A red and black logo

Description automatically generated

Hacettepe University

Computer Engineering Department

BBM415 IMAGE PROCESSING LAB. - 2023 Fall

**Instructor**: Aydın Kaya

**TA**: Burçak Asal

**ASSIGNMENT 1**

Giving Cartoon Effect to Colorful Images

**Student name:**

Hasan Malkoç

**Student Number:**

2200356826

**Giving Cartoon Effect to Colorful Images**

Image filtering is a fundamental task in image processing, used for tasks like image smoothing and edge detection. Smoothing involves removing high-frequency components from an image using low-pass filters like the Gaussian filter. Edge detection aims to identify abrupt changes in brightness using filters like Sobel and Prewitt. Image editing tools, such as Photoshop, use special filters for artistic effects like cartoon or pencil drawing images. In studies like "Real-Time Video Abstraction" by Winnemoller et al., cartoon-like images are created by smoothing a color image with a non-linear filter, extracting edges, quantizing the smoothed image to reduce colors, and combining edges with the quantized image for the cartoon effect.

**Approach**

We’ll be doing edge extraction with given steps in assignment description (Details, Edge Detection) after smoothing the image with either median or gaussian filtering with using allowed libraries. Then we are going to quantize our smoothed image with different values of quantization levels. After then, we are going to combine the extracted edges and quantized images and get a result of cartoon effect on colorful images.

**Experiments:**

**1.Image smoothing**

Median filtering

A collage of buildings

Description automatically generated

A collage of buildings

Description automatically generated

**Result:** More the kernel size increases more the image gets blurred. Also my image gets black and white, I believe this is because of different techniques and applications of scipy.ndimage.median\_filter.

Gaussian Filtering

A collage of buildings

Description automatically generated

A collage of buildings

Description automatically generated

**Result:** More the sigma value more the image gets blurred. Since I got better results with gaussian filtering, I’ll be using this instead of median filtering for the next experiments and result.

**2. Edge Detection**

Threshold Value Experiments:

**A black background with white text

Description automatically generatedA collage of buildings

Description automatically generated**

**A black background with white text

Description automatically generated**

**Results**:I kept the sigma and k values the same and change the threshold. When threshold gets smaller, my implementation gets more detailed edges and I don’t want that for my cartoon effect. So, I might use relatively bigger value for thresholds. (like 15-20, It won’t change too much between 15-200’s)

Sigma Value Experiments:

A black background with white text

Description automatically generated A collage of buildings

Description automatically generated

A black background with white text

Description automatically generated A collage of buildings

Description automatically generated

A black background with white text

Description automatically generated A collage of buildings

Description automatically generated

**Results:** If k and the threshold values remain the same, the larger sigma value may result in fewer detected edges. I can change this value for different pictures in order to get better results related to the edge details in the picture.

k Value Experiments:

A black background with white text and numbers

Description automatically generated A collage of buildings

Description automatically generated

A black background with white text and numbers

Description automatically generated A collage of buildings

Description automatically generated

A black background with white text

Description automatically generated A collage of buildings

Description automatically generated

**Results:** If sigma value and the threshold value remain the same, the larger sigma value may result noise in detected edges. When it gets larger it finds unnecessary edges like in the sky as seen in the picture. So for final results I’ll be choosing smaller k values.

**3. Quantization**

A collage of buildings

Description automatically generated

A collage of buildings

Description automatically generated

A collage of buildings

Description automatically generated

**Results:** A higher quantization level means a larger step size, resulting in fewer distinct levels in the image and brings more coarsely quantized image. Higher level of quantization brings more simplified and low resolution image.

**4. CARTOONIZE IMAGE**

In this step I’ll take the inverse of the estimated edges values and multiply it with the quantized image for each channel.

A collage of buildings and a picture of a city

Description automatically generated

A collage of buildings

Description automatically generated

**Result:** My cartoonized image brought good results where the edges are obvious, but work inefficient for sky scenes in pictures. It gets darker than it should be. Even though my edge algorithm and quantization algorithm gives me the results those they should be, when combining quantized image and edge image together some details in the picture doesn’t bring me the results that I expect. It might be related to changes I do between image colors like RGB to BGR and even changing to GRAYSCALE.