Method-1(Report COP290)

Sourav(2019CS10404),Harikesh(2019CS10355)

March 2021

1 Introduction

In implementation of this method we have calculated the queue density and dynamic density using the **absdiff** function.
We also use some **Matplotlib** and **numpy** for plotting the graph and finding **Utility**.
Baseline:
Baseline for this method is the code used in Assignment1 part2 i.e. processing every 3rd frame.

2 Analysis

In this method we use x as where x is the number of frames skipped .
We have used absolute error as a metric for measuring utility.

$$Runtime = totaltimetakentorunthecode$$

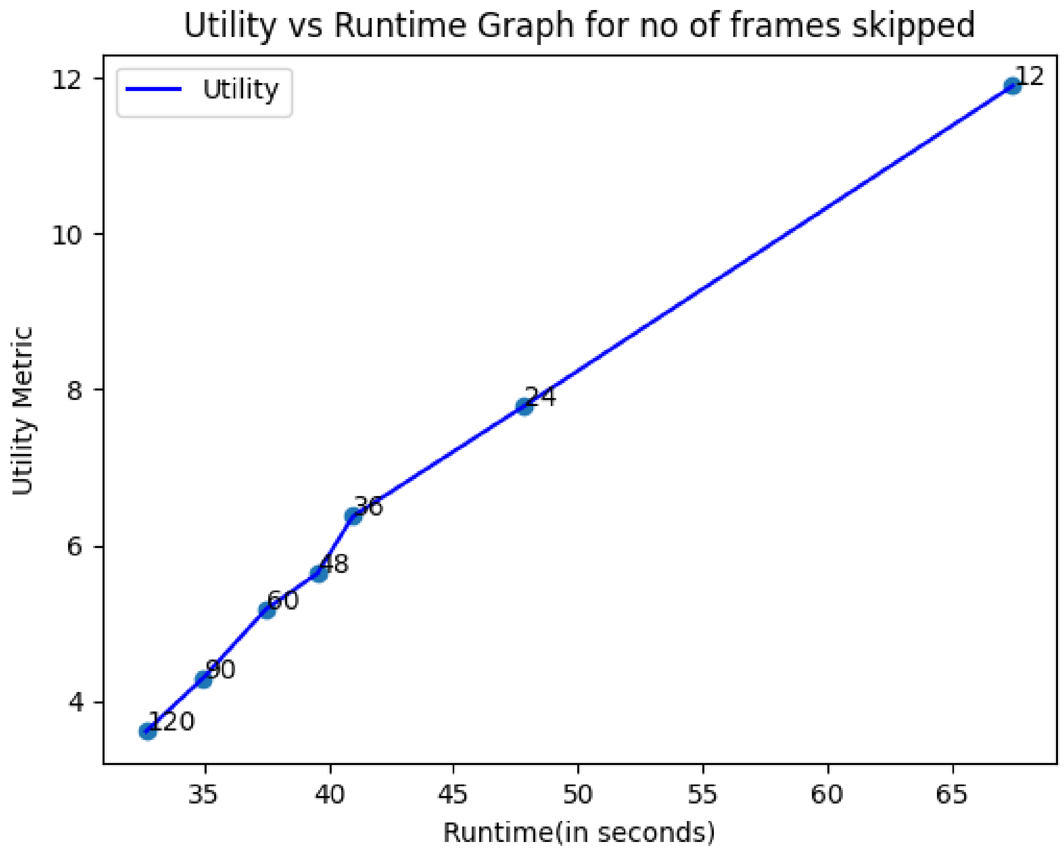
$$queuedensitydeviation = \sum_{allframes} |BaselineDensity - NewDensity|$$

$$dynamicdensitydeviation = \sum_{allframes} |BaselineDensity - NewDensity|$$

$$AbsoluteError = \frac{queuedensitydeviation + dynamicdensitydeviation}{totalframes}$$

$$UtilityMetric = \frac{1}{AbsoluteError}$$

2.1 Utility vs Runtime curve for parameter number of frames skipped



The graph shown above is the graph for different values of x with their calculated Utility and Runtime values and plot the curve. where X-Axis contains the Runtime for every x and Y-Axis contains the Utility Metric for every x.

From the graph we clearly see that the Utility and Runtime both increase approximately lineraly with each other.And from the graph we are cleraly see the it is not lineraly increase this is due to the number of frames skipped are more and some frames takes more time to process so that's why the Utility is also lower in some cases we cannot find the exact linear graph.

For most of the values,the graph is linearly increasing because when Runtime is increase then x is decreased that implies that more frame is processed which leads to a better accuracy and hence more Utility. more Utility implies that Utility is also increase with increase with Runtime.