Practical Exercise

19/07/2017

The goal of the exercise is to implement a program in python to visualize a graph representing image descriptors and their relation, together with its evolution along the video sequence.

INPUT:

the path of a video file to analyze

the type of the descriptor (SIFT, SURF, etc.) to extract from the images

OUTPUT:

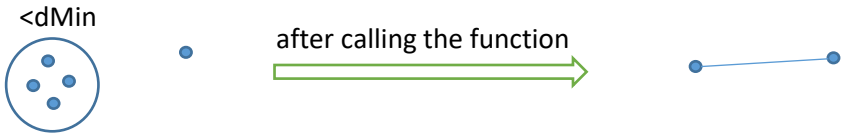
A window showing on the screen the video and the constructed graph on each frame

---

We will go step by step in in increasing order of difficulty to achieve the final goal.

1. Write a python program for opening and displaying a video on the screen
    a. The path of the video can be a global parameter of the python program
    or
    b. More difficult, it can be an input argument given by the user

2. Add to the previous step the detection of descriptors and the visualization of them on each frame of the video
    a. Introduce a new parameter to allow the user to choose the descriptor (e.g. 1/ for SIFT, 2/ for SURF, etc.); in OpenCV there are several description detection algorithms
    b. Again the parameter can be a global parameter of the python program of it can be an input argument given by the user
3. Write a function that, given the image with the descriptor points (nodes) , connect with a line (edges) all points (to form a graph); use this function in the main program in order to visualize a graph of descriptor points for each frame
4. Add to the previous function (question 3) a parameter *dMax* in order to create only connections (edges) between points (nodes) that have a distance less than *dMax*
5. Add to the function wrote at the point (4) a parameter *dMin* in order to keep a single point for a set of points that are distant less than dMin each other's (see figure below for a visual explanation)



6. Modify the function in order to create, for each point only a connection with its nearest neighbor

---

Optionally, if you have time, try to visualize, at each time t, the graph of frame t and the one of frame t-1, side by side, in order to allow the user imagine a possible matching between them.