Basic image processing

Point operators

Image Feature Extraction Techniques

Piotr Fulmański

FACULTY OF MATHEMATICS
AND COMPUTER SCIENCE
University of Lodz

Brightness, contrast and histogram

Brightness and contrast

Luminance is a photometric *measure of the luminous intensity per unit area of light travelling in a given direction*. It describes the amount of light that passes through, is emitted from, or is reflected from a particular area, and falls within a given solid angle.

The SI unit for luminance is **candela per square metre** (cd/m2), as defined by the International System of Units. A non-SI term for the same unit is the **nit**.

Brightness is the term for the *subjective impression* of the objective luminance.

For 8-bit pixels, the brightness ranges from zero (black) to 255 (white).

Contrast is the *difference in luminance or colour that makes an object (or its representation in an image or display) distinguishable*.

Histogram

The histogram plots the number of pixels with a particular brightness level against the brightness level.

[img_1_1]

Histogram

This histogram shows us how many available grey levels we have used (or not used).

For example, we can stretch the image to use them all, and the image would become clearer.

This can be considered as a kind of cosmetic operation performed on image to make the image's appearance better. It's true but **making the appearance better, especially in view of later processing, is the focus of many basic image processing operations**, as will be covered in this lecture.

Histogram

Histogram as the most simple image descriptor

The intensity histogram shows how individual brightness levels are occupied in an image;

the image contrast is measured by the *range* of brightness levels.

Histogram

Examples - typical image

[img_1_1]

Histogram

Examples - only white

[img_1_2]

Histogram

Examples - only black

[img_1_3]

Histogram

Examples - only gray [128]

[img_1_4]

Histogram

Examples - black + white (very high contrast)

[img_1_5]

Histogram

Examples - less black + less white (less contrast)

[img_1_6]

Histogram

Examples - less less black + less less white (less less contrast)

[img_1_7]

Histogram

Examples - finally we will get only gray 128

[img_1_8]

Histogram

Examples - finally we will get only gray 128

Reducing contrast you make histogram more concentrated.

Point operators

Point operators

The most basic operations in image processing are *point operations* where **each pixel value is replaced with a new value obtained from the old one.**

Brightness operator

The most basic operations in image processing are *point operations* where **each pixel value is replaced with a new value obtained from the old one.**

Brightness operator

Basic brightness operator

The output pixel $q(x, y)$ result from adding Δ_{bright} to the input pixel $p(x, y)$:

$$q(x, y) = p(x, y) + \Delta_{bright}$$

Note:

Coordinates

[img_1_3]

Point $(0,0)$ is located at the top left corner. You read image in row order, column by column.

[img_1_4]

Brightness operator

Basic brightness implementation

!!! tutu code

Brightness operator

Playing with brightness

Mapping where the output is a direct **copy** of the input

[img_1_5_1]

Brightness operator

Playing with brightness

Mapping for brightness **inversion** where dark parts in an input image become bright in an output and vice versa

[img_1_5_2]

Brightness operator

Playing with brightness

Mapping for **addition**

[img_1_5_3]

Brightness operator

Playing with brightness

Mapping for **multiplication**

[img_1_5_4]

Brightness operator

Playing with brightness

Any **nonlinear** mapping

[img_1_5_5]

Brightness operator

General brightness formula

$I_{out} = o_{bright}(I_{in})$

which for linear mapping takes the form:

$o_{bright}(I) = k \cdot I + l$

or equivalently in pixel form:

$q(x, y) = k \cdot p(x, y) + l$

where $p(x, y) \in I_{in}$ and $q(x, y) \in I_{out}$.

This point operator replaces the brightness at points in the picture according to a **linear brightness relation**.

- The level l controls **overall brightness** and is the minimum value of the output picture.
- The gain k controls the **contrast**, or range, and if the gain is greater than unity, the output range will be increased, this process is illustrated in

For example the image, processed by $k = 1.2$ and $l = -25$ will become brighter and with better contrast.

Brightness operator

General brightness formula

$q(x, y) = k \cdot p(x, y) + l$

- The level l controls **overall brightness** and is the minimum value of the output picture.

- The gain k controls the **contrast**, or range, and if the gain is greater than unity, the output range will be increased, this process is illustrated in

For example the image, processed by $k = 1.2$ and $l = -25$ will become brighter and with better contrast

[img_1_5_6]

Brightness operator

Examples

tutu tutorials

Histogram normalization

In this case, the original histogram is stretched, and shifted, to cover all the 256 available levels.

Note: Keep in mind: you operate on the image not the histogram.

If the original histogram of an old picture (input picture) I_{in} starts at I_{min} and extends up to I_{max} brightness levels, then you can scale up the image so that the pixels in the new picture I_{out} lie between a new minimum output level O_{min} and a maximum level O_{max}, simply by scaling up the input intensity levels according to:

$$q(x, y) = \frac{O_{max} - O_{min}}{I_{max} - I_{min}}(p(x, y) - I_{min}) + O_{min}.$$

Because in most cases you want your output image brightness to range from $O_{min} = 0$ to $O_{max} = 255$ that is the maximum range for images that use a byte per pixel, so the formula simplifies to:

$$q(x, y) = \frac{255 \cdot (p(x, y) - minValue)}{inputBrightRange},$$

where:

- $inputBrightRange = I_{max} - I_{min}$
- $minValue = I_{min}$

Histogram normalization

Examples

tutu tutorials

Histogram equalisation (flattening histogram)

Idea

Histogram Equalization is a computer image processing **technique used to improve contrast in images.** Histogram equalisation aims to change a picture in such a way as to **produce a picture with a flatter histogram**, where all levels are equiprobable.

https://en.wikipedia.org/wiki/Histogram_equalization

Histogram equalisation (flattening histogram)

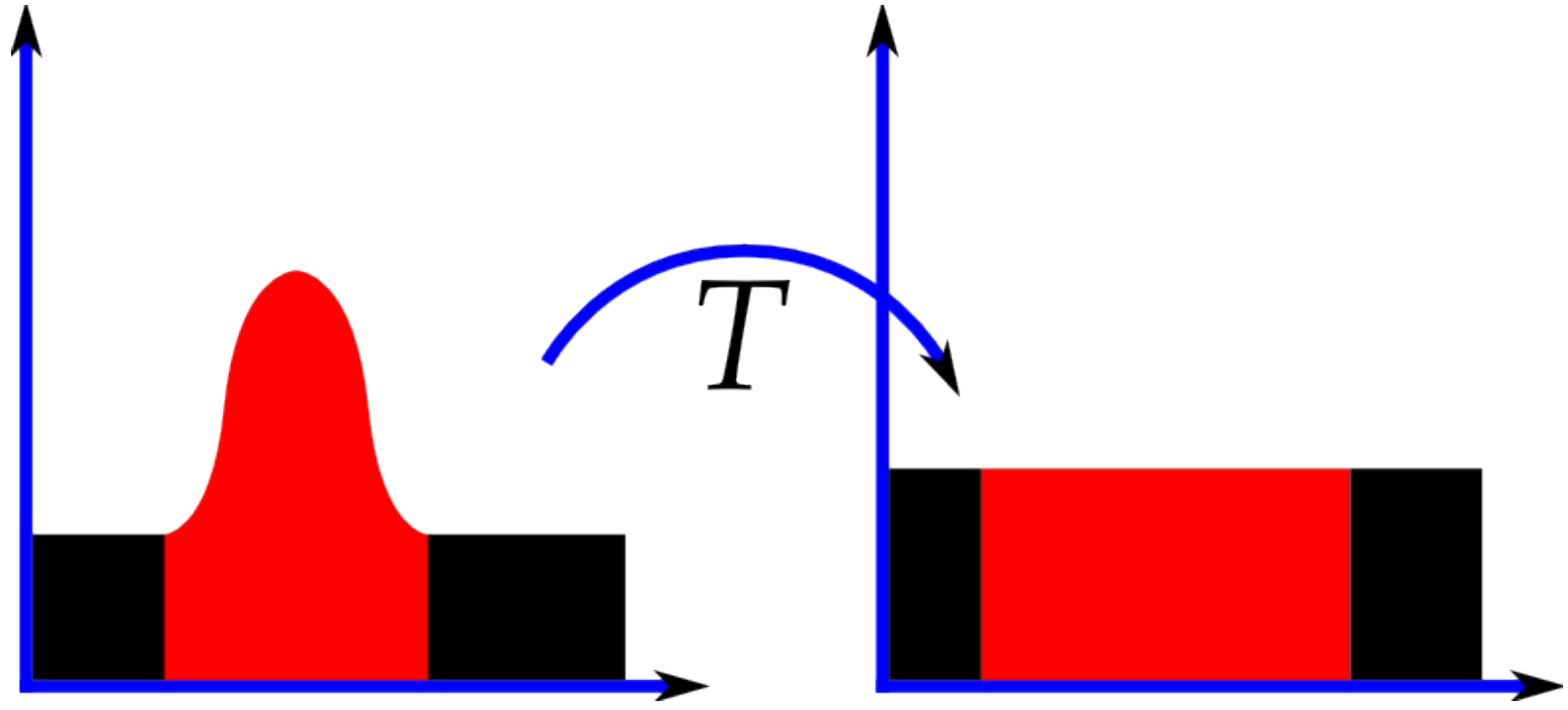
Idea

Flatt histogram and consequences

[img_1_6_1]

Histogram equalisation (flattening histogram)

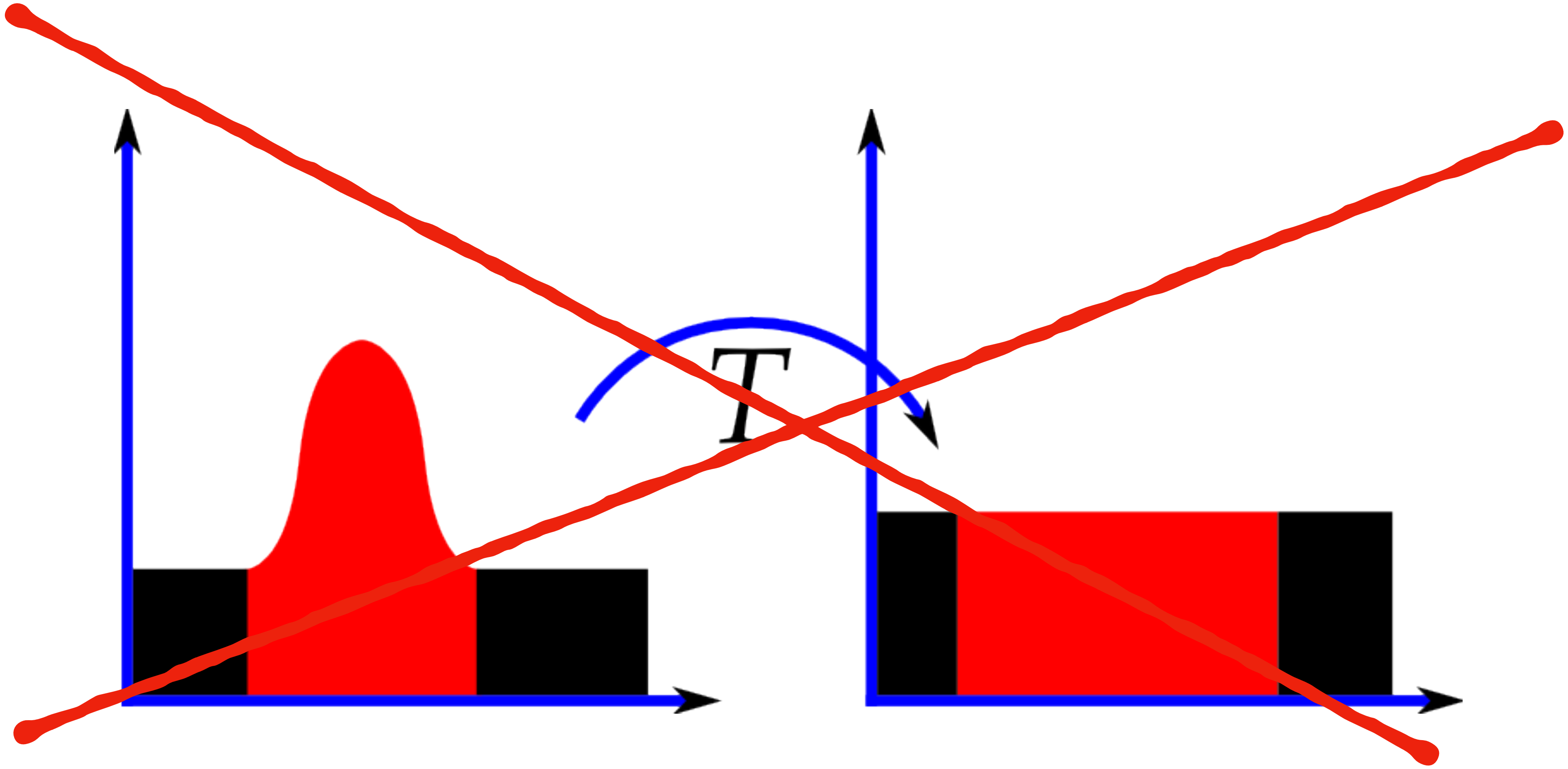
Idea



???

Histogram equalisation (flattening histogram)

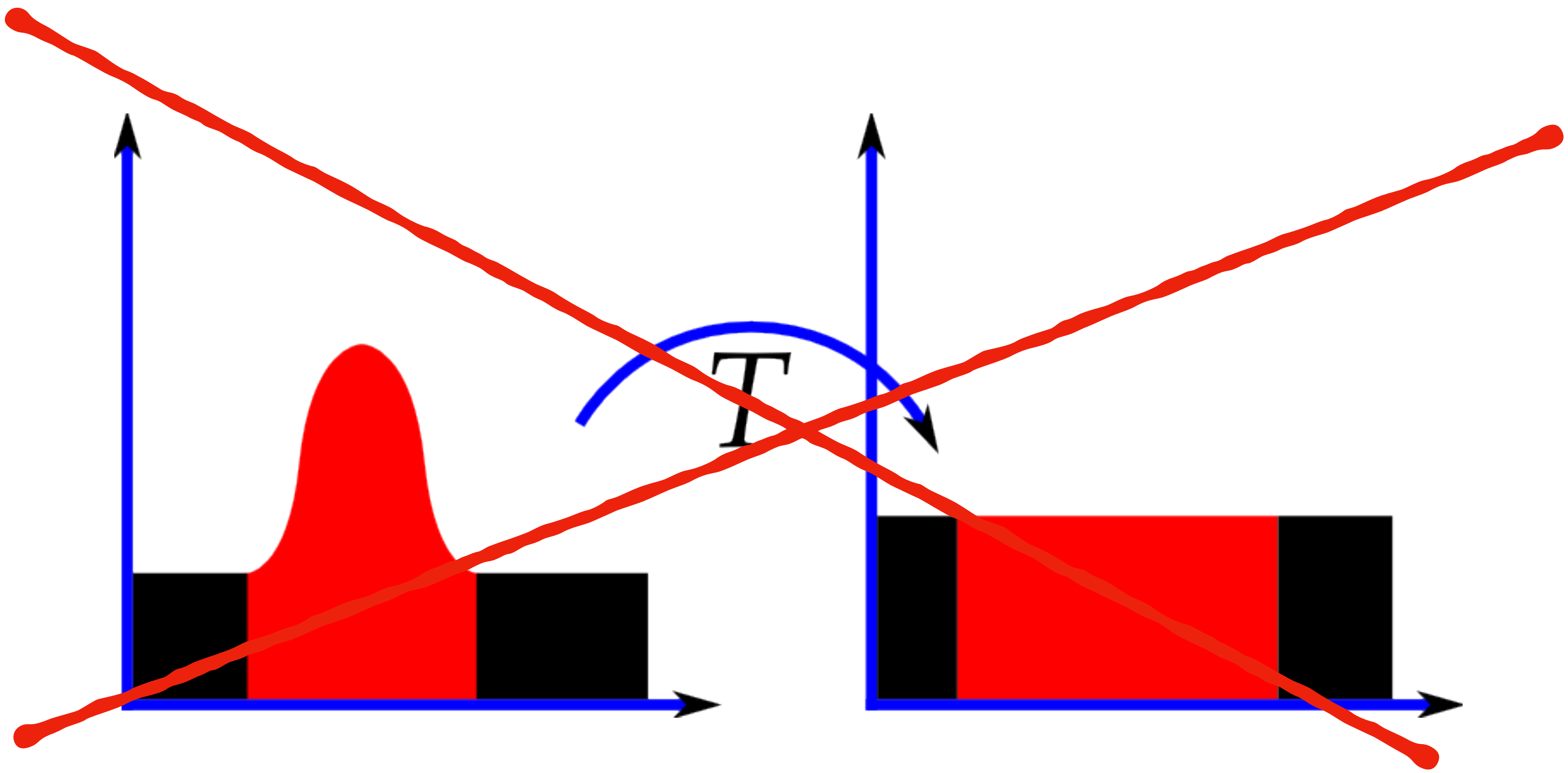
Idea



NO!!!

Histogram equalisation (flattening histogram)

Idea



NO!!!

Flattening histogram doesn't mean fattening histogram!

Why?

Histogram equalisation (flattening histogram)

Idea

Why?

Flattening histogram effect (uncontrolled brightens change; multiple possible results!!!)

[img_1_6_3]

Histogram equalisation (flattening histogram)

Idea

Flattening histogram doesn't mean fattening histogram!

This means changing the histogram H so that the cumulative histogram H_{cum} looks as if histogram H is a flat histogram.

Histogram equalisation (flattening histogram)

Idea

In order to develop the operator, we can first inspect the histogram:

[img_1_6_4_1]

Histogram equalisation (flattening histogram)

Idea

Calculate cumulative values for histogram components:

Component	Value	Cumulative value
130	6	6
140	10	16
150	4	20
160	7	27
170	5	32

Histogram equalisation (flattening histogram)

Idea

Because cumulative histogram H_{cum} should looks as if histogram H is a flat histogram, it means that **calculated cumulative values should lie on the flat histogram cumulative histogram line**

Component	Value	Cumulative value
130	6	6
140	10	16
150	4	20
160	7	27
170	5	32

Histogram equalisation (flattening histogram)

Idea

[img_1_6_4_2]

Histogram equalisation (flattening histogram)

Idea

Move histogram component to get its cumulative histogram which looks as if it is a flat histogram

[img_1_6_4_3]

Histogram equalisation (flattening histogram)

Idea

How to get new "positions" for histogram components?

You have to find an inverse function; a function that "reverses" *typical histogram* function transforming possible gray levels to the number of pixels (per gray level).

Histogram equalisation (flattening histogram)

Idea

For flat histogram you can find *typical histogram function* doing simple math:

1. Two-point form equation of a line defined by two points (x_0, y_0) and (x_1, y_1) lying on that line is given by the formula:

$$\frac{x - x_0}{x_0 - x_1} = \frac{y - y_0}{y_0 - y1}.$$

2. Because in our case

$$(x_0, y_0) = (0,0)$$

$$(x_1, y_1) = (255, C)$$

where C is a highest cumulative value, so:

$$\frac{x}{255} = \frac{y}{C}$$

3. Finally you get:

$$y = \frac{C}{255}x,$$

where x represents histogram components (typically from 0 to 255) and y represents cumulative histogram value (from 0 to sum of all pixels for a given image).

Histogram equalisation (flattening histogram)

Idea

Equation for the **inverse** *typical histogram function* takes the form:

$$x = \frac{255}{C} y,$$

where x represents histogram components (typically from 0 to 255) and y represents cumulative histogram value (from 0 to sum of all pixels for a given image).

Histogram equalisation (flattening histogram)

Idea

Now you can do some calculations to get new "positions" for histogram components. In our case $C = 32$ so

$$x = \frac{255}{C}y = \frac{255}{32}y = 7{,}96875y = 7{,}97y$$

Component	Value	Cumulative value	New "position"
130	6	6	7,97 * 6 = 47,82
140	10	16	7,97 * 16 = 127,52
150	4	20	7,97 * 20 = 159,4
160	7	27	7,97 * 27 = 215,19
170	5	32	7,97 * 32 = 255,04

Histogram equalisation (flattening histogram)

Idea

Compare computed new positions with our intuition

again [img_1_6_4_3]

Histogram equalisation (flattening histogram)

Idea

One negative aspect of this procedure is that you never reach 0 component.

It would be nice to stretch histogram so it starts at 0 and ends at 255.

Again simple math comes to your rescue. To displace components so the lowest component c is located at 0 and the highest stays at 255 you have to again linearly scale numbers. Using two-point form equation of a line defined by two points:

$(x_0, y_0) = (c, 0)$

$(x_1, y_1) = (255, 255)$

you get:

$$\frac{x - c}{c - 255} = \frac{y - 0}{0 - 255}$$

$$y = \frac{c - x}{c - 255} 255$$

Histogram equalisation (flattening histogram)

Idea

In our case $c = 47.82$ and finally you have:

$$y = \frac{c - x}{c - 255}255 = \frac{47.82 - x}{47.82 - 255}255 = -(47.82 - x) \cdot 1.23 = (x - 47.82) \cdot 1.23$$

Component	Value	Cumulative value	New "position"	Updated new "positions"
130	6	6	7,97 * 6 = 47,82	0
140	10	16	7,97 * 16 = 127,52	97.39
150	4	20	7,97 * 20 = 159,4	137.24
160	7	27	7,97 * 27 = 215,19	205.87
170	5	32	7,97 * 32 = 255,04	254.9

Histogram equalisation (flattening histogram)

Idea

Compare computed new positions with our intuition

tutu [img_1_6_4_4]

Histogram equalisation (flattening histogram)

Ready to use formula

All previous transformations can be encompassed in one clean formula [1]:

$$y = \frac{x - c_v}{S - c_v} \cdot (L - 1),$$

where:

- S is the number of pixels,
- c_v cumulative value of the minimum component c,
- L is the number of grey levels used (in most cases 256)
- x is a cumulative value for a given component.

In our case $S = 32$, $L = 256$, $c_v = 6$ so:

$$y = \frac{x - 6}{32 - 6} \cdot 255 = (x - 6) \cdot 9.8$$

Histogram equalisation (flattening histogram)

Ready to use formula

In our case $S = 32$, $L = 256$, $c_v = 6$ so:

$$y = \frac{x - 6}{32 - 6} \cdot 255 = (x - 6) \cdot 9.8$$

Component	Value	Cumulative value	New "position"	Updated new "positions"	New formula
130	6	6	7,97 * 6 = 47,82	0	0
140	10	16	7,97 * 16 = 127,52	97.39	(16-6) * 9.8 = 98
150	4	20	7,97 * 20 = 159,4	137.24	(20-6) * 9.8 = 137.2
160	7	27	7,97 * 27 = 215,19	205.87	(27-6) * 9.8 = 205.8
170	5	32	7,97 * 32 = 255,04	254.9	(32-6) * 9.8 = 254.8

Tresholding

This operator selects pixels which have a particular value:

$$q(x,y) = \begin{cases} 255 & \text{if} \quad p(x,y) > treshold \\ 0 & \text{if} \quad p(x,y) \leq treshold \end{cases}$$

or are within a specified range:

$$q(x,y) = \begin{cases} 255 & \text{if} \quad p(x,y) \in range \\ 0 & \text{if} \quad p(x,y) \notin range \end{cases}$$

where $range = (treshold_{min}, treshold_{max})$.

Bibliography

Bibliography

1. https://en.wikipedia.org/wiki/Histogram_equalization