SweeTile: Efficient Tiling for 360° Video Streaming on Light-Weight VR Devices

Cheng-Yeh Chen, Hung-Yun Hsieh

National Taiwan University Mobile Networks and Wireless Communications Lab (TONIC)

Jun 1 2023

Outline

- Introduction
  - Tile-based 360° video streaming
  - Metrics and tradeoff in tiling mechanism
- Related work and motivation
  - Dynamic tiling and fixed tiling
  - Motivation for light-weight VR devices
- Observation and analysis
  - Observation of human field of view and viewport position
  - Real-world dataset analysis
- SweeTile configuration
  - Combination and rotation of sweet spot
  - Client-side deployment
- Evaluation
- Conclusion

Tile-based 360° video streaming

- 360° video streaming aims to support full coverage of field of view (FoV) without limitation on users' head movement.
- Since only 20% of panoramic video would be viewed by a user [1], transmitting only the visible part of video could substantially save the transmission and computation resources.
- By splitting the whole panoramic video into separate "tiles", the video player could flexibly determine which tiles to transmit.

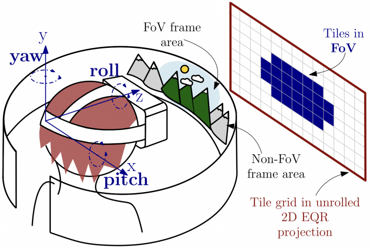


Figure: Scenario of 360° videos streaming [2]

Mapping 3D visual content onto a 2D plane

- To apply existing video compression and encoding techniques, the 3D panoramic video is first mapped onto a 2D plane.
- There are several ways to transform 3D visual content onto its 2D projection. We focus on the most common projection: **EquiRectangular Projection (ERP)**.

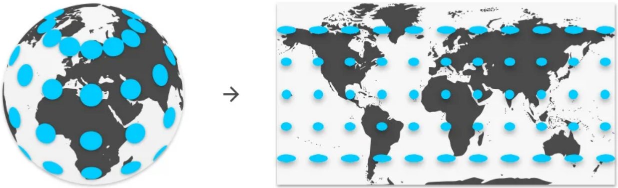


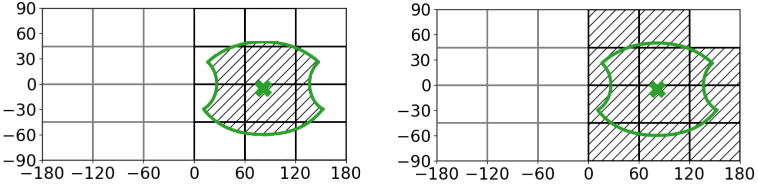
Figure: Demonstration of ERP [3].

Metrics and tradeoff in tiling mechanism

- **Two key metrics** should be addressed to properly achieve the desired save of transmission and computation resources:
  - Transmission efficiency
  - Encoding efficiency
- The interplay of these two metrics forms **the main tradeoff in tiled-based streaming: the granularity of tiles**.

Key metrics in tiling mechanism

- **Transmission efficiency:**
  - This metric evaluates the ratio of the area between the viewport and the transmitted tiles.

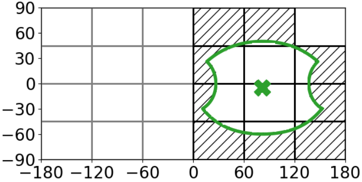


(a) Viewport (b) Actual transmitted tiles

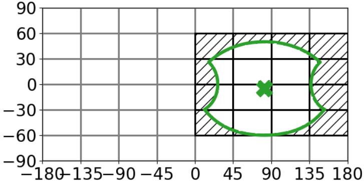
  - There exists mismatch between the viewport and the aggregated shape of transmitted tiles. Such mismatch results in **waste of unviewed area** and lowers the transmission efficiency.

Key metrics in tiling mechanism

- **What affects transmission efficiency?**
  - Different tiling setting results in different mismatch of the viewport and the transmitted tiles.



(a) Waste of 6 × 4 tiling: 57% of transmitted area.

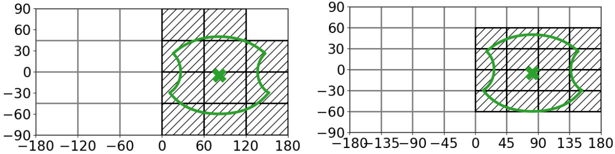


(b) Waste of 8 × 6 tiling: 44% of transmitted area.

  - Typically, coarse-grained tiling results in lower transmission efficiency, while fine-grained tiling results in better transmission efficiency.

Key metrics in tiling mechanism

- **Encoding efficiency:**
  - This metric evaluates the efficiency of encoding all the required tiles.
- **What affects encoding efficiency?**
  - To enable tile-based transmission, each tile should be **independently encoded and decoded**.
  - **The number of tiles** affects the encoding efficiency [4].

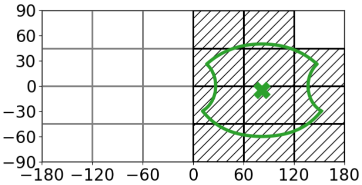


(a) 11 tiles in total (b) 16 tiles in total

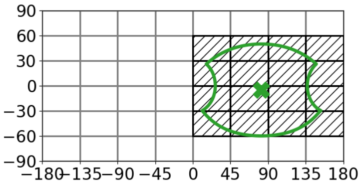
  - Given the same viewport, the higher the number of tiles is, the smaller each tile would be, and the lower the **spatial redundancy** can be utilized to compress and encode a tile.

Tradeoff of tiling mechanism

- The tradeoff of tiling mechanism lies in the granularity of tiles:
  - **Coarse-grained tiles** benefit from higher encoding efficiency but suffer from lower transmission efficiency.
  - **Fine-grained tiles** benefit from higher transmission efficiency but suffer from lower encoding efficiency.



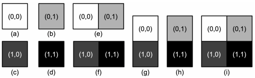
(a) Lower number of tiles (11 tiles) but higher wasted area (57%).



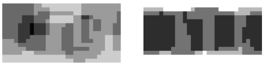
(b) Higher number of tiles (16 tiles) but lower wasted area (44%).

Existing tiling mechanisms: Dynamic tiling

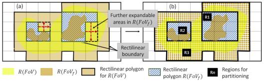
- Dynamic tiling [4, 5, 6]
  - Split the whole panoramic video into dynamic (usually rectilinear) shapes of tiles according to video content, user's visual attention, storage capacity, and transmission capacity.
- Advantage
  - Higher flexibility and adaptivity to strike a balance between transmission and encoding efficiency.
- Drawback
  - Impose huge preprocessing overhead on the server.
  - Unscalable for large-scale real-time content, especially for multiple clients with diverse viewing behaviors requesting live 360° videos.



(a) OpTile [4]



(b) ClusTile [5]



(c) VASTile [6]

Light-weight is a necessity, not just a benefit

- Although VR/AR with 360° video streaming has been envisioned as an upcoming revolution that will change how people interact with the world, the revolution has never been triggered for most consumers.
  - The most popular headset, Meta's Quest, is struggling for its selling units (20 million estimated) and retention rate (10% estimated) [7].
  - Apple's long-rumored "Reality Pro" has been postponed from 2021 to 2023 and still not yet unveiled [8].
- Existing wireless VR headset has burdening weight (>500g) and limited battery capacity (2-3 hours) [9], which are physical limitations restricting the popularity of VR headset.
- **Light-weight both in hardware and software is a necessity for the revolution of VR headset.**

Existing light-weight tiling mechanisms: Fixed tiling

- Fixed tiling [10, 11]
  - Split the whole video into fixed shapes of tiles regardless of video content, visual attention, etc.
- Advantage
  - Efficient both on the server side and the client side (due to fixed shapes, simple tiling mechanisms, and lower encoding and decoding overhead).
  - Applicable on real-time streaming application.
- Drawback
  - The coarse-grained tiles still suffer from low transmission efficiency.
  - The fine-grained tiles still lower the encoding efficiency and increase the rendering overheads on the client side.

Observation of human field of view

- To better strike a balance between transmission efficiency and encoding efficiency, we first observe how the human FoV interacts with actual tiles.
- Typically, human FoV is split into three regions, cotagorized by their degrees of span with respect to the FoV center:
  1. fovea and near-periphery ($0^\circ \sim 30^\circ$),
  2. mid-periphery ($30^\circ \sim 60^\circ$),
  3. far-periphery ($60^\circ \sim$).
- Visual acuity for pattern and color recognition degrades as the angle of view from the FoV center increases.

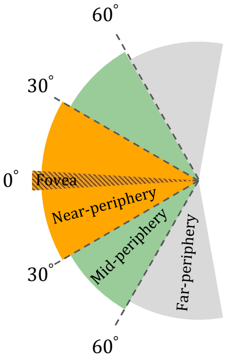


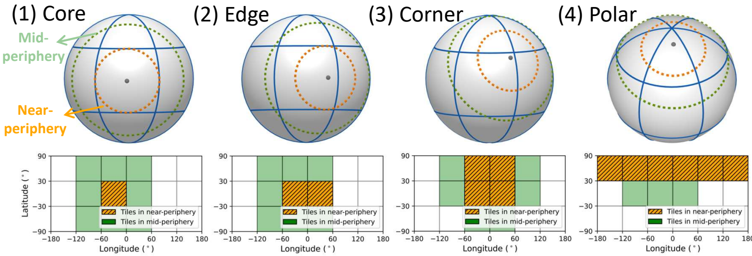
Figure: Human FoV [12]

Efficient coarse-grained tiling

- Since visual acuity degrades as the angle of view from the FoV center increases, most existing tile-based streaming adapts the required bitrate accordingly.
- It is reasonable to transmit high-quality content only in near-periphery and basic-quality content in mid-periphery to save bandwidth consumption without severely affecting the user's quality of experience (QoE).
- To develop efficient tiling mechanism, we choose **6×3 tiling based on equirectangular projection (ERP), a coarse-grained fixed tiling**, as the reference layout.
- Based on 6×3 tiling, we will propose a tiling mechanism **breaking the tie of the tradeoff between transmission and encoding efficiency**.

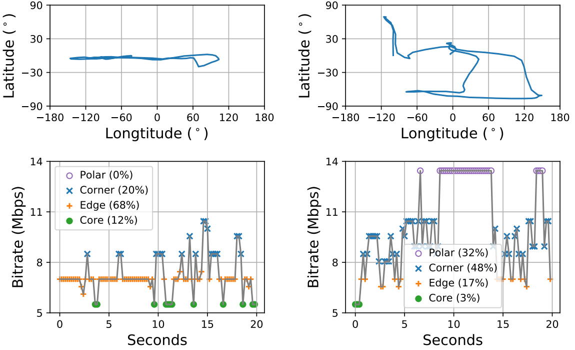
Viewport position matters!

- We first categorize the relative viewport position into four regions and calculate their required bitrates if 4K tiles are transmitted for near-periphery and 1080p tiles are transmitted for mid-periphery.
  - Core region: 5.50 Mbps (good transmission efficiency)
  - Edge region: 6.89 Mbps (good transmission efficiency)
  - Corner region: 9.55 Mbps (poor transmission efficiency)
  - Polar region: 13.38 Mbps (poor transmission efficiency)



Real-world dataset analysis

- We further verify how often the four regions would be traversed in a real-world dataset [13].

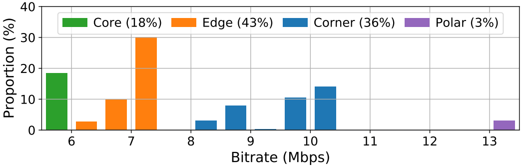


(a) User 1 watching video `PortoRiverside`.
Ave. bitrate: 7.26 Mbps

(b) User 46 watching video `GazaFishermen`.
Ave. bitrate: 10.27 Mbps

Real-world dataset analysis

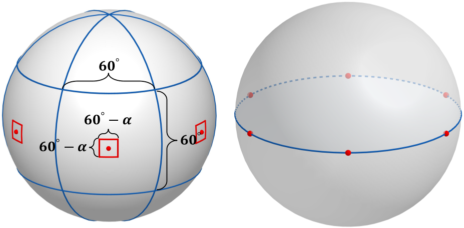
- Throughout the dataset (57 users watching 19 videos), the percentage of duration traversing each region is listed below:



- The corner and polar region suffer from extremely low transmission efficiency due to
  - the distortion of equirectangular projection,
  - the mismatch of aggregated shape of tiling.
- Our goal is to **increase the coverage of the core region** (or later defined as the "sweet spot") to increase the duration of viewpoint traversing in the core region.

Definition of a sweet spot

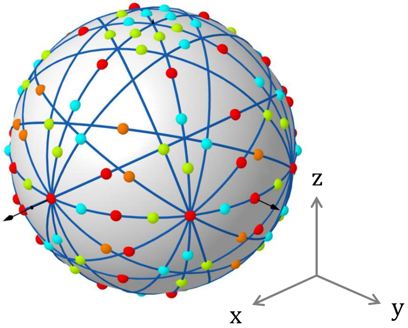
- We define **the core region of an equatorial tile** as the "sweet spot" since a viewpoint traversing into this spot perfectly enjoys full coverage of near-periphery by a single tile.
- Let α define the diameter in degree that near-periphery covers. The defined sweet spot is a square region with a $(60^\circ - \alpha)$ span both in longitude and latitude:



- Such a sweet spot repeats for 6 times along the equator by every 60° in one 6×3 ERP.

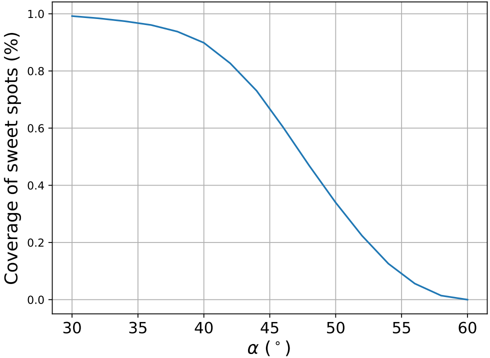
SweeTile: Combination of sweet spots

- The concept of SweeTile is intuitive:
  - If we can cover the visual sphere with sweet spots, we could leverage the benefit of sweet spots no matter where the user's viewpoint is.
- The proposed SweeTile is a combination of 24 versions of 6×3 ERP:



Overall coverage of sweet spots for SweeTile

- In our evaluation, we assume the diameter α of near periphery (high-quality region) to be 40°.
- For the SweeTile configuration, the sweet spots with $\alpha = 40°$ achieves an coverage of 91% of the visual sphere.



SweeTile: Rotate sweet spots to cover the visual sphere

- Define the visual sphere as a unit sphere centered at $(0, 0, 0)$.
- Let $f_p : R^2 \rightarrow R^3$ denote a mapping from (λ, ψ) to (x, y, z) for an ERP p, where
  - (λ, ψ) represents the longitude and latitude of ERP p,
  - (x, y, z) corresponds to the Euclidean coordinate.
- In order to rotate various versions of ERP, we define

$$R_a(\theta), \forall a \in \{x, y, z\}$$

as a rotation matrix to rotate the mapping f_p along the a-axis by an angle of θ to get a new mapping.

SweeTile: Rotate to cover the visual sphere

- Note that the rotation matrix is a 3×3 matrix defined along each axis:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

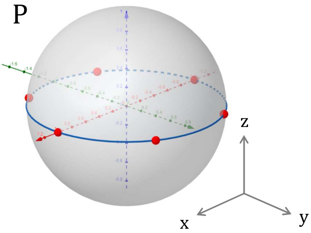
SweeTile: Reference projection P

- Specify P as a reference ERP with $f_P(\lambda, \psi) = (x, y, z)$ where

$$x = \cos(\lambda)\cos(\psi), \quad y = \sin(\lambda)\cos(\psi), \quad z = \sin(\psi),$$

and its six equatorial tiles located at

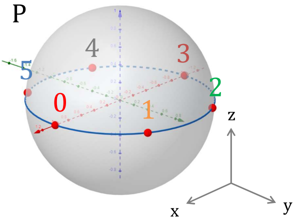
$$\begin{aligned}(x_c, y_c, z_c) =& f_p(c \times 60^\circ, 0) \\ =& \big(\cos(c \times 60^\circ), \sin(c \times 60^\circ), 0\big), \forall c \in \{0, 1, \cdots, 5\}\end{aligned}$$



SweeTile: Sweet spots of P

- The six centers of equatorial tiles of P:

$$
\begin{aligned}
(x_0, y_0, z_0) =& f_p(0 \times 60^\circ, 0) \\
=& \big(\cos(0 \times 60^\circ), \sin(0 \times 60^\circ), 0\big) = (1, 0, 0), \\
(x_1, y_1, z_1) =& f_p(1 \times 60^\circ, 0) \\
=& \big(\cos(1 \times 60^\circ), \sin(1 \times 60^\circ), 0\big) = (0.5, 0.87, 0), \\
& \dots \\
(x_5, y_5, z_5) =& f_p(5 \times 60^\circ, 0) \\
=& \big(\cos(5 \times 60^\circ), \sin(5 \times 60^\circ), 0\big) = (0.5, -0.87, 0).
\end{aligned}
$$



SweeTile: Rotation of sweet spots

- Note that each version of ERP has only 6 sweet spots covering the equatorial region of the corresponding projection. We need to rotate some ERPs to cover the whole visual sphere.
- We classify all the 24 ERPs into 3 categories
  - **Vertical sets V** $= \{\mathbf{V_1}, \mathbf{V_2}, \cdots, \mathbf{V_9}\}$: covers the polar region of P and the region with longitude $c \times 60^\circ, \forall c \in \{0, \cdots, 5\}$.
  - **Horizontal sets H** $= \{\mathbf{H_1}, \mathbf{H_2}, \mathbf{H_3}\}$: covers the equatorial region of P
  - **Expanding sets E** $= \{\mathbf{E_1}, \mathbf{E_2}, \cdots, \mathbf{E_{12}}\}$: covers the remaining region of P.
- Before heading into all the 24 versions of ERP, we first take V_1 (the first version in the vertical set) for example to demonstrate how the rotation matrix works.
- We define $f_{V_1}(\lambda, \psi) = R_x(90^\circ) f_P(\lambda, \psi)$, which is to rotate P by 90° along the x-axis.

SweeTile: Rotation of sweet spots

- For notation clarity, we treat (x, y, z) as a column vector.
- The centers of six equatorial tiles for V_1 can be written as

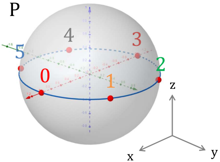
$$f_{V_1}(0 \times 60^\circ, 0) = R_x(90^\circ) f_P(0 \times 60^\circ, 0)$$

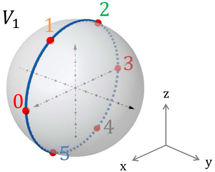
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{pmatrix} (1, 0, 0) = (1, 0, 0),$$

$$f_{V_1}(1 \times 60^\circ, 0) = R_x(90^\circ) f_P(1 \times 60^\circ, 0)$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{pmatrix} (0.5, 0.87, 0) = (0.5, 0, 0.87),$$

$\ldots$





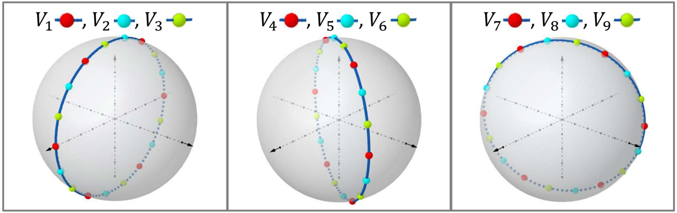
SweeTile: Rotation of sweet spots

- **Vertical sets ($\mathbf{V_1} \sim \mathbf{V_9}$)**: covers the polar region of P and the region with longitude $c \times 60^\circ, \forall c \in \{0, \cdots, 5\}$.

$$f_{V_1} = R_x(90^\circ)\mathbf{f_P}, f_{V_2} = R_y(20^\circ)f_{V_1}, f_{V_3} = R_y(40^\circ)f_{V_1}.$$

$$f_{V_4} = R_z(60^\circ)f_{V_1}, f_{V_5} = R_z(60^\circ)f_{V_2}, f_{V_6} = R_z(60^\circ)f_{V_3}.$$

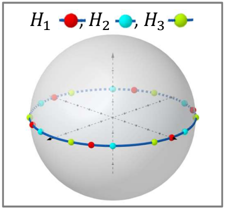
$$f_{V_7} = R_z(120^\circ)f_{V_1}, f_{V_8} = R_z(120^\circ)f_{V_2}, f_{V_9} = R_z(120^\circ)f_{V_3}.$$



SweeTile: Rotation of sweet spots

- **Horizontal sets ($\mathbf{H_1} \sim \mathbf{H_3}$)**: covers the equatorial region of P.

$$f_{H_1} = R_z(90^\circ)\mathbf{f_P},\ f_{H_2} = R_z(15^\circ)f_{H_1},\ f_{H_3} = R_z(-15^\circ)f_{H_1}.$$



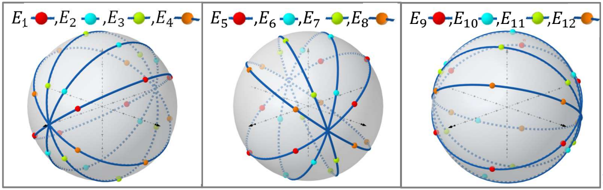
SweeTile: Rotation of sweet spots

- **Expanding sets ($\mathbf{E_1} \sim \mathbf{E_{12}}$)**: covers the remaining region.

$$f_{E_1} = R_x(36^\circ) f_{H_1}, \qquad f_{E_2} = R_x(72^\circ) f_{H_1},$$
$$f_{E_3} = R_x(108^\circ) f_{H_1}, \qquad f_{E_4} = R_x(144^\circ) f_{H_1}.$$

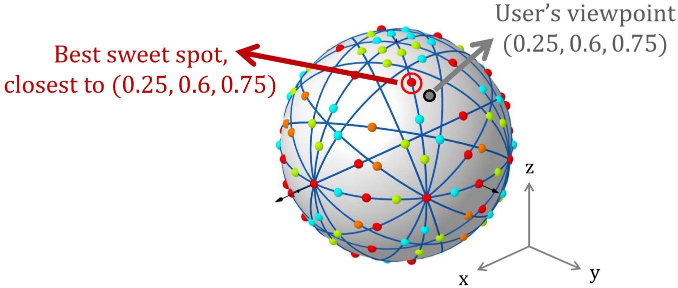
$$f_{E_5} = R_z(60^\circ) f_{E_1}, \qquad f_{E_6} = R_z(60^\circ) f_{E_2},$$
$$f_{E_7} = R_z(60^\circ) f_{E_3}, \qquad f_{E_8} = R_z(60^\circ) f_{E_4},$$

$$f_{E_9} = R_z(120^\circ) f_{E_1}, \qquad f_{E_{10}} = R_z(120^\circ) f_{E_2},$$
$$f_{E_{11}} = R_z(120^\circ) f_{E_3}, \qquad f_{E_{12}} = R_z(120^\circ) f_{E_4}.$$



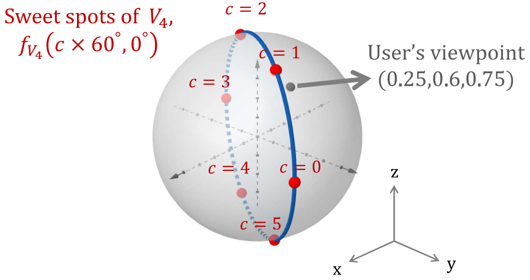
Client-side deployment

- The optimal tiling for SweeTile is to find the best sweet spot among the 144 sweet spots (24 versions × 6 sweet spots per version), which is **the nearest sweet spot with respect to the user's viewpoint**.
- For example, given the user's viewpoint at $(0.25, 0.6, 0.75)$:



Client-side deployment

- Instead of looping over 144 sweet spots in search of the best sweet spot, we propose an efficient way requiring only 24 iterations to find the best sweet spot.
- We introduce how to perform only 1 search to find the closest sweet spot among 6 sweet spots within the same projection.
- Take the user's viewpoint at $(0.25, 0.6, 0.75)$ and projection V_4 for example,

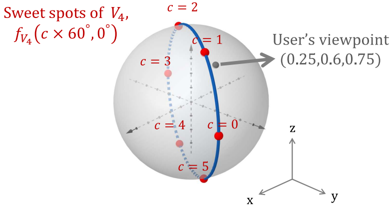


V_4 under the coordinate of P

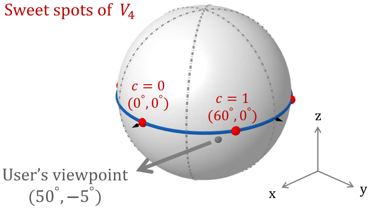
Client-side deployment

- Denote $\mathcal{S}$ as the set of 24 ERP versions in SweeTile.
- Define $f_p^{-1} : R^3 \to R^2$ as an inverse function of f_p for $p \in \mathcal{S}$.
- We first inversely project $(0.25, 0.6, 0.75)$ back to ERP coordinate according to V_4:

$$f_{V_4}^{-1}(0.25, 0.6, 0.75) = (50^\circ, -5^\circ).$$

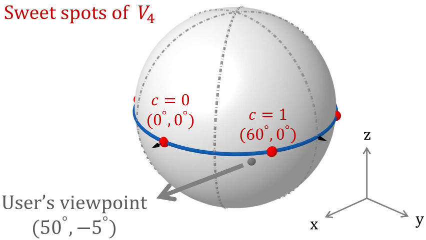


V_4 under the coordinate of P



V_4 under the coordinate of V_4

Client-side deployment



V_4 under the coordinate of V_4

- The closest sweet spot for $(\lambda, \psi) = (50^\circ, -5^\circ)$ is

$$(\lfloor(\lambda + 30^\circ)/60^\circ\rfloor \times 60^\circ, 0^\circ)$$
$$= (\lfloor(50^\circ + 30^\circ)/60^\circ\rfloor \times 60^\circ, 0^\circ) = (60^\circ, 0^\circ).$$

- Note that $\lfloor(\lambda + 30^\circ)/60^\circ\rfloor$ is to find the closest index of the sweet spot with respect to λ.

Client-side deployment

Algorithm 1 Efficient tiling selection algorithm

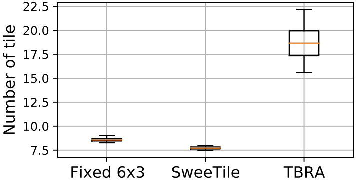
```
Input: $\mathcal{S}$, $(x_0, y_0, z_0)$.
Initialization: $d_{\min} = 2$.
for $p \in \mathcal{S}$ do                                   ▷ Loop over 24 version of ERPs.
    $(\lambda_0, \psi_0) \leftarrow f_p^{-1}(x_0, y_0, z_0)$.  ▷ Find projection point under p.
    $(x_1, y_1, z_1) \leftarrow f_p(\lfloor(\lambda_0 + 30°)/60°\rfloor \times 60°, 0°)$.  ▷ Find the closest sweet spot in p.
    if $\|(x_1, y_1, z_1) - (x_0, y_0, z_0)\|_2 < d_{\min}$ then   ▷ Update $d_{\min}$ and $p_{sweet}$.
        $d_{\min} \leftarrow \|(x_1, y_1, z_1) - (x_0, y_0, z_0)\|_2$.
        $p_{\text{sweet}} \leftarrow p$.
    end if
end for
Return: $p_{\text{sweet}}$.
```

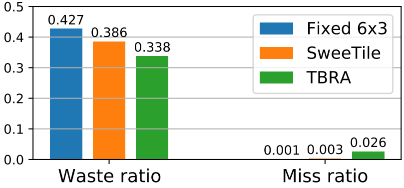
Evaluation

- Simulation environment
  - We apply a real-world head movement dataset [13] comprising 19 videos viewed by 57 users.
  - We implement SOTA viewport predictor [14] with a prediction window of 5s.
  - We replay the throughput trace of a 4G/LTE dataset [15] to simulate the underlying bandwidth fluctuation.
- Benchmark tiling mechanisms
  - **Traditional fixed tiling** (Fixed 6×3 tiling)
  - **Adaptive fixed tiling** (TBRA [11]): TBRA adaptively selects the best tiling among 4×4, 5×5, $6 \times 6, \cdots, 10 \times 10$ tiling with respect to viewport prediction error and bandwidth condition.

Evaluation on efficiency

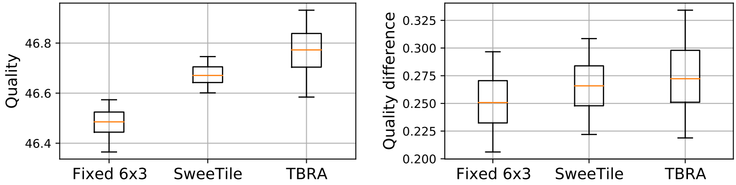
- Number of required tiles
  - SweeTile reduces the required number of tiles by 10% and 40% compared to fixed 6×3 and TBRA, and hence achieves the highest encoding efficiency.
- Waste and miss ratio
  - Similarly, SweeTile achieves higher transmission efficiency by reducing the waste ratio by 9% compared to fixed 6×3.
  - Although TBRA achieves even higher transmission efficiency (lower waste ratio), such improvement comes at the cost of higher miss ratio due to the error of viewport prediction.





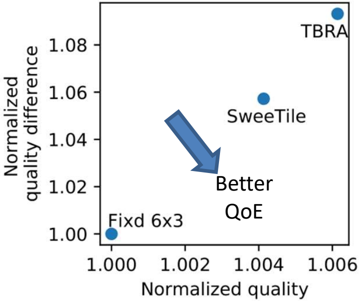
Achievable quality

- We verify the achievable quality by measuring the video quality in (1) peak signal-to-noise ratio (PSNR) and (2) the difference of PSNR between adjacent video segments.
  - PSNR: TBRA achieves the highest PSNR since it wastes less resource by fine-grained tiling.
  - Difference of PSNR: fixed 6×3 achieves the highest stability.



Achievable quality

- We further plot the tradeoff between quality and quality difference normalized with respect to fixed 6×3 tiling.



- We could see a clear tradeoff between quality and quality difference, where TBRA achieves the best quality while fixed 6×3 tiling achieves the lowest quality difference.

Achievable quality

- These results indicate that SweeTile may not be the first option if the QoE is the first priority. However, SweeTile is suitable for light-wiegth VR devices with limited computation and power consumption since **SweeTile requires substantially lower number of tiles to cover users' FoV**.

Conclusion

- Compared to fixed 6×3, SweeTile *breaks the tie of the well-known tradeoff and improves both the encoding and transmission efficiency*.
- If the computation resource and power consumption are limited on the client side (commonly for light-weight VR devices), SweeTile pops up as a cost-effective solution since
  - SweeTile achieves superior encoding efficiency,
  - the overall computation overhead to determine the best tiling is extremely light.
- SweeTile could serve as a promising building block for viewport prediction, tile selection, and rate adaptation for 360° streaming.

Reference I

[1] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a Moving Target: Motion-Prediction-Based Transmission for 360-Degree Videos," in *Proc. IEEE International Conference on Big Data*, 2016, pp. 1161–1170.

[2] C. Perfecto, M. S. Elbamby, J. D. Ser, and M. Bennis, "Taming the Latency in Multi-User VR 360°: A QoE-Aware Deep Learning-Aided Multicast Framework," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, 2020.

[3] C. Brown, "Bringing pixels front and center in vr video," 2017. [Online]. Available: https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/

Reference II

[4] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OpTile: Toward Optimal Tiling in 360-Degree Video Streaming," in *Proceedings of the 25th ACM International Conference on Multimedia*, ser. MM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 708–716. [Online]. Available: https://doi.org/10.1145/3123266.3123339

[5] C. Zhou, M. Xiao, and Y. Liu, "ClusTile: Toward Minimizing Bandwidth in 360-degree Video Streaming," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 962–970.

Reference III

[6] C. Madarasingha and K. Thilakarathna, “VASTile: Viewport Adaptive Scalable 360-Degree Video Frame Tiling,” in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 4555–4563. [Online]. Available: https://doi.org/10.1145/3474085.3475613

[7] J. Wöbbeking, “Leak reveals sales figures for meta quest devices,” 2023. [Online]. Available: https://mixed-news.com/en/leak-reveals-sales-figures-for-meta-quest-devices/

[8] J. Porter, “Everything we know about apple’s mixed reality headset,” 2023. [Online]. Available: https://www.theverge.com/23689334/apple-mixed-reality-headset-augmented-virtual-reality-ar-vr-rumors-s

Reference IV

[9] A. Robertson, "Oculus quest vs. oculus quest 2: what's the difference?" 2020. [Online]. Available: https://www.theverge.com/21433030/oculus-quest-2-vr-headset-specs-comparison-htc-valve-microsoft

[10] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 Video Delivery over Cellular Networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ser. ATC '16. Association for Computing Machinery, 2016, p. 1–6. [Online]. Available: https://doi.org/10.1145/2980055.2980056

Reference V

[11] L. Zhang, Y. Suo, X. Wu, F. Wang, Y. Chen, L. Cui, J. Liu, and Z. Ming, "TBRA: Tiling and Bitrate Adaptation for Mobile 360-Degree Video Streaming," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 4007–4015. [Online]. Available: https://doi.org/10.1145/3474085.3475590

[12] Y. Sauer, A. Sipatchin, S. Wahl, and M. García García, "Assessment of Consumer VR-headsets' Objective and Subjective Field of View (FOV) and its Feasibility for Visual Field Testing," *Virtual Reality*, vol. 26, no. 3, p. 1089–1101, 2022.

Reference VI

[13] Y. Rai, J. Gutiérrez, and P. Le Callet, "A Dataset of Head and Eye Movements for 360 Degree Images," in *ACM Multimedia Systems Conference (MMSys)*, 2017, p. 205–210.

[14] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze Prediction in Dynamic 360° Immersive Videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.