ETH Zürich
Eidgenössische Technische Hochschule Zürich

Computer Vision
Department of Computer Science

Lab 5:
Image Segmentation

Christos Dimopoulos - 23-941-834
cdimopoulos@student.ethz.ch

November, 2023

Contents

Task 1: Mean-Shift Algorithm

In the first part of the lab exercise, we implement the Mean-Shift algorithm for segmentation of an input image, that illustrates ETH, in the CIELAB color space (downscaled by a factor of 0.5 for faster computations). Mean shift is an unsupervised learning algorithm that is mostly used for clustering. Its main steps for the image segmentation task are described below:

- First of all, we compute the Euclidean distance between a given point and all other points (including current point) that are within a specific radius. For simplicity, here we consider this radius to be $+\infty$ for an easier implementation. This process is implemented in the *distance(x, X)* function, that gets as inputs a specific point x and the set of all image points X.
- Next, in the *gaussian()* function, we compute weight of every point as a Gaussian kernel function of distance:
$$K_N(x) = exp(-\frac{(x - X)^2}{2 * Bandwidth^2}) \quad (1)$$
The role of the bandwidth parameter is discussed later on.
- Finally, in the *update_point()* function, we compute weighted mean of all points (including current point), then update current point with this weighted mean.

The above unsupervised procedure is repeated for 20 iterations. In Figure 1 we see the results of image segmentation, after experimenting with different values of **bandwidth = [1, 2.5, 3, 5, 7]**.

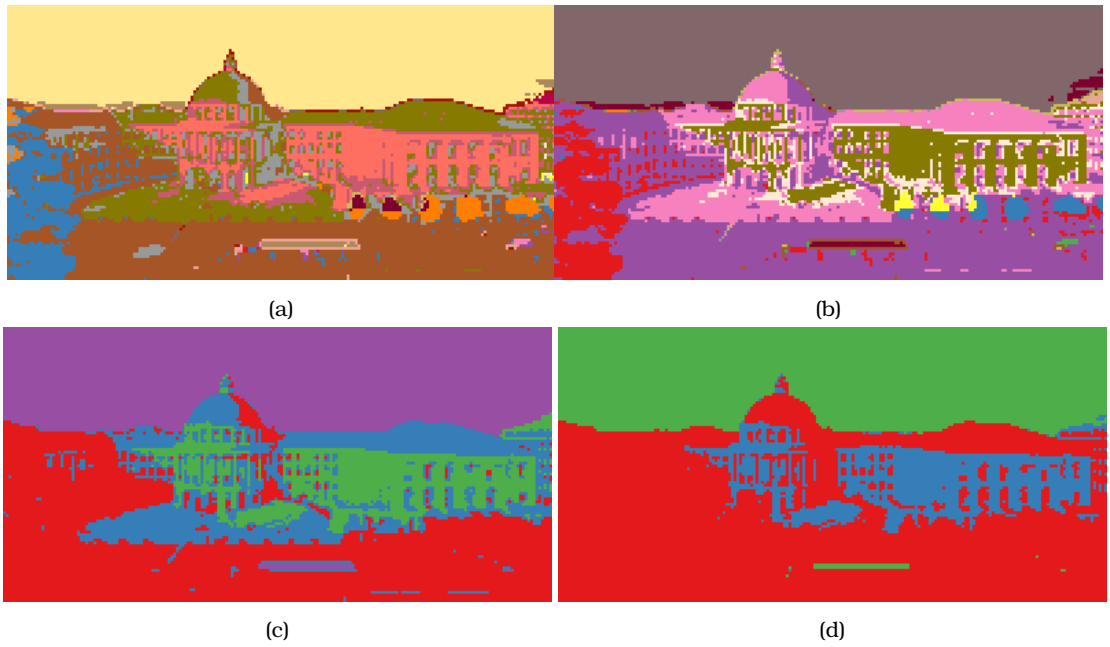

Figure 1: Mean-Shift Image Segmentation results for bandwidth (a) 2.5 (b) 3, (c) 5, (d) 7.

The bandwidth parameter in the mean shift algorithm controls the size of the spatial neighborhood used to compute the mean shift. It influences how far the algorithm looks for points to consider in the update process for each point.

As it is obvious in Figure 1 **smaller values of bandwidth result in the detection of more clusters to segment**. It is worth mentioning that for **bandwidth=1** the code is **not executable**, since *colors[labels]* raises the error *index 266 is out of bounds for axis 0 with size 24*. That is too many clusters are being detected and thus we have not defined enough colors to mark them in the given *colors.npz* file. A possible solution is to add more colors in the given file or increase the bandwidth value.

Nonetheless, a larger bandwidth means a larger spatial neighborhood, and points from a wider region will contribute to the update of the current point. This can lead to smoother and more gradual convergence, but it might also result in a slower algorithm (See Table 1). Also, if the bandwidth is too large, the Gaussian weights calculated in the algorithm may become very small for some points, leading to numerical instability. This can result in extremely small or zero weights, causing division by very small numbers or zero in the update step, leading to numerical instability.

Conversely, a smaller bandwidth restricts the spatial neighborhood, causing the algorithm to focus on a narrower region around each point. This can lead to faster convergence, but it may result in a less smooth output, potentially capturing more detailed structures in the data.

Bandwidth	Time (sec)
2.5	478.88
3	487.50
5	512.57
7	527.252

Table 1: Execution Time of Mean-Shift Algorithm for different bandwidth values.

Ultimately, choosing the appropriate bandwidth is often a crucial aspect of using the mean shift algorithm, and it depends on the characteristics of the data we are working with. It's common to experiment with different bandwidth values to find the one that produces the desired level of smoothing or detail in the output.