Cartoon effect

Student: **Balus Dan**, gr. 30237

Supervisor: **Pop Cristian Alexandru**

25.05.2018

# Introduction

Description:

* The goal of this project is to implement an algorithm who convert a photo into a cartoon one. The algorithm is designed to provide artistically and comically appealing results on as wide a range of pictures as possible.

Context:

* The user of application select an photo and receive a converted photo namely the picture cartoon.

Motivation:

* My motivation when I worked on the project was that the theme was enjoyable and funny at times.

# State of the art

During the search on the Internet, I came across many methods in which the requirement can be implemented, but many of them required complex or too simplistic operations.

# Proposed method

* Describe your approach (suggestion: design a diagram block containing the main steps in your approach and then detail in text each of the mentioned step)
* The solution approached is taken from ([1], n.d.) and has the following stages:

- Edges

- Colors

- Recombine

A) Edges

Finding smooth continuous contours is an important component to achieving the overall effect. The following steps are taken to provide contour detection that works in an artistically pleasing manner on a wide variety of images. All edge processing tasks are performed with a single-channel grayscale image derived from the luminance values of the input.

1. Median Filter: Before any further processing, a median filter is applied in order to reduce any salt and pepper noise that may be in the image. At this stage, the median filter kernel is a 7×7.
2. Edge Detection: This algorithm uses the popular Canny edge detector. The benefits of using the Canny edge detector instead of a Laplacian kernel is that the edges are all single pixel edges in the resulting image.
3. Morphological Operations: Currently, the only morphological operation employed by the algorithm at this stage is dilation with a small 3 × 3 structuring element. to both bolden and smooth the contours of the edges slightly.

B) Colors

The other important aspect of the cartoon effect is that of blockish color regions. In this branch of the algorithm, the colors are repeatedly smoothed to create homogenous color regions.

1. Bilateral Filter: This filter is the key element in the color image processing chain, as it homogenizes color regions while preserving edges, even over multiple iterations. The bilateral filter works similarly to a Gaussian filter in that it assigns to each pixel a weighted sum of the pixel values in the neighborhood. In this way, a pixel that is close in color to the centroid pixel will have a higher weight than a pixel at the same distance with a more distinct color. This extra step in the weight calculation is important because it means that sharp changes in color (edges) can be preserved, unlike with a simple Gaussian blur.
2. Quantize Colors: The final step in the color image processing chain is to requantize the color palette of the image. This is achieved with each color channel by applying the following formula, where p is the pixel value, and a is the factor by which the number of colors in each channel is to be reduced: Pnew = [p/a] x a. Admittedly, this has the unwanted effect of truncating colors with pixel values close to the maximum, but in practice the absence of max-value colors is hardly noticeable. In this algorithm, a scale of a = 24 was chosen

C) Recombine

Once both the color and edge image processing chains are complete, the only task left is to overlay the edges onto the color image. The final algorithm simply draws on all the contours in black.

* Detail the theoretical concepts / algorithms.

Median Filter:

The median filter is a nonlinear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise (but see discussion below), also having applications in signal processing.

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). ([2], n.d.).

Canny edge detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1.Apply Gaussian filter to smooth the image in order to remove the noise

2.Find the intensity gradients of the image

3.Apply non-maximum suppression to get rid of spurious response to edge detection

4.Apply double threshold to determine potential edges

5.Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges. ([3], n.d.).

Dilatation:

The dilation process is performed by laying the structuring element B on the image A and sliding it across the image in a manner similar to convolution (will be presented in a next laboratory). The difference is in the operation performed. It is best described in a sequence of steps:

1. If the origin of the structuring element coincides with a 'background' pixel in the image, there is no change; move to the next pixel.

2. If the origin of the structuring element coincides with an 'object' pixel in the image, make (label) all pixels from the image covered by the structuring element as ‘object’ pixels. ([4], n.d.).

Bilateral Filter

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

# Experimental results

Before:



After:



Before:

After:

# Conclusions

* I implemented an algorithm who convert a photo into a cartoon one using algorithms like Bilateral Filter, Median Filter, Canny Edge Detector.
* Pro-s and the limitations of my approach:
  + Low time processing
  + A good looking results.
  + Sometimes are too many edges detected.
* Future developments

Color edge and better edge filter.

# Bibliography

* [1] <https://stacks.stanford.edu/file/druid:yt916dh6570/Dade_Toonify.pdf>
* [2] <https://en.wikipedia.org/wiki/Median_filter>
* [3] <https://en.wikipedia.org/wiki/Canny_edge_detector>
* [4] <http://users.utcluj.ro/~tmarita/IPL/IPLab/IPLAB.htm>
* [5]<http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html>
* [6] <https://unsplash.com/>
* [7] [www.youtube.com](http://www.youtube.com)/
* [8] <http://inside.mines.edu/~whoff/courses/EENG510/projects/2013/Anil_slides.pdf>