1 Color Constrast Occurence

Here we will see the different steps that we have to follow to compute the C$_2$O descriptor. There will be three main steps that we will have to study, The transformation through a perceptual space, the computation of the coocurence matrix and the computation of the signature vector.

1.1 Transformation through a perceptual space

First of all, we need to pass our image in the L*a*b* space to get directly a representation of our image in a space which split the luminance and the chromatic information.

For doing that, we first need to pass the image in the XYZ space and choose the correct standard illuminant A.

$$A = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix}$$

That parameter depends on the characteristics of the device used to take the picture: it represents how the device interpret the real colors. In our case, there are many different devices that have been used to take the different pictures so we will have to make a choice. Some standard illuminant are defined by the illumination of the scene like D50 which corresponds to an "horizon light" contrary to D65 which correspond to a "noon light". Ideally, we will have to make the choice for each image but in front of the huge quantity of images, it will be need to make a choice.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = A * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1}$$

After doing this transformation, we can easily transform our image through the L*a*b* space.

$$L^* = \begin{cases} 116 * (\frac{Y}{Y_0})^{\frac{1}{3}} - 16 & si \frac{Y}{Y_0} > 0.008856 \\ 903.3 * (\frac{Y}{Y_0}) & si \frac{Y}{Y_0} < 0.008856 \end{cases} \tag{2}$$

$$a^* = 500 * \left[f(\tfrac{X}{X_0}) - f(\tfrac{Y}{Y_0}) \right] \tag{3}$$

$$b^* = 300 * \left[f(\tfrac{Y}{Y_0}) - f(\tfrac{Z}{Z_0}) \right] \tag{4}$$

1.2 The coocurence matrix

To obtain the concurrence matrix, we have to calculate an image of the difference of color. For that, the program will take the difference between each point of the image and the point which is at a distance following a Δ vector from it. This calculation will has the same result if we compute the difference between the original image and a copy of it translated following Δ. (ColorDiff)

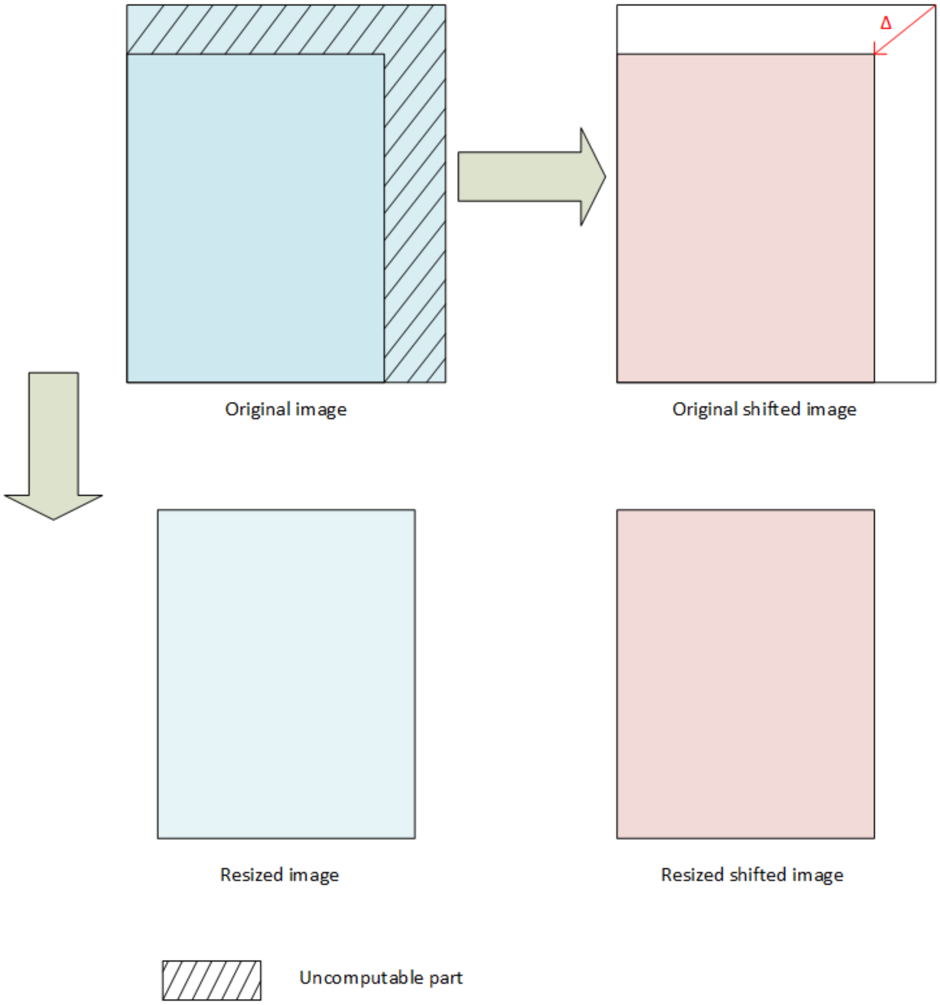

Figure 1: Color Difference illustration

To do that, the program will calculate the equivalent of the translation in horizontal and vertical pixel translation as show below .

$$\begin{aligned} \sin\theta &= dY/\|\Delta\| \\ dY &= \sin\theta * \|\Delta\| \\ \cos\theta &= dX/\|\Delta\| \\ dX &= \cos\theta * \|\Delta\| \end{aligned} \tag{5}$$

The values of $\|\Delta\|$ and θ have to being choose in the aim of get an entire number of pixel.

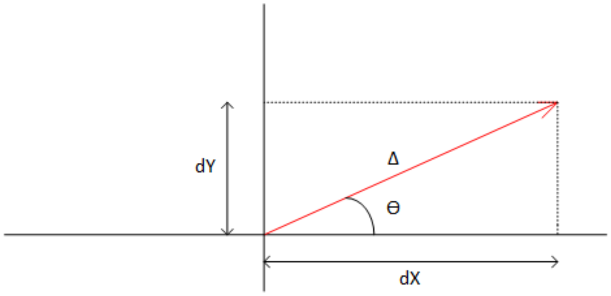

Figure 2: Vector to pixel distance

This computation gives us the C_2O matrix which corresponds to the cloud of point described in the State of art. To validate this part, we have to compare our results with theory. It's known that the L*a*b* space has different component :

- L* which is an achromatic component
- a* which express the opposition between the red and the green
- b* which express the opposition between the blue and the yellow

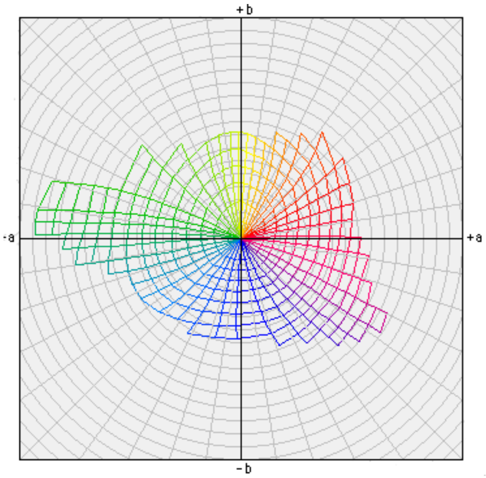

Figure 3: Lab space illustration

We define two test images :

So if we are testing our program on images like the two one shown below which are containing the two same colors opposed on the same component of the L*a*b* space, the result on the difference must show that :

- For the first one which is in green and red, there will be two points spaced on the a* component but at constant levels on the L* and the b* component.
- For the second one which is in blue and yellow, there will be two points spaced on the b* component but at constant levels on the L* and the a* component.

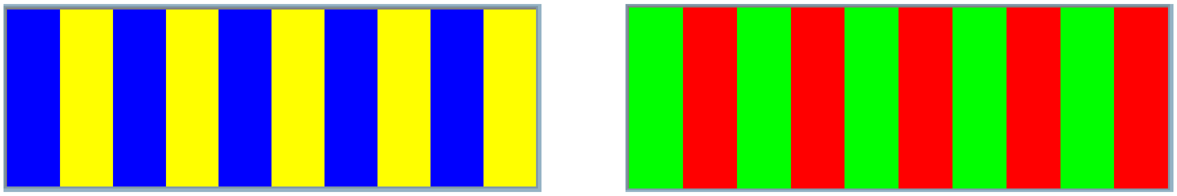

Figure 4: Test images

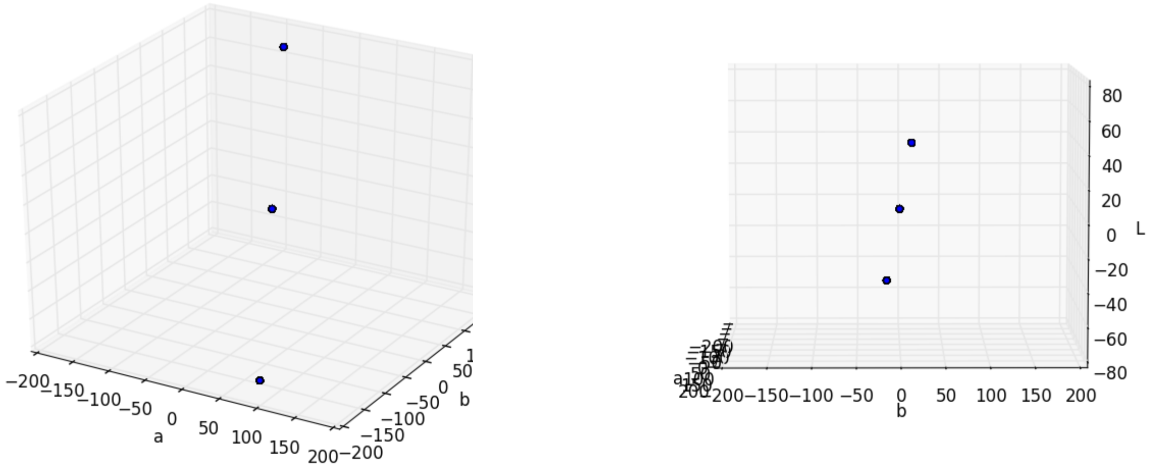

Figure 5: Coorcurence matrix obtained from test images

With this step validate by the theory, we will have to test with more colors to verify that we obtain the right number of points on the coocurence matrix for the corresponding number of colors. (Test to add once the will be done (Appendix ?))

1.3 The signature computing

After doing that, we have to compute the spherical quantization of the probability matrix. For that, we have to transform our difference image for Carthesian coordinates to Spheric coordinates (from these one, the spherical quantization will be easier to compute). So we consider our Lab space like a 3 dimensional repository and for each points, it's calculate the norm of the vector formed by the distance between it and the point (0,0,0), the orientation α formed by the angle between the vector and the a plan and finally the orientation β formed by the angle between the vector and the b plan.

So for each pixel we obtain :

- Δ E the color distance (approximately equivalent to the contrast)
- α the orientation of the Λ vector on the a, b plan (that give us an image of the hue)

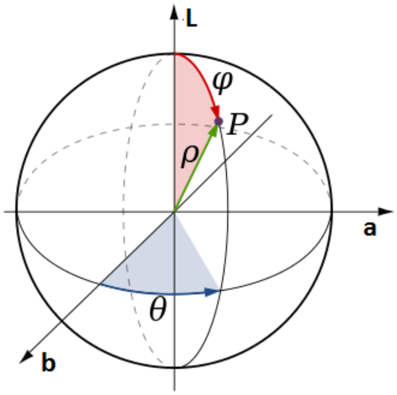

Figure 6: Spherical coordinates

- β the orientation of the Λ vector on the L, b plan (that give us an image of the luminance)

In that way, we obtain our cloud of points in a spherical repository. Once we get it, we need to calculate our C2O signature by quantifying the cloud of point we obtain by a spherical quantization.

It's computed in three times :

There will we 4 interval of radius for the sphere :

$$\begin{aligned} 0 \leq \Delta E < 3 \\ 3 \leq \Delta E < 6 \\ 6 \leq \Delta E < 9 \\ 9 \leq \Delta E < infinite \end{aligned} \tag{6}$$

Each sphere will be split by α interval to concentrate the information as show below in the sectional view following the $(L^*, b^{)}$ plan. Each α interval will measure $\Delta\alpha = 360/6$

This quantization has to be done for each value of β interval which will measure $\Delta\beta = 180/3$.

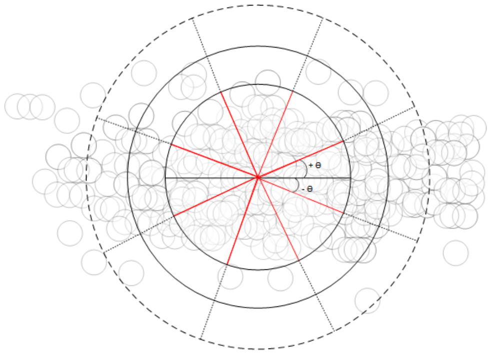

Figure 7: Spheric quantizaton

Once this quantization is done, the signature can be constituted by concatenate the values in a vector following a spiral for each value of β. The histograms obtained for each interval of β will be concatenate to obtain the whole signature.

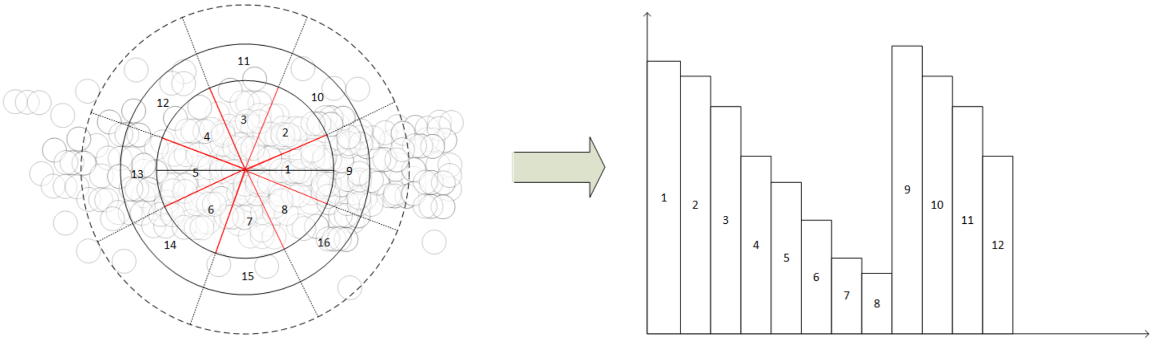

Figure 8: Spheric quantizaton to signature

In this way, we obtain one unique vector of 72 values to describe the image.