PARSEC: Streaming 360° Videos Using Super-Resolution

Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, Samir R. Das
Department of Computer Science

https://www3.cs.stonybrook.edu/~mdasari/parsec
360° Video Streaming

Central to many immersive applications (e.g., VR/AR)

Image credit: Oculus

Immersive Experience

$ Billion Market

Popularity of 360° Video is on the Rise!

http://blog.dsky.co/tag/head-tracking/
Grand Challenge

- 360° videos require 8x bandwidth compared to regular videos for the same perceived quality

Image from Rollercoaster video

http://blog.dsky.co/tag/head-tracking/
Current Solutions

- Viewport-adaptive streaming
  - Divide video into tiles (e.g., 192x192 pixels)

Flare [MobiCom’18], Rubiks [MobiSys’18], MOSAIC [IFIP Networking’19]
PANO [SIGCOMM’19], ClusTile [INFOCOM’19]
Current Solutions

Viewport-adaptive streaming
- Divide video into tiles (e.g., 192x192 pixels)
- Predict viewport tiles based on head tracking and video saliency analysis
- Stream only viewport specific tiles using ABR algorithm

Flare [MobiCom’18], Rubiks [MobiSys’18], MOSAIC [IFIP Networking’19]
PANO [SIGCOMM’19], ClusTile [INFOCOM’19]
Limitations of Current Solutions

Viewport Prediction (VP)
- Predicting user head movement is hard
- Fetch more tiles to avoid the tile misses
- Fetching more tiles competes for bandwidth and reduces video quality

Network is the only resource for achieving good video quality

Can we improve client’s video quality without relying much on network?
Opportunity 1: Super-resolution

- Use low resolution image/video, hallucinate the details to produce high resolution

  Idea dates to the 90s
  - Currently benefiting from deep neural networks (DNNs)

  DNNs are computationally expensive

https://amundtveit.com/2017/06/04/deep-learning-for-image-super-resolution-scale-up/
Opportunity 2: Computation

- Significant improvement in GPU capacity over the decade
  - Often underutilized
- Leverage this compute capacity on the client to do super-resolution

Is this compute power enough to do super-resolution?
Super-resolution Challenges

- Bulky DNN models
  - Slower inference (e.g., less than 2FPS for a 1-minute 4k video)
  - Large model sizes

- Large variance in quality enhancement

How to make the models smaller, faster & better?
Lightweight Micro-models for Super-resolution

- Train a model for each segment
- Fetch the model along with segment download
- Enhance the quality of few viewport-specific tiles instead of whole frame
Lightweight Micro-models for Super-resolution

- **Benefits**
  - Small model footprint
  - Faster inference

- **Key Questions**
  - Which tiles to download and at what quality?
  - Which tiles to generate (using super-resolution)?
  - Which tiles to ignore?

- **Additional challenges**
  - Still only few tile/sec inference rate

Need a new ABR algorithm that combines compute and network resources.
Neural-Aware ABR

Compute Capacity

Network Capacity

Player Buffer

Client Player

Video Server

Viewport Prediction

[IFIP Networking’19]
Neural-Aware ABR

Expected Quality

\[ E(Q) = \sum_{i=1}^{N} p_i (q_{i,D}r_{i,D} + q_{i,G}r_{i,G}) \]

Tile miss ratio

\[ E(M) = \sum_{i=1}^{N} p_i r_{i,M} \]

Quality switches

\[ V_s = \sum_{i=1}^{N} \text{StdDev}[p_i (q_{i,D}r_{i,D} + q_{i,G}r_{i,G})] \]

Overall experience

\[ QoE = E(Q) - \beta E(M) - \xi (V_s + V_t) \]

How to Find a Solution Fast?

Greedy Algorithm

Download ULR and enhance using SR

Maximize
Putting Everything Together

Only these components present in state-of-the-art 360° video streaming

```plaintext
Offline Super Resolution Model Training

HTTP Server
- ULR Tiles & Micro-Models
- HEVC Encoded Segments

Client
- Neural-Aware ABR Algorithm
  - Bitrate Selection for Downloaded Tiles
  - Inference Scheduler for Generated Tiles
- Decoded Playback Buffer
  - Render and Display

Viewport Prediction
- Network State
- Compute Capacity
- Generated and Downloaded Tile Qualities

14```
Implementation & Evaluation

- Linux server
  - Node.js
- Client
  - Pixel3 phone
- Super-resolution model
  - Keras with Tensorflow backend
- Diverse network conditions
  - Real traces: WiFi & 4G/LTE
  - FCC & Belgium traces
- 360° video dataset
  - 10 videos
  - MMSYS’17 head movement dataset
Performance Comparison

- VP Only [NOSSDAV’17]
  - Download only viewport-specific tiles

- FLARE [MobiCom’18]
  - Fetch additional tiles to accommodate VP inaccuracy

- NAS-regular [OSDI’18]
  - A recent regular video streaming system using super-resolution

- NAS-360
  - A modified version of NAS-regular for 360° video
Performance Comparison
Average Quality and Tile Misses

30% improvement compared to Flare [MobiCom’18]

26% improvement at the 90th percentile compared to Flare [MobiCom’18]
Overall QoE Performance

37% improvement compared to Flare

[MobiCom’18]
Impact of Computation

PARSEC performs better as we increase the computing power
Conclusion

• PARSEC
  • A panoramic video streaming system
  • DNN based super-resolution
  • Neural-aware ABR algorithm

• PARSEC provides high QoE compared to the state-of-the-art solutions

For more details please visit:
https://www3.cs.stonybrook.edu/~mdasari/parsec