

IPV week 1 - Lloen Ferreira 3170969, Aleix Mariné 3760480

Besides the example, can you find two images with the same histogram? Write down your script(.m file)and explain what you understand?

First we need to understand clearly what a intensity histogram is and what mathematical operations are involved to be able to check if that situation is possible.

The intensity histograms are a column graph representation showing the counting of the number of pixels of a certain intensity. Being I the intensity and n the number of bits used for encoding the intensity:

$$\{0 \leq I < 2^n\}.$$

Data from the histogram is stored in an array of length n .

As usual in a graphical representation, the dimension of data is simplified using a (hash) function in order to highlight different features of data. In this case we have a set of pixels defined by the variables (x, y, I) . We are going to delete the dimension corresponding to the position of the pixel (x, y) and we will add another dimension corresponding to the number of pixels of that intensity value. Being s the number of pixels of the same intensity the hash function will act as follows:

$$f(x, y, I) \rightarrow (s, I)$$

We also can notice that variable n is not calculated using x or y variables.

Since we are sending information from a three-dimension space to a two-dimension space, collision are assured, making it clearly possible to have the same histogram of different images.

The condition of having a collision (and therefore the same histogram) is given when two images contain the same **set** of pixels, independently of their position, since the counting function will return the same value for each intensity.

Also, we can see that it is impossible to have images of different size and same intensity histogram. The bigger image will always have more counted pixels in some of the columns, making the histograms different.

Due to that, in the following script we will demonstrate our conclusions in two different scenarios in which the previously described condition is met:

- The first situation is if the intensity histogram remains the same if we rotate the image.
- The second situation is if the intensity histogram remains the same if we rearrange randomly the rows.

```
I = imread('pout.tif');
checkHistogram(I)
I2 = randi(100, 100, 100, 100);
checkHistogram(I2)
I3 = imread('example.tif');
checkHistogram(I3)
```

```
function checkHistogram(I)

%Check if the same image rotated has the same histogram
if imhist(I) == imhist(rot90(I, 1))
    disp("The histogram of the same transposed remains the same")
else
    disp("The histogram of the same transposed image is different")
end

%Check if a random rearranged image (within the same row) has the same histogram
if imhist(I) == imhist(I(randperm(size(I, 1)), :))
    disp("The histogram of the a randomized image remains the same")
else
    disp("The histogram of the same randomized image is different")
end
end
```