COP290 TASK 1 REPORT

TRAFFIC DENSITY ESTIMATION USING OPENCV FUNCTIONS

Ritika Hooda (2018CS10375)
Pragya Dechalwal (2018CS10366)

1. Metrics

1.1 Utility

For methods 1-4 (based on queue density estimation):
Firstly, we define error as the absolute values of differences in queue density at a given time, compared to baseline. Utility is defined such that when error is low, utility is high and vice versa. Mathematically,
Error = $|(q - q_0)|$ where q_0 is the queue density in baseline and q is the queue density of the output.
Utility = e^{-Error}
For Extra Credit Problem (based on dynamic density estimation):
We define error as the absolute values of differences in dynamic density at a given time, compared to baseline. Utility is defined such that when error is low, utility is high and vice versa. Mathematically,
Error = $|(d - d_0)|$ where d_0 is the dynamic density in baseline and d is the dynamic density of the output.
Utility = e^{-Error}

1.2 Runtime Metrics

- **Latency:** The time taken by the code for generating the output.
- **Accuracy:** The average utility of the output as compared to baseline.

2. Methods

Method1: Parameters-background image and int N.
processing every x frame i.e. process frame N and then frame N+x, and for all intermediate frames just use the value obtained for N. Processing is done at 5 fps
Method2: Parameters-background image, X, Y.
Reduces resolution for each frame to the specified XxY resolution before processing. Processing is done at 5 fps
Method3: Parameters-background image and int N
split work spatially across threads (application level pthreads) by giving each thread part of a

frame to process. Number of threads is given by N. Processing is done at 5 fps
Method4: Parameters-background image and int N
split work temporally across threads (application level pthreads), by giving consecutive frames to different threads for processing. Number of threads is given by N. Processing is done at 5 fps
Method5: Parameters - string ("sparse" or "dense").
finds dynamic density using sparse or dense optical flow denoted by the parameter. Sparse optical flow is implemented using Lucas Kanade method. Dense optical flow is calculated using Gunner Farneback's algorithm. Processing is done at 3 fps

3. Trade-off Analysis

3.1 Benchmark

Traffic video uploaded on COP290 homepage is the benchmark for trade-off analysis

3.2 Baseline

For methods 1-4 (based on queue density estimation):
Baseline is the queue density estimation in part b.
For Extra Credit Problem (based on dynamic density estimation):
Baseline is the dynamic density estimation using dense optical flow.

3.3 Results

Table 1: Results

Method	Parameters	Time taken (in s)	Utility
Baseline for methods 1-4	-	57.3992	-
Baseline for method 5	-	805.11	-
Method1	N=10	28.09	0.925526
Method2	X=400, Y=600	55.9409	0.889521
Method3	N=10	28.0321	0.950792
Method4	N=10	203	0.800013
Method5	sparse	38.9596	0.946734

Figure 1: Queue density vs time plot for baseline for methods 1-4

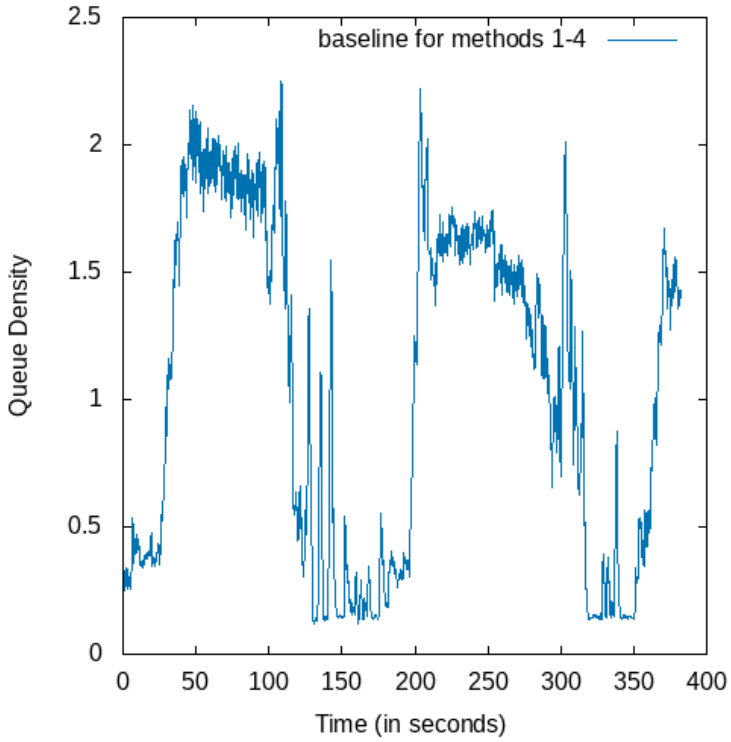

Figure 2: Dynamic density vs time plot for baseline for method 5

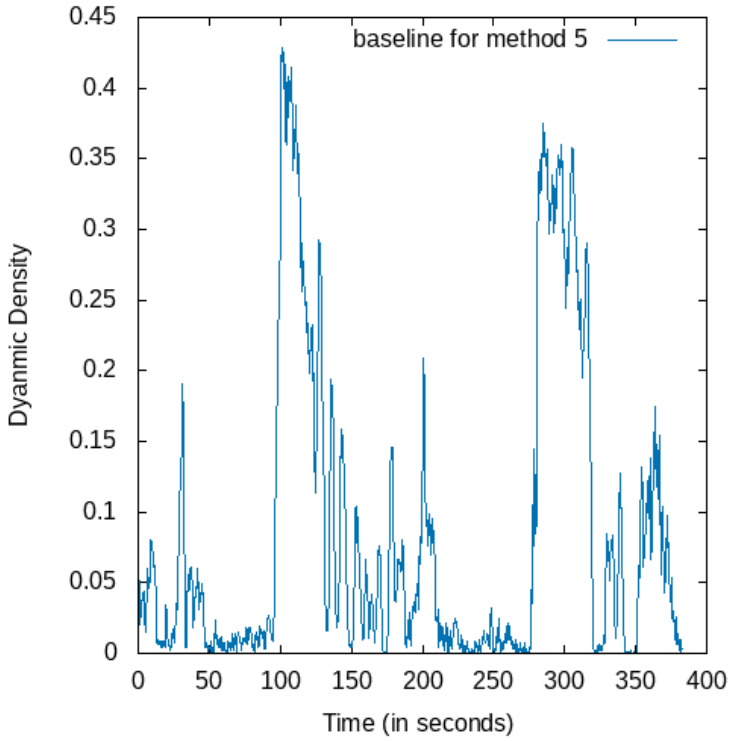

Figure 3: Queue density vs time plot for method 1

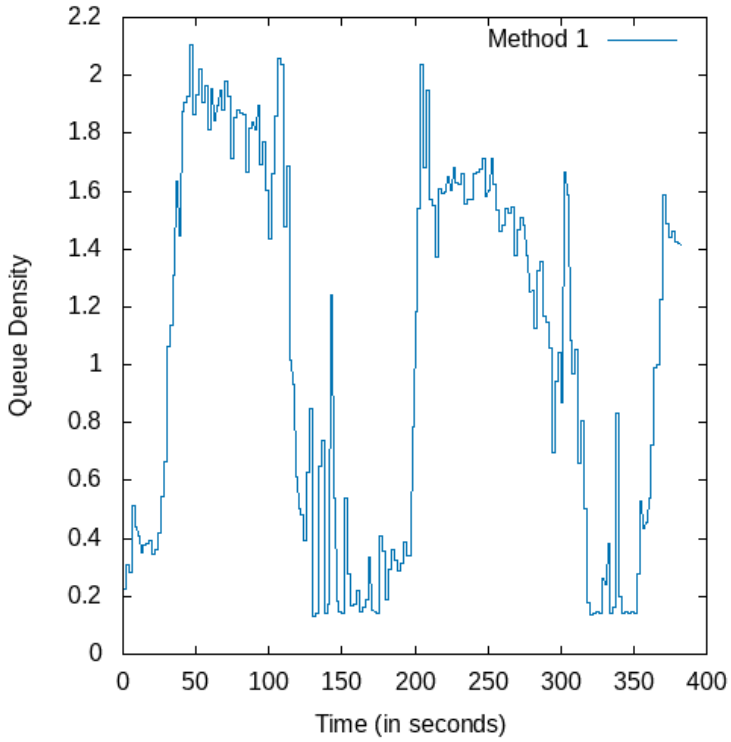

Figure 4: Queue density vs time plot for method 2

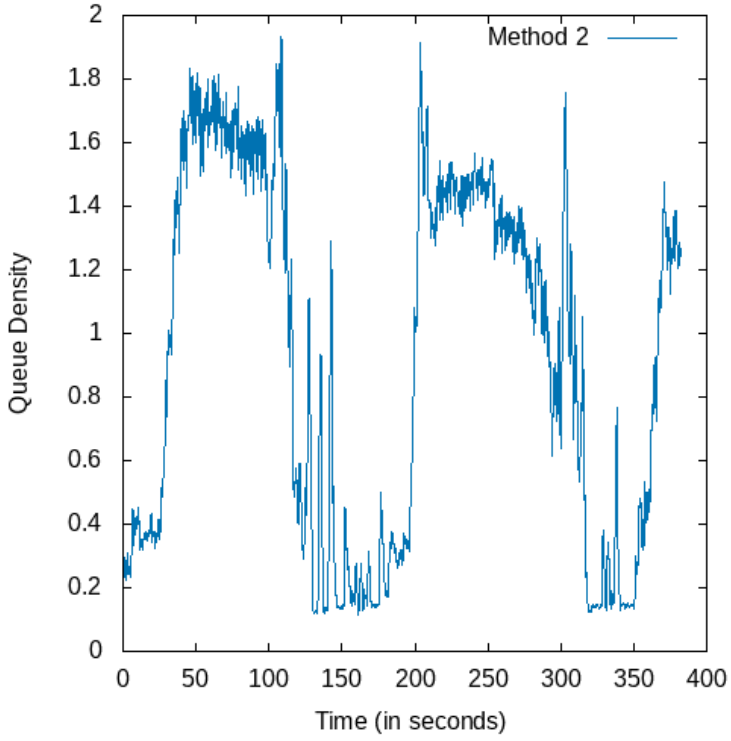

Figure 5: Queue density vs time plot for method 3

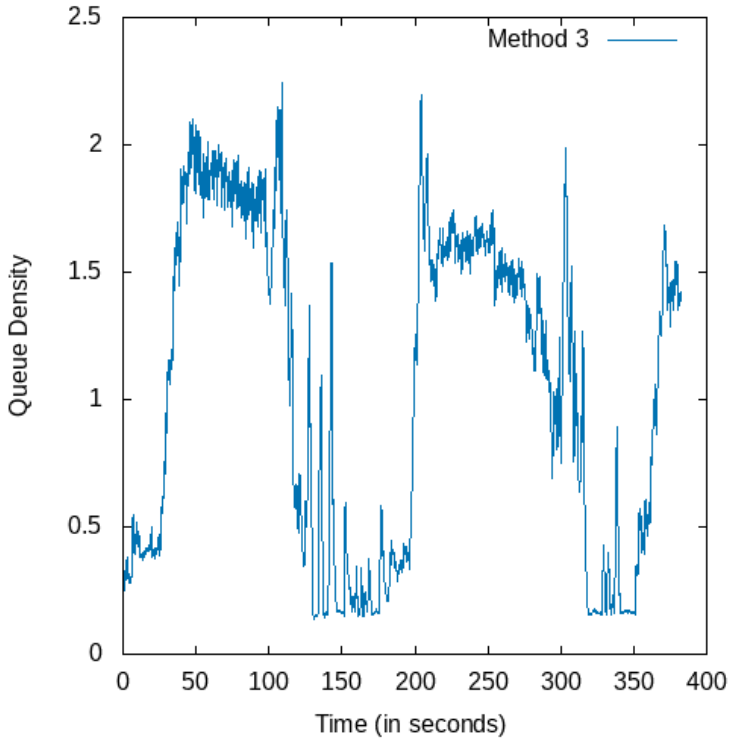

Figure 6: Queue density vs time plot for method 4

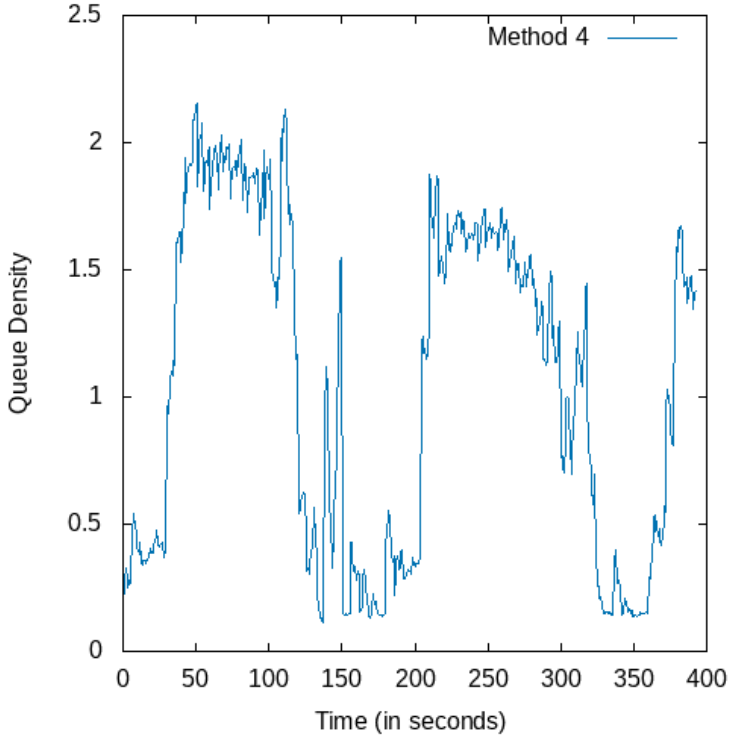

Figure 7: Dynamic density vs time plot for method 5

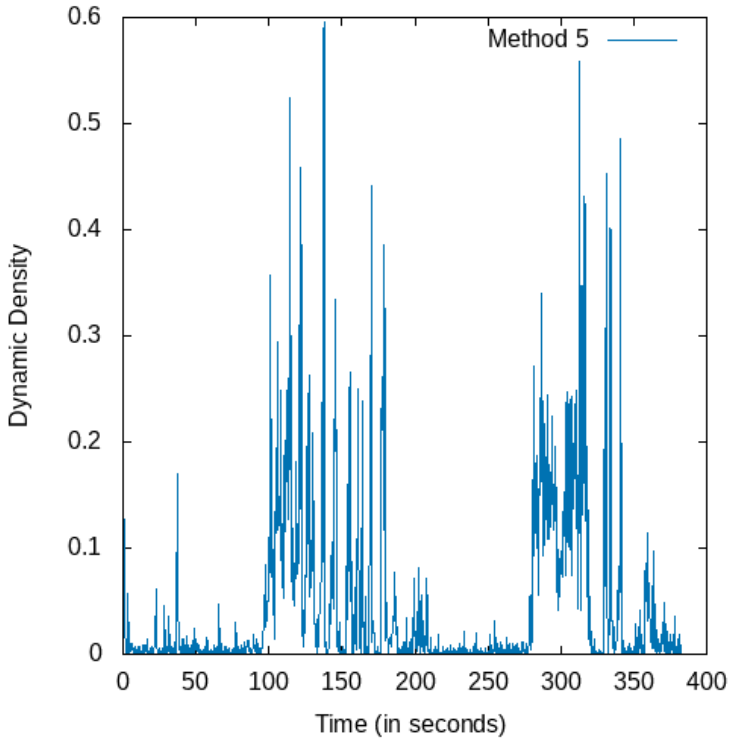

Figure 8: Utility vs time plot for methods1-4