Laboratory 2:
Video Segmentation

Video Processing, Winter 2023

Overview

The goal of this session is to practice on visual segmentation. To this end, we will work on:

- Background subtraction for animal motion segmentation.
- Graph-cut for video object segmentation.
- Subspace Clustering for video scene segmentation.

First of all, we must download the codes and supplementary materials from the link *Lab 2 Materials* on Aula Global of *Video Processing.*

FAQs:

- **What do I need to send?** A report with your work, including missing codes, an explanation where you discuss your work, as well as your results and conclusions. Extra codes and analysis are also welcome.
- **How?** A single report is needed. Please, do not send partial reports. Once your work is finished, you should upload it to the *Lab 2 Deliverable* on the Aula Global, in a single zip file. The submitted zip file has to contain your report and your source code (and all files needed to run it, such as images, other dependencies, etc). **Be clear and concise.** Your final score is not directly proportional to the number of pages in your report. The work in this session can be performed by two-people groups.
- **When?** Anytime until 7 February 2022 (10:29 UTC+2) or 8 February 2022 (12:29 UTC+2) for groups P101 and P102, respectively. After that, nothing will be received.

References

- Video-Segmentation slides of the course on *Video Processing.*
- Lazy Snapping algorithm based on graph cut.
- Subspace Clustering for sequential data.
- Semantic and Instance annotations for supervised visual segmentation.

1 Introduction

The main aim of this practice is to get familiar with some algorithms to solve visual segmentation. This practice consists of four parts:

- In part #1 you will implement and analyze a background subtraction algorithm for foreground segmentation.
- In part #2 you will understand the main ideas of a graph-cut algorithm, while user interaction is exploited to solve the problem. Analyze the solution in terms of visual similarities and temporal consistency.
- In part #3 you will implement and understand the main ideas of a video scene segmentation algorithm by applying subspace clustering. Order-varying filters are considered to constrain the solution, and the clusters are modeled by an affinity matrix.
- In part #4 you will study how the annotation step in deep supervised learning approaches is performed.

1.1 First Part

In this task, the goal is to practice on the technique background subtraction that was introduced in class and that we will exploit to solve segmentation and tracking. Go into the folder "BS" and look into the Matlab function **main.m**. To apply the algorithm, we will use the dataset in the folder "data", where some dynamic objects appear along video. Particularly, we can find the video "input_video.avi" and eleven frames from another scenario.

- Tasks [M - Mandatory, O - Optional]:

  M You need to understand what this function does in terms of functionality, input and output. Then you need to implement the missing instructions in the files **main.m** and **extract_object.m**. The location of the code that needs to be implemented is indicated by the text "MISSING CODE HERE".

  M Try to run your code in the provided example (**using the frames**), and report the full trajectory of the animal in a spatial domain. You can see some visualizations about the partial detections, and the corresponding final estimation. Explain the results and provide the optimal threshold.

  M **Considering the video** "input_video.avi", first of all you should extract the frames using the provided functions in the function **main.m**, and then try to solve the problem for different conditions in the input data (changes of illumination, small motion of clouds and/or trees, and so on). Explain your solution in terms of template computation and the corresponding results.

  O Explain how you could extend your method to detect time-varying shadows.

1.2 Second Part

Now, the goal is to practice on segmentation by using a graph-cut algorithm. Go into the folder "GC" and look into the Matlab function **main.m** and the folder "data". Two datasets are provided: 1) a flamingo motion and, 2) a camel one. In both cases, ten images and the corresponding ground-truth solutions are included.

- Tasks [M - Mandatory, O - Optional]:

M You need to understand what this function does in terms of functionality, input and output. The algorithm exploits user interaction in order to obtain the final solution. To this end, you basically have to provide some bounding boxes as samples of both foreground and background. Please, follow the function to know as the corners to encode every bounding box have to be selected. Run the algorithm for the pictures *baby* and *peppers*, provide and comment the results, as well as try to reduce the user interaction as much as possible.

M Alternatively, from the Matlab toolstrip, you can look for the Image Segmenter App. In the Image Processing and Computer Vision section, click Image Segmenter to open the App. Try the graph-cut algorithm but now by including lines or points as user interaction. Provide and comment the results like in the previous point.

M Use the algorithm in the videos *flamingo* and *camel*, considering: 1) a frame-by-frame estimation (no temporal consistency needs to be considered) where user interaction is only provided in the first image and, 2) video estimation where again the user interaction is only provided in the first image. In both cases, you should provide the quantitative error with respect to the ground truth images. You can consider the number of foreground labels correctly labeled to do that. Comment the results and give an explanation of the error in segmentation you potentially obtained in both videos.

1.3 Third Part

Finally, you are going to practice on video scene segmentation by using a subspace clustering algorithm. Go into the folder "SC" and look into the Matlab function **main.m** and the folder "data". This time we use two datasets: 1) a synthetic one and, 2) a real one composed of twelve images.

- **main.m**: This file implements a function to solve the video scene segmentation. Synthetic data generation and error-computation functions are also provided.
- **osc_relaxed.m**: This file implements a function to solve the optimization problem and recover the affinity matrix for segmentation.
- **ncutW.m**: This file implements a function to interpret the affinity matrix into different clusters.
- Tasks [M - Mandatory, O - Optional]:

  M You need to understand what all the functions do in terms of functionality, input and output. The functions *spmtimesd* and *sparsifyc* are implemented in C, and therefore you have to generate the corresponding *mex* files to run them in Matlab.

  M You need to implement the missing instructions in the file **osc_relaxed.m**. The location of the code that needs to be implemented is indicated by the text "MISSING CODE HERE". Basically you should add a second- and fourth-order filter. It is worth noting that in the current version a first-order filter was given as an example.

  M For every type of filter $[1, 2, 4]$, and every level of noise $[0 : 0.02 : 0.18]$, provide the corresponding error as a direct comparison with the ground truth.

  M Change the code to handle real images as input, where you could use for every image its pixel intensities to define data. Use the pictures in the folder "data" in this order $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. In addition to that, you could use the pictures #11 and #12 in your group of five images with the same character. Provide and analyze the results for every case.

O Use a real video you can acquire for video scene segmentation, and use features to encode data.

1.4 Fourth Part

The goal is to use the **labalme** tool to annotate images in the context of video segmentation. This package can be downloaded from the website LabalMe, and is implemented in Python. At least, we can obtain a coarse annotation by this toolbox. Please, follow the installation block point-by-point to run the code.

Go into the folder "SS" and look into the folder "boxing". This time we have up to five images.

- Tasks [M - Mandatory, O - Optional]:

  M You need to obtain the annotations (a pixel-by-pixel annotation) of the first frame in the folder "boxing" in terms of semantic and instance segmentation. To that end, you should explain your particular solution according to your own ideas. For instance, you could propose and analyze a particular application where the provided annotations are a good point to learn the model.

  M According to the last point, do you consider your annotations are enough to solve a panoptic segmentation problem? Please, explain your answer carefully.