

Developing an Application to Convert Real Images into Cartoons

By: Benjamin Melone, Eimhin McDonald, Joseph Dooley, Ryan Kane

Summary

The project goal is to create a cartoon version of a real life image. This will be accomplished by applying a graph-based segmentation algorithm to separate different elements of the image, an alpha image matting algorithm for adding any desired background and a custom re-colouring algorithm to achieve a unique effect on image elements of choice. All algorithms will be programmed as a part of this project to better understand them.

Background

The inspiration for this idea comes from Figure 5.12 in the class textbook. The figure shows an example of a rotoscoped image being hand-drawn. Our idea is not to hand-draw a rotoscoped image but to try and achieve a similar effect computationally.

As in this example, we plan to use image segmentation to identify the different objects in the image so that we can isolate the subject from the background. As a starting point we will use the algorithm highlighted in this paper:

Efficient Algorithms for Hierarchical Graph-Based Segmentation Relying on the Felzenszwalb–Huttenlocher Dissimilarity.

By E. Cayllahua Cahuina, J. Cousty, et al. Published April 9 2019:

<https://www.worldscientific.com/doi/pdf/10.1142/S0218001419400081>

For our project we will be doing our own unique implementation inspired by this paper. Our project is more than just an implementation of the graph segmentation algorithm because that implementation is only the first step towards achieving the cartoon effect.

The Challenge

This problem is challenging for many reasons. First, implementing the graph-based segmentation algorithm will involve understanding a research paper and converting that understanding into code. It will be difficult to recreate the performance obtained in the paper, while maintaining the correctness of the algorithm. Next, this project wants to provide an ability to remove the background from the image. This step will need to investigate if an image matting algorithm can be used to remove unwanted segments from the image. This step is something this project hopes to achieve, and it will depend on how difficult it is to apply an image matting algorithm. Finally, this project needs to create a way to make the resulting image look more like a cartoon. A simple way to achieve this is to colour the segments based on the average colour

in the segment, but more interesting approaches will be explored. Facing these challenges will help to learn about segmentation algorithms and heuristic techniques for modifying images.

Goals and Deliverables

We plan to create a working application that can accept any real life image and convert it into a cartoon stylized picture using segmentation techniques and re-coloring algorithms. We hope to add a feature that allows users to select the custom background to be added into the picture using image matting algorithms. We also hope to add a feature that allows users to select different stylizations in addition to the default cartoon style.

Success would involve selecting a real life picture, choosing a background and style, and then having the application output a cartoon styled version of the real image. To validate the success of our project, we will include multiple before and after images of real life to cartoon transformations. The success of these transformations will be measured in two ways. First, by qualitatively by ensuring that each segment from the original image has been uniquely altered to look like a cartoon. Second, by performing speed benchmarks on the application to see if the performance is similar to the performance of the segmentation algorithm described in the paper.

These goals are realistic to achieve in the allotted time. The image segmentation is a very explored topic and is getting updated every year, so finding information and tips for implementing segmentation will be easy. The image matting seems more difficult to understand and apply so it will be achieved if time permits. In conclusion the project seems realistic to achieve in the allotted time.

Schedule for completion

Week of:

Feb 3

- Decide on C++ or Python
- Set up a project template in github
- Read the paper in completion
- Find some samples of segmented images for testing (pre and post segmenting)

Feb 10

- Agree on best practices for version control and code
 - Create basic outline (no code) for implementation of graph segmentation algorithm from paper
 - Investigate desired look for the cartoon effect
 - Learn more about the concepts involved in the graph segmentation algorithm by reading related papers
-

Feb 17

- Start implementation of image segmentation algorithm
 - Outline algorithm for achieving the cartoon look
 - Decide what kind of user interface the app will have
 - Design the application that will use the algorithms to apply them to an image
-

Feb 24

- Continue implementation of image segmentation
 - Start implementing algorithm for achieving cartoon effect
 - (2 people) develop application
-

March 2

- Continue/Test implementation of image segmentation
 - Continue/Test algorithm for achieving cartoon effect
 - (2 people) develop application
-

March 9

- Finalize implementation of image segmentation
 - Finalize algorithm for achieving cartoon effect
 - (2 people) develop application
-

March 16

- Integrate segmentation algorithm with the application
 - Integrate algorithm for achieving cartoon effect with the application
-

March 23

- Test application all together (4 people)
-

March 30

- Create slideshow presentation
 - Finish application (3 people)
-

April 6

- Finalize presentation slideshow
 - Divide the presentation into parts for each person to present
 - Practice presentation as a group
 - Present
-

Project github link: <https://github.com/BMelone/COMP4102-Project>