**Introduction**

A picture is composed of different attributes (height, width, name, etc.) and a set of pixels. There are different ways to define a color (RGBA, RGB, HSLA, etc.). The pixel in an image with an RGBA (red, green, blue, alpha) color encoding needs 32 bits, 8 bits for each color and for the alpha value (transparency). Through different filters it is possible to only show the red, green or blue parts of an image. In this project we are going to implement different filters applying them directly on pixel data of an image.

### Grayscale

\[ \forall P \in \text{Pixels}, P_{\text{RGB}}^{\text{grayscale}} = P_R + P_G + P_B \]

This filter transforms the original colors in shades of black and white.

### Luminance

\[ \forall P \in \text{Pixels}, P_{\text{RGB}}^{\text{lum}} = 0.2126 \cdot P_R + 0.7152 \cdot P_G + 0.0722 \cdot P_B \]

We tried to optimize the grayscale filter by using the concept of luminance of a color.

### Brightness

\[ \forall P \in \text{Pixels}, \forall C \in [R, G, B], P_C^{\text{brightness}} = P_C + \text{factor} \]

By adding an integer factor to all three colors the image becomes brighter.

### Convolution

Convolution computes the pixels color, based on the pixels surrounding it. Sobel Filter is usually used for edge detecting, and is a composition of multiple filters.

\[ \forall P \in \text{Pixels}, \forall C \in [R, G, B], P_C^{\text{conv}} = \sum_{i=1}^{\text{height}} \sum_{j=1}^{\text{width}} W_{i,j} \cdot P_{C(i,j)} \]

### Negative

\[ \forall P \in \text{Pixels}, \forall C \in [R, G, B], P_C^{\text{negative}} = 255 - P_C \]

The negative filter is an involution function, it inverts the colors of the picture.

### Contrast

\[ \forall P \in \text{Pixels}, \forall C \in [R, G, B], P_C^{\text{contrast}} = (P_C - 128) \cdot \text{factor} + 128 \]

It makes the most luminous color brighter while it darkens the others.

### Threshold

\[ \forall P \in \text{Pixels}, \begin{cases} P_{\text{RGB}}^{\text{threshold}} = 0 & \text{if } P_{\text{RGB}} < \text{factor} \\ P_{\text{RGB}}^{\text{threshold}} = 255 & \text{otherwise} \end{cases} \]

It converts dark colors to black and light colors to white depending on a threshold factor.

### ASCII

We converted the pixels to ASCII (American Standard Code for Information Interchange). Over 16M different colors were changed into 10 characters. The set of characters can be changed dynamically.

**Conclusions**

Besides the processing of the image we learned CSS, HTML and JavaScript. These technologies are supported by all operating systems that can run a Web browser complying with the latest standards. Through out this week learned a lot of new things and we enjoyed it.