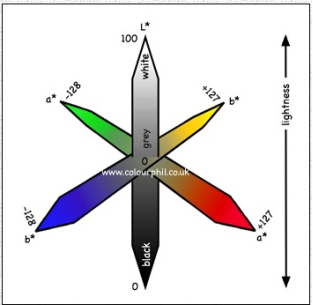
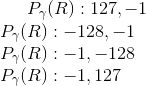
**Progress report from 20th May till 22nd June 2015.**

1-Re-devloping the algorithm needed for computing the features color as the first implementation was inaccurate due to using false prototype values . In this step, the RGB image will be converted into CIELAB color system ”shown below in the figure”, in this color system, the distance in coordinates is fair guide to the difference between two colors, and this color system imitates how the human visual system perceives the color. The next step is, to compute the RGBY values from the CIELAB image for each scale {s2, s3, s4} and then combine them by applying the cross addition, and finally apply Center-Surround Filter OnOff *only* for {3, 7} on the resultant images.

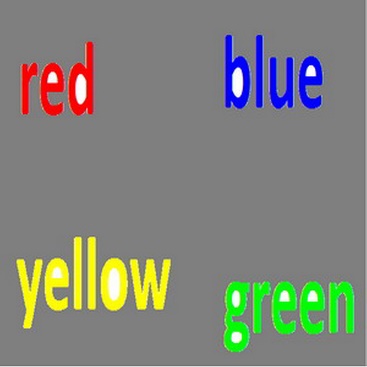


The prototypes used:

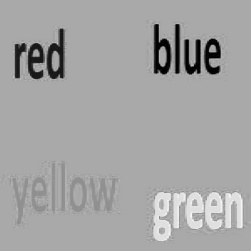


Then, subtract 128 from each pixel value in the L\*a\*b\* color space and finally normalize the image from 0 to 255.

For the following input image:



Here are the output images for R, G, B, and Y respectively:

2- Conspicuity Maps Computations. In this step we try to promote/raise the rare/unique features in the image to increase their influence and suppress the redundant ones, this could be done by computing the uniqueness weight of each map {Intensity (on, off), Orientation (0, 45, 90,135), Color(R, G, B, Y)} and then combine them.

For the following input image:



Conspicuity Map for Intensity:



Conspicuity Map for Orientation:



Conspicuity Map for Color:



3**-**computing the Saliency map, in this step the conspicuity maps {intensity, Orientation, Color} are weighted again using the following uniqueness weight function and then summed up into the final bottom-up saliency map.

**D:\Benutzer\baam1011\Downloads\rare.jpg**

**Where X:** is the map

**m:** is the number of the local maxima that exceeds a pre-defined threshold

For the following input image:



Here is the saliency map:



5- Determining the Features Of Attention “FOA”, first the most Salient Region “MSR” is determined by locating the global maxima in the image, then locating the local maxima that exceeds a pre-defined threshold. Finally, we apply the “seedGrowing Algorithm” recursively for the whole candidate maxima found in the image.

After applying the seedGrowing Algorith:



