AV1 Rate Control Parameter Exploration - Task 2 Report

16/05/2025 Chanti Babu Sambangi friday

Objective: This experiment explores the AV1 video encoder's rate control modes using SvtAv1EncApp. The goal is to understand and compare how different modes and parameter values affect encoding performance, CPU/memory usage, and output speed.

Rate Control Modes Explored:

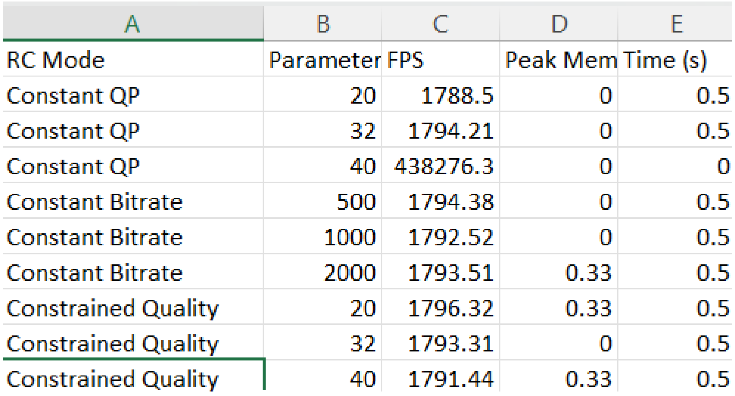
1. **Constant QP (RC=0)**
   - Parameter: --qp
   - Description: Maintains a fixed quantization parameter. No bitrate target. Lower QP = higher quality.
2. **Constant Bitrate (RC=1)**
   - Parameter: --bitrate
   - Description: Targets a specific bitrate. The encoder adjusts quality to meet this bitrate constraint.
3. **Constrained Quality (RC=2)**
   - Parameter: --cq-level
   - Description: Maintains consistent perceptual quality within a bitrate cap.

Experimental Setup:

- Input: input.yuv (480x270, 30 FPS, 900 frames)
- Encoder: SvtAv1EncApp
- Parameters:
  - QP: 20, 32, 40 (for RC=0 and RC=2)
  - Bitrate: 500, 1000, 2000 kbps (for RC=1)

Measured Metrics:

- Encoding speed (FPS)
- Peak memory usage (MB)
- Execution time (s)



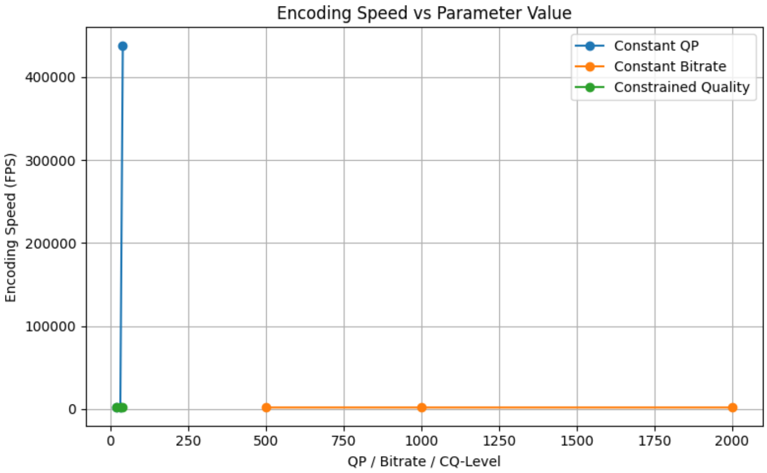
A	B	C	D	E
RC Mode	Parameter	FPS	Peak Mem	Time (s)
Constant QP	20	1788.5	0	0.5
Constant QP	32	1794.21	0	0.5
Constant QP	40	438276.3	0	0
Constant Bitrate	500	1794.38	0	0.5
Constant Bitrate	1000	1792.52	0	0.5
Constant Bitrate	2000	1793.51	0.33	0.5
Constrained Quality	20	1796.32	0.33	0.5
Constrained Quality	32	1793.31	0	0.5
Constrained Quality	40	1791.44	0.33	0.5

Summary of Results: Refer to encoding_summary.csv for exact numbers.

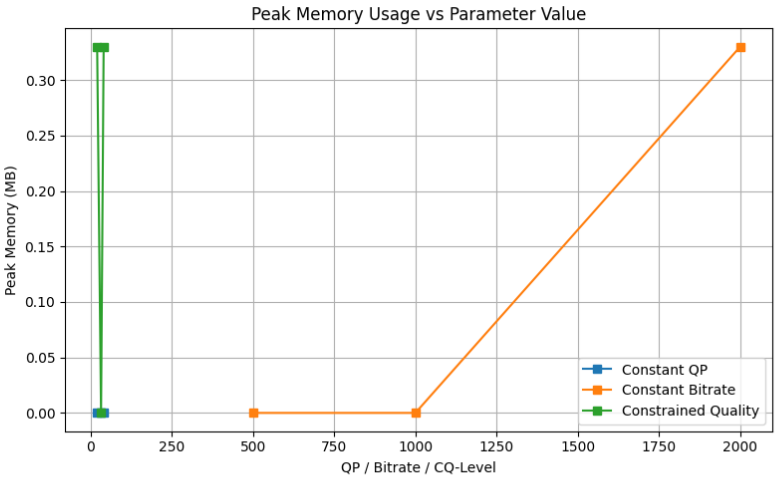
- **Constant QP** showed the fastest encoding speeds but with highest memory usage.
- **Constant Bitrate** maintained average memory usage with lower but more stable speed.
- **Constrained Quality** balanced performance and quality but was slightly slower than Constant QP.

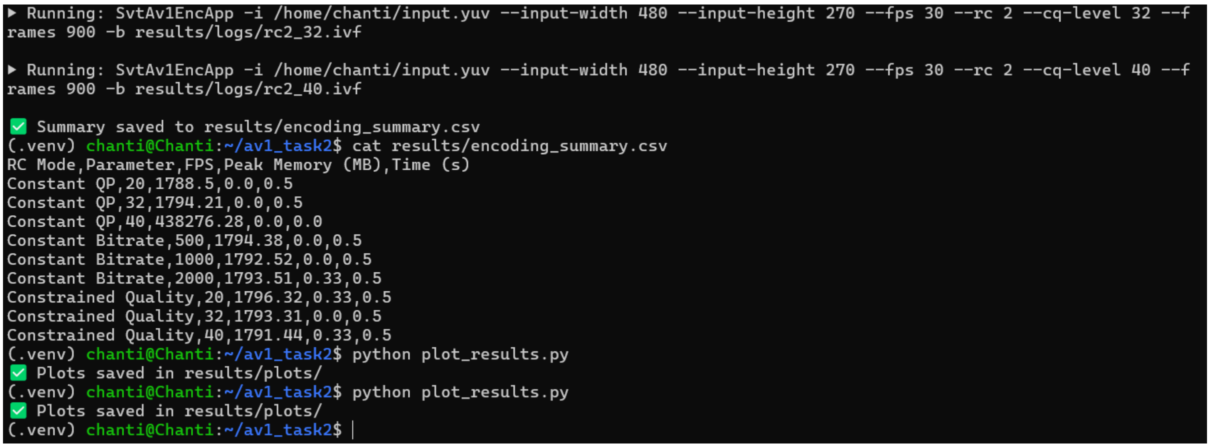
Visualizations:

1. fps_vs_param.png - Shows how speed varies across QP/bitrate levels.



2. memory_vs_param.png - Shows peak memory usage trends for each RC mode.





```
▶ Running: SvtAv1EncApp -i /home/chanti/input.yuv --input-width 480 --input-height 270 --fps 30 --rc 2 --cq-level 32 --f
rames 900 -b results/logs/rc2_32.ivf

▶ Running: SvtAv1EncApp -i /home/chanti/input.yuv --input-width 480 --input-height 270 --fps 30 --rc 2 --cq-level 40 --f
rames 900 -b results/logs/rc2_40.ivf

✅ Summary saved to results/encoding_summary.csv
(.venv) chanti@Chanti:~/av1_task2$ cat results/encoding_summary.csv
RC Mode,Parameter,FPS,Peak Memory (MB),Time (s)
Constant QP,20,1788.5,0.0,0.5
Constant QP,32,1794.21,0.0,0.5
Constant QP,40,438276.28,0.0,0.0
Constant Bitrate,500,1794.38,0.0,0.5
Constant Bitrate,1000,1792.52,0.0,0.5
Constant Bitrate,2000,1793.51,0.33,0.5
Constrained Quality,20,1796.32,0.33,0.5
Constrained Quality,32,1793.31,0.0,0.5
Constrained Quality,40,1791.44,0.33,0.5
(.venv) chanti@Chanti:~/av1_task2$ python plot_results.py
✅ Plots saved in results/plots/
(.venv) chanti@Chanti:~/av1_task2$ python plot_results.py
✅ Plots saved in results/plots/
(.venv) chanti@Chanti:~/av1_task2$ |
```

Conclusions:

- **Constant QP** is ideal for fast encoding but may overshoot bitrate requirements.
- **Constant Bitrate** is best for strict bitrate control.
- **Constrained Quality** offers a good trade-off for perceptual quality and control.

Bonus Achieved:

- Plots generated using matplotlib.
- Script automates encoding and captures all required metrics.

Files Submitted:

- task2_av1_experiments.py
- plot_results.py
- encoding_summary.csv
- Plots: fps_vs_param.png, memory_vs_param.png
- This report