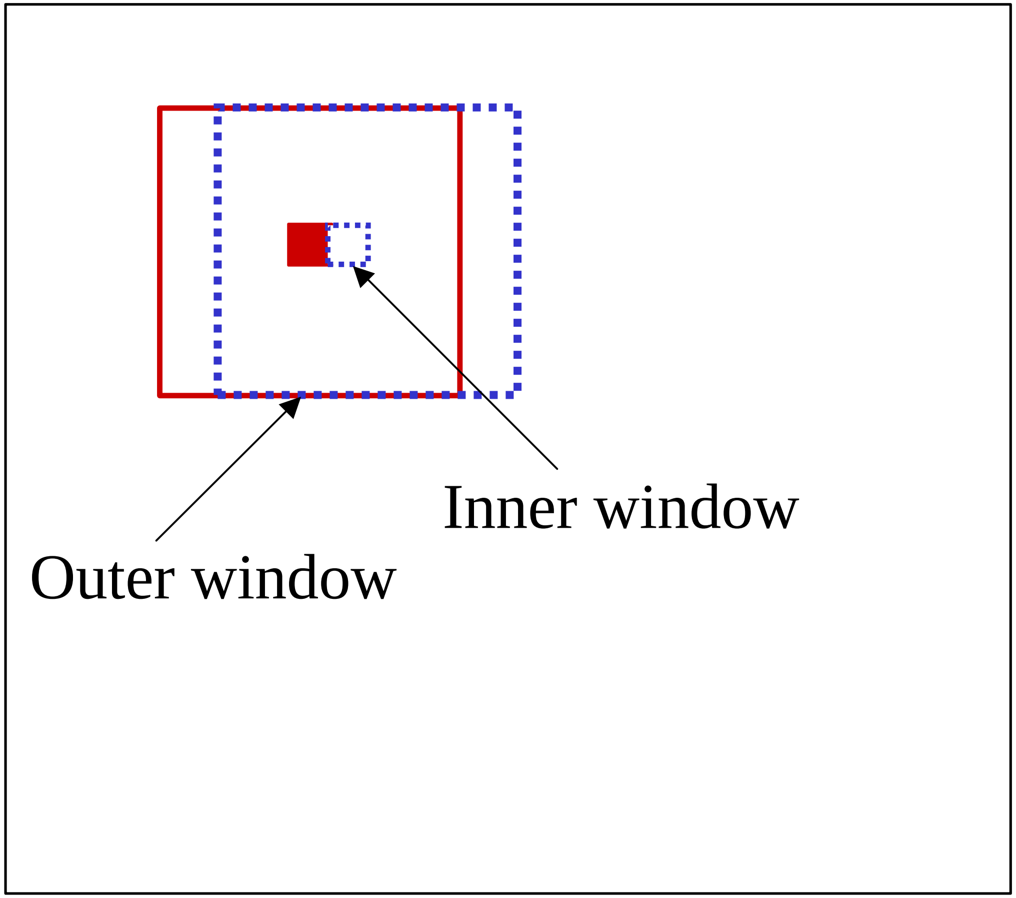
Histogram Equalized Image




Adaptive HE 128:128

Moving Window AHE

The HE LUT for large outer window (e.g., 128 × 128) is computed, but applied only to the small inner window (e.g., 16 × 16). The inner window is then shifted and the process is repeated.



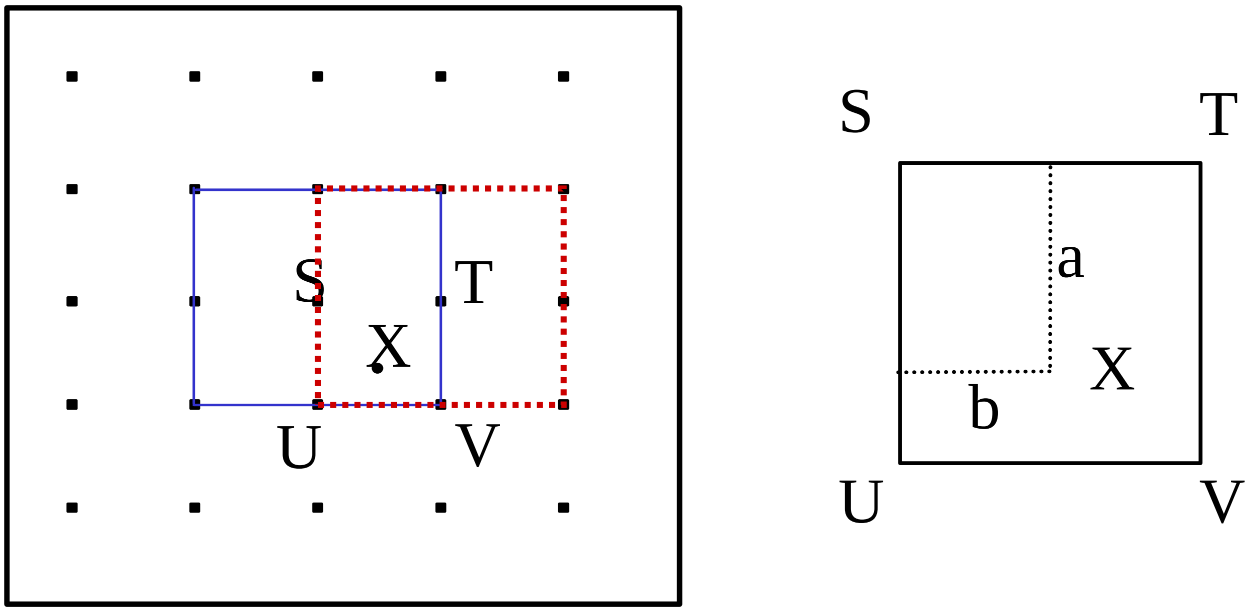

Original 512x512 8-bit


Adaptive HE 128:16


Adaptive HE 128:4


Adaptive HE 128:2

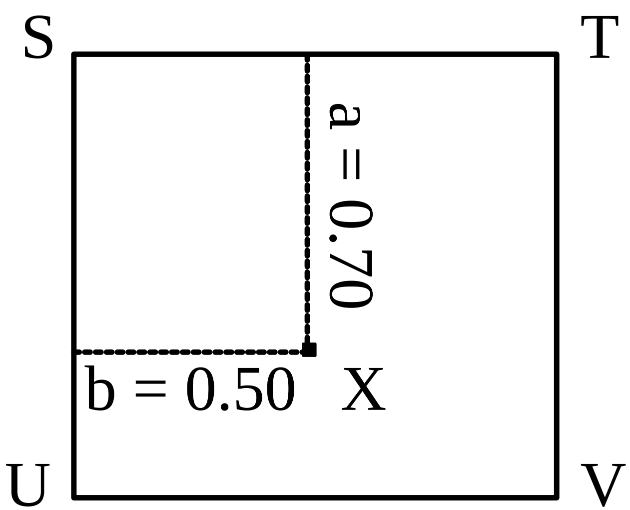
Pizer’s Adaptive Histogram Equalization



- A coarse grid (e.g., 64×64 or 128×128) is superimposed on the image to identify sample points equally distributed in both directions. For each grid point, the HE LUT for a rectangular window surrounding it with a dimension twice the grid spacing is determined. The windows have a 50% area overlap.

Pizer's Adaptive Histogram Equalization

Due to the 50% overlap between windows, each pixel belongs to four neighborhoods. The values based on the HE LUT of the four neighborhoods are bi-linearly interpolated to find the enhanced output pixel value. As an example consider:



$$X = 100$$

$$g_S(100) = 120, \quad g_T(100) = 95$$

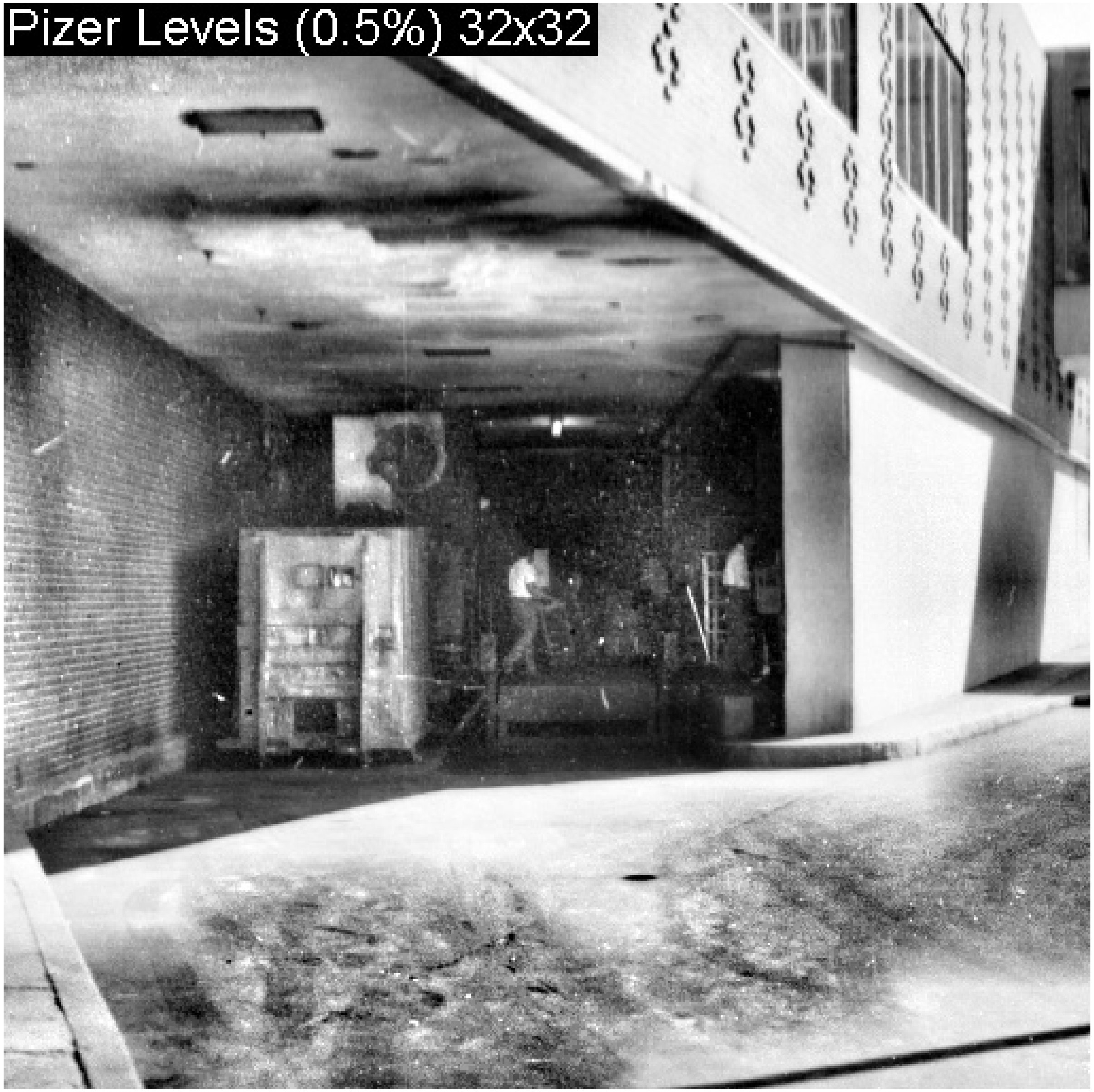
$$g_U(100) = 80, \quad g_V(100) = 78$$

$$g = (1-a)(1-b)g_S + a(1-b)g_U + \\ + (1-a)bg_T + abg_V$$

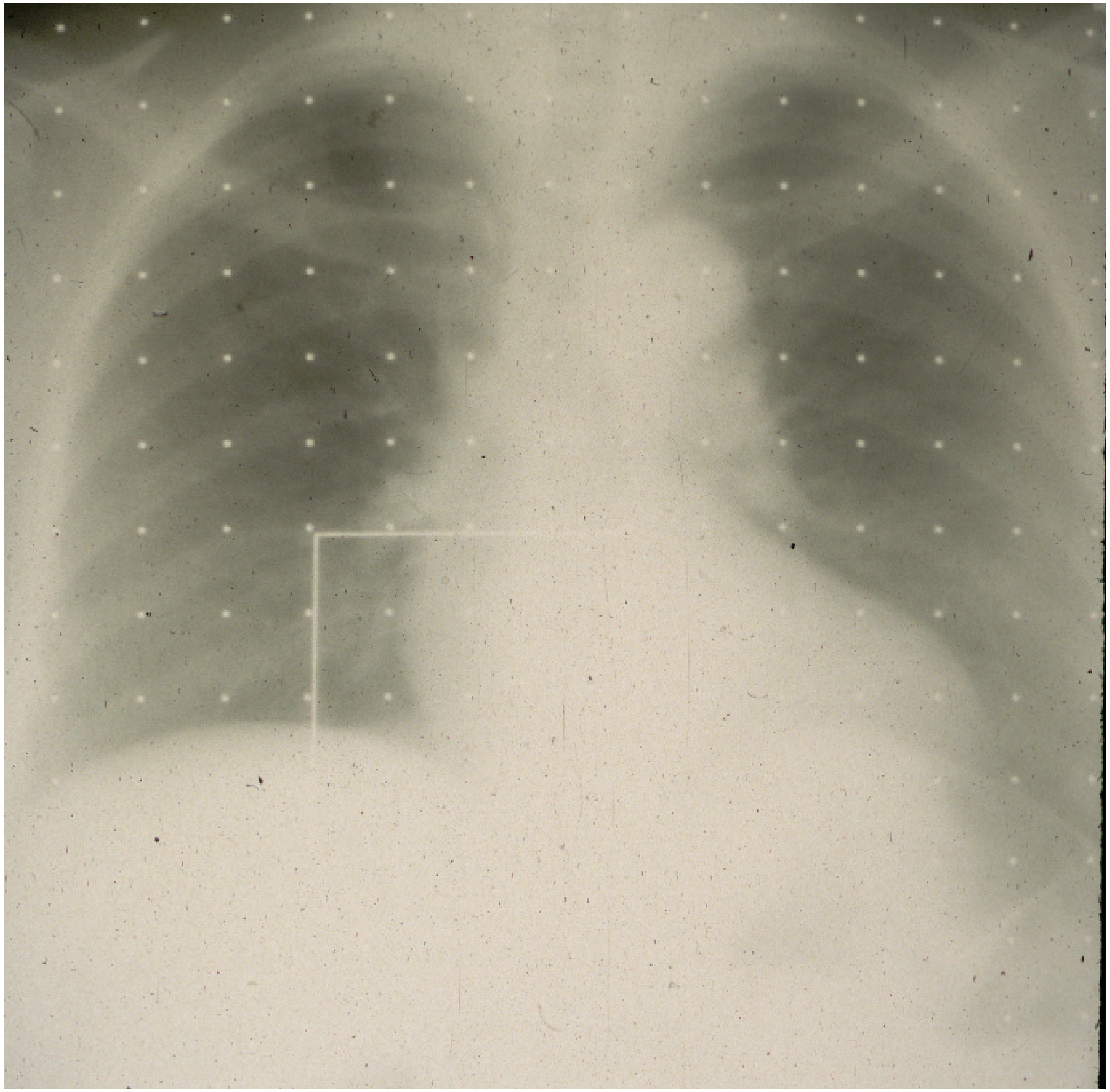
$$g(100) = 120 \times 0.15 + 80 \times 0.35 + 95 \times 0.15 + 78 \times 0.35 = 88$$

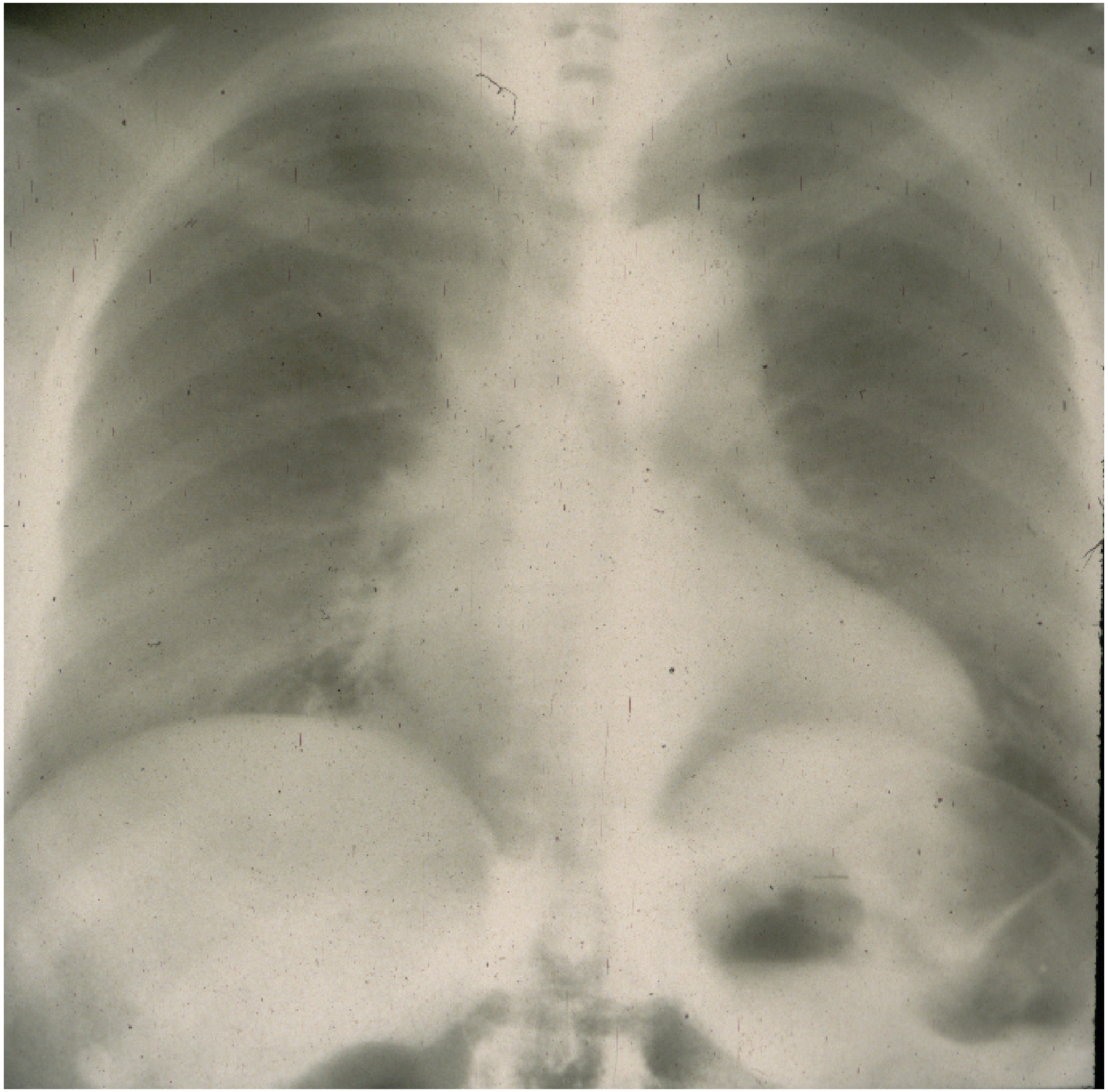

Pizer AHE 96x96


Pizer AHE 128x128


Pizer Levels (0.5%) 32x32


Pizer Levels (0.5%) 64x64





8-bit sRGB from scanned film

8-bit sRGB image from left, contrast enhanced adaptively

12-bit erimm from scanned film, contrast enhanced adaptively

Statistical Differencing

- The following transformation creates an image with a desired local standard deviation (s.d.) while maintaining the same local mean:

$$g(k,l) = \left[f(k,l) - \bar{f}(k,l) \right] \frac{s_d}{s_f(k,l)} + \bar{f}(k,l)$$

where $\bar{f}(k,l)$ and $s_f(k,l)$ denote the local mean and the local s.d. of the image and s_d denotes the desired s.d..

- The local mean is usually approximated by blurring or smoothing the image with a low-pass filter, while the local s.d. is calculated in a local neighborhood surrounding the pixel. The desired s.d. is constrained to be the same throughout the image.

Wallis Algorithm

$$g(k,l) = \left(f(k,l) - \bar{f}(k,l) \right) \left[\frac{A s_d}{A s_f(k,l) + s_d} \right] + \left[a \bar{f}_d + (1 - a) \bar{f}(k,l) \right]$$

- The values $\bar{f}(k,l)$ and $s_f(k,l)$ denote the mean and s.d. (computed over a local neighborhood) of the pixel $f(k,l)$.
- The values $\bar{f}_d$ and s_d are the desired mean and desired s.d. of the enhanced image.
- The parameter A is a gain factor that prevents overly large output values when $s_f(k,l)$ is small.
- The parameter α is a weighting factor that controls the ratio of the local mean to the desired mean.

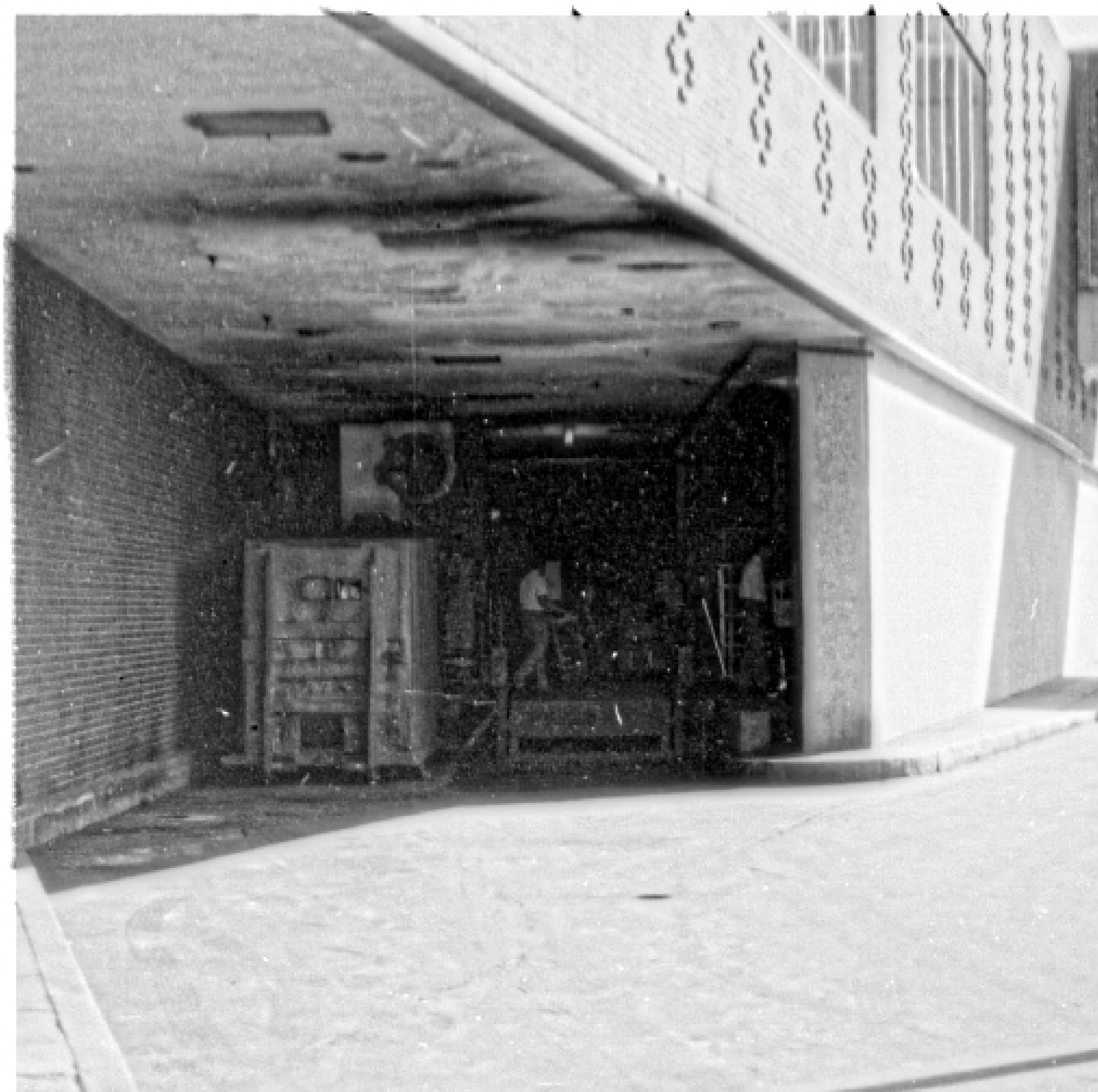
Original Image

Window = 13×13, A=5, sd = 50, α = 0

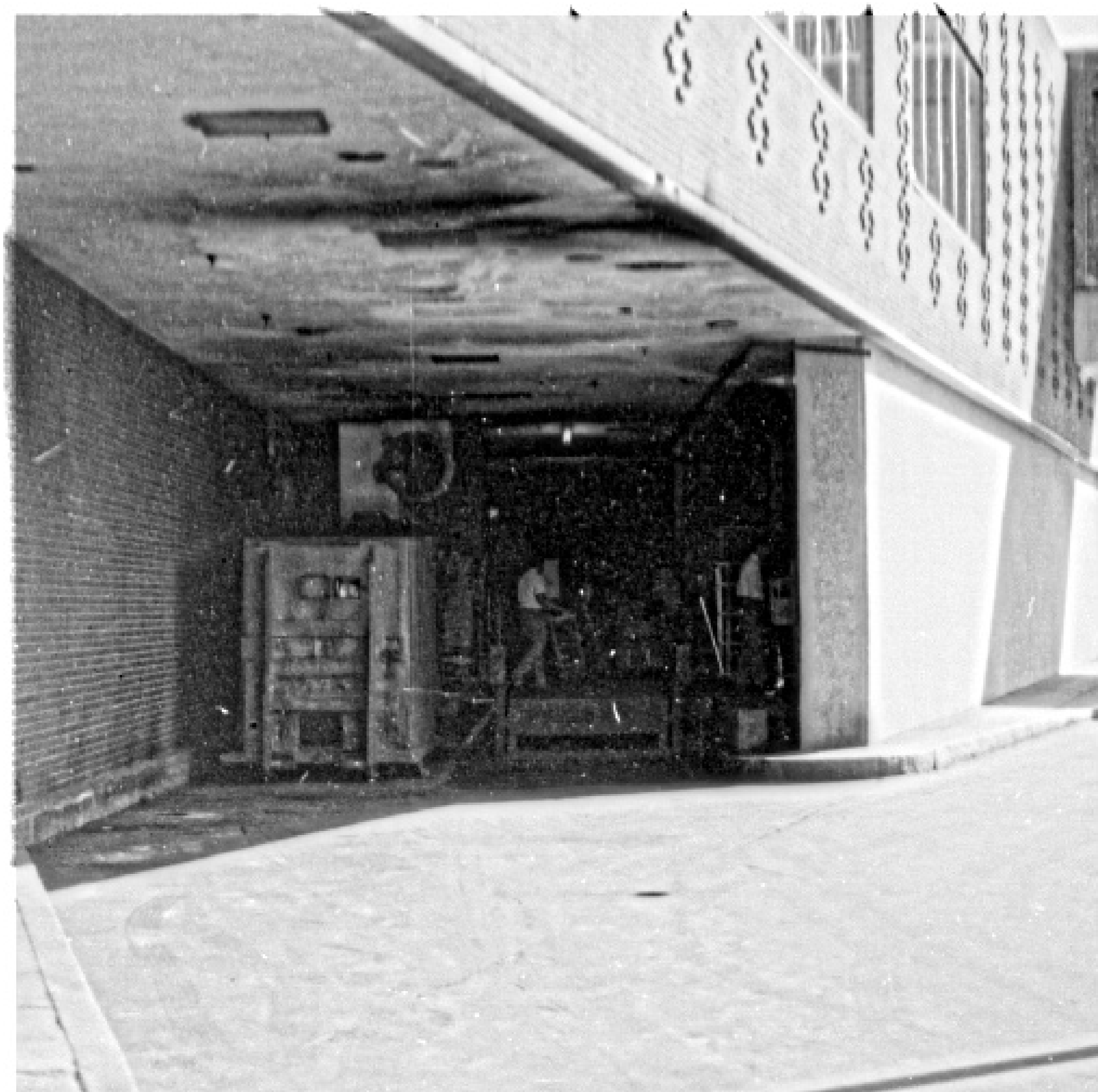
Window = 13×13, A=5, sd = 80, α = 0

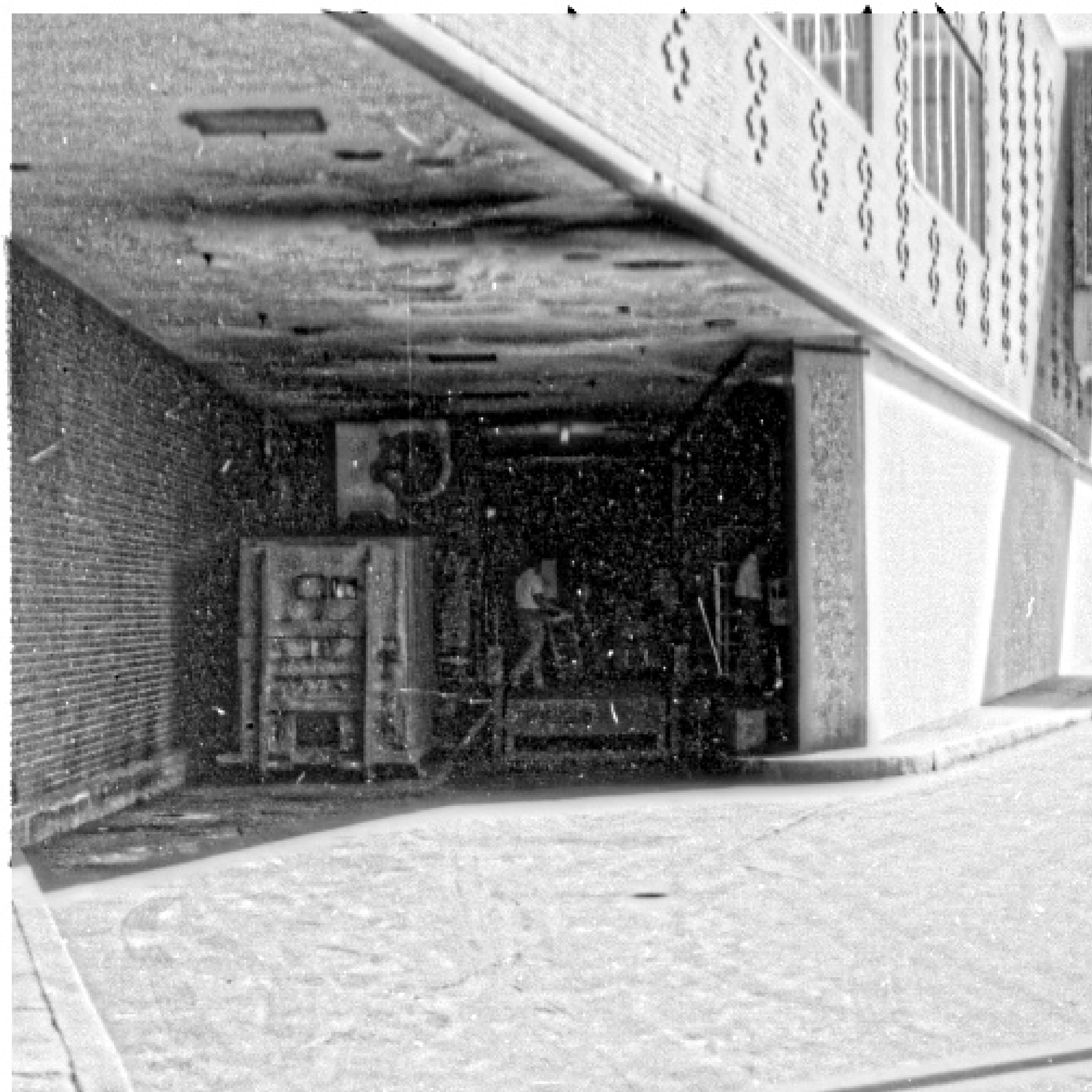
Original Image

Window = 13×13, A=5, sd = 50, $\alpha = 0$



Window = 13×13, A=5, sd = 80, $\alpha = 0$





Window = 13×13, A=10, sd = 50, $\alpha = 0$

Adaptive Dynamic Range Compression (DRC)

- Contrast enhancement can be done by breaking up the image into a low-frequency (pedestal) and high-frequency (detail) component and applying a LUT to pedestal image.