

Report - Background Subtraction

By **Anoop, 2015CS10265**

Test Data has been taken from "http://bmc.iut-auvergne.com/?page_id=24"

1. Boring parking, active background
2. Big trucks
3. Wandering students
4. Rabbit in the night
5. Snowy Christmas
6. Beware of the trains
7. Train in the tunnel
8. Traffic during windy day
9. One rainy hour

Algorithm implemented as part of Assignment - Understanding Background Mixture Models for Foreground Segmentation by **P. Wayne Power and **Johann A. Schoonees****

Algorithm implementations for comparison are taken from latest opencv (version - 3.3.0-dev)

1. Gaussian Mixture Models (GMM) -
createBackgroundSubtractorMOG2()
Based on -
 1. **"Improved Adaptive Gaussian Mixture Model for Background Subtraction"** by **Zoran Zivkovic**
 2. **"Efficient Adaptive Density Estimation per image pixel for the task of Background Subtraction"** by **Zoran Zivkovic** and **Ferdinand van der Heijden**
2. K-Nearest Neighbors - createBackgroundSubtractorKNN()

Brief Working

The background subtraction algorithm that has been implemented uses Gaussian Mixture Models to model the pixel intensities. It uses a simple threshold heuristic to classify some gaussians as background gaussians. Next frame pixels are then classified as background or foreground. The weights are updated according to the new frame and matching that has been obtained. In case of no significant match, a new gaussian based on this new pixel value replaces the least match gaussian.

Observations

Note - Videos have been observed for first 5 minutes in case they are longer.

1. Foreground motion -

- Slow motion - My implementation of the algorithm does equally well as compared with the OpenCV implementations.
- Fast motion - OpenCV implementations does capture tiny rapid motions which are not captured by my implementation. Fast motion of large objects is detected in my implementation as well as OpenCV implementations.

2. Shadow Capturing -

OpenCV implementations capture shadows to a great extent and represents them in grey color. My implementation does not get affected by shadows. Also my implementation gives 0/1 output mask.

3. Sudden Illumination -

OpenCV implementations capture sudden illumination even though this is not part of any motion. My implementation does

not get affected with faint illumination changes. Large illumination changes are detected by my implementation.

4. Persistence of detected motion -

KNN based algorithm has a greater persistence of detected motion as compared to GMM based algorithm. Persistence of detected motion is quite less in my implementation of the algorithm.

OpenCV implements algorithms that are based on the paper that I have implemented as part of the assignment. So they have some extra capabilities like shadow detection and persistence of detected motion. But the paper that has been implemented performs well to detect motion which is the main aim.