EC504 Project Report

Image foreground/background segmenta-on using max-flow

Chenxi Li

Documentation

Description of the problem

In this project, I'm trying to use max-flow algorithm to solve image segmentation. The key concept is to utilize the disparity between intensity of neihboring pixels to construct energy flow. Each pixel will be modeled as a node, and there is always a edge between neighboring pixels. Besides, there are two imaginary terminal pixels representing the source and sink of the flow. In this way, image segmentation is formulated as a max-flow problem, and the min-cut of the residual graph will be used as contour that separates image foreground and background.

Relevant References and background materials

"Interactive Image Segmentation using Graph Cuts" by Mayuresh Kulkarni and Fred Nicolls
"Graph Cuts and Fred Nicolls

"Graph Cuts and Efficient N-D Image Segmentation" by Yuri Boykov and Gareth Funka-Lea

Implementation, design decisions

The program is written in python, because there are plenty of existing open libraries for computer vision and matrix manipulation, such as OpenCV, numpy. I decided to build the graph using PyMaxflow, which optimizes the use of memory and GPU a lot. The size of the graph will be at least a few hundred thousand bytes. After my laptop struggled to process a 50x50 image when I built my own graph with primitive 2D arrays, because only one core was used and RAM was running out of space. To process a normal size image, I had to get help from those existing libraries.

Boykov algorithm is used to find the min-cut, at first I used Edmond-Karp with BFS, but it takes forever to find min-cut for a 100x100image, because the graph size is 10^8.

My implementation of max-flow algorithm is based on Boykov's paper mentioned above, as the key was to find the appropriate weights for the edges connecting the pixels.

Features

The finished program has two parts

Part1: Training Gaussian mixture model

There is a simple GUI program for users to select parts of the images as foreground and background. Information about pixels selected will be stored in a text file for later training.

Part2: Separating a image's foreground.

Training starts when user run the segmentation program "segmentor.py." The program will take a image from its directory and place the output image here after it's finished.

As I completed this project individually, there has been a tough time constrain, so I did not develop a standalone CLI, but I did used progress bar for each stage of the graph computation, so user knows how long it takes.

For training, run "python parts_selector.py -i <image_path>"

For segmentation, run "python segmentor.py -i <image_path>"

Supporing Files

Github Repo: https://github.com/ChainZeeLi/max-flow_image_segmentation/blob/master/segmentor.py

Python 2 is required, does not support Python3

To run the programs, user need to install all the required python libraries first.

Required libraries:

sklearn

scipy

numpy

PyMaxflow

Pillow

matplotlib

opencv-python

progressbar

(Easiest way to acquire them is to use pip and just do “pip install <library_name>” in the terminal)

Work Breakdown

I finished the project on my own