Offline on Template Matching

Task in brief: In this assignment, you are required to track a reference object (given in **reference.jpg)** in a video file (given in **input.mov**). The output will be another video file **output.mov** showing the location of the reference object using bounding boxes in each frame.

Input Files:

1. reference.jpg
2. input.mov

Output File:

1. output.mov

The input movie file **input.mov** contains multiple frames. Each frame contains the reference image exactly once. Also, the reference image is present in the frames exactly, meaning that you do not have to employ any deformation based template matching here.
Your task is to read the input video file, separate the frames, track the reference object in the frame, use a red-inked bounding box (☐) to mark the location of the reference object in the frame, and merge the frames to produce a output video file **output.mov**.

Methods to implement:

You have to do this entire job using the following methods:

1. Exhaustive search technique
2. 2D Logarithmic search
3. Hierarchical search

Though Method#1 requires every frame to be searched in an exhaustive manner which is pretty straightforward, but for the other two methods, the reference object in the first frame should also be identified using exhaustive search. However, **for all the methods**, searching in a frame should be localized to a window area $[-p, +p] \times [-p, +p]$ centered to the location

where the reference object is found in the previous frame. Definitely, there is no previous frame for the first frame which requires exhaustive search.

Performance Measure:

In addition to the output file **output.mov**, you also have to show the performance of these methods for different window sizes. You can vary the size of *p* (which is defined above) and record some performance metric (say, number of times the reference is searched in an entire frame). Then for all the frames, you have a certain reading for a certain *p*. Taking average, you will get an estimation of that performance metric for a single value of *p*. Then, for another *p*, repeat the entire process. Finally, plot the estimations against *p*. Do this for all three methods. For example, say there are 3 frames in the video. And with $p = 2$, Method *X* has to search the reference frame 2 times in Frame#1, 3 times in Frame#2 and 1 time in Frame#3. Then for $p = 2$, *X* has the value: $(1+2+3)/3 = 2$. Similarly you have to calculate them for Methods *Y* and *Z*.

So to sum up, your program should read **reference.jpg** and **input.mov**; and

Task1: output a file **output.mov**.

Task2: output sufficient numerical data on another file showing comparison amongst the three methods. A sample comparison:

p	Exhaustive	2D Log	Hierarchical
p_1	x_1	y_1	z_1
p_2	x_2	y_2	z_2
.	.	.	.
.	.	.	.

Guidelines:

It is suggested that you finish **Task1** completely first, and then move on to **Task2**. Even before starting **Task1**, try to work with two images. Search a large image and try to find a reference in it. When you are able to do so, then move on to dealing with the video. When you can deal with individual images, then dealing with a video should come to you naturally.

Coding:

Use any language you like. But for submission, put your student ID in the source code files. For image/video processing, you can use any library you like. **However, you MUST DO the template matching codes yourselves.**

Submission Deadline:

Submit by 9 am on 7 November 2020.