Image Procesing and Computer Vision Notes

by Mattia Orlandi

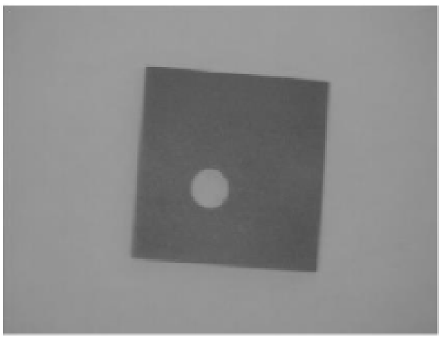
3. Intensity Trasformation

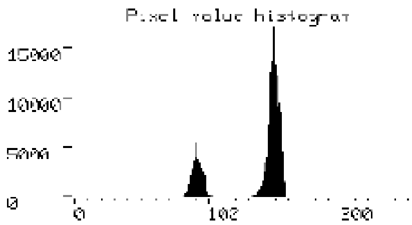
3.1. Gray-Level Histogram

- Intensity Trasformations, or **Point Operators**, are image processing operators aimed at enhancing the quality (e.g. contrast) of input image, which rely on the computation of gray-level histogram (intensity histogram) of input image.
- The **gray-level histogram** is a **function associating to each gray-level** the **number of pixels taking that level** in the image.
- Straightforward computation:

```
int histogram[256];
...
for (int i = 0; i < N; i++)
  for (int j = 0; j < M; j++)
    histogram[image[i][j]]++;
```

- The histogram will be affected by noise:





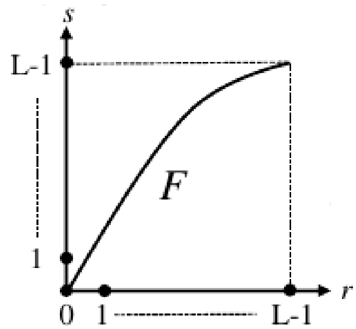
- It provides useful information on the image content, but it **does not encode** any **information related to spacial distribution** of intensities.
- **Normalization** of histogram entries by total number of pixels yield relative frequencies of gray-level occurrences, which can be **interpreted as** their **probabilities** ⇒ **Probability Mass Function** of the **discrete random variable** given by **randomly picked pixels** in the image.

3.2. Point Operators

- Image processing operator which considers only the single pixel.
- It computes intensity of a pixel in output image as a function of intensity of corresponding pixel in input image $\Rightarrow$ it **maps a gray-level into a new gray-level**:

$$s = F(r)$$

where r is the input gray-level and s is the output gray-level.



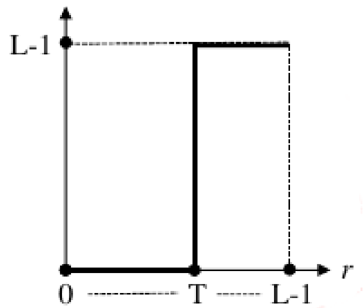
- Any point operators can be implemented as a **Look-Up Table** (LUT):

```
int lut[256];
...
for (int i = 0; i < N; i++)
  for (int j = 0; j < M; j++)
    out_image[i][j] = lut[in_image[i][j]];
```

Thresholding

- Point operator which maps pixels whose intensity is below a given threshold to a certain gray-level (usually black), and those whose intensity is beyond that threshold to another gray-level (usually white):

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



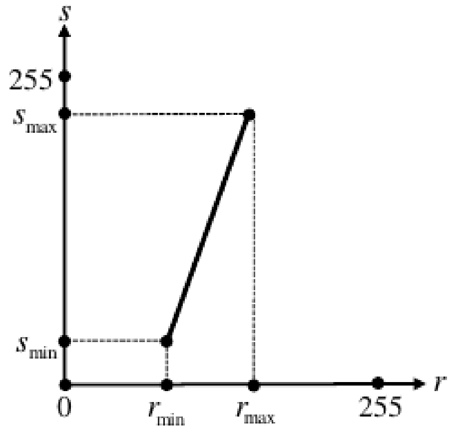
- Particularly **useful to identify objects** in an image (assuming a dark background and a uniform light on the objects).

Linear Contrast Stretching

- A point operator which enhances the contrast of an image.

- Given an image featuring a small gray-level range (poor contrast), it can be enhanced by **linearly stretching the intensities** to span a larger interval:

$$s = \frac{s_{max} - s_{min}}{r_{max} - r_{min}}(r - r_{min}) + s_{min}$$
$$s_{min} = 0,\ s_{max} = 255 \Rightarrow s = \frac{255}{r_{max} - r_{min}}(r - r_{min})$$



- In a scenario in which most pixels lie in a small interval while there exist a few dark and bright outliers, the linear function is ineffective, since it approximates an identity.

- Therefore, r_{min} and r_{max} are taken equal to some percentiles of the distribution (e.g. $5\%, 95\%$), s.t. the pixels outside the interval are neglected and mapped to s_{min} (if $< r_{min}$) or s_{max} ($> r_{max}$).

3.3. Histogram Equalization

- The **purpose of Histogram Equalization** is not to get a flat histogram (which is not possible), but to **make the image use the full range** of gray-levels (achieved by improving contrast).
- It **spreads uniformly pixel intensities** across the whole available range, which improves the contrast.
- Unlike linear stretching, histogram equalization **does not require manual intervention** to set r_{min} and r_{max}.
- HE **maps the gray-levels** of the source s.t. the **histogram** of the target turns out **ideally flat**.
- To find the mapping:

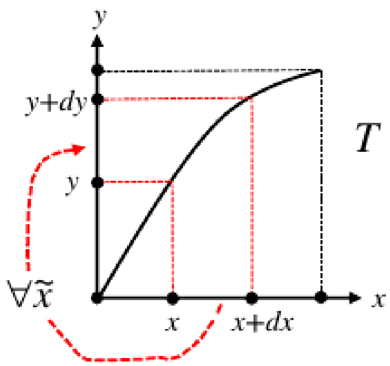
  - Consider a continuous random variable x and a strictly monotonically increasing (i.e. invertible) function T:

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

  - Denote as $p_x(x)$ and $p_y(y)$ the Probability Density Function 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 $y = T(x), y + dy = (Tx + dx)$.



- 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:

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

  where $\frac{dx}{dy}$ is the derivative of inverse function $x = T^{-1}(y)$.

- 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:

$$\frac{dy}{dx} = \frac{dT}{dx}(x) = \frac{d}{dx}\left(\int_0^x p_x(\xi)d\xi\right) = p_x(x)$$
$$p_y(y) = p_x(x)\frac{dx}{dy} = p_x(x)\frac{1}{dy/dx} = \frac{p_x(x)}{p_x(x)} = 1$$

  thus y turns out uniformly distributed in $[0, 1]$.

- In conclusion, by **mapping any continuous random variable through** its **CDF** (assumed strictly increasing) the **result** is a **uniformly distributed random variable**.

- The previous result is **discretized by considering** the **Cumulative Mass Function** (CMF) of the **discrete random variable associated** with the **gray-level** of a pixel, whose **PMF** is **given by normalized histogram**:

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

  where $j \in [0, 1]$, so to map it in $[0, L-1]$ it is necessary to multiply it by $L - 1$:

$$j = \frac{L-1}{N}\sum_{k=0}^{i} h(i)$$

- Due to the several approximations involved, the above function does not perfectly equalize the histogram, but it is **effective in spreading the intensities over a wider range** so as to improve image contrast.