

CMSC 426, Image Processing

Project 4: MyRotoBrush

Due on: 11:59:59PM on Tuesday, Nov 22 2016

Prof. Yiannis Aloimonos,
Nitin J. Sanket and Kiran Yakkala

November 11, 2016

The aim of this project is to implement an end-to-end pipeline to do image video object segmentation (segmenting accurate object boundaries in each frame of a video sequence) given the segmentation of the object in the first frame.

In the next few sections, we will detail how this has to be done along with the specifications of the functions for each part.

1 Track Object Bounding Box - 30Pts

As mentioned earlier, you know the object boundary in the first image for each sequence (use MATLAB function `roipoly` to extract this), you can easily obtain the bounding box from this mask using the MATLAB command `regionprops`. Your task is to track the bounding box across all frames in the video sequence (note you are given each frame as an image and not a video). Plot a red rectangle where you think the object is in each frame and make a video out of these frames. For this part you can use any available code with appropriate citation and detailed explanation of the algorithm with step-wise outputs in your report.

2 Track Exact Object Boundaries - 70Pts

Your task in this section is to track exact object boundaries, i.e., the contour/outline of the object. A sample output is shown in Fig. 1. For this part you can follow any paper and you'll have to implement most part (at-least 50%) of the code. You need to highlight the object boundaries in green color in each frame (along with the red box from part 1). However you are free to use codes for the following tasks: Computing Optical Flow, Kalman Filter, Particle Filter, Mean-Shift Tracker, KLT Tracker, Active Contours (Chan-Vese Segmentation), GrabCut, any super-pixel segmentation code, Gaussian Mixture Models and so on. If you have a query about "if you can use built-in or third party code" for something feel free to ask the TAs.



Figure 1: Example of tracking exact object boundary. Taken from <http://val.serc.iisc.ernet.in/seamSeg/seamSeg.html>

Here are some links to codes which might be useful for both parts are you are allowed to use them:

1. Optical Flow

MATLAB's in-built: <https://www.mathworks.com/discovery/optical-flow.html>

Michael J. Black's code: <http://cs.brown.edu/~black/code.html>

Optical Flow using Neural Networks: <http://damienteney.info/cnnFlow.htm>

Ce Liu's PhD Optical Flow code from MIT: <https://people.csail.mit.edu/celiu/OpticalFlow/>

2. Mean-Shift Tracker

Sylvain Bernhardt's code: <https://www.mathworks.com/matlabcentral/fileexchange/35520-mean-shift-video-tracking>

Rajshekar Kumar's code: <https://www.mathworks.com/matlabcentral/fileexchange/45066-designing-and-implementation-of-highly-efficient-object-tracking-system-using>

3. KLT Tracker

MATLAB's in-built: <https://www.mathworks.com/help/vision/ref/vision.pointtracker-class.html>

Yazan Jadaa's code: <https://github.com/yjadaa/KLT>

4. Active Contours

MATLAB's in-built: <https://www.mathworks.com/help/images/ref/activecontour.html>

Shawn Lankton's Code: <https://www.mathworks.com/matlabcentral/fileexchange/>

[19567-active-contour-segmentation](#)

5. GrabCut

Xiuming Zhang's GrabCut Code: <https://github.com/xiumingzhang/grabcut>

6. Super-pixel Segmentation: MATLAB's in-built code for SLIC Superpixels: <https://www.mathworks.com/help/images/ref/superpixels.html>

Greg Mori's SLIC Superpixels code: <https://www.cs.sfu.ca/~mori/research/superpixels/>

Peter Kovesi's SLIC Superpixels code: <http://www.peterkovesi.com/projects/segmentation/>

Rama Chellapa's Entropy Rate Superpixel segmentation code: <https://github.com/mingyuliutw/ers>

7. Other useful Matlab functions:

`gmdistribution`, `bwmorph`, `watershed`, `regionprops`, `bwboundaries` and other computer vision and image processing toolbox functions.

3 Philosophy behind the design of this project

Video object segmentation is extensively used in augmented reality, hollywood, computer graphics imagery and Adobe is one of the companies at the forefront of this research. Your job here is to implement your own version of Adobe's RotoBrush.

4 Some Useful Papers

Try to implement the algorithms given in the following papers or try to find your own, these papers should serve as a starting point. You don't have to implement the complete paper, talk to the TAs to see how a paper can be simplified or come up with your own ideas to simplify stuff. We know that all of you are not familiar in reading papers, the TAs can help you with this. Here are the papers: [1] (We would highly recommend this paper as we think it's simple and provides good result), [2].

5 Starter Code

There is no starter code for this project.

6 Submission Guidelines

The Data Folder has 5 sub-folders named **Frames*** where * goes from 1 to 5. Each Frame sequence has frames number from 1 though total number in .jpg format. The images are rescaled such that you can run your codes fast. Feel free to make them smaller if you want. If you want a higher resolution image for some reason contact the TAs. Your job is to plot

a red bounding box (part 1) for each frame of all sequences of where your object has been tracked. In addition to the red bounding box, you need to plot a green object boundary (part 2) for each frame of all sequences. Now create a video for each of the sequences and name them as `VideoOutput*.mp4` where `*` goes from 1 to 5. Please make sure the format is mp4 so we won't have any codec issues. What object you need to track in each sequence is mentioned next.

1. **Frames1**: Track the turtle. (Friend's Video)
2. **Frames2**: Track the motorcycle and motorcyclist. Taken from <https://www.youtube.com/watch?v=jhxCJLm4C-0>
3. **Frames3**: Track the gymnast. Taken from <https://www.youtube.com/watch?v=gMNCwgXtQIw>
4. **Frames4**: Track the powerlifter only (without the weights). Taken from <https://www.youtube.com/watch?v=EiqEFUFM-KI>
5. **Frames5**: Track the lizard along with it's tail. Taken from <https://www.youtube.com/watch?v=Mp1R4Lxoj5c>

We want you to write a very detailed report with a **LOT** of images (step-wise outputs if possible, after all this is a computer vision project). Submit your code (all the codes with package dependencies) along with the report in pdf format and output videos in mp4 format in a .zip file with the naming convention `YourDirectoryID_proj4.zip` onto ELMS/Canvas (**Please compress it to .zip and no other format**). You should also include a detailed README file explaining how to run your code, if you use third party packages which are hard to install, please write guidelines on how to use them. If we are unable to run your code we might ask you to come and run the code during our Office Hours.

If your code does not comply with the above guidelines, you'll be given **ZERO** credit.

Also make a presentation if you want to present it in the class **Note: Every student should present AT LEAST once in the class (you can volunteer to present for more than one project) and can choose to do for any of the Projects from 1 to 4. Good presentations will receive cookie points.**

Also, this is the last project you can use any late days you have.

7 Collaboration Policy

You are restricted to discuss the ideas with at most two other people. But the code you turn-in should be your own and if you **DO USE** (try not it and it is not permitted) other external codes/codes from other students - do cite them. For other honor code refer to the CMSC426 Fall 2016 website.

DON'T FORGET TO HAVE FUN AND PLAY AROUND WITH IMAGES!.

Acknowledgements

This fun project was inspired from Nitin's research at Indian Institute of Science.

References

- [1] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics (TOG)*, volume 28, page 70. ACM, 2009.
- [2] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 376–383. IEEE, 2014.