Exercise 2: Mean-Shift tracking

Advanced Computer Vision Methods

2024/2025

In this exercise you will implement a simple tracker that utilizes Mean-shift algorithm for finding the region with the best similarity between the region and a stored visual model of the target object. The Mean-shift provides a robust deterministic way of finding a mode of a probability density function.

In the first assignment you will familiarize yourself with Mean-shift algorithm and test it on a simple example, for second assignment you will use the algorithm to write a simple tracker and test it.

Submission instructions

The exercise should be submitted on-line on the course website. The submission should contain a report and the source code. Do not submit the data which was given as a part of the instructions, except you add some data that you collect from other sources. The report is the most important part of the submission, so make sure that you spend enough time on it, after you are done with coding and experiments. Note that a strict page limit of the report is **two pages maximum**. Detailed description of grading can be found at the end of the document.

The submissions should be done by the deadline. Late submissions are possible, however, a strict deadline is one week after the first one. A baseline for the late submission is 70%. After that you cannot submit the exercise anymore. The assistant will review the submissions and provide a feedback within a week after the deadline (unless stated otherwise). Each exercise must be done individually and all submissions will be checked for plagiarism. A student will be notified about the grade in the submission feedback. Passing all five exercises is required to pass this part of the course.

Assignment 1: Mean-shift mode seeking

For the first assignment you will implement mean-shift algorithm that you have heard about at the lectures. To demonstrate the validity of the implementation and to test its properties, use the tabulated "function landscape" that you can load by calling the function `generate_responses_1()` from the file `ex2_utils.py`. Test the methods using different starting positions, kernel shapes and sizes and termination criteria. *Write your observations in the report.*

Keep in mind that from the implementation point of view all that you need is the derivative of a mean-shift kernel from the slides. So, if you say that you are using Epanechnikov kernel, you will implement your mean-shift algorithm using it's derivative – the con-

stant kernel. Also, do not try do derivate kernels numerically, instead look at the exercise slides for their appropriate derivated counterparts.

Assignment 2: Mean-shift tracker

For the second assignment you will implement a tracker that uses the Mean-shift method from Assignment 1. The basic idea of using optimization in tracking scenario is to initialize the method at the last known position of the target and let it converge to (hopefully) correct new position.

Implement a mean-shift tracker (described in paper [1] and on lecture slides) that uses color histogram and histogram backprojection[1] to construct a foreground similarity distribution. Additionally you can also test some improvements, like modeling background color distribution as it is described in [1] (section 6.1). Write the tracker in the separate class, which inherits the Tracker class from `ex2_utils.py`, similarly as in the NCC example.

Test the implemented tracker on the sequences from VOT (use the script `run_tracker.py`). The sequences for different VOT challenges can be dowloaded on these links (choose one challenge):

- VOT13: `http://box.vicos.si/vot/vot2013.zip`
- VOT14: `http://box.vicos.si/vot/vot2014.zip`
- VOT15: `http://box.vicos.si/vot/vot2015.zip`
- VOT16: `http://box.vicos.si/vot/vot2016.zip`

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, May 2003.

[1] Use the attached function `extract_histogram` to assemble a histogram and `backproject_histogram` to get a backprojection using an image and an existing histogram.

Grading

The tasks marked with *Req.* are required to successfully complete the exercise. The number in the brackets represents number of points of other tasks while *Add.* stands for additional tasks which can bring you more than 100 points

- (Req.) Implement mean-shift mode seeking method. Compare different starting points, kernel sizes and termination criteria and report where do they converge. Write your observations of the convergence speed (in number of steps needed) in different setups. Can you speed up the convergence?

- (Req.) Run the tracker on at least 5 sequences from VOT(14) and report tracking performance on these sequences (in terms of number of failures of a tracker).
  For the reference: our basic implementation of a mean-shift tracker fails 38 times in total on the VOT2014 dataset. Average speed is approximately 310FPS on a standard laptop.

- (15) MS mode-seeking: include at least one of your own functions and show convergence there.

- (15) Identify failure cases and discuss how would you improve the tracker. (Do not just list these sequences, but rather try to find what the properties of the sequences and what do they have in common. Think about the reasons for good/bad performance.)

- (20) Find good setting of the parameters and discuss how do they impact tracking performance (number of histogram bins, termination criterion / number of iterations, kernel size, update speed). You can even show graphs: tracking performance w.r.t. parameter values.

- (Add.) Try different color spaces (RGB, HSV, YCbCr, Lab), modeling the background (as background histogram and combine it with foreground)