COL780:A1: Background Subtraction

Sidharth Agarwal 2019CS50661

February 4, 2023

Libraries Used:

OpenCV and standard python libraries

Run:

To run without foreground pixel aggregation : **python 2019CS50661_A1.py {input_dir} {output}**
To run with foreground pixel aggregation, add **"filter"** as an argument in the end.

Method Used:

I had Y = 1 and I implemented the original approach, of the paper "Adaptive background mixture models for real-time tracking examples" as discussed in class.

Failure Cases:

So my algorithm performs really well in detecting the boundary of the foreground objects. But if the object is moving really slow then it is not able to detect the inner body of the object as we can see the difference between the detection of the person moving in IBM dataset and CAVIAR dataset. Where we saw that my algorithm performed poor on CAVIAR dataset because the person was moving very slow. Whereas in the IBM and Highway datasets, the objects detected were more complete and accurate. This could be because my algorithm starts assuming the inner body of the object to be background.

Other than this the algorithm performed really bad in the Candela dataset where the person who was walking earlier now sits on a bench, and according to our algorithm it becomes background. This is cannot be solved by our core approach of GMM, however we can reduce its affect by using very low value of alpha. But still the way our algorithm works this shortcoming was expected.

Foreground pixel aggregation:

For foreground pixel aggregation I used method of integral images to first calculate the prefix sum over the whole image after background subtraction. Then I chose a window, threshold and moved that window with some stride across the image. If the sum of pixels in that position of window is less than the threshold then I initialize all the pixels to be zero. In this way I removed most of the noise from the image but it also came up at the cost of some removal of the edges of the foreground.

Output Link