University of Bologna

Computer Vision Laboratory

Intensity Transformations

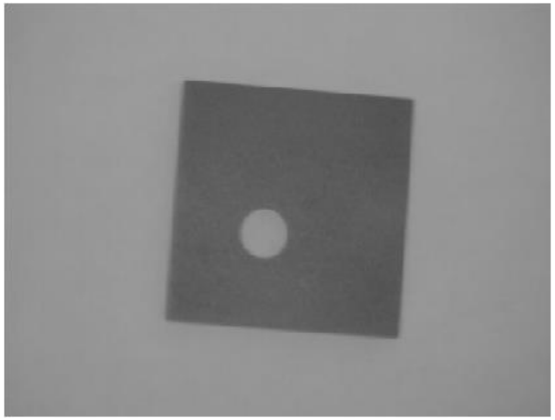
Luigi Di Stefano (luigi.distefano@unibo.it)

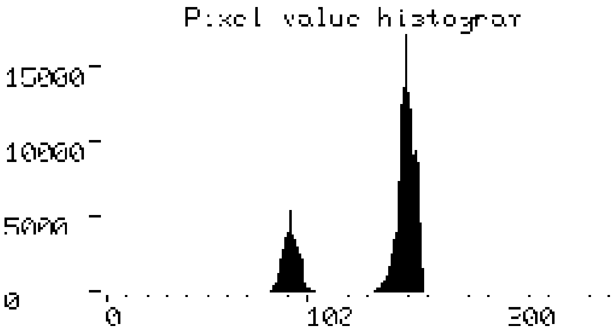
The gray-level histogram (1)

- **Intensity Transformations (aka Point Operators) are image processing operators aimed at enhancing the quality (e.g. the contrast) of the input image.**
- **As most such operators rely on the computation of the gray-level histogram (intensity histogram) of the input image, we start by defining this useful and widespread function.**
- **The gray-level histogram of an image is simply a function associating to each gray-level the number of pixels in the image taking that level.**
- **Computing the histogram is straightforward: we define a vector (H) having as many elements as the number of gray-levels and then scan the image (I) to increment the element of the vector corresponding to the gray-level of the current pixel (p):**

$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix}, \qquad \forall \mathbf{p} \in \mathbf{I}: \quad \mathbf{H}\big[\mathbf{I}[\mathbf{p}]\big] + +$$

The gray-level histogram (2)





- **The histogram provides often useful information on image content, although it must be highlighted that it does not encode any information related to the spatial distribution of intensities, so that, e.g., if we shuffle randomly the pixels of an image we end up with a new image having exactly the same histogram as before.**
- **Normalization of histogram entries by the total number of pixels yields relative frequencies of occurrence of gray-levels, which can be interpreted as their probabilities. Accordingly, the normalized histogram can be thought of as the *pmf* (probability mass function) of the discrete random variable given by the gray-level of a randomly picked pixel in the image.**

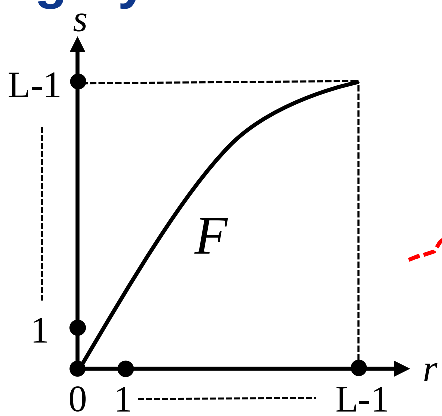
Intensity Transformation (Point Operator)

- **An image processing operator whereby the intensity of a pixel of the output image is computed based on the intensity of the corresponding pixel of the input image only. As such, this operator is just a function which maps a gray-level into a new gray-level:**

$$s = F(r);$$

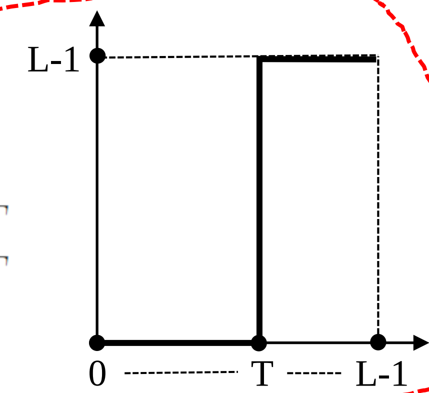
r : input gray - level,

s : output gray - level



e.g.: Thresholding operator

$$s = \begin{cases} 0 & \text{if } r \leq T \\ L-1 & \text{if } r > T \end{cases}$$

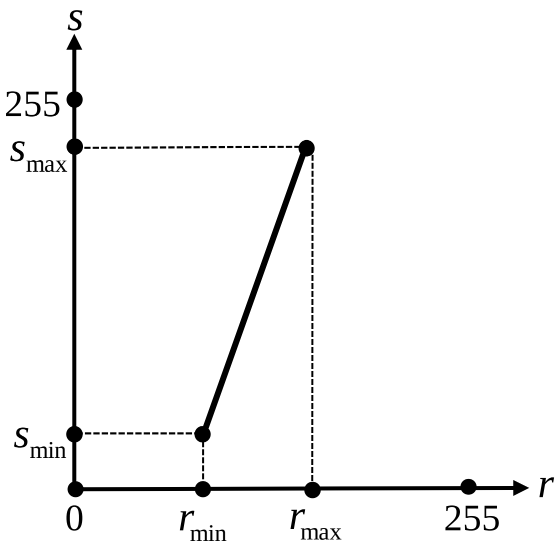


- **Any such operator can be implemented as a Look-Up Table (LUT), which is often convenient especially in case of hardware implementation:**

$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix}, \forall \mathbf{p} \in \mathbf{I}: \ \mathbf{O}[\mathbf{p}] = \mathbf{LUT}\big[\mathbf{I}[\mathbf{p}]\big]$$

Linear Contrast Stretching

- **An image featuring a small gray-level range has likely poor contrast. The contrast can be enhanced by linearly stretching the intensities to span a larger interval (typically, the full available range).**



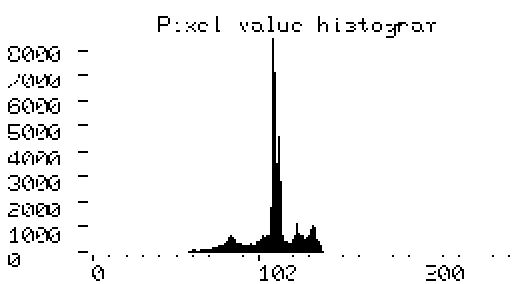
$$s = \frac{s_{\max} - s_{\min}}{r_{\max} - r_{\min}} \left(r - r_{\min} \right) + s_{\min}$$

$$s_{\min} = 0,\ s_{\max} = 255 \rightarrow s = \frac{255}{r_{\max} - r_{\min}} \left(r - r_{\min} \right)$$

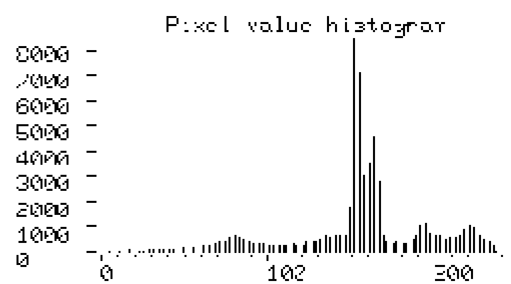
- **Should most pixel lie in a small interval while there exist a few dark and bright outliers, the linear function would approximate the identity and thus be ineffective. In such a case, r_{min} and r_{max} can be taken equal to some percentiles of the distribution (e.g. 5% and 95%), with smaller and higher input intensities mapped to s_{min} and s_{max} respectively.**

Examples (1)

Original Image

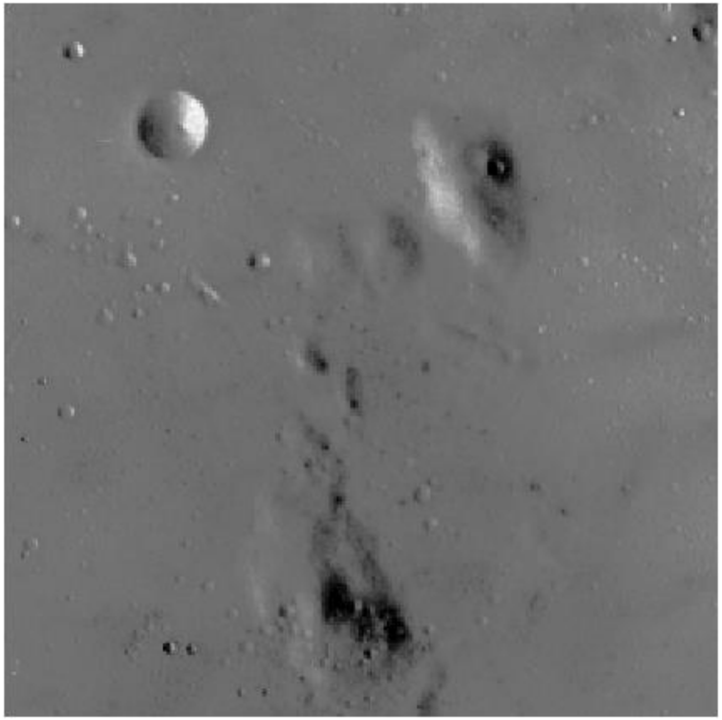


Enhanced Image

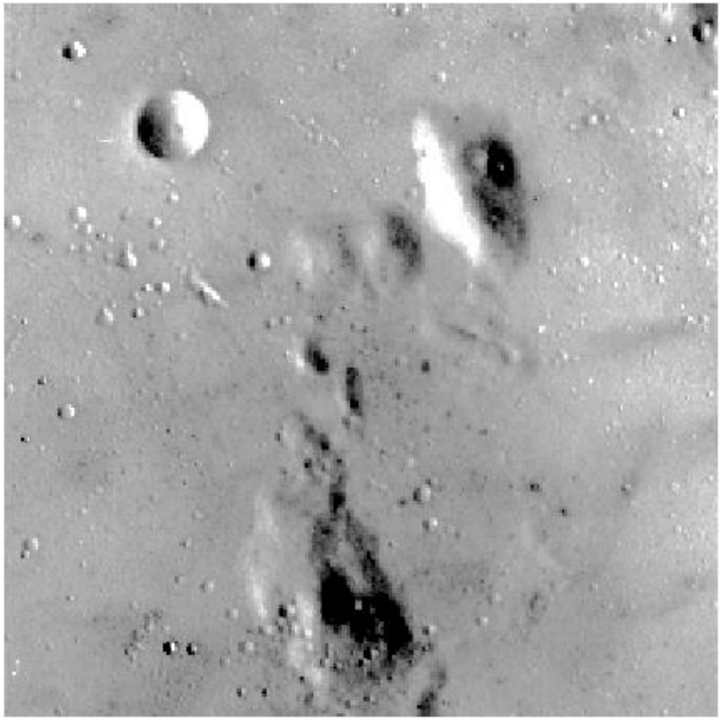


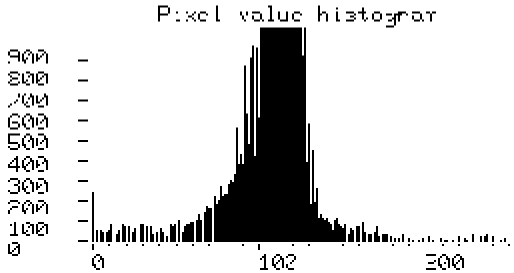
Examples (2)

Low-contrast image of the moon's surface



Enhanced Image

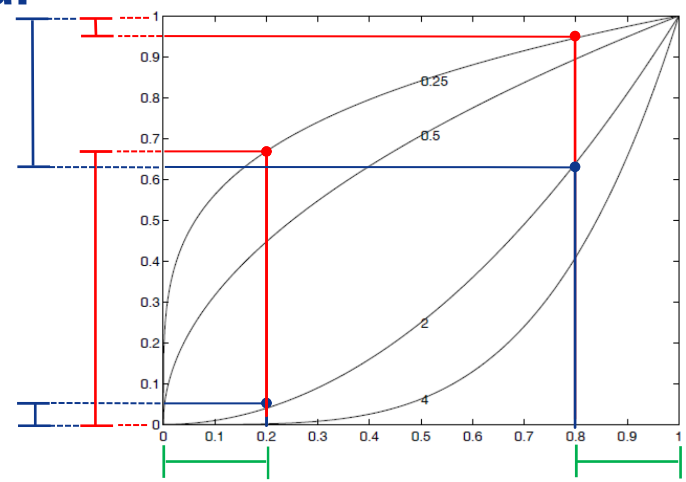




Due to the presence of black and white pixels, we take r_{min} and r_{max} at 1% and 99% of the distribution respectively, so as to obtain effective contrast enhancement by a linear mapping.

Exponential Operator (1)

- **It is sometimes desirable to selectively enhance the contrast in either dark (under-exposed) or bright (over-exposed) areas of the image. Linear stretching, though, would expand the contrast uniformly.**
- **However, a non-linear mapping, such as the Exponential Operator, can be deployed.**



$$y = x^r$$

$$x \in [0,1],\ r = 0.25,\ 0.5,\ 2,\ 4$$

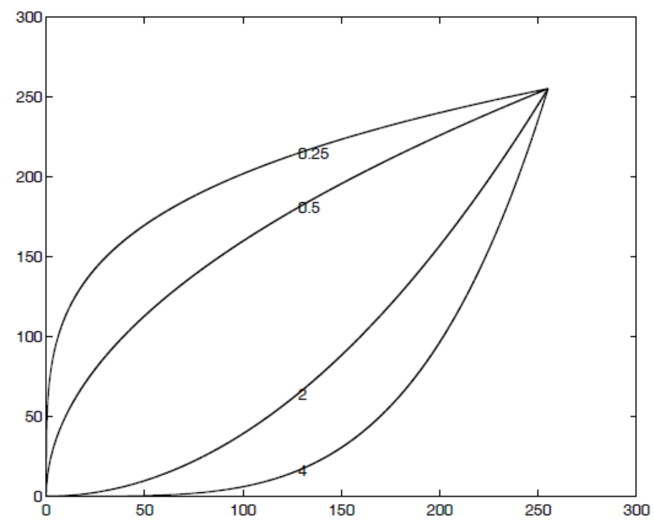
- **Thus, taking *r<1* would stretch the intensity dynamics of dark areas and shrink that of bright ones, which turns out effective to improve under-exposed images. The opposite behaviour is achieved with *r>1*, which therefore can be used to enhance over-exposed images.**

Exponential Operator (2)

- **As the gray levels of an image indeed range within [0,255]:**

$$y \to y/255,\ x \to x/255: \quad \frac{y}{255} = \left(\frac{x}{255}\right)^r \to y = 255^{1-r}x^r$$

$$P_{out} = 255^{1-r}P_{in}^r$$



- **The operator is also known as *gamma correction*, due to its widespread use to compensate for the exponential voltage-to-intensity mapping ($I=V^{\gamma}$) of old CRT monitors (often γ=2.2, i.e. darkening), which can be compensated by brightening the image ($\gamma'=1/\gamma=0.45$) before displaying.**

Examples (1)

Original Images (under-exposed)

Enhanced Images (r = 0.5)

Examples (2)

Original Images (over-exposed)

Enhanced Images (r = 1.5)

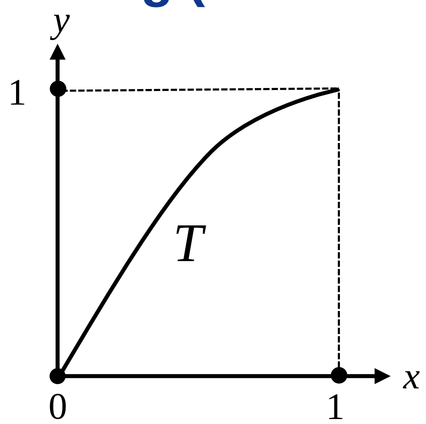
Histogram Equalization (1)

- **Spreads uniformly pixel intensities across the whole available range, which usually improves the contrast. Unlike linear stretching, which may require manual intervention to set r_{min} and r_{max}, HE provides fully automatic contrast enhancement.**

- **More formally, HE maps the gray-level of the input image so that the histogram of the output image turns out (ideally) flat.**

- **To find the mapping, let us consider a continuous random variable, *x*, and a strictly monotonically increasing (i.e. invertible) function, *T*, such that:**

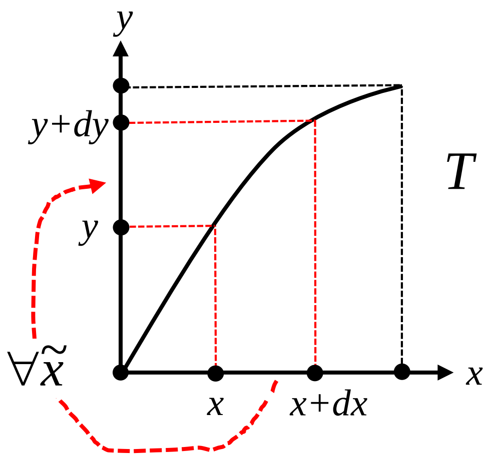
$$x \in [0,1] \rightarrow y = T(x) \in [0,1]$$



Histogram Equalization (2)

- **Let us now denote as $p_x(x)$ and $p_y(y)$ the probability density function (*pdf*) of x and y, respectively. As T is monotonically increasing:**

$$\forall \tilde{x} \in [x, x+dx] \rightarrow \tilde{y} = T(\tilde{x}) \in [y, y+dy]$$
$$with \quad y = T(x),\ y+dy = T(x+dx)$$



- **Therefore, the probability of x and y to belong to their –infinitesimal- intervals is exactly the same, which allows deriving the *pdf* of y as a function of T and the *pdf* of x:**

$$p_y(y)dy = p_x(x)dx \rightarrow p_y(y) = p_x(x)\frac{dx}{dy}$$

derivative of the inverse function
$x = T^{-1}(y)$

Histogram Equalization (3)

- **Let us now consider a specific mapping function T, i.e. the cumulative distribution function (*cdf*) of *x*, which is guaranteed to map into [0,1] and be monotonically increasing:**

$$y = T(x) = \int_0^x p_x(\xi)\, d\xi$$

- **Assuming also strict monotonicity**

$$p_y(y) = p_x(x)\frac{dx}{dy} = p_x(x)\frac{1}{\mathrm{dy}/\mathrm{dx}} = \frac{p_x(x)}{p_x(x)} = 1$$

we are lead to notice that *y* turns out uniformly distributed in [0,1].

- **We have thus found that by mapping any continuous random variable through its *cdf* (assumed strictly increasing) we attain a uniformly distributed random variable. Next, we shall see how to make use of this theoretical result to equalize an image.**

Histogram Equalization (4)

- **We proceed by discretizing the previous result, i.e. by considering the cumulative probability mass function of the discrete random variable associated with the the gray-level of a pixel, whose *pmf*, as already pointed out, is given by the normalized histogram:**

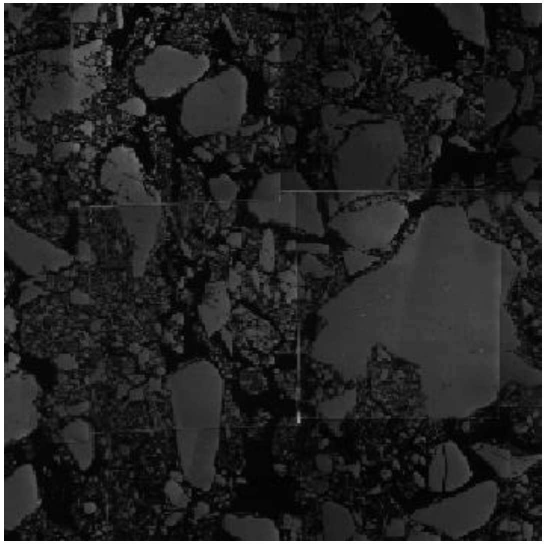
to map j in [0..L-1]

$$\begin{cases} N = \sum_{i=0}^{L-1} h(i) \\ \mathrm{p}(\mathrm{i}) = \dfrac{h(i)}{N} \end{cases} \rightarrow j = T(i) = \sum_{k=0}^{i} p(k) = \frac{1}{N}\sum_{k=0}^{i} h(k) \rightarrow j = \frac{L-1}{N}\sum_{k=0}^{i} h(k)$$

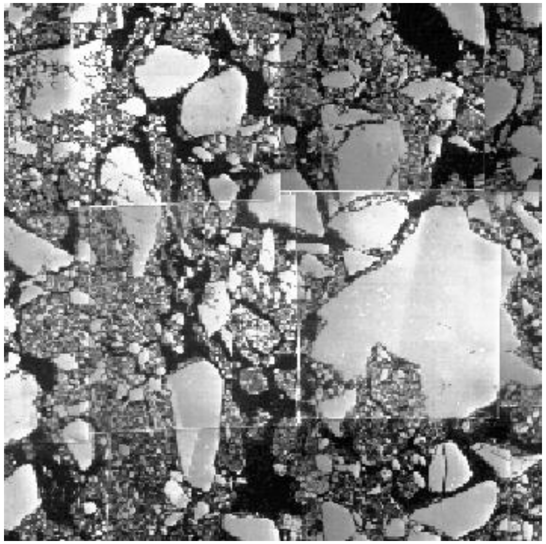
- **The above function does not perfectly equalize the histogram due to the several approximations involved (proof in the continuous case, strict monotonicity assumption, rounding errors). Nonetheless, it is usually effective in spreading the intensities over a wider range so as to improve image contrast.**

Examples

Original Images (low-contrast)



Equalized Images



also some posterization effects due to reduction of gray levels caused by rounding

Histogram Specification/Matching (1)

- **Starting from an arbitrary input image, we wish to transform it into a new image having a given, desired histogram.**

- **Let us denote as**

$$x,\ p_x(x) \quad \text{e} \quad z,\ p_z(z)$$

the random variables associated respectively with the gray-level of a pixel in the input and output image, and let's consider also the random variables attainable by equalization of both:

$$y = T(x) = \int_0^x p_x(\xi)\, d\xi$$

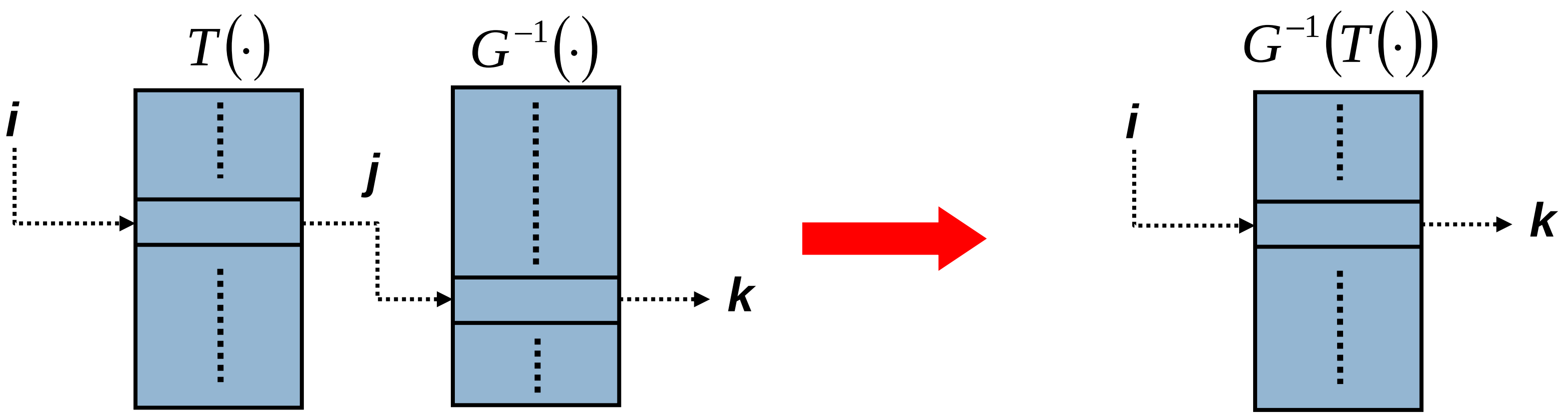
$$w = G(z) = \int_0^z p_z(\xi)\, d\xi$$

Histogram Specification/Matching (2)

- As *y* and *w* are uniformly distributed within the same interval:

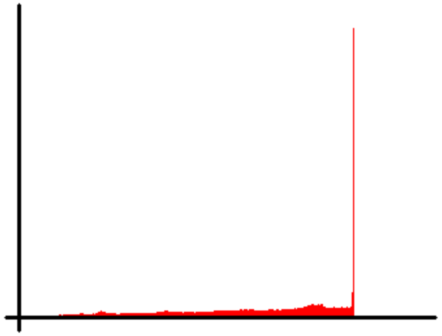
$$z = G^{-1}(w), \qquad \Rightarrow$$

- We can notice that $T(\cdot)$ can be obtained by equalizing the histogram of the input image, (which is known), $G^{-1}(\cdot)$ by equalization of the (known) desired histogram and then inversion of the obtained transformation.

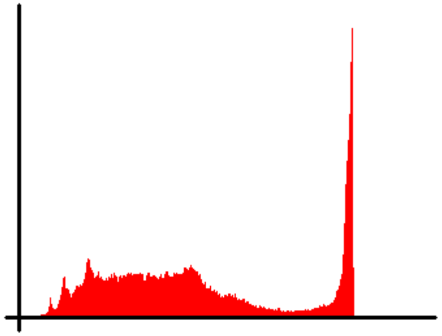


Tonal registration by HS

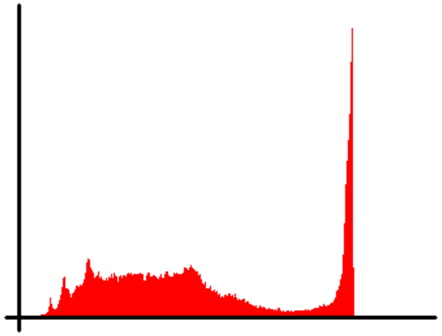
Source Image



Target Image



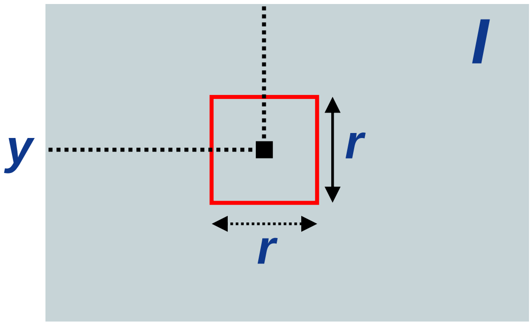
HS Image



Local histograms

- **The methods described thus far can fail in enhancing details in small image areas, due to pixels belonging to such areas being too small a fraction of the overall pixels quantity to be able to influence significantly the estimated global transformation.**

- **To enhance small image details the studied methods (*contrast stretching, histogram equalization/specification*..) can instead be applied to local histograms calculated over a *sliding window*.**



Accordingly, the intensity transformation to map a pixel is estimated based upon the histogram of a window centred at the pixel. Once the current pixel has been mapped, the window is slid over the next one to compute a new transformation.

- **For the sake of computational efficiency, it is worthwhile computing local histograms through incremental calculation schemes.**

Main References

1) R. Gonzales, R. Woods, “Digital Image Processing – Third Edition”, Pearson Prentice-Hall, 2008.

2) R. Fisher, S. Perkins, A. Walker, E. Wolfart, “Hypermedia Image Processing Reference”, Wiley, 1996 (http://homepages.inf.ed.ac.uk/rbf/HIPR2/)