

Smart Streaming of Panoramic Videos

Students: HW Ma, YZ Cai

Kandao Tech: XK Jiang, R Ma, ZY Ma

Presenter: DM Chiu

August 25, 2018

Background

- Smart cameras
 - Multiple lenses, plus computer vision/AI software
 - Panoramic video, VR, plus other features
 - Our collaborator, Kandao Technology
 - Award-winning camera, Obsidian
- Machine learning applied to networking algorithms
 - Predict available bandwidth to optimize adaptation of DASH [Pensieve, Sigcomm 2017]
 - In panoramic video or games, predict eye focus to stream only what matters



Different approaches

- Foveated streaming
 - Track user's gaze, and encode video accordingly in real time, usually in online video games
- Tile-based streaming
 - Partition each frame of video into a set of tiles, and encode each tile in multiple quality levels (offline)
 - Stream high quality tiles depending on user's eye focus (real time)
- Copy-based streaming (the approach we study)
 - Create multiple copies of each frame, optimizing for different focus (offline)
 - Stream the copy best match current user eye focus (real time)

Pros and cons of copy-based approach

- Requires considerably more storage at server
- But requires no encoding/decoding in real time
- Potentially better quality (subjective?)

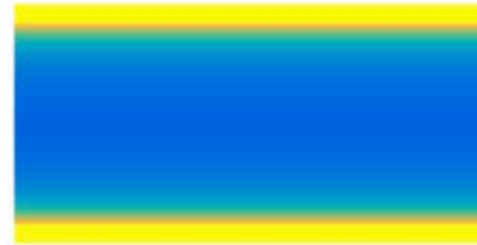
Example implementation

- <http://home.ie.cuhk.edu.hk/~dmchiu/dynamicvideo.mp4>

How each copy is made (1) - projection

- The panoramic videos are generated by fish-eye cameras covering a spherical view
- We need to first project the spherical view into a planar view
- Some projection schemes

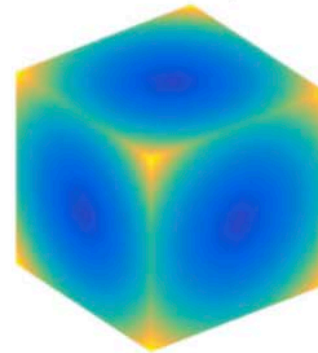
Equirectangular



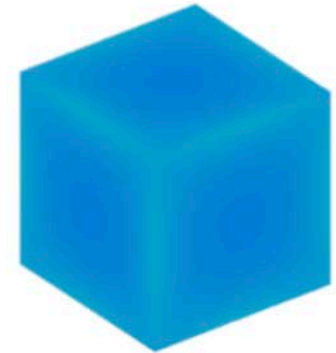
Barrel



Cubemap

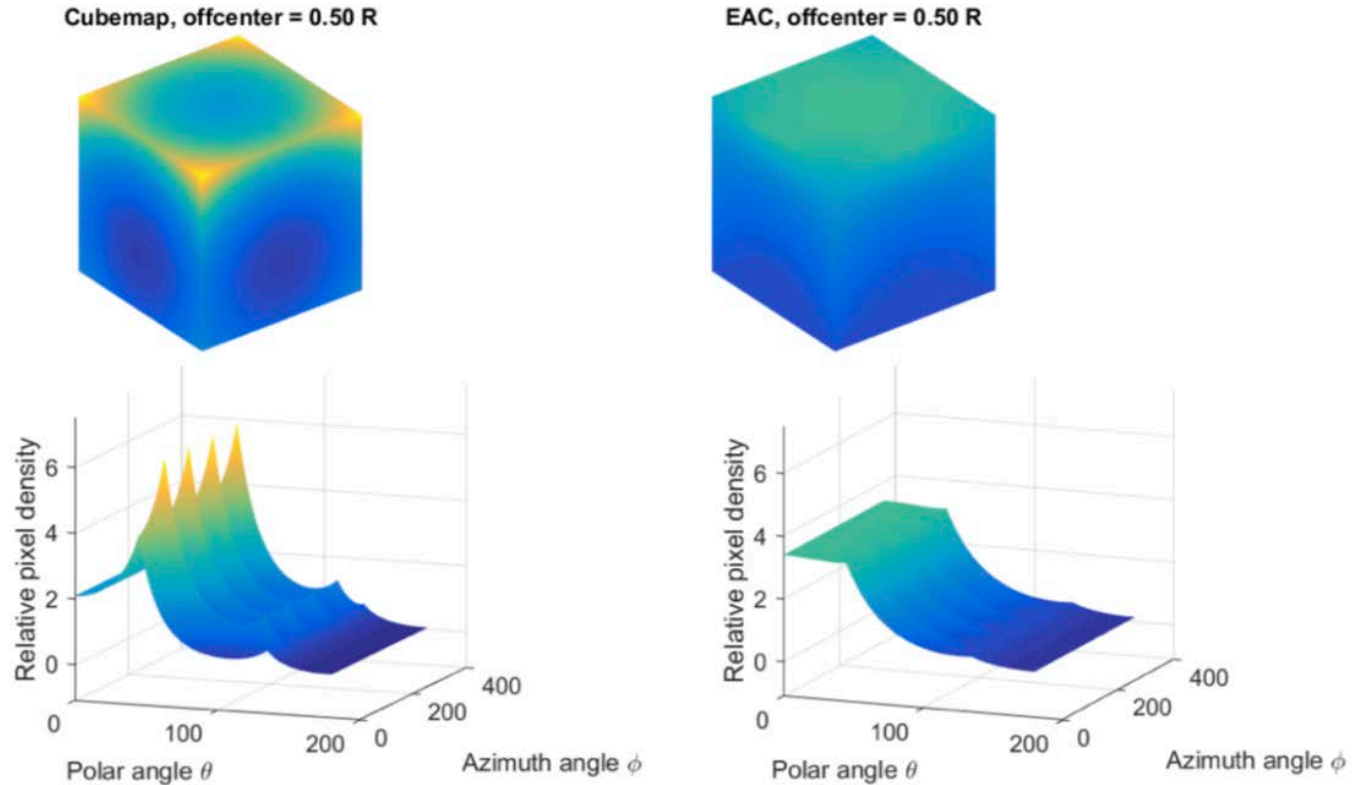


EAC



How each copy is made (2) – pixel allocation

- Allocate more pixels to that part of the planar view matching user's attention. (e.g. the front rectangle)
- Equi-angular: Smoothly continuous pixel allocation across different parts of the planar view.
- This is the approach taken by Kandao Technology, and is what we assume in defining the pixel allocation function.



Assumption and problem definition

- Assume the available bandwidth between server and player is given and is greater than the video playback rate
 - Allow us to focus first on adapting user attention
- Two problems
 - Copy-switching algorithm design - online algorithm
 - Segmentation design - offline problem

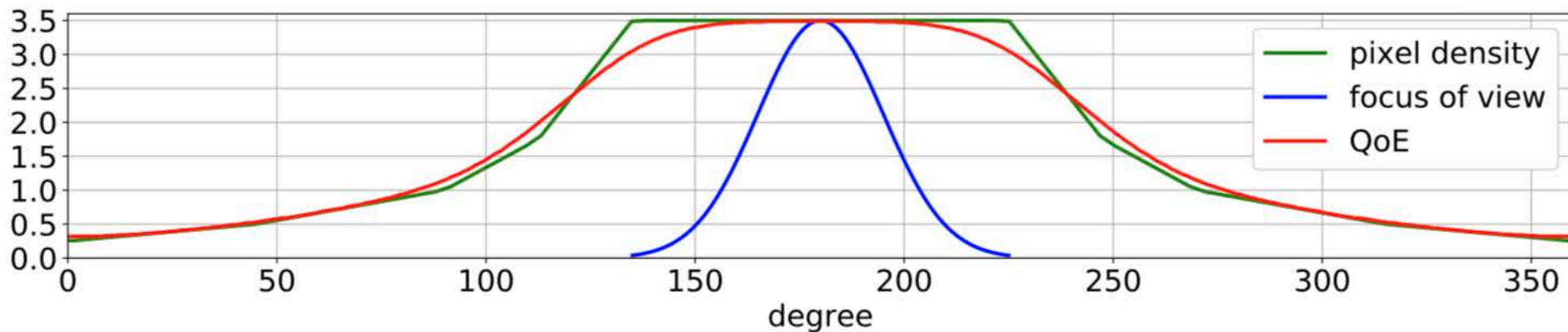
Summary of what we did

- Simulate copy-switching strategies for different attention trajectories
 - Define a QoE metric
 - Measure user attention trajectory
 - Propose polar representation for visualization
 - Compare different strategies under network delay
- Simulate what-if scenarios
 - Different segmentation
 - Different bandwidth conditions

A measure of QoE

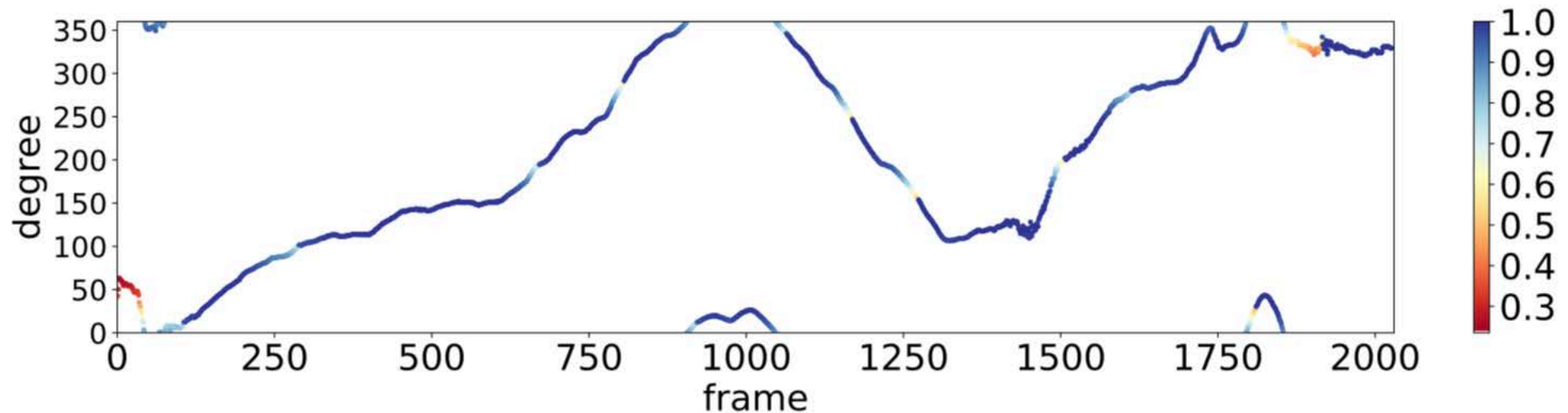
- For simplicity, assume video is 360 only in horizontal direction
- For each copy, the allocation of pixels is a pixel density function $p_y(x)$, where $y(t)$ is the center of the copy at time t
- User's attention for the video can be captured by a focus of view function $f_z(x)$ where $z(t)$ is the center of the user's focus at time t

$$Q(t) = \int_0^{2\pi} f_z(x)p_y(x)dx$$



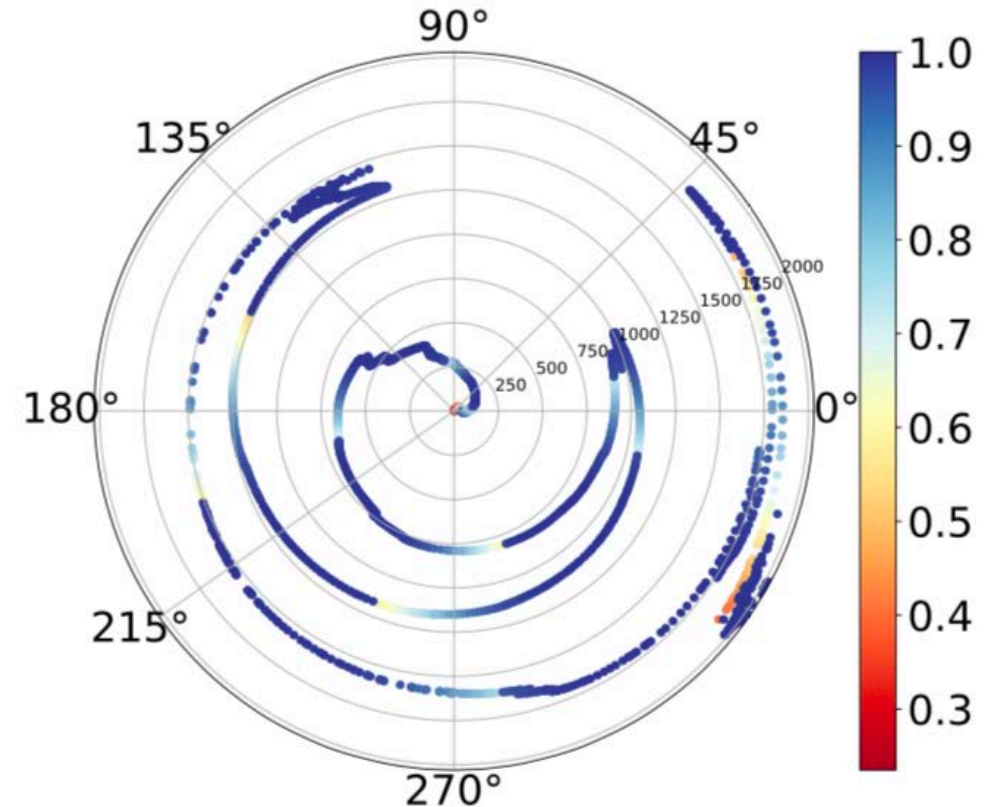
User attention trajectory

- User attention trajectory, denoted by $u(t)$, $0 < u(t) < 2\pi$
- X axis is time
- Color represents the value of QoE
- A recorded real user attention trajectory
- Wrap around problem



Polar representation

- The distance of the dot from the origin measures the time t whereas the direction of $u(t)$ is the polar coordinate
- Each copy's coverage is approximated a sector of the circle
- The same recorded user attention trajectory



Other benchmark trajectories

Define two artificial trajectories for comparison

- **Scan trajectory** - user's attention scans from position 0 towards position 2π at a regular speed
- **Random trajectory** - user's attention is a random walk, this is generated in a similar way as the scan trajectory, except the direction of change can be either left or right.

We determine the average speed of the real trajectory, and use that for the benchmark trajectories.

Modeling copy switching with delay

We assume the video streaming mechanism is based on a request-response protocol such as HTTP.

- A small delay between request and start of playback at the player even no initial buffering is done, similar when switching.
- At least two other factors affect the delay
 - GoPs (Group of Picture): the starting point of playing a new copy must be at the beginning of a GoP
 - A good amount of the current copy may already be buffered at the player side.
- Simply use a suitable value for the minimum initial delay from request to playback.

Switching algorithm

- Switching strategy is an online algorithm, the knowledge of user attention available to the algorithm is only the current position $u(t)$.
- The correct copy to switch to depends on the knowledge of $u(t')$ where t' is the time switching happens, **but value of $u(t')$ is unknown.**

Switching algorithm

For our analysis, we evaluate the following switching strategies:

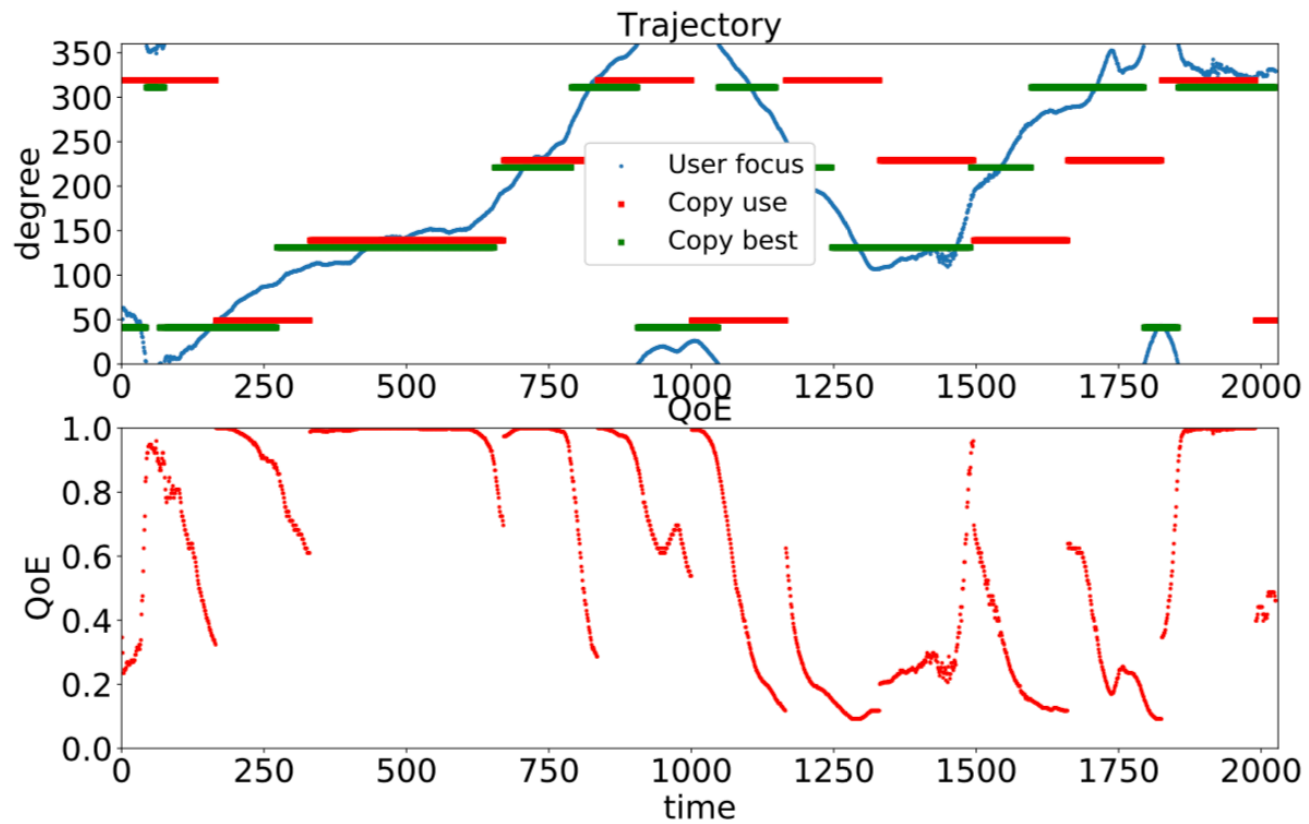
- **Baseline strategy (B):** assume $u(t') = u(t)$ for $t' > t$ when calculating Q for different copies;
- **Simple Markov estimator strategy (M):** make a directional adjustment with based on a simple Markov model.
- **Distance estimator strategy (D):** the position of $u(t')$ is estimated for some expected t' (when switching is expected to happen). The estimation is based on the speed of attention change in the recent past.
- **Offline optimal strategy (O):** assume the best copy is used at each t .

Simulation

- We study these strategies based on simulation, using different types of user attention trajectories.
- Assume there are four copies of the video, each covering a different 90 degree quadrant, and the center of these four copies are respectively 45, 135, 225, 315 degrees.
- At the beginning of viewing, a random copy is used.
- The available bandwidth is $B = 33$ and playback rate is $R = 30$ (unit not given as only relative magnitude matter).

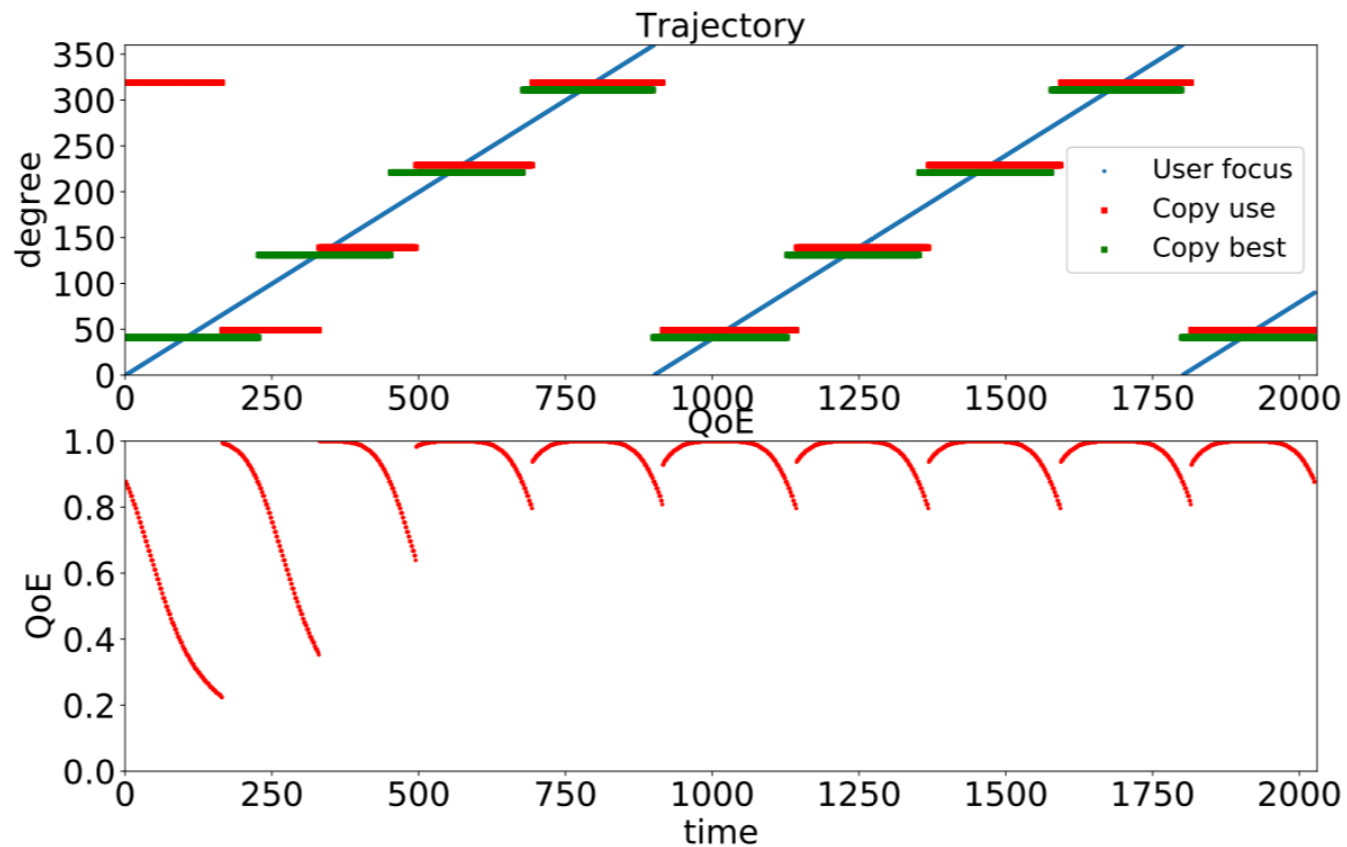
Benchmark: measured trajectory

Performance of baseline strategy on real user trajectory, average Q is 0.63



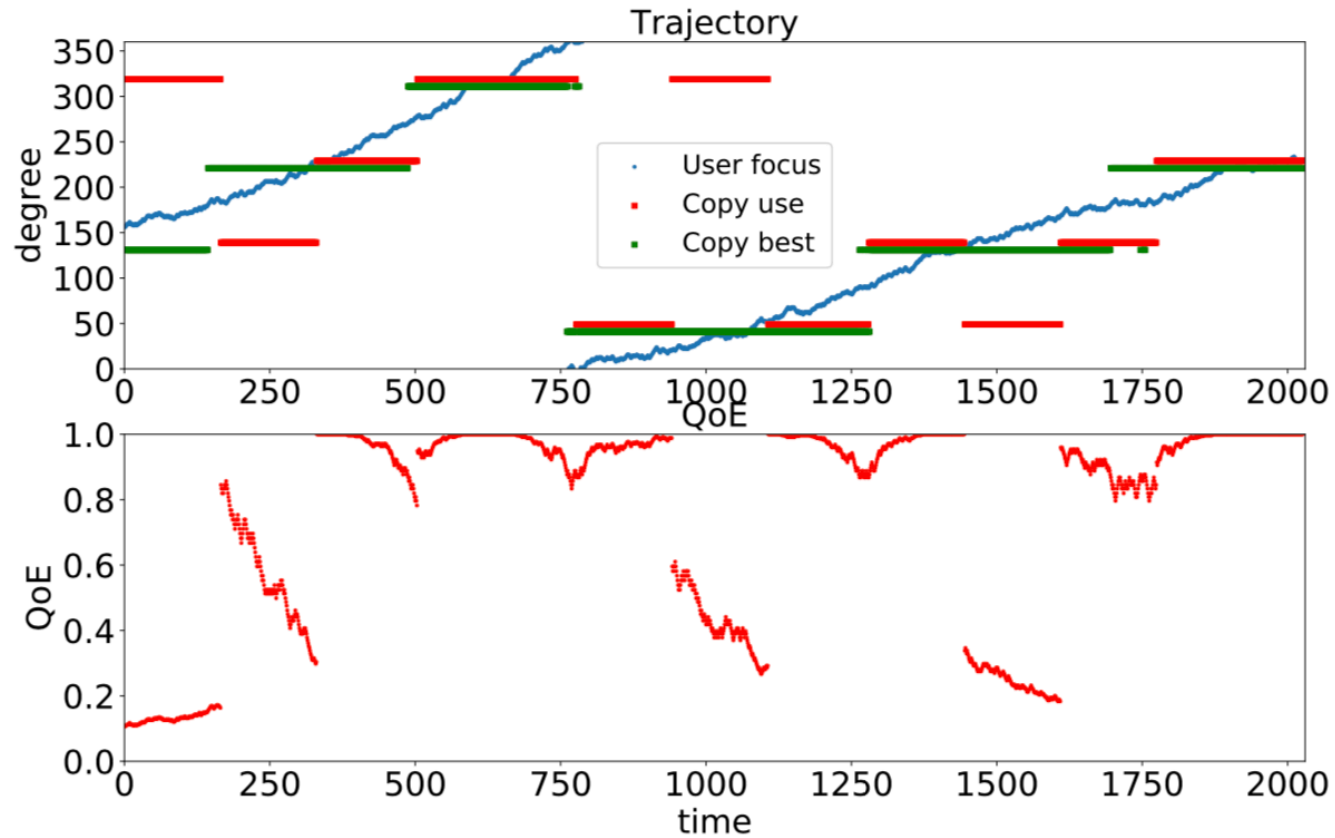
Benchmark: scan trajectory

Performance of baseline strategy on scan trajectory, average Q is 0.91



Benchmark: random trajectory

Performance of baseline strategy on random trajectory, average Q is 0.76



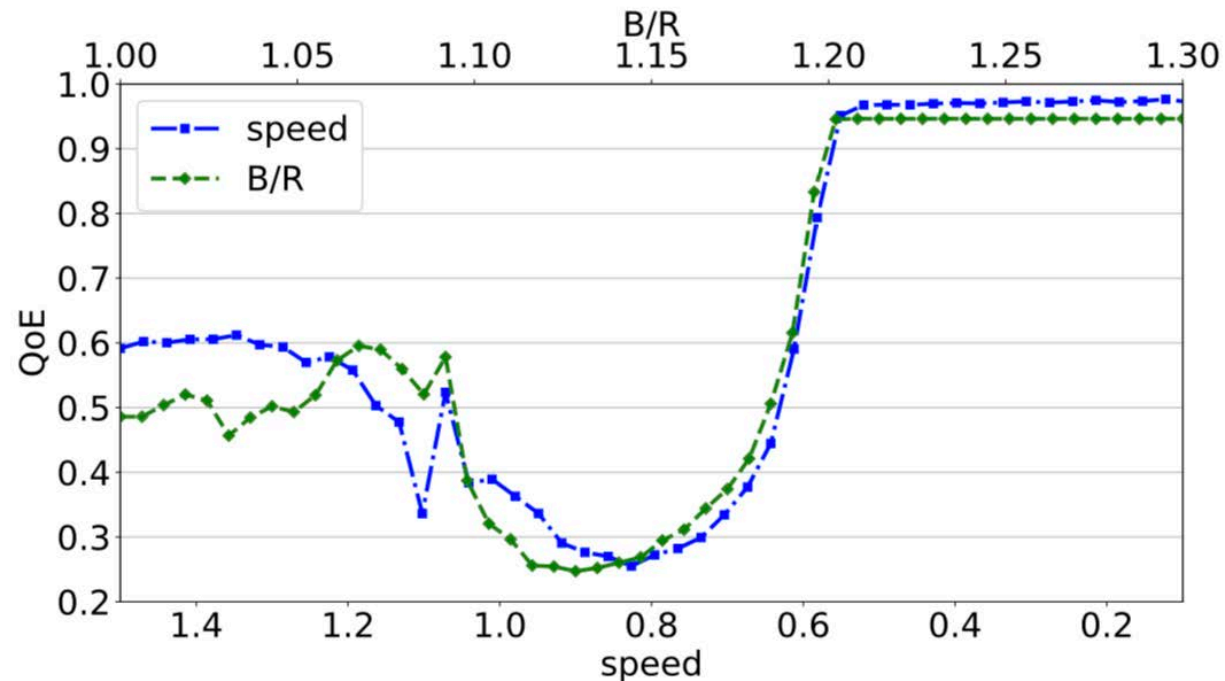
Discussion of simulation results

- Scan trajectory is most predictable, its performance is also most regular.
- For the random trajectory, the performance varies more, and occasionally a wrong copy is streamed and it takes some time to correct.
- Real user trajectory performed quite poorly, even worse than the random case at this particular B and R ratio (33 and 30)
- To improve QoE, obviously we need to either speed up switching (hard to do) or do better prediction.

Effect of speeding and bandwidth

The effect of some system parameters

- The relative value of playback rate R to available bandwidth B
- The speed of change of user attention, to switching performance



Comparing different switching strategies

- Markov model improves the baseline only slightly
- The Distance algorithm improves QoE significantly more
- Offline optimal is not 100% either, since we are only using four copies to cover the 360 degree video.

| | real | scan | random |
|----------|-------|-------|--------|
| baseline | 0.63 | 0.91 | 0.76 |
| Markov | 0.68 | 0.89 | 0.83 |
| Distance | 0.89 | 0.92 | 0.86 |
| Offline | 0.980 | 0.975 | 0.971 |

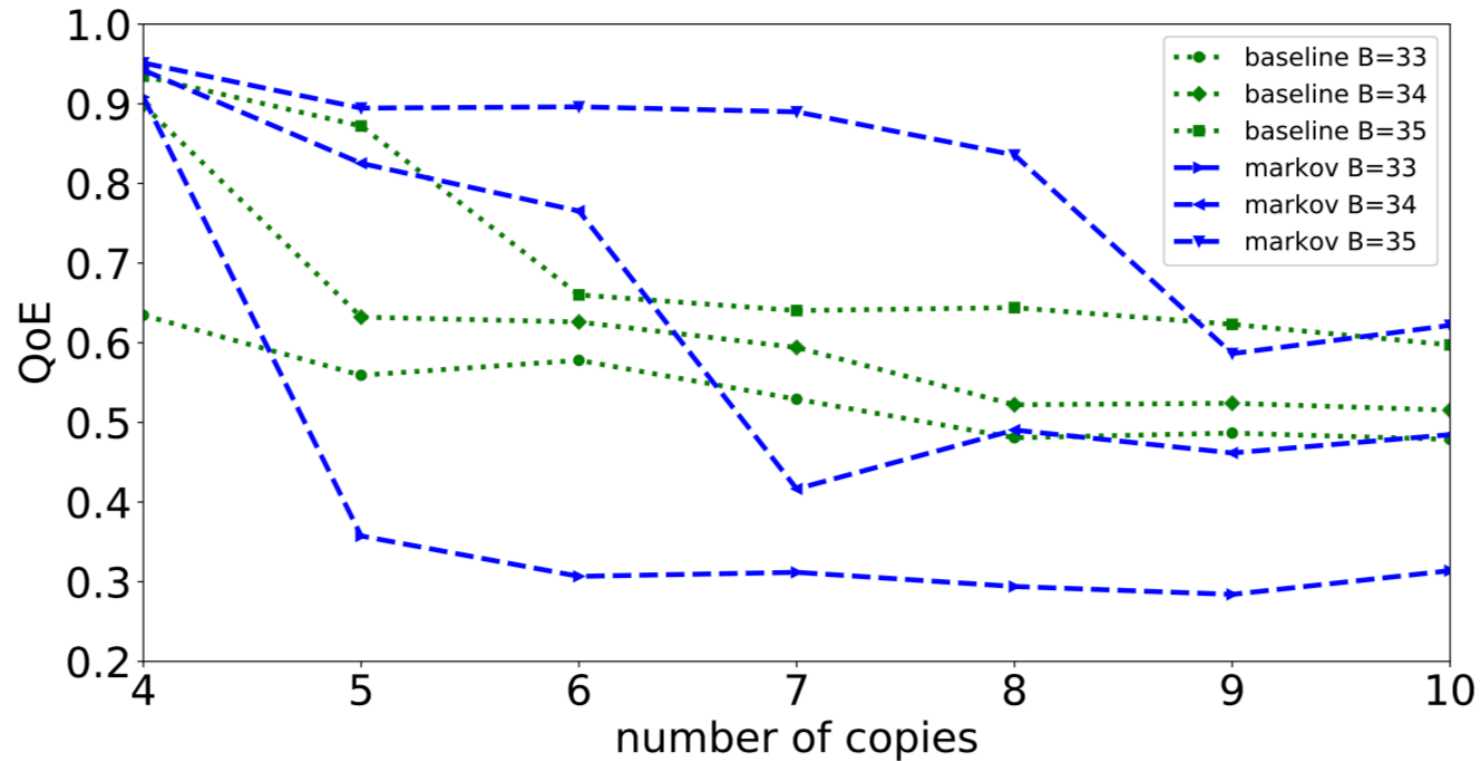
Table 1: Comparison of user attention prediction algorithms using different user attention trajectories

Segmentation

Segmentation: how many copies to use and what each copy covers.

- Experiment using more copies and see if QoE will improve.
- When increasing number of copies, we assume the copies are used to cover evenly divided intervals of the 360% spectrum.
- Shift their center (together) so that a best alignment that minimizes number of switching.
- In practice, if there is knowledge of where user attention tends to focus on, e.g. if we know the heat map, then it is possible to be smarter in using more copies and alignment.

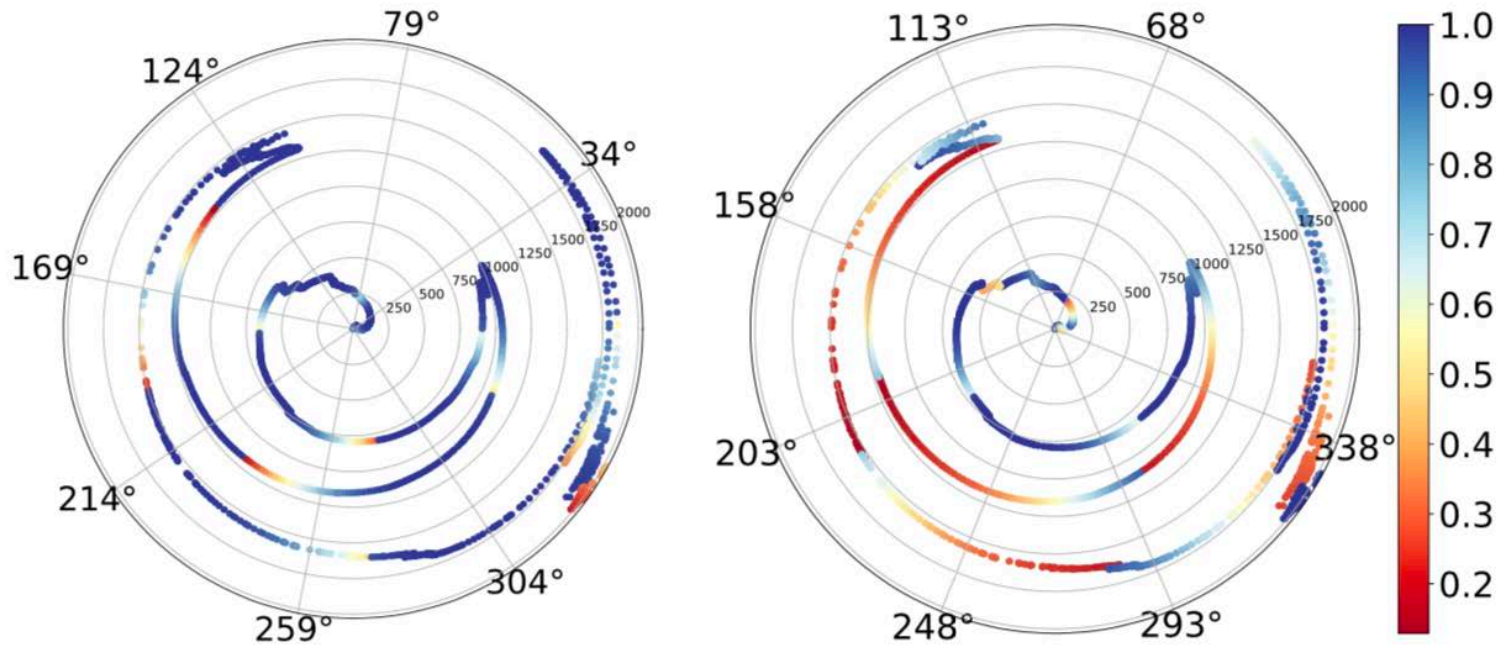
Effect of increasing the number of copies



More copies means more challenge in designing good switching algorithms.

Shifting copy coverage to improve QoE

Clearly that aligning the copies to start at 34 degrees works much better than 68 degrees.



(a) Copies shifted to start at 34 degrees
(b) Copies shifted to start at 68 degrees

Conclusion

We studied the problem of streaming panoramic video (based on the copy-based approach) by simulation, making some assumptions and problem abstraction

Future directions:

- Remove horizontal only assumption
- Improve Q, e.g. to include smoothness
- Consider bandwidth adaptation together

Thank you!