COP290

Ujjawal Sankhwar 2019CS10412, Vivek Choudhary 2019CS10413

March 2021

1 Introduction

We will convey different aspects of how a program can be modified to get the best general output which will consume less power, takes less time to execute, less deviation in error or in short trade-off analysis of new program with the base program.

2 Metric

2.1 Utility

Error is taken as the mean of the absolute difference between the queue density of baseline and given method for every frame.
Lower the error, higher is the utility of the given method.

$$\text{Error} = \sum_{0}^{allframes} |BaselineQueueDensity - MethodQueueDensity| \frac{1}{TotalFrames}$$

2.2 Runtime

The total time taken to process the complete video by a given method.

3 Methods

3.1 Method 1: Sub-sampling of Frames

Processing every x frame i.e. after processing N^{th} frame, the $(N+x)^{th}$ frame is processed and the queue density values obtained for N^{th} frame is used for all the intermediate frames from N to N+x.
Parameter for this method is "x" i.e., the number of frames skipped for calculation of queue density values.

3.2 Method 2: Resolution Reduction

Resolution is reduced by resizing the frames to be processed. Original resolution of the frame is 328x778.
Parameter for this method is "Area Ratio" i.e., the ratio of area of new resolution to that of original resolution.

3.3 Method 3: Splitting Work Spatially across Threads

Each frame is divided into different parts and each part of this frame is processed by a different thread.
Parameter for this method is "Number of Threads" which is also same as the number of parts the frame is divided.

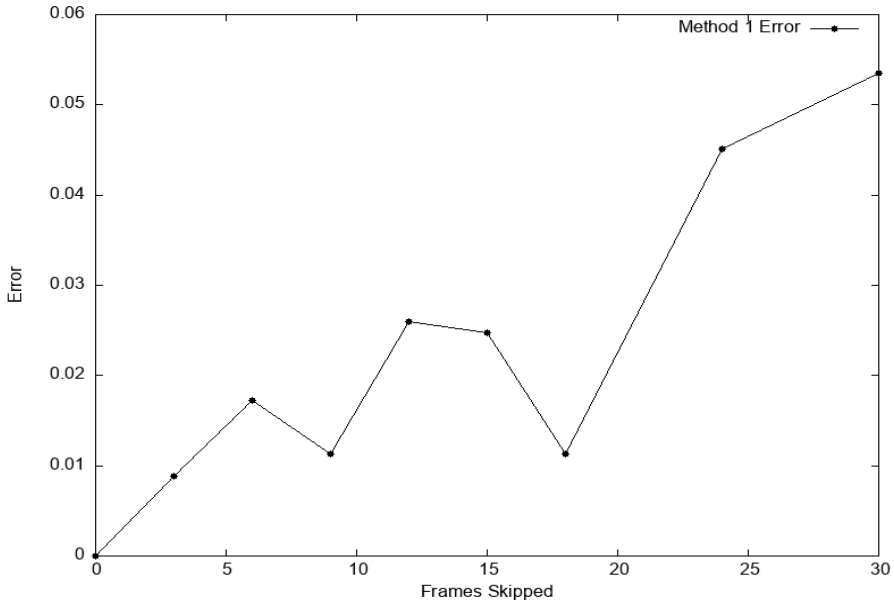
3.4 Method 4: Splitting Work Temporally across Threads

Consecutive frames are processed by different threads and the total work is divided among them.
Parameter for this method is "Number of Threads" among which the total work is divided.

4 Trade-off Analysis

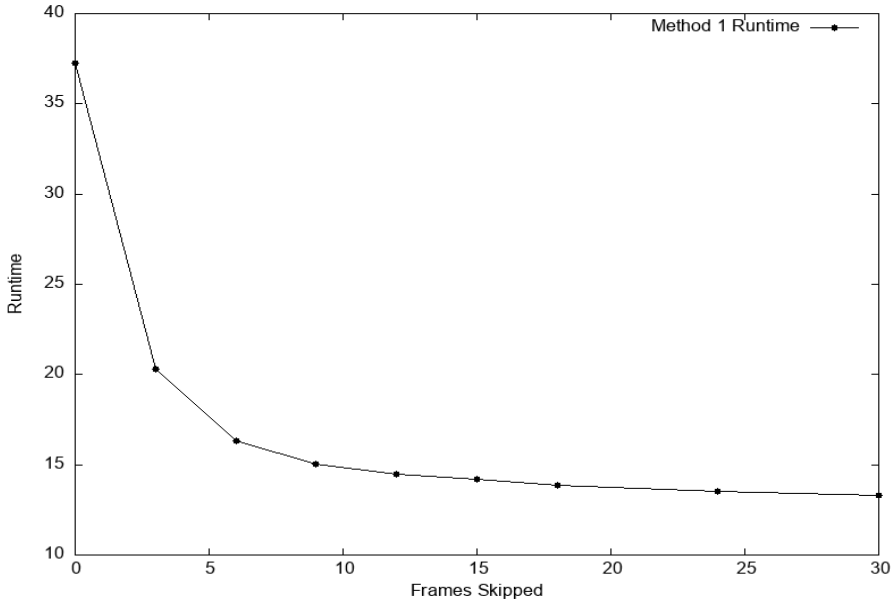
4.1 Method 1: Sub-sampling of Frames

Parameter (Frames Skipped)	Error	Runtime (in sec.)
0	0	37.2846
3	0.00888966	20.3017
6	0.0172564	16.3354
9	0.01133	15.0559
12	0.0259718	14.5045
15	0.0247516	14.1716
18	0.01133	13.8361
24	0.0451455	13.5194
30	0.0535123	13.3297



Method 1: Error vs Frames Skipped

Error increases because when frames are skipped then there is a loss of information of intermediate frames.

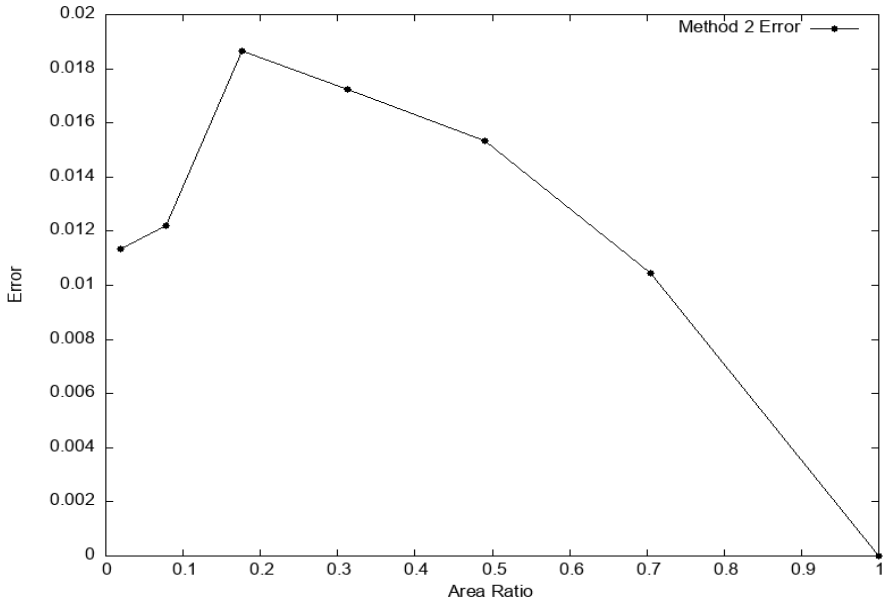


Method 1: Runtime vs Frames Skipped

As the number of frames processed is decreasing, the runtime is also decreasing.

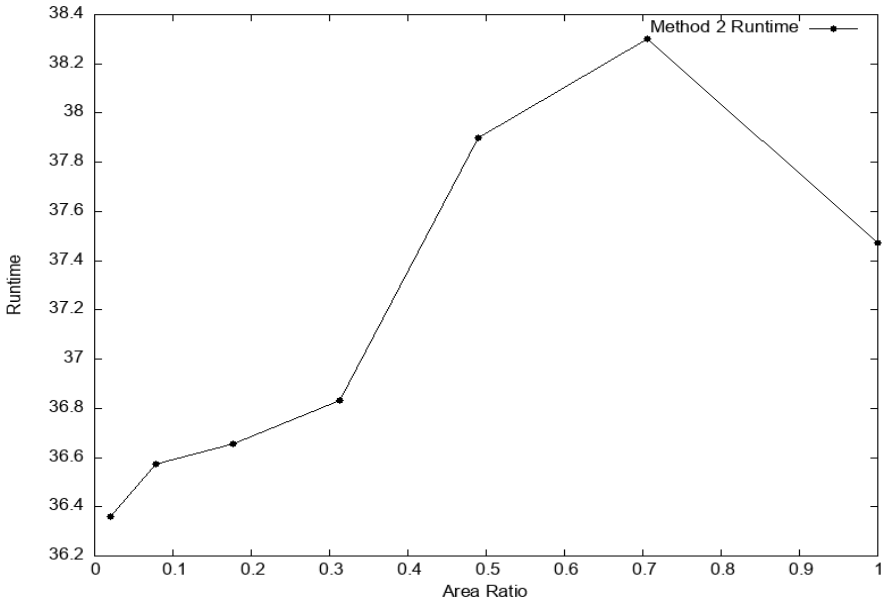
4.2 Method 2: Resolution Reduction

Resolutiom	Parameter (Area Ratio)	Error	Runtime (in sec.)
50x100	0.0195937	0.01133	36.3588
100x200	0.0783748	0.0122015	36.5753
150x300	0.176343	0.0186509	36.6549
200x400	0.313499	0.0172564	36.8325
250x500	0.489843	0.015339	37.8994
300x600	0.705373	0.0104584	38.3016
328x778	1	0	37.4709



Method 2: Error vs Area Ratio

Error decreases as area ratio increases because we process a frame which is more close to the actual size of the Base frame.

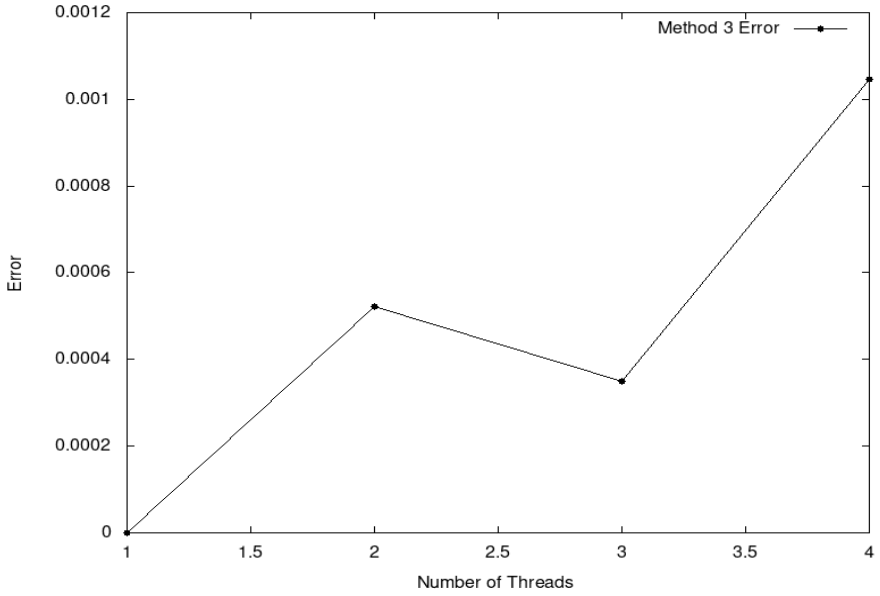


Method 2: Runtime vs Area Ratio

As resizing of frames also takes certain amount of time, the runtime is initially less the Baseline which increases on increasing area ratio and gradually becomes more than the Baseline after a particular area ratio.

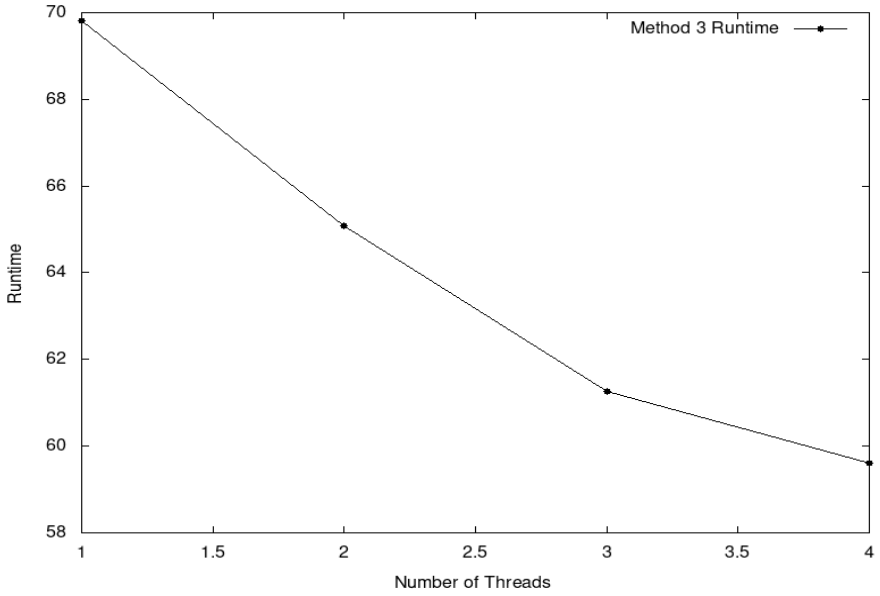
4.3 Method 3: Splitting Work Spatially across Threads

Parameter (Number of Threads)	Error	Runtime (in sec.)
1	0	69.8026
2	0.000522921	65.0947
3	0.000348614	61.253
4	0.00104584	59.598



Method 3: Error vs Number of Threads

Negligible error is generated as each frame is divided into certain parts and each part is processed by different thread in parallel.

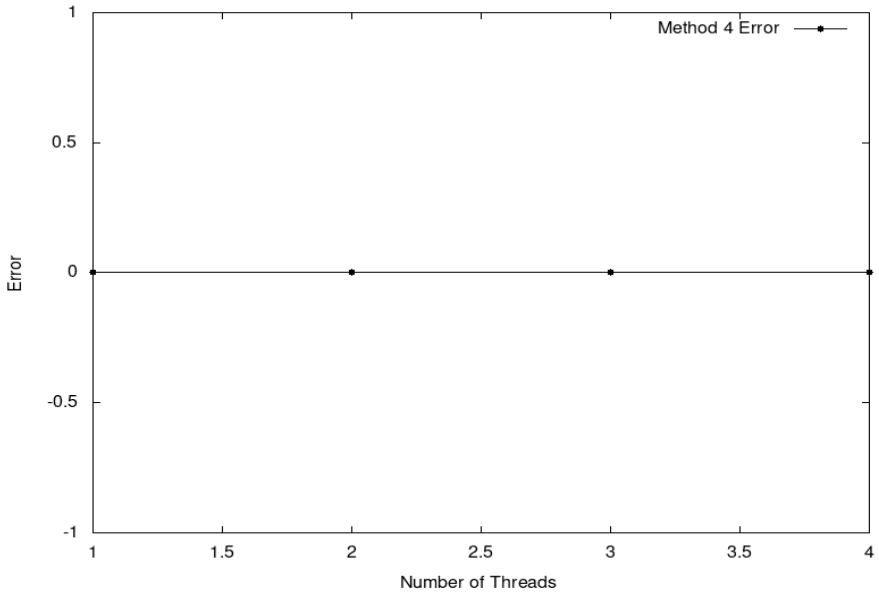


Method 3: Runtime vs Number of Threads

Runtime decreases as each frame is divided into certain parts and each part is processed by different thread in parallel.

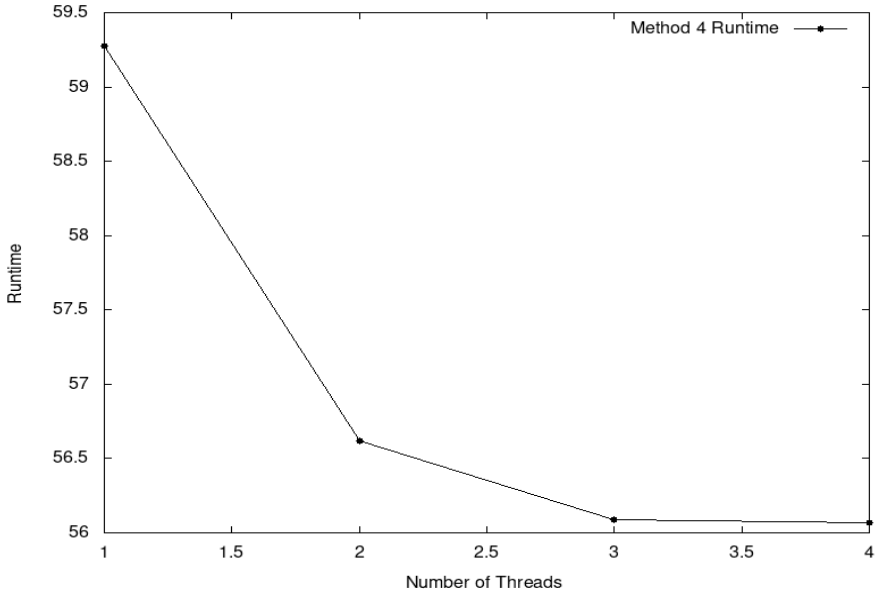
4.4 Method 4: Splitting Work Temporally across Threads

Parameter (Number of Threads)	Error	Runtime (in sec.)
1	0	59.2752
2	0	56.6209
3	0	56.0901
4	0	56.0684



Method 4: Error vs Number of Threads

Error is 0 because each frame is processed by a single thread without any change made to the frame.



Method 4: Runtime vs Number of Threads

Runtime decreases as consecutive frames are processed by different threads in parallel.

5 Summary

5.1 Method 1: Sub-sampling of Frames

There is significant trade-off between utility and runtime. A higher runtime ensures more utility but skipping frames generates lesser runtime with more error and hence less utility.

5.2 Method 2: Resolution Reduction

There is a significant trade-off between utility and runtime. On decreasing the resolution of frame significantly the runtime decreases along with the utility as new frame is very different from the original one.

5.3 Method 3: Splitting Work Spatially across Threads

There is minor trade-off between utility and runtime. On increasing the number of threads used some negligible error is generated but runtime decreases.

5.4 Method 4: Splitting Work Temporally across Threads

No trade-off is seen between utility and runtime. On increasing the number of threads used, the runtime decreases with no decrease in the utility.