Geometric and 3D Computer Vision
Assignment 1

Filippo Bergamasco

12/10/2022

Image segmentation

The goal of this assignment is to perform automatic foreground/background segmentation to extract the silhouette of a target object from a video file. This task is part of the operations required for the final project of the previous year's course (more info **here**).

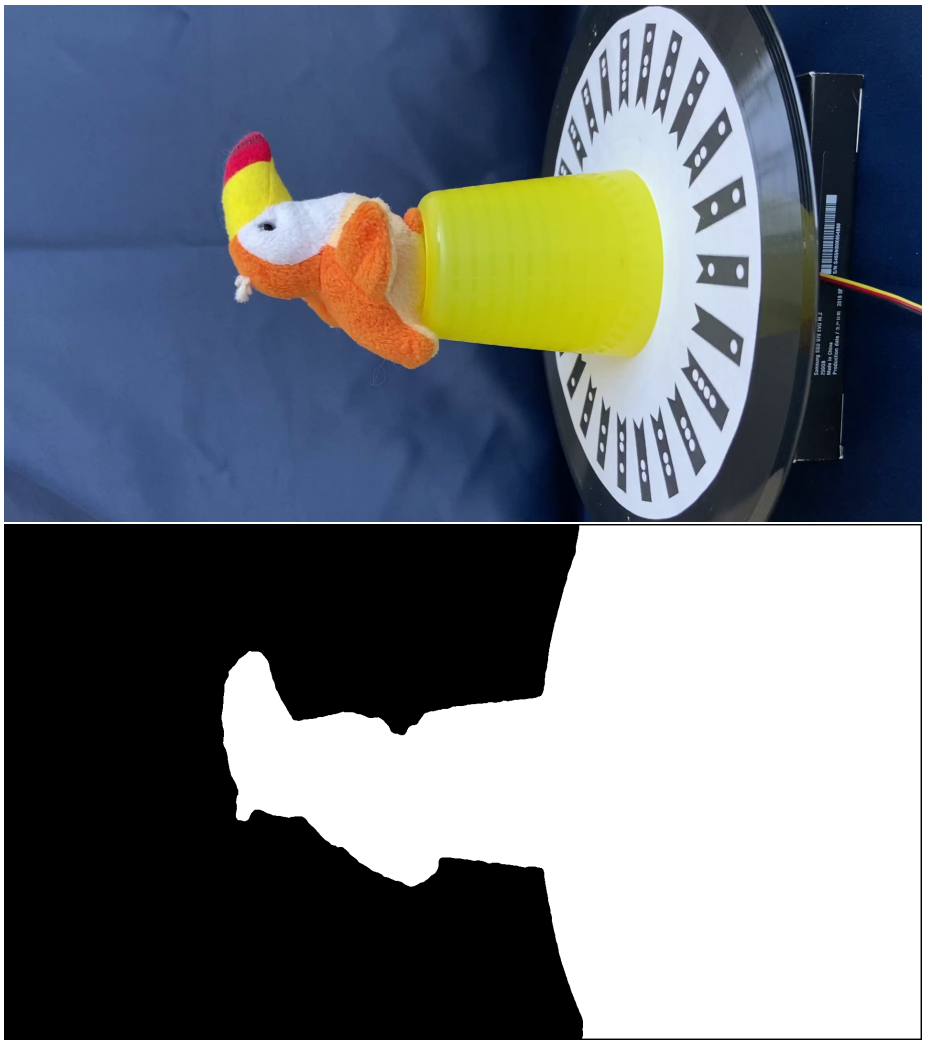
Begin by downloading the dataset from:

https://www.dais.unive.it/~bergamasco/teachingfiles/G3DCV2022/data.7z

In the package you will find video sequences of 4 objects on a rotating turntable (`obj1.mp4` ... `obj4.mp4`).

Your task is to create a Python program (or notebook) that:

- opens one of the video files cited above
- computes a binary thresholded image for each frame in which the white pixels correspond to the object in the turntable and the black pixels to the background
- saves the thresholded images onto a video named `obj1_mask.mp4` ... `obj4_mask.mp4`
- evaluates the results against a set of ground truth images.



Segmentation methods

Your program should implement two different segmentation approaches, selectable before the execution:

1. seed-based watershed algorithm, implemented by the OpenCV watershed() function and described in [1]
2. a custom algorithm, designed by yourself, combining basic operations like color-space conversion, thresholding, filtering, etc.

Both the methods can use tunable parameters for each video file. For example, the list of seeds (or markers) used by watershed can be hardcoded. However,

such parameters cannot change throughout the processing of the whole video sequence.

Evaluation 

Together with the processing of whole video sequences, you have to evaluate the two approaches against some ground truth images extracted from the original videos:

1. Download the **test dataset** here:

   https://www.dsi.unive.it/~bergamasco/teachingfiles/cvstuff/assignment1_test_dataset.7z

2. Perform the segmentation of each ground truth image with both the methods
3. Compare the result against the ground truth masks.

For comparison, consider as `positive` the pixel classified as object (ie. white) and `negative` the pixels of the background (ie. black). Compute the confusion matrix containing the number of false positives, false negatives, true positives, and true negatives. Then, compute **precision**, **recall** and **accuracy**.

Submission

Package the produced source code in a zip file named `name_surname_assignment1.zip` and submit it via Moodle at the official course page. Please, **do not submit any data related to the project, like video files, images, etc.**

Report is not required, just submit the source code. Please comment every function so that I can clearly understand what you are doing. **I suggest to add a Readme file if you need to provide additional instructions on how to use/run your programs**.

Notes for the exam

During your final oral exam, you will be asked to run your code and discuss the performance of both the approaches. You are free to use any OpenCV function, as long as you fully understand the related algorithms. For example, I can ask you to discuss how the watershed algorithm works. So, ensure to read and understand the paper [1].

For any question, feel free to mail me at filippo.bergamasco@unive.it.

`Last revision: 11/10/2022`

Bibliography

1. Fernand Meyer. Color image segmentation. In Image Processing and its Applications, 1992., International Conference on, pages 303–306. IET, 1992.