Software Atelier: Simulation, Data Science & Supercomputing

Spring 2019

Image Segment Detection

Liudmila Karagyaur karagl@usi.ch
Lorenzo Ferri ferril@usi.ch
Vanessa Braglia braglv@usi.ch

March 11, 2019

Introduction

Nowadays edge detection techniques have several applications. For example, a content based video retrieval is widely required for searching digital information in large databases, in order to improve text based retrieval systems [1]. It also has an important role in the development of self-driving vehicles, that need to perform a real-time feature detection [2], and in medical field, in the analysis of digital images of pathological conditions, such as tumors [3].

The ultimate goal of this project is to find the main features in a video and to be able to follow their trajectories. In particular, we will analyse single video frames using edge detection, which is an image segmentation technique.

There exists a great variety of clustering algorithms that can perform an efficient image segmentation, which is a fundamental step in feature extraction. The first step consists in analysing different clustering algorithms and detect the most suitable one to the problem at hand.

Since the frames could be noisy or blurry, the next stage will focus on possible ways to improve the result of the clustering algorithm, by looking for an optimal sharpening operator. In this step some deblurring techniques will be examined.

Finally, the edge detection will be executed in parallel on each frame of the video, in order to improve the performance.

1 Project tasks

1.1 Image clustering

Clustering ia a technique that allows, given a dataset of points, to classify them into different groups depending on their features. In our case the datapoints that have to be classified are the pixels of the given image. The algorithms that will be examined are k-means, DBSCAN and spectral clustering.

1.1.1 k-means

The k-means algorithm is the simpliest and the most popular clustering algorithm. Its ultimate goal is to partition n initial datapoints into k clusters. It requires that the user provides the number k of clusters that have to be found. The algorithms uses an iterative techniques:,starting from an initial set of random centroids. At each iteration it generates the clusters by minimizing the Euclidean distance D between each point x_i and the centroids c_j:

$$\arg\min_j D(x_i, c_j) \quad j = 1, \dots, k$$

Then, for each cluster C_j a new centroid c_j is computed as the mean of all n_j points x_i assigned to the cluster in the previous step:

$$c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

1.1.2 DBSCAN

Density-Based Spatial Clustering of Applications with Noise algorithm needs two parameters in order to perform clustering on data: the maximum radius (*eps*) of the neighbourhood and the minimum number of points (*MinPts*) in the *eps*-neighbourhood of a point. All the points are classified as:

- **core point**: if it has equal or more than *MinPts* points in its *eps*-neighbourhood

- **border point**: if it has less than *MinPts* points in its *eps*-neighbourhood, but it is in the neighbourhood of a core point

- **noise point**: any point that is not a core or a border point.

Then the datapoints are assigned to clusters in the following way. Any two core points that belong to each other's neighbourhood (located within a distance *eps*) are assigned to the same cluster; any border point that belongs to the *eps*-neighbourhood of a core point is assigned to the same cluster of the core point. Noise points are discarded.

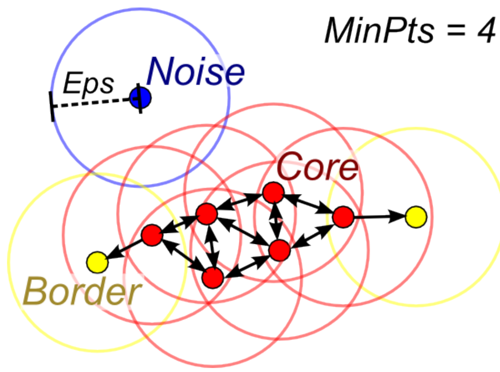

Figure 1: An illustration of DBSCAN algorithm

1.1.3 Spectral clustering

The spectral clustering makes use of the spectrum (eigenvalues) of the adjacency matrix L of the data, that specify all the connections between the image pixels. The eigenvector corresponding to the first eigenvalue $\lambda_1 = 0$ is the vector of all ones and it does not provide information about the graph structure. The second lowest eigenvector, instead, called the "Fiedler eigenvector," is used by spectral partitioning to divide into sub graphs. The pixels that correspond to a negative value in the eigenvector will be in one group and the ones corresponding to a positive value in the other. For this clustering algorithm, unlike the others, require the construction of the adjacency matrix of the image.

1.2 Image deblurring

After the selection of the clustering technique, the next step will be trying to sharpen of the video frames to see if the algorithm works better. The video frames will be deblurred with different operators and the most suitable one will be chosen. This process consists in solving a linear system

$$Ax = b$$

where b is the blurred image, x is the clear image that we want to obtain and A represents the deblurring operator we want to apply. For the solution of the linear system different algorithms will be used (e.g. steepest descent, conjugate gradient).

1.3 Feature extraction and machine learning

1.4 Parallelization

The last step consists in the parallelization of the program to improve the performance.

References

[1] B.V. Patel and B.B. Meshram, *Content Based Retrieval Systems*, International Journal of UbiComp (IJU), Vol.3, No.2, April 2012

[2] H. Cho, Y. Seo, B. V. K. V. Kumar and R. R. Rajkumar, *A multi-sensor fusion system for moving object detection and tracking in urban driving environments*, IEEE International Conference on Robotics and Automation (ICRA), 2014

[3] Ed-Edily Mohd, Azhari,Muhd. Mudzakkir Mohd. Hatta1, Zaw Zaw Htike1andShoon Lei Win, *Tumor detection in medical imaging: a survey* , International Journal of Advanced Information Technology (IJAIT) Vol. 4, No. 1, February 2014