Composing Error Concealment Pipelines for Dynamic 3D Point Cloud Streaming

I-Chun Huang (ichun0627@gmail.com)

Networking and Multimedia Systems Lab

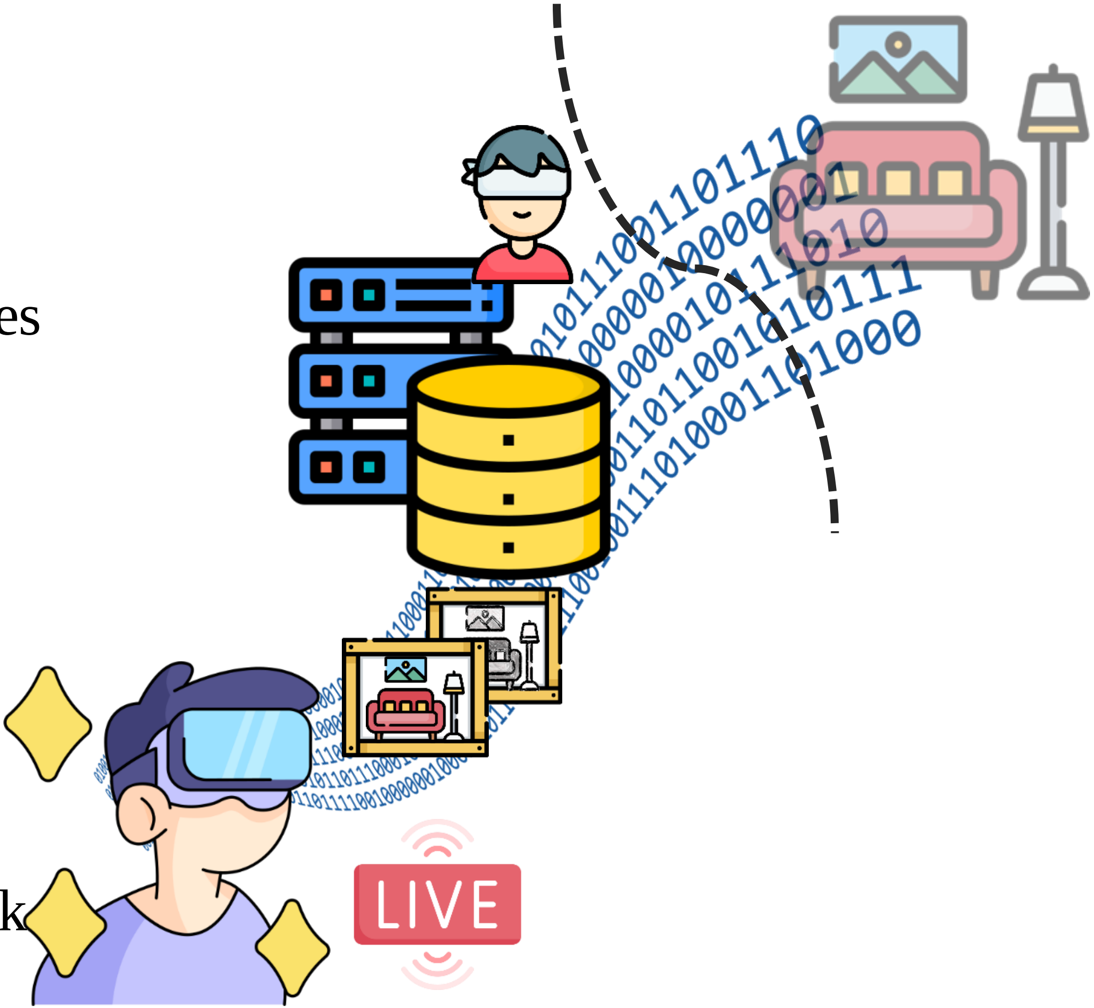
Department of Computer Science, National Tsing Hua University

NMSL@NTHU

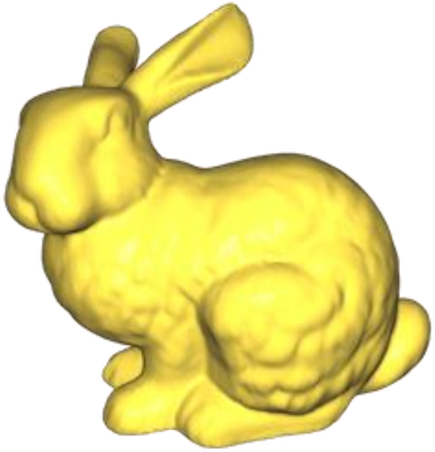
Networking and Multimedia Systems Lab

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



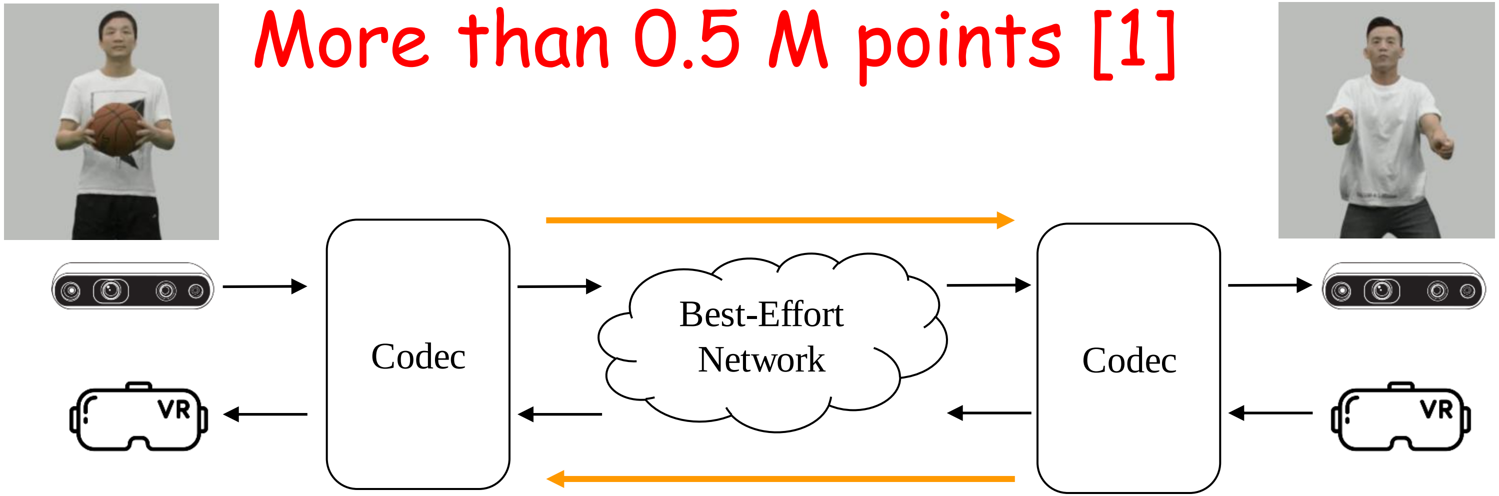
3D Representations



- Meshes:
  - Approximates the surfaces of objects through polygons
  - Not native output data types of any capturing sensors
- Point Cloud:
  - Represent in multiple points (coordinate + attribute)
  - Native data format from some sensors
  - Easy to interact, such as: voxelization, motion estimation, and segmentation……
  - Better compression method for dynamic point cloud

Point Cloud Teleconferencing

More than 0.5 M points [1]

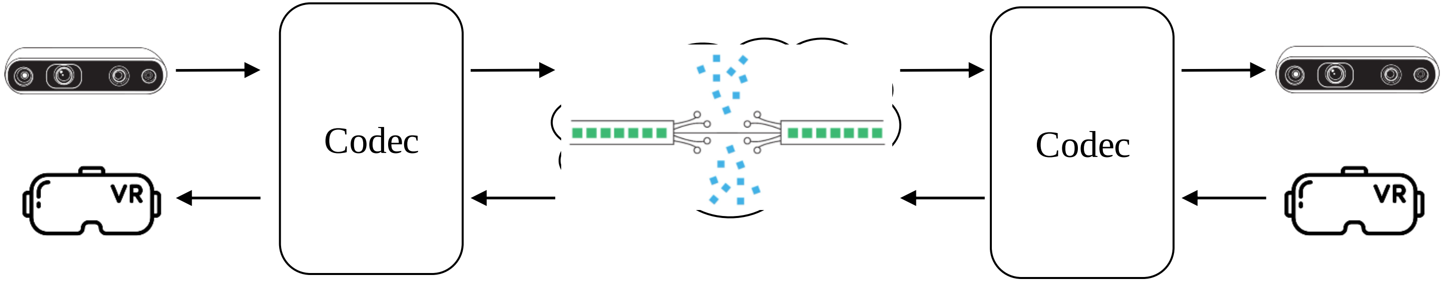


So compression before streaming is a must!

[1] C. Cao, M. Preda, and T. Zaharia, “3D point cloud compression: A survey,” ACM International Conference on 3D Web Technology (Web3D’19), pages 1–9, July 2019.

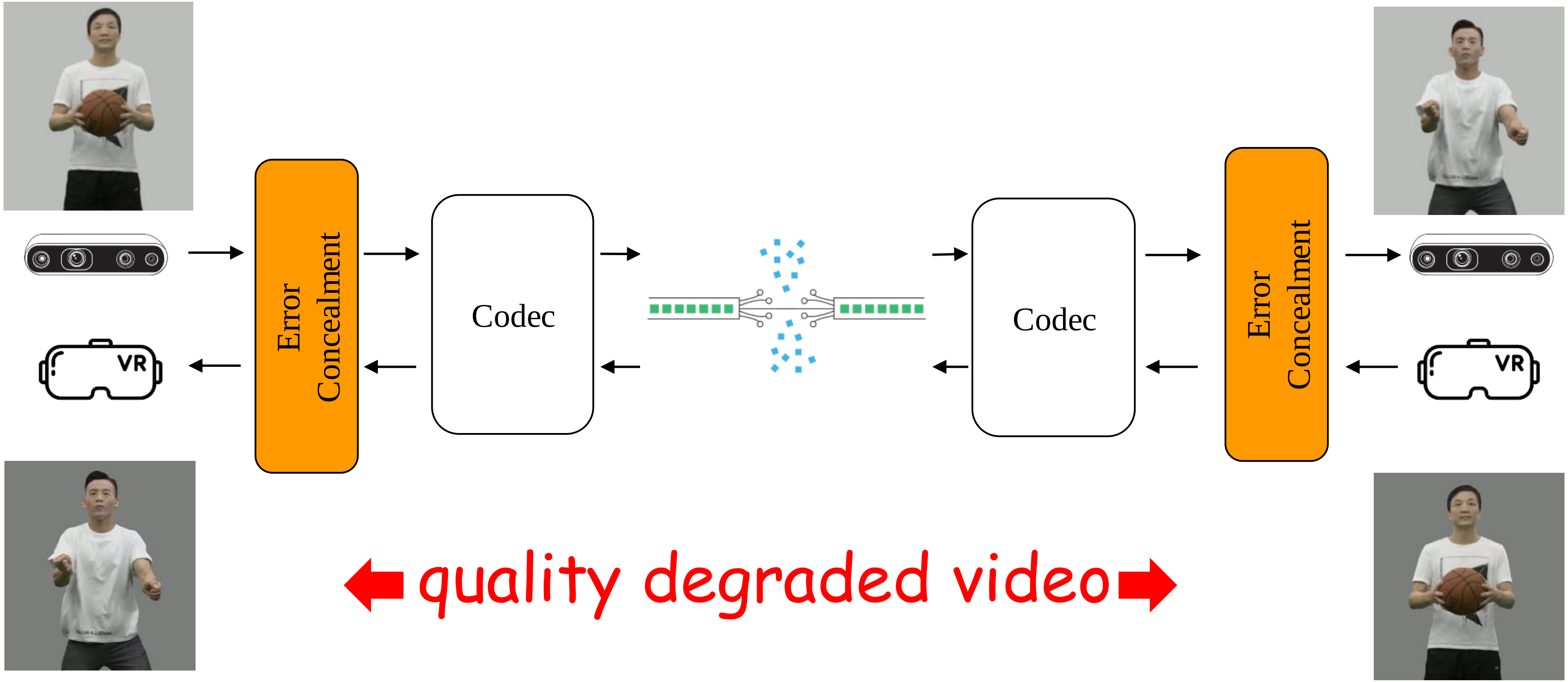
Lost or Late Packets

Packet loss!!

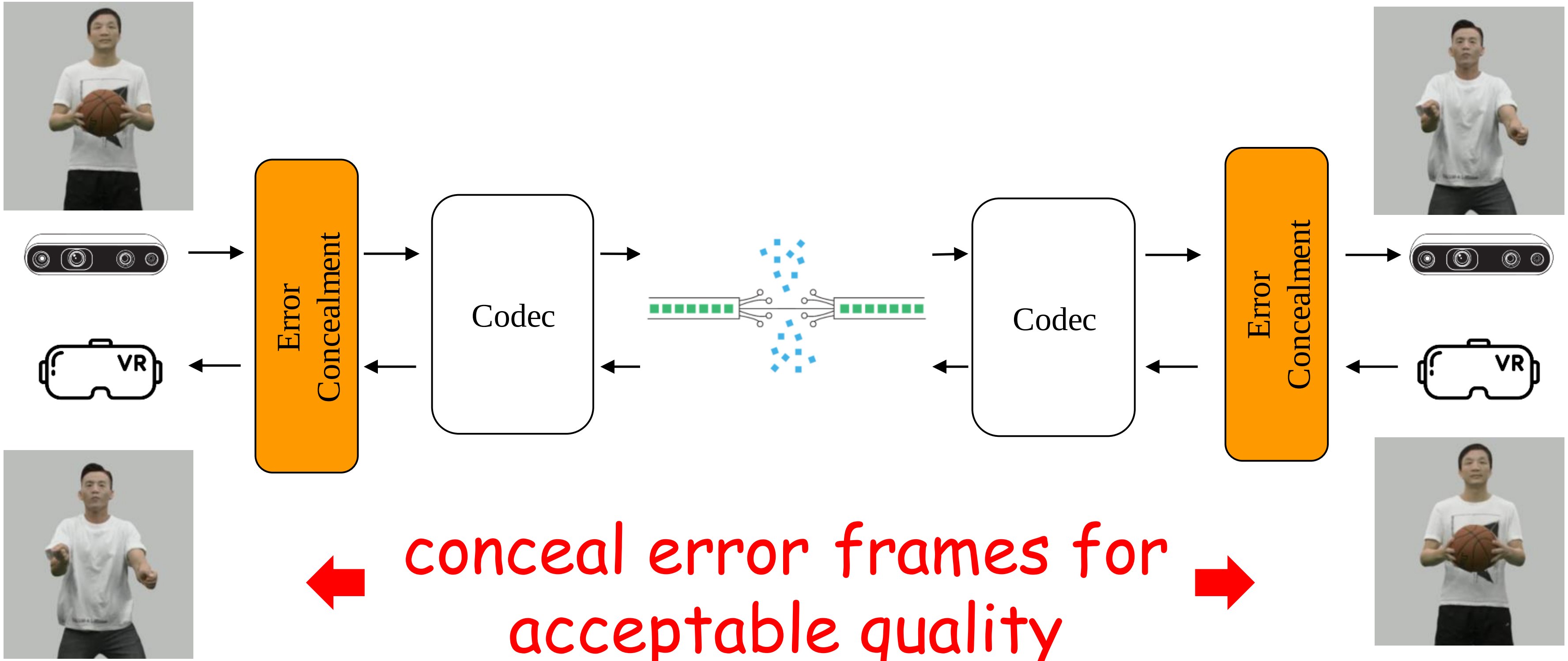


degrades visual quality

Error Concealment Module (1/2)



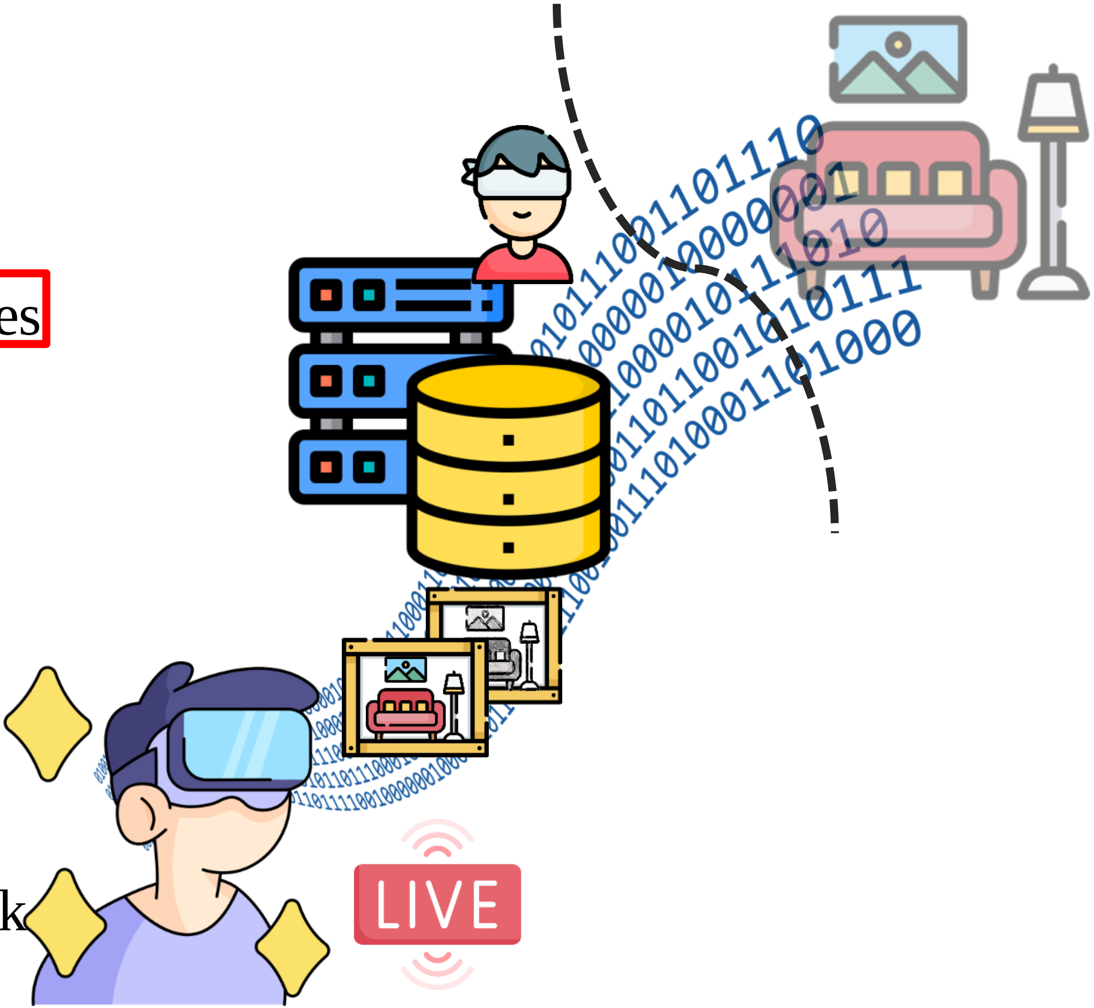
Error Concealment Module (2/2)



conceal error frames for acceptable quality

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Video-based Point Cloud Compression (V-PCC)

- V-PCC [1] bitstreams consist of Network Abstraction Layer Units (NALUs)
  - Geometry Video Data (GVD)
  - Attribute Video Data (AVD)
  - Occupancy Video Data (OVD)
- Two types of concealment
  - Attribute concealment (easier [2])
  - ***Geometry concealment (critical)***

[1] MPEG 3DGC. V-PCC codec description v12. International Organization for Standardization Meeting Document, 2020.
[2] T.-K. Hung, I.-C. Huang, S. R. Cox, W. T. Ooi, and C.-H. Hsu, "Error Concealment of Dynamic 3D Point Cloud Streaming," in Proc. of the ACM MM, October 2022.

Error Concealment for 3D Point Clouds

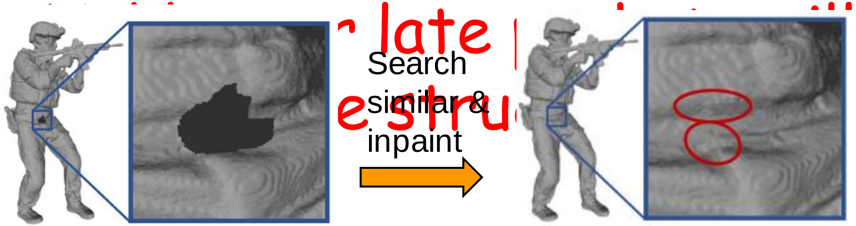
- Traditional error concealment methods:
  - Frame copy
  - Temporal concealment
  - Spatial concealment
- Can’t apply method in V-PCC since patches are at different places [1]

[1] Li. et al. "Advanced 3D Motion Prediction for Video-Based Dynamic Point Cloud Compression," in IEEE Transactions on Image Processing, 2020

Related Work

Can we apply them to our work?

- Repair errors within a point cloud frame:
  - Point cloud completion [1]
  - Point cloud inpainting [2]
- Generate a new point cloud frame:
  - Interpolation [3, 4]



No! Limit to dataset or point number

We subjectively compare our proposed methods with [3, 4] later

[1] Jingdao. Et al. "Point Cloud Scene Completion of Obstructed Building Facades with Generative Adversarial Inpainting," Sensors, 2020.
[2] Wei. et al. "Local Frequency Interpretation and Non-local Self-similarity on Graph for Point Cloud Inpainting," IEEE Transactions on Image Processing 2019
[3] Y. Zeng, Y. Qian, Q. Zhang, J. Hou, Y. Yuan, and Y. He, "IDEA-Net: Dynamic 3D Point Cloud Interpolation via Deep Embedding Alignment," in Proc. of the CVPR'22, June 2022.
[4] A. Akhtar, Z. Li, G. Van der Auwera, and J. Chen, "Dynamic Point Cloud Interpolation," in Proc. of the ICASSP'22, May 2022.

Our Solution Approach (1/2)

- Challenges:
  - Cannot apply traditional error concealment methods in 2D space
  - Current point cloud completion/inpainting papers limit to small region
  - Algorithms that suit for different usage scenario are not exist
  - Seldom work compared between objective and subjective results
- Contributions:
  - Propose a general multi-stage pipeline framework for error concealment
  - Develop and quantitatively compare a suite of algorithms for individual stages
  - Evaluate the end-to-end performance of several representative pipelines and conduct a user study

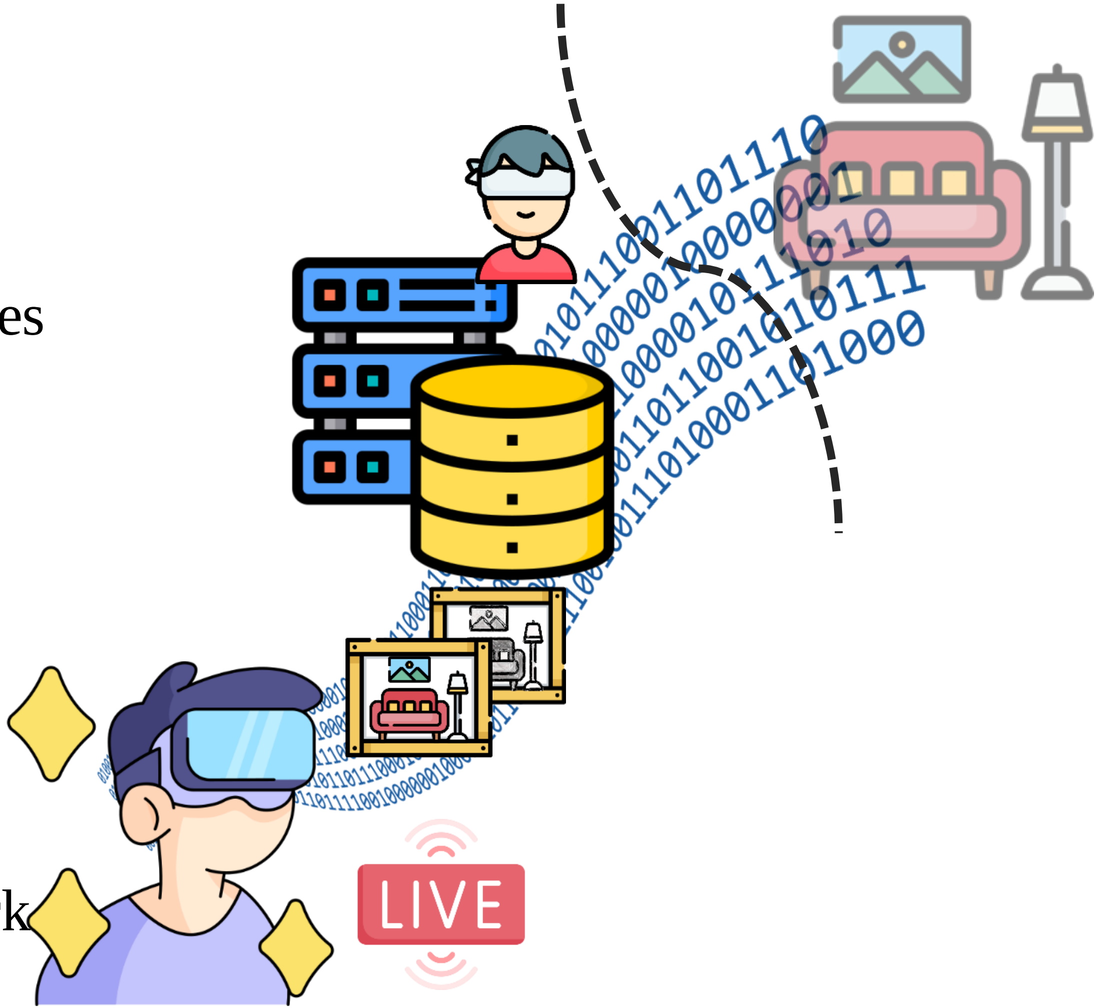
Our Solution Approach (2/2)

- Different with prior work [1]:
  - We proposed multiple algorithms for individual stages, which can be mixed and matched into different error concealment algorithms
  - We proposed new metrics to evaluate the effectiveness of our algorithms in different stages
  - We redesigned and conducted new objective experiments to evaluate our proposed representative pipelines
  - We conducted a user study to subjectively compare our proposed representative pipeline against the state-of-the-art learning-based interpolation algorithms
  - We shared our source code with the research community on GitHub

[1] T.-K. Hung. On Error Concealment of Dynamic 3D Point Cloud Streaming. Master's thesis, National Tsing Hua University, 2022.

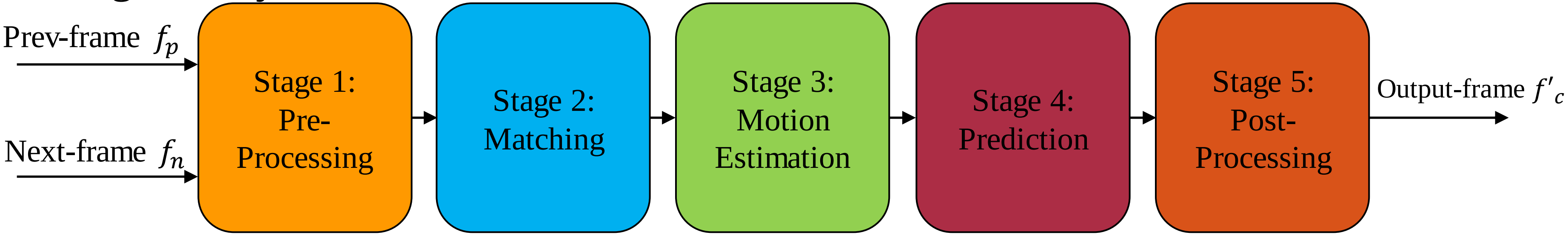
Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Error Concealment Framework

- We propose a suite of error concealment algorithms for concealing geometry distortion



- Target of individual stages:
  1. Pre-processing: downsamples point clouds to reduce the computational overhead
  2. Matching: creates a matching table, which provides the point-to-point correspondence
  3. Motion Estimation: generates motion vectors for either individual points or cubes
  4. Prediction: produces an interpolated point cloud
  5. Post-processing: apply refinement algorithms to further improve the concealed quality

Design Objective

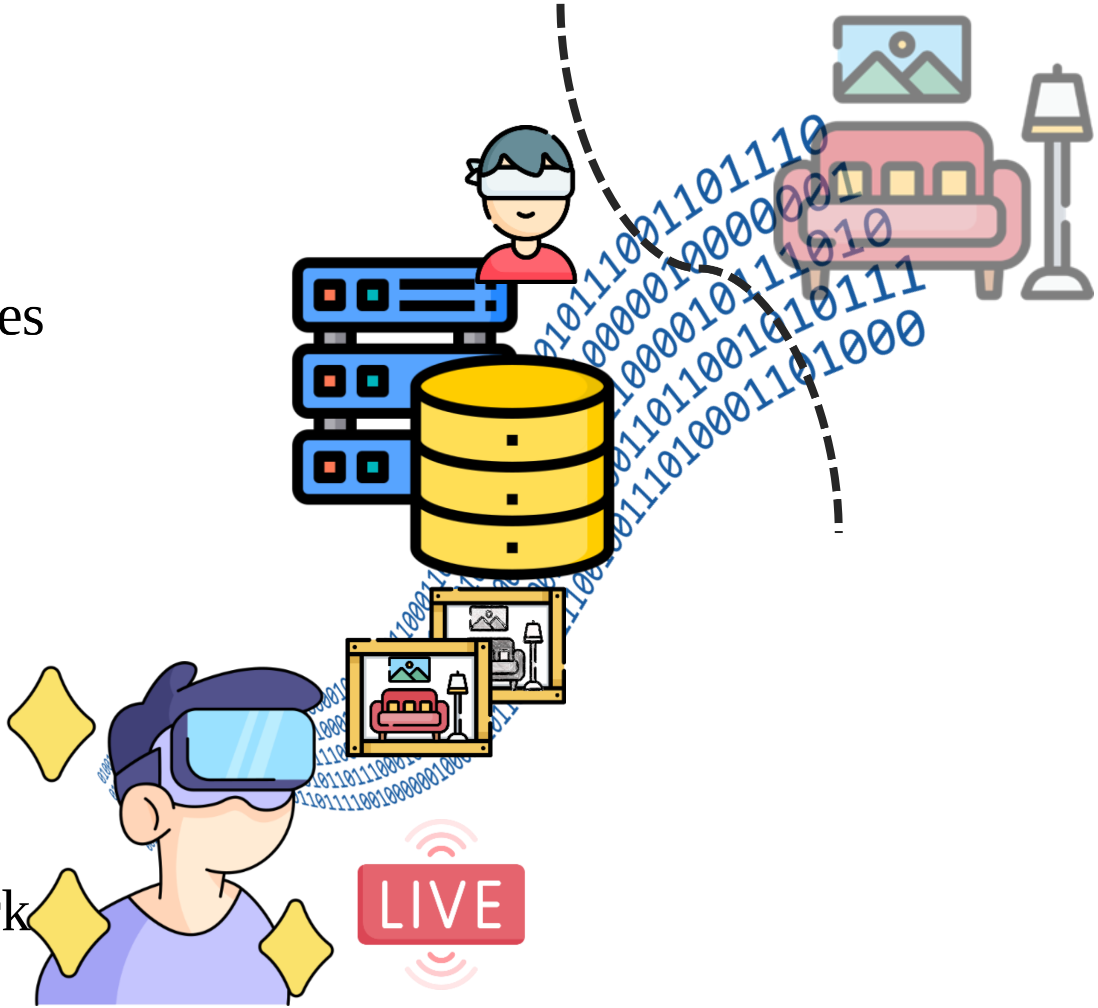
- Better visual quality:
  - 3D metrics: GPSNR (Geometry-PSNR), Hausdorff distance, CPSNR (Color-PSNR)
  - 2D metrics : PSNR, SSIM, VMAF, where we use Open3D to project the point cloud into 8 horizontal 2D views [1].
- For each stage, we propose different algorithms and evaluate the performance on a synthetic dataset [2], for example, we define:
  - Spatial smoothness: $d_g = \Delta_{cor}(p, p')$
  - Temporal smoothness $d_r = \Delta_{angle}(p, p')$

p: estimated point
p': ground-truth point
Δ_{cor}: geometry difference
Δ_{angle}: direction difference

[1] E. Zerman, I. Kulkarni, and A. Smolic. "User Behavior Analysis Of Volumetric Video In Augmented Reality," in Proc. of the International Conference on Quality of Multimedia Experience (QoMEX'21), 2023.
[2] Y.-C. Sun, I.-C. Huang, Y. Shi, W. T. Ooi, C.-Y. Huang, and C.-H. Hsu, "A dynamic 3D Point Cloud Dataset for Immersive Applications," in Proc. of the ACM MMsys'23, June 2023.

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Stage2: Matching Algorithms 1 ~~2~~ ~~3~~ 4

- Goal: Complete the matching table from the anchor frame to current frame
- Input: f_p, f_n, Output: $t_{p\rightarrow n}$ (matching table)
- Alternative algorithms :
  - Nearest-Neighbor (NN):
    - $t_{p\rightarrow n} = \{argmin_{q\in f_n} \Delta_{cor}(p,q) | p \in f_p\}$
  - Query-Radius (QR):
    - $s(p,\tau) = \{q \in f_n | \Delta_{cor}(p,q) < \tau\}$
    - $\Delta(p,q) = \alpha\Delta_{cor}(p,q) + (1-\alpha)\Delta_{rgb}(p,q) + \beta t(q)$
  - Adaptive Query-Radius (AQR):
    - Adaptively select τ in QR to reduce temporal redundancy and increase temporal/spatial smoothness

Ground-truth Matching

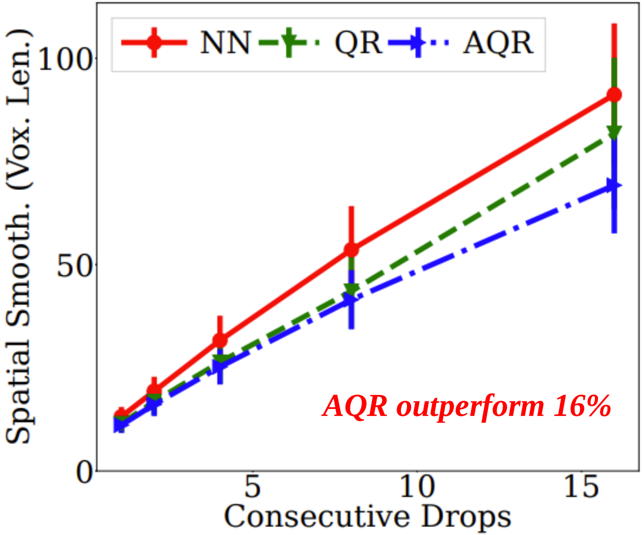
NN Matching

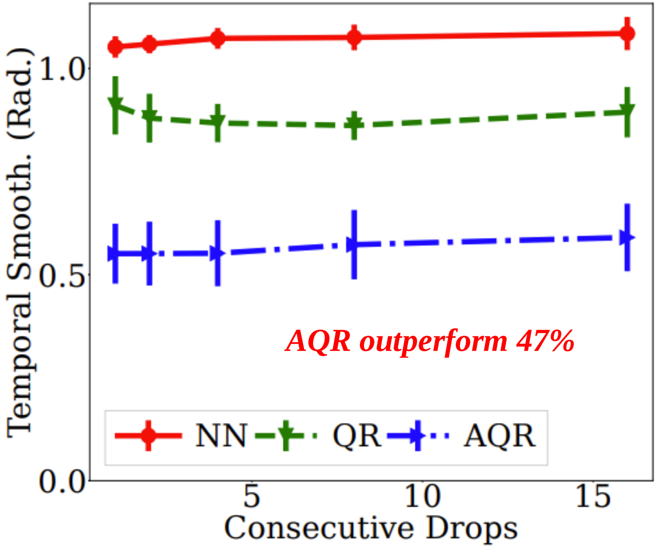
Δ_{cor}: geometry difference Δ_{rgb}: color difference
τ: the query radius $s(.)$: a set of points
α, β: adjustable factor $t(q)$: matching time table

Stage 2: Matching Evaluations

- Quality comparison

Average height of the avatar is 1024 voxel length





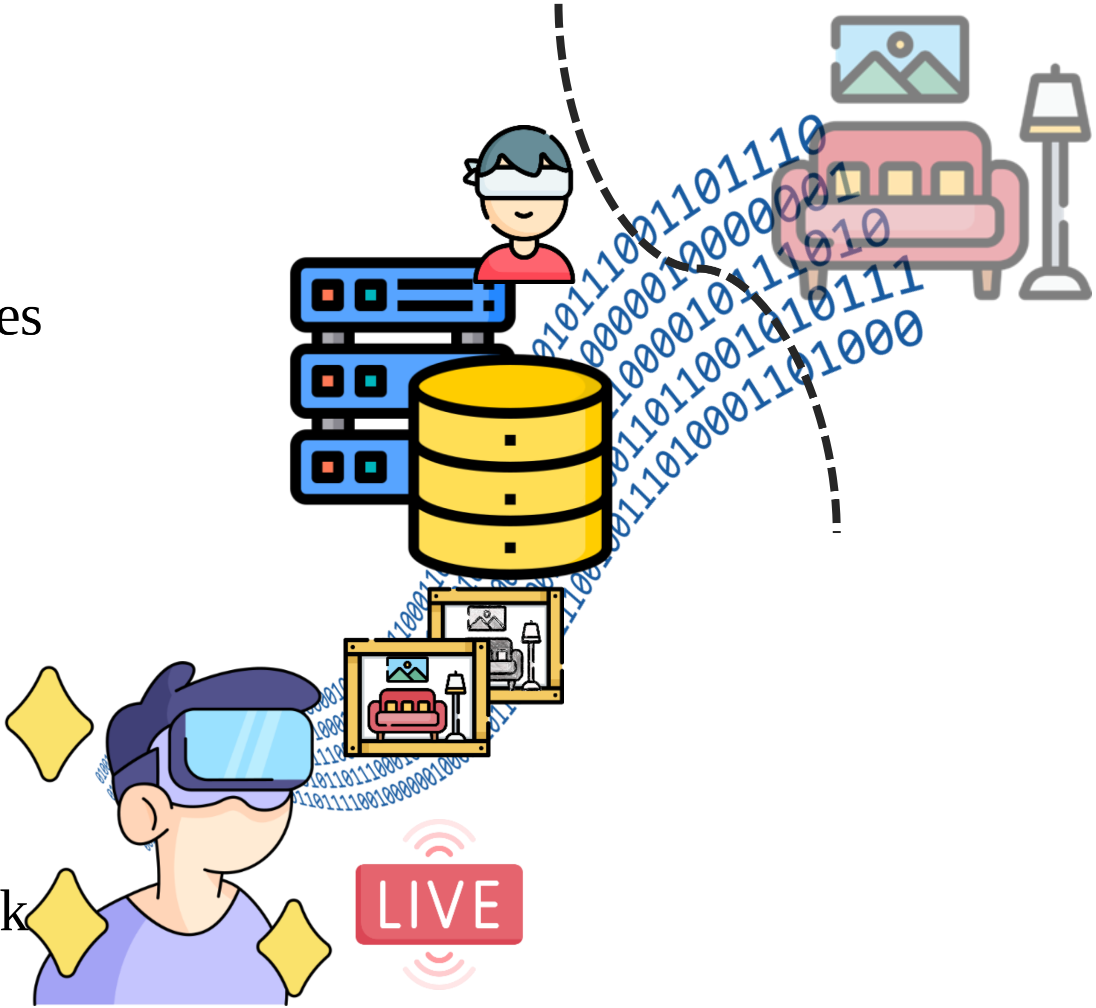
- Time complexity comparison:
  - NN: $O(|f|log|f|)$
  - QR: $O(|f|^2)$
  - AQR: $O(|f|^2)$

$|f|$: number of points in the frame

- AQR perform better in spatial smoothness when frame drops increase
- AQR consistently perform better in temporal smoothness

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work

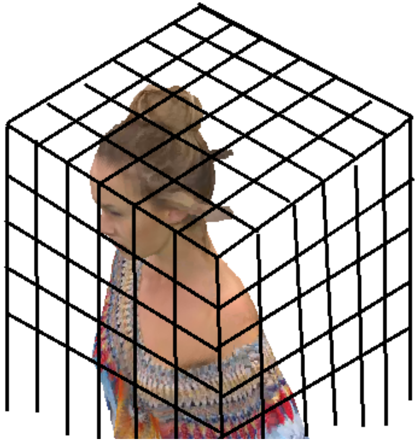


Stage3: Motion Estimation Algorithms

- Goal: calculate motion vectors from the anchor frame to current frame
- Input: $f_p, f_n, t_{p\to n}$ Output: $m_p\ or\ m_c$
- Alternative algorithms :
  - Point Motion (PM) :
    - $m_p = \{\overrightarrow{p_i t_{p\to n}(p_i)}\,|p_i \in f_p\}$
  - Cube Motion (CB) :
    - To mitigate the outliers in matching tables, we slice f_p into multiple cube c
    - $\mathbf{m}_c = \left\{\overrightarrow{\Sigma_{p_i \in c_j} p_i t_{p\to n}(p_i)}/|c_j| \,\middle|\, c_j \in \mathbf{c}\right\}$

m_p: per point motion vector

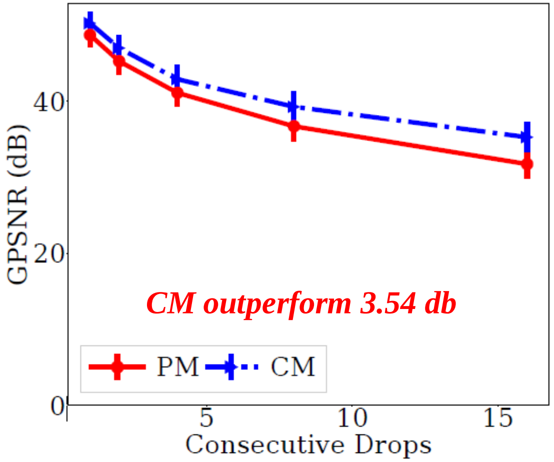
m_c: per cube motion vector

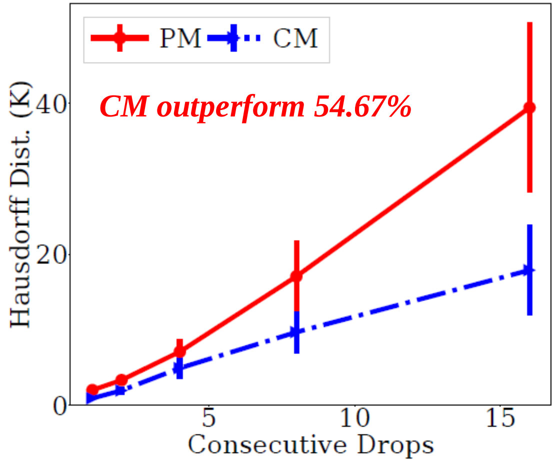


Slice point cloud into multiple cubes

Stage 3: Motion Estimation Evaluations

- Quality comparison (using NN matching)



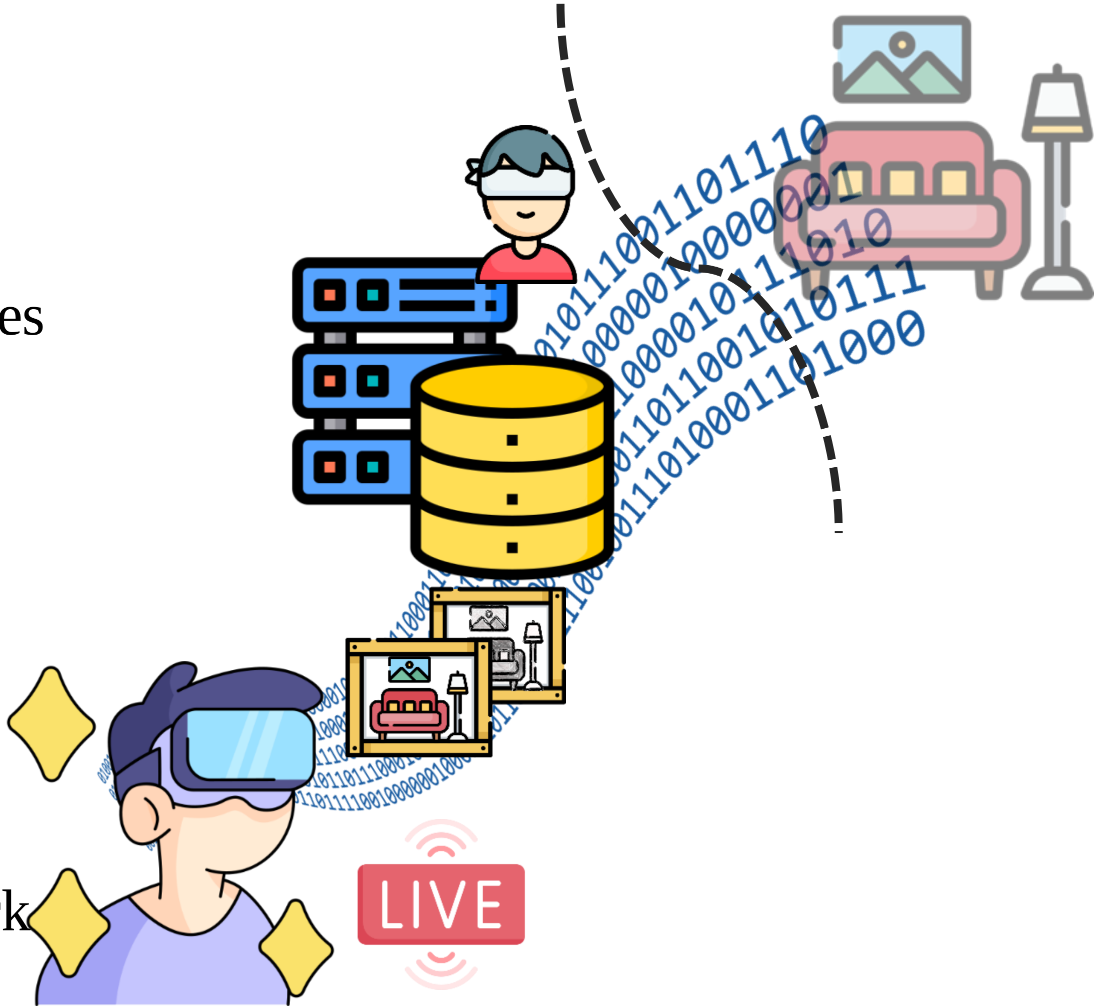


- Time complexity comparison:
  - PM: $O(|f|)$
  - CM: $O(|f|)$

- CM mitigate the errors in point matching

Outline

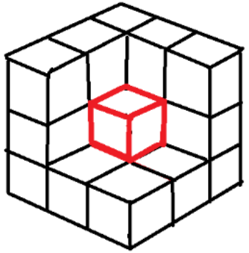
- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Stage4: Prediction Algorithms

- Goal: to generate output frame with a good visual quality
- Input: $f_p, f_n, m_p/m_c$ Output: f'_c
- Alternative algorithms: r: ratio of f_c between f_p
  - Point-based Prediction (PP):
    - $\mathbf{f}'_c = \{p_i + \mathbf{f}_c.r \times \mathbf{m}_p(p_i) | p_i \in \mathbf{f}_p^{l_2}\}.$
  - Cube-based Prediction (CP):
    - $\mathbf{c}_c = \{c_j + \mathbf{f}_c.r \times \mathbf{m}_c(c_j) | c_j \in \mathbf{c}\}$
    - $\mathbf{f}'_c = \{p_i + \mathbf{f}_c.r \times \mathbf{m}_c(c_j) \big| p_i \in \mathbf{f}_p^{l_2}\}$
  - Neighboring-cube-based Prediction (NP):
    - $e_{i,k} = 1/\Delta_v(p_i, c_k),$
    - $\mathbf{f}'_c = \left\{ p_i + \mathbf{f}_c.r \times \frac{\sum_{c_k \in \mathbf{n}_j \cup \{c_j\}} e_{i,k} \mathbf{m}_c(c_k)}{\sum_{c_k \in \mathbf{n}_j \cup \{c_j\}} e_{i,k}} \middle| p_i \in \mathbf{f}_p^{l_2} \right\}$

Irregular surface between cubes

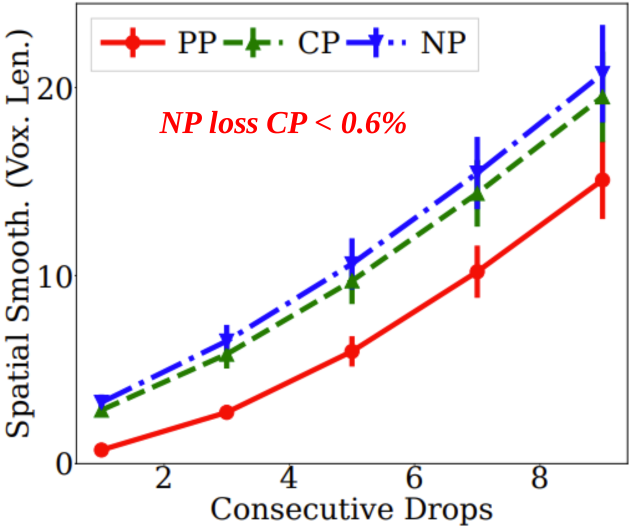


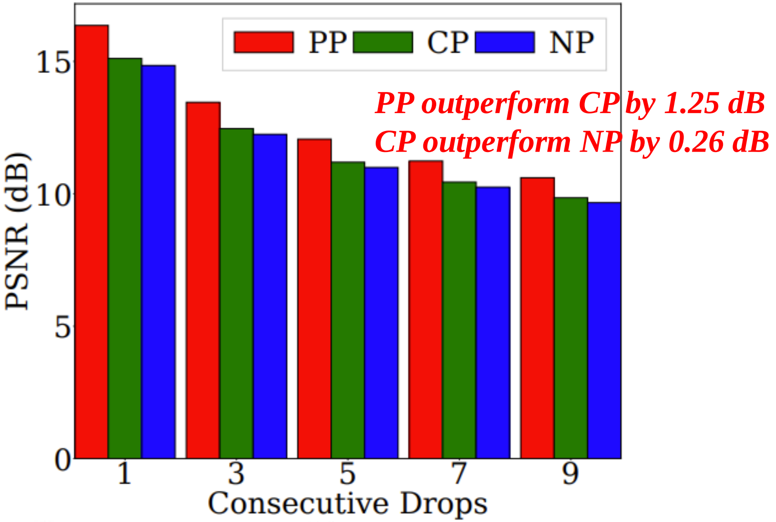
Center cube (red) and its neighboring cubes (in total 26)

Stage 4: Prediction Evaluations

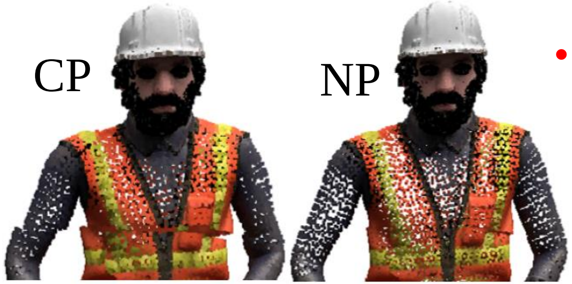
Average height of the avatar is 1024 voxel length

- Quality comparison (with ground-truth matching)





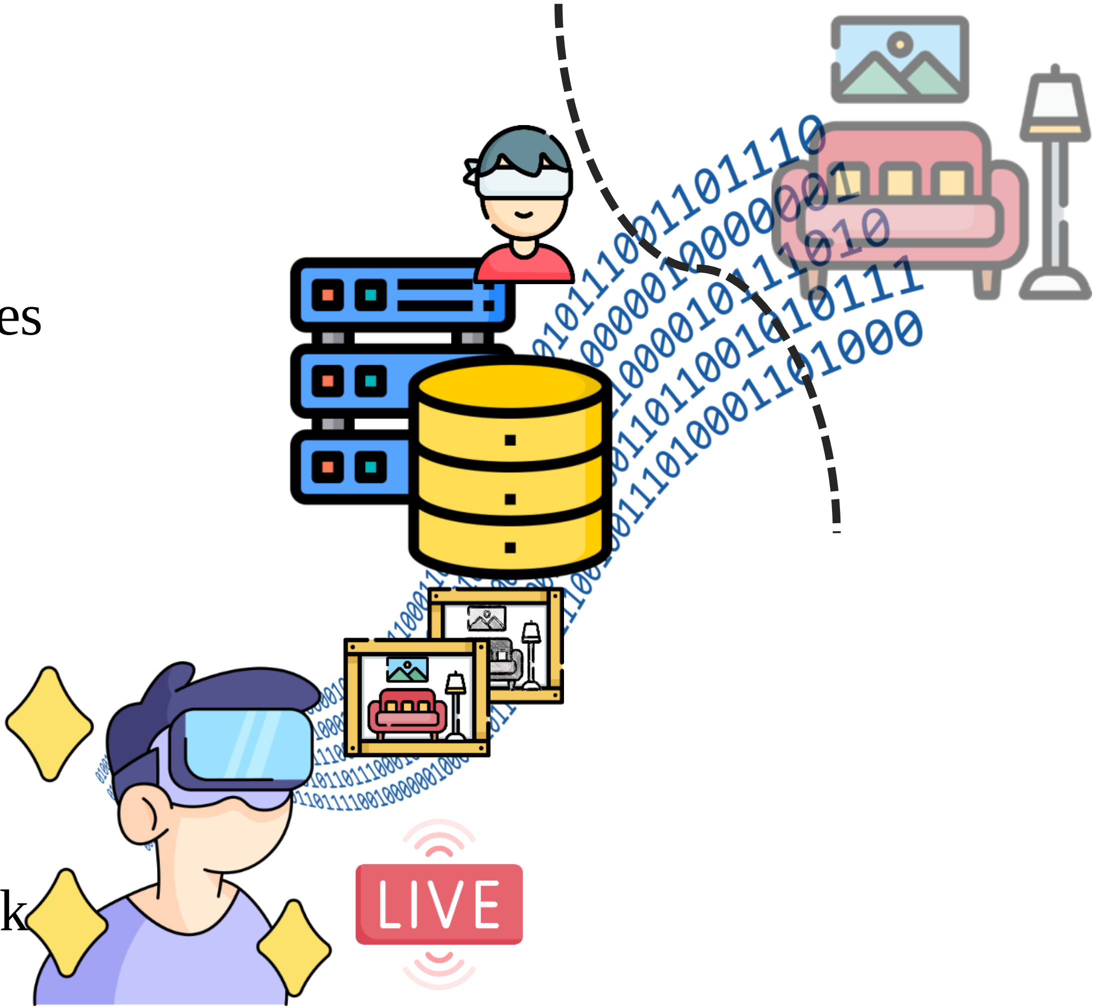
- Time complexity comparison:
  - PP: $O(|f|)$
  - CP: $O(|c|log|c| + |f|)$
  - NP: $O(|c| + |f|)$



- NP loss a little quality but mitigate the not aligned problem

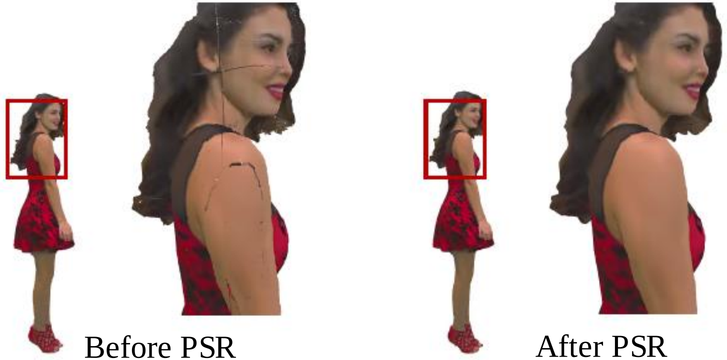
Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Pre- and Post-Processing Stage

- Goal: To mitigate the high complexity caused by too many points & further increase spatial smoothness
- Alternative algorithms:
  - Pre-processing: voxel down-sample [1]
  - Post-processing: poisson surface reconstruction (PSR) [2]



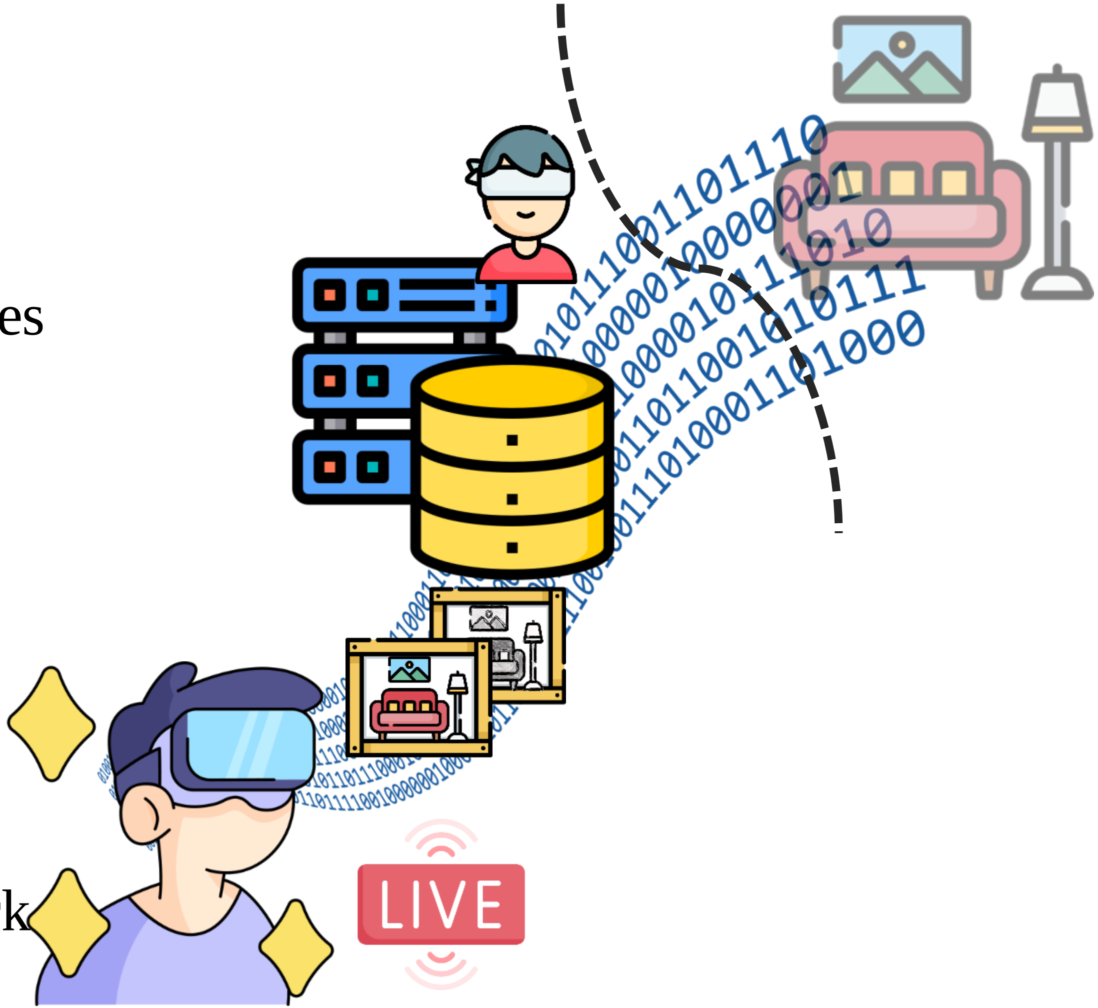
Before PSR After PSR

- Downsample can reduce the temporal redundancy
- PSR can further mitigate the error on the point cloud surface

[1] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern Library for 3D Data Processing," 2023, http://www.open3d.org/.
[2] M. Kazhdan and H. Hoppe, "Screened Poisson Surface Reconstruction," ACM ToG, vol. 32, no. 3, 2013.

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Experimental Setup

- Representative Pipelines

	Name	Pre-Proc.	Matching	M. Est.	Pred.	Post-Proc.
Pipeline	F (Fast)	-	QR	PM	PP	-
	B (Balance)	-	NN	CM	CP	-
	Q (Quality)	Downs.	AQR	CM	NP	-
	Q+	Downs.	AQR	CM	NP	PSR

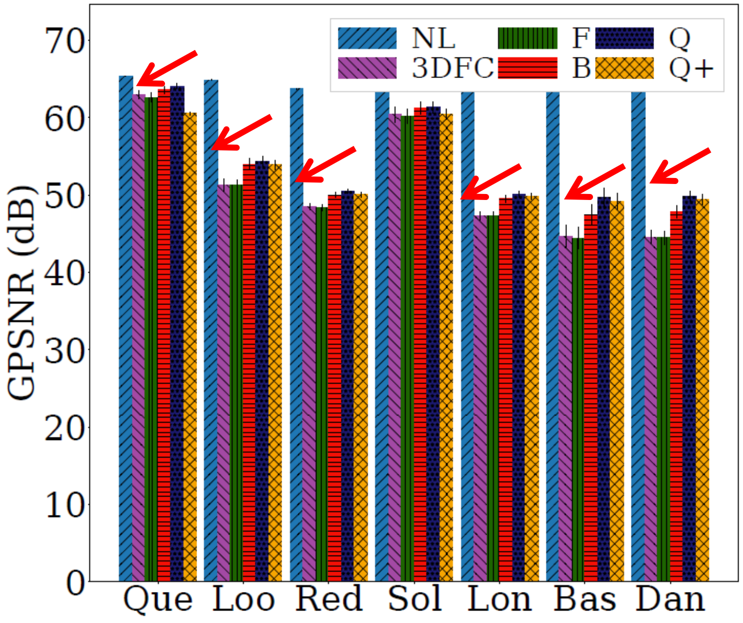
- Pipeline Fast (F): aims for the best time efficiency
  - Query Radius (QR): produces smaller matching than NN and runs fast by selecting good τ
  - Point Motion (PM) and Point Prediction (PP): the most efficient algorithm for interpolation
- Pipeline Balance (B): aims for a balance between running time and quality
  - Nearest Neighbor (NN): runs fast, but it may produce noisy matching tables
  - Cube Motion (CM) and Cube-based Prediction (CP): use cube-based methods to mitigate the noise
- Pipeline Quality (Q): aims for the best visual quality
  - Adaptive-QR (AQR): the best quality matching algorithms
  - Cube Motion (CM) and Neighboring-cube-based Prediction (NP): achieve the best spatial smoothness and good visual quality
- Pipeline Quality+ (Q+): aims for similar quality as Q, but runs faster
  - Pre-processing: downsamples to reduce the computational overhead
  - PSR: applies a refinement algorithm to further improve the concealed quality

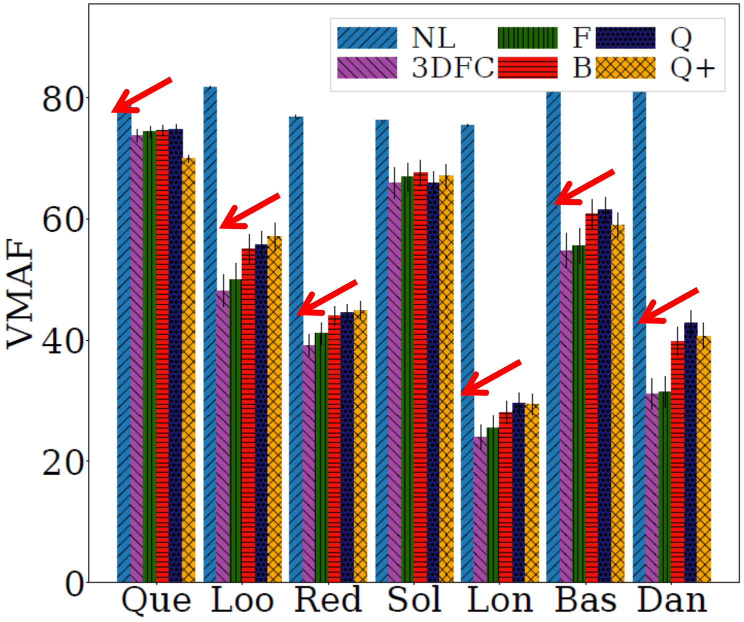
Performance Metrics and Dataset

- Baseline:
  - 2DFC (implemented in video codec), 3DFC
- Distortion Type:
  - Only **geometry distortion** (simulate 1~5 consecutive frame drop)
- Visual metrics
  - 3D: GPSNR (Geometry-PSNR), Hausdorff distance, CPSNR (Color-PSNR)
  - 2D: PSNR, SSIM, VMAF
- Running time (sec):
  - Average running time on 24 random frames using a single CPU core without optimization
- Dataset:

	Queen	**Loot**	**Red&Blk**	**Solider**	**Longdress**	**Basketball**	**Dancer**
Complexity	Low	Low	Low	Low	Medium	High	High
Point Num	1.00 M	0.78	0.70	1.50	0.80	2.90	2.60

Overall Quality of Concealed Frames

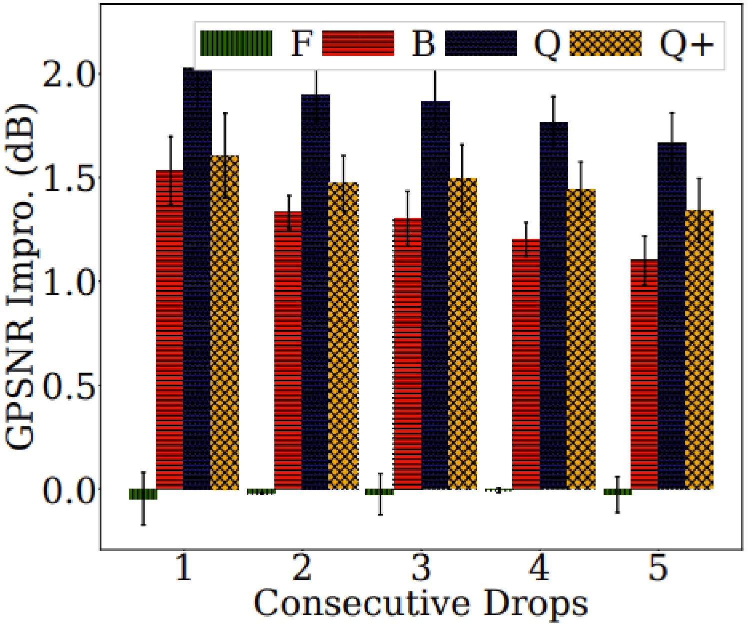




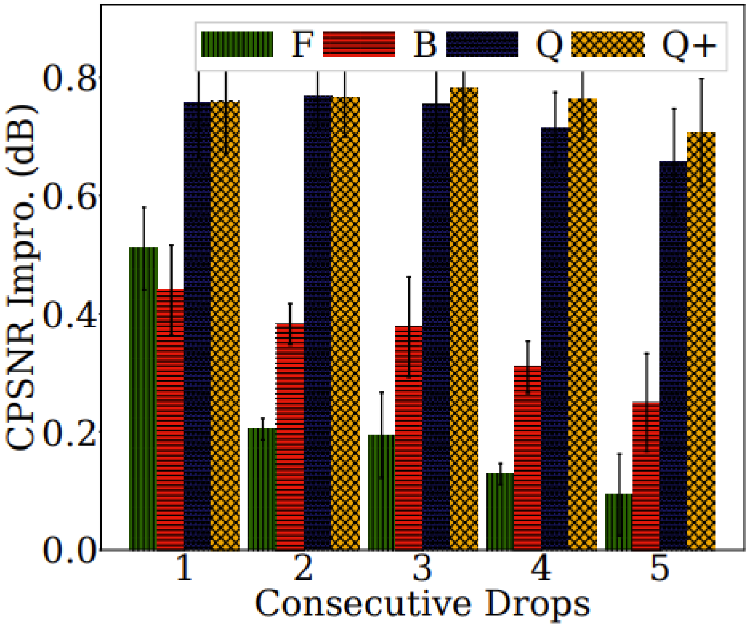
- B and Q outperform 3DFC, but F trails 3DFC in some sequences
- Q outperforms B, B outperforms F in most cases

3D Metrics Improvement from Low-Complexity Red&Blk

Q outperforms 3DFC at most 2.03dB



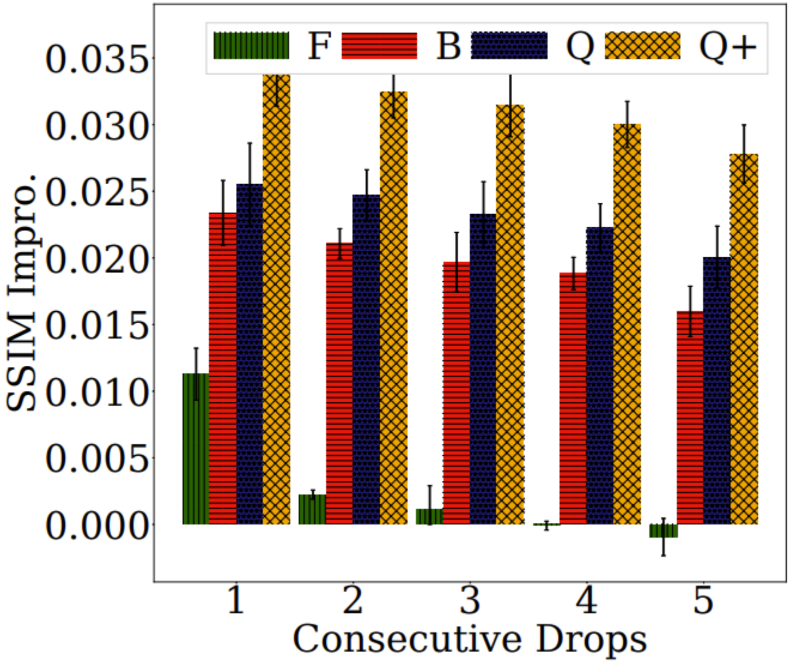
Q outperforms 3DFC at most 0.76dB



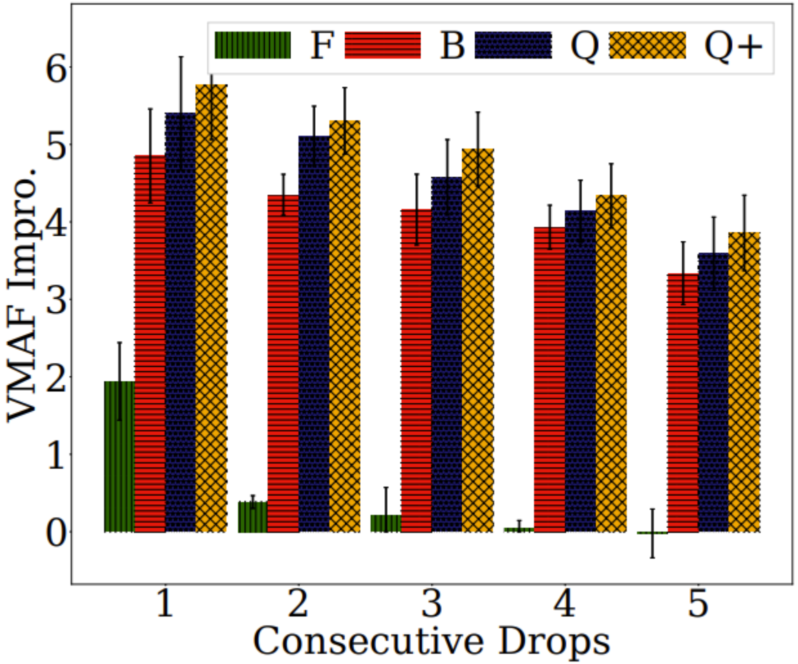
When number of frame drops increases, quality improvement is still observed

2D Metrics Improvement from Low-Complexity : Red&Blk

Q outperforms 3DFC at most 0.025

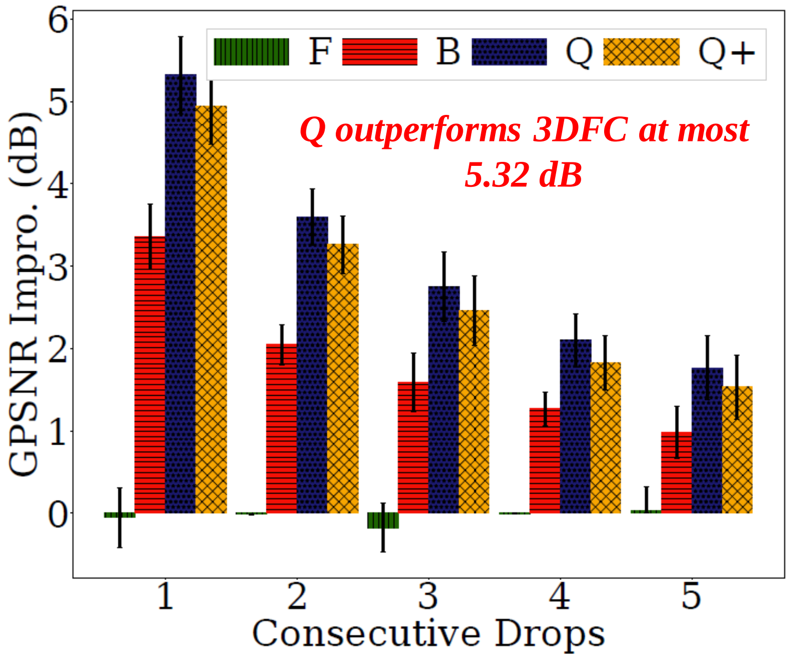


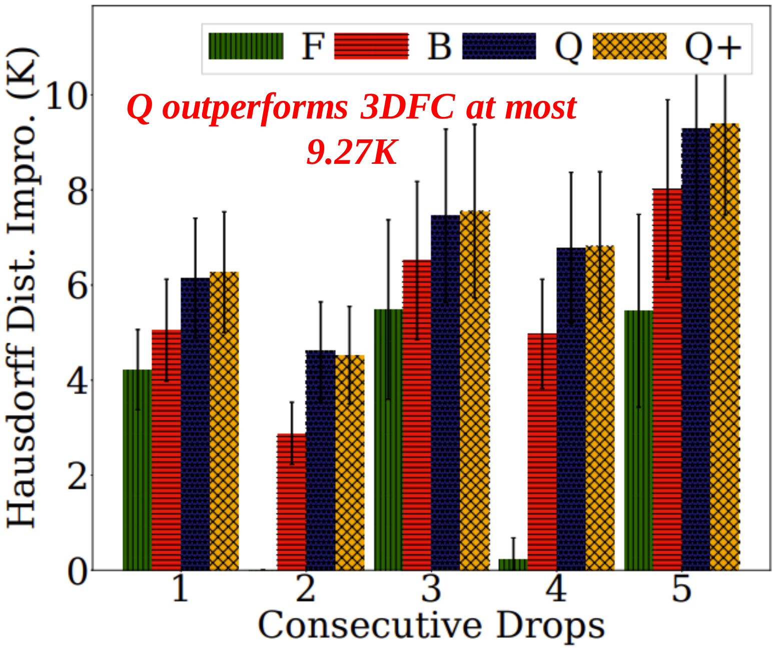
Q outperforms 3DFC at most 5.4



When number of frame drops increases, quality improvement is still observed

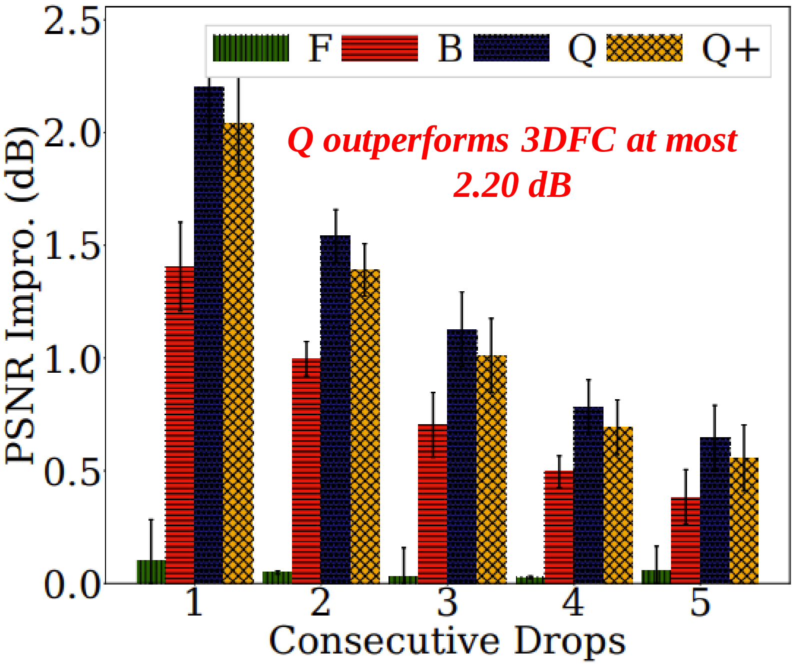
3D Metrics Improvement from High-Complexity Dancer

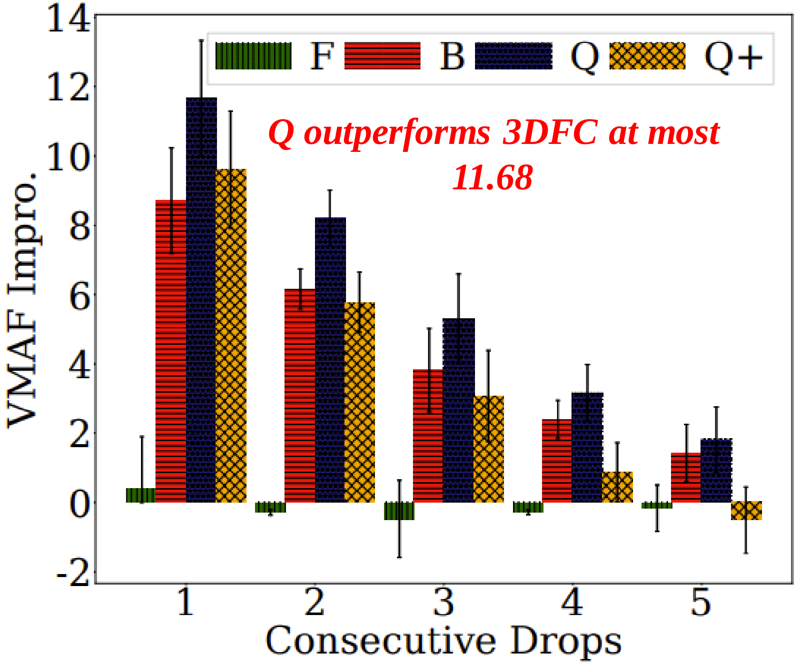




When number of frame drops increases, quality improvement may decrease

2D Metrics Improvement from High-Complexity Dancer





When number of frame drops increases, quality improvement may decrease

Issue of Pipeline Q+

Before PSR

Bad normal estimations

After PSR

Running Time: Seconds

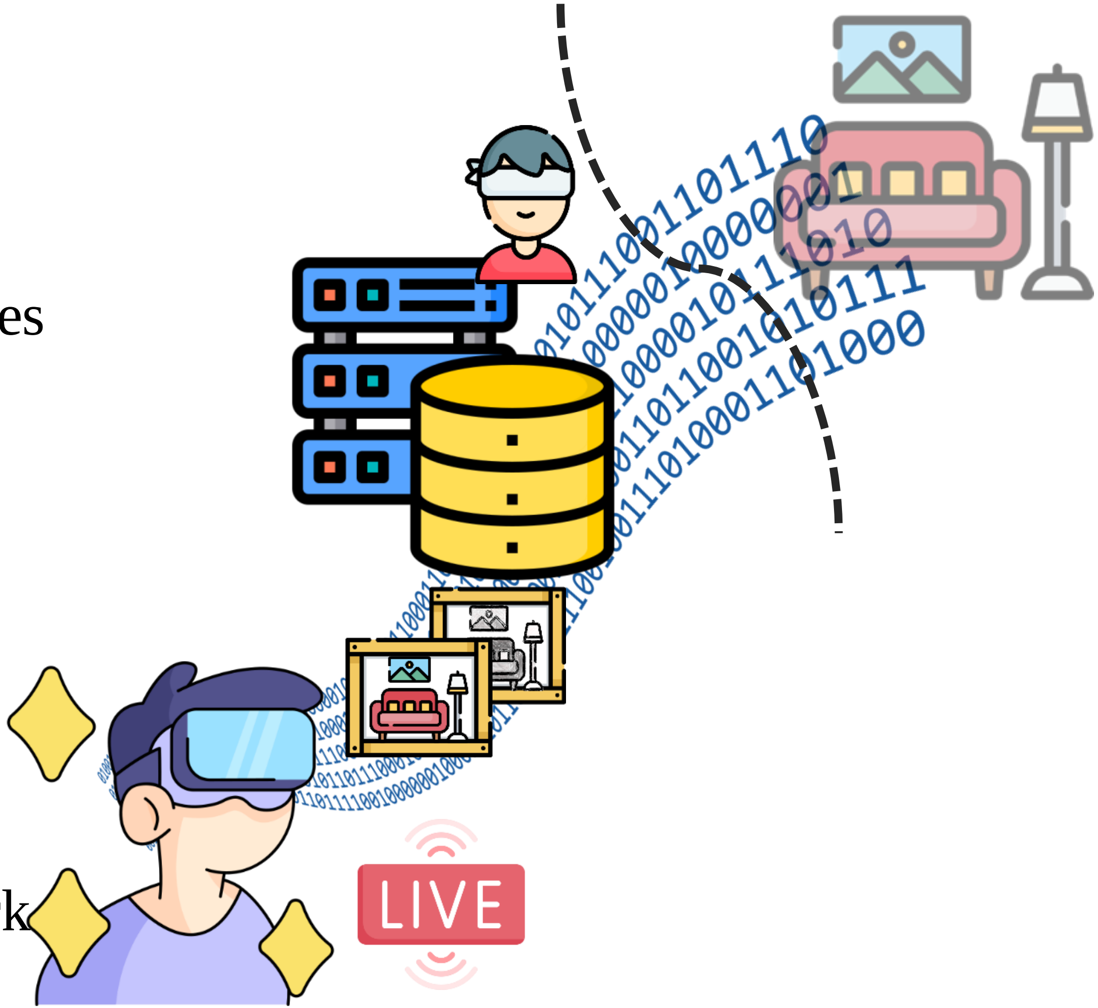
Loot/LongDress/Dancer

Stage / Pipeline	Pre-Proc.	Matching	M. Est.	Pred.	Post-Proc.	Total
Fast	-	QR 1.05/0.96/2.66 (93%/93%/93%)	PM 0.02/0.02/0.03 (2%/2%/1%)	PP 0.06/0.06/0.19 (5%/5%/6%)	-	$\boxed{1.13/1.03/2.88}$
Balance	-	NN 0.99/1.34/14.64 (25%/30%/54%)	CM 1.13/1.11/4.52 (27%/25%/17%)	CP 1.98/1.98/7.93 (48%/45%/29%)	-	$\boxed{4.11/4.44}$/27.08
Quality	Downs 0.04/0.01/0.15 (<1%/<1%/<1%)	AQR 1.94/2.17/18.81 (18%/18%/33%)	CM 0.01/0.01/0.03 (<1%/<1%/<1%)	NP 8.69/9.49/36.10 $\boxed{(81\%/81\%/66\%)}$	-	10.67/11.71/55.09
Quality+	Downs 0.11/0.11/0.36 (2%/2%/2%)	AQR 1.77/1.89/9.53 (36%/36%/43%)	CM 0.01/0.01/0.03 (<1%/<1%/<1%)	NP 2.80/3.07/11.48 (58%/58%/52%)	PSR 0.17/0.18/0.52 (3%/3%/2%)	$\boxed{4.86/5.27}$/21.92

- F faster than B, B faster than Q, and Q+ faster than Q
- Larger buffers (few seconds) and parallelized executions could enable real-time

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



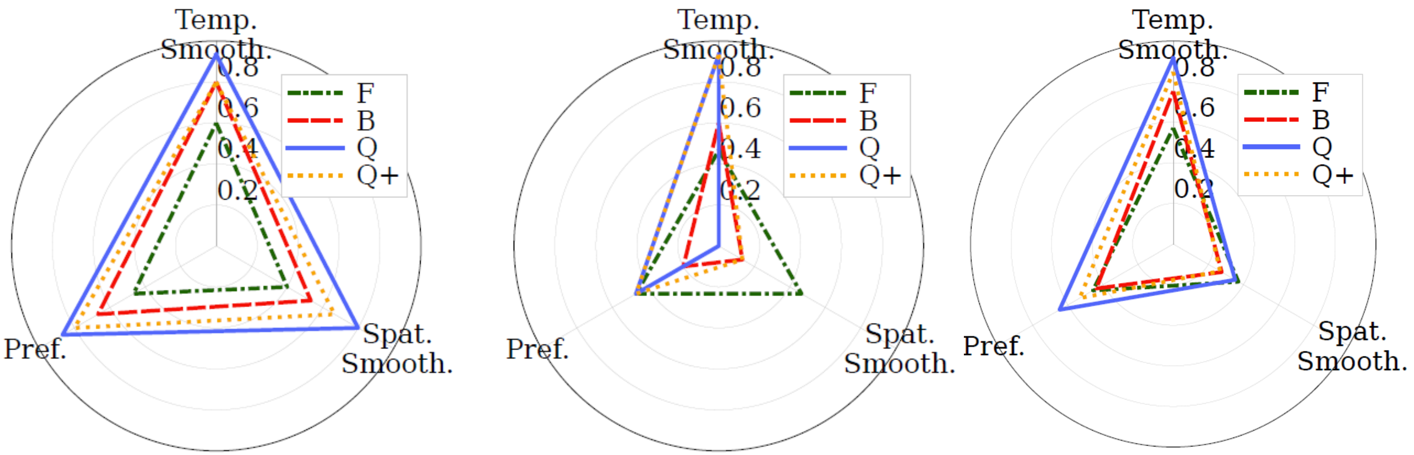
User Study Design

- Subjects: 11 male and 4 female (average 23.83 years old)
- Three subjective questions:
  1. Spatial smoothness: artifacts like cracks, irregular surfaces, and blurred edges
  2. Temporal smoothness: varying frame rate and stalls
  3. Preference: overall quality
- Test 1: Our pipeline (F, B, Q, Q+) against 3DFC under 5 consecutive frame drops
- Test 2: Our pipeline (Q) against two learning-based methods [1], [2] under a single frame drop

[1] Y. Zeng, Y. Qian, Q. Zhang, J. Hou, Y. Yuan, and Y. He, “IDEA-Net: Dynamic 3D Point Cloud Interpolation via Deep Embedding Alignment,” in Proc. of the CVPR’22, June 2022.
[2] A. Akhtar, Z. Li, G. Van der Auwera, and J. Chen, “Dynamic Point Cloud Interpolation,” in Proc. of the ICASSP’22, May 2022.

Winning Rates of Three Subjective Questions

Compared with 3DFC

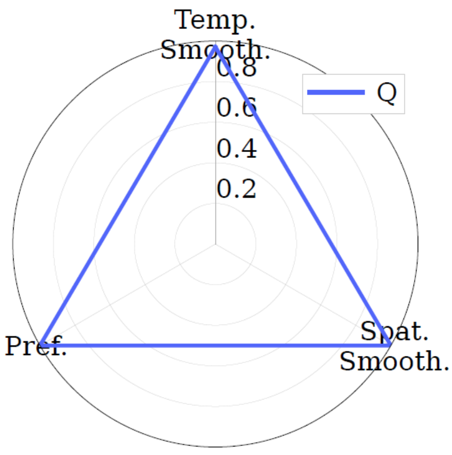


(a) Best-performing Red&blk

(b) Worst-performing Basketball

(c) Average across all sequences

Compared with Learning-based



(d) Average across all sequences

- Better temporal smoothness over 3DFC
- Lower spatial smoothness over 3DFC but understandable
- Q is more preferred, as user are more sensitive to spatial smoothness

Pipeline Q outperform learning-based method in all 3 questions

No Loss

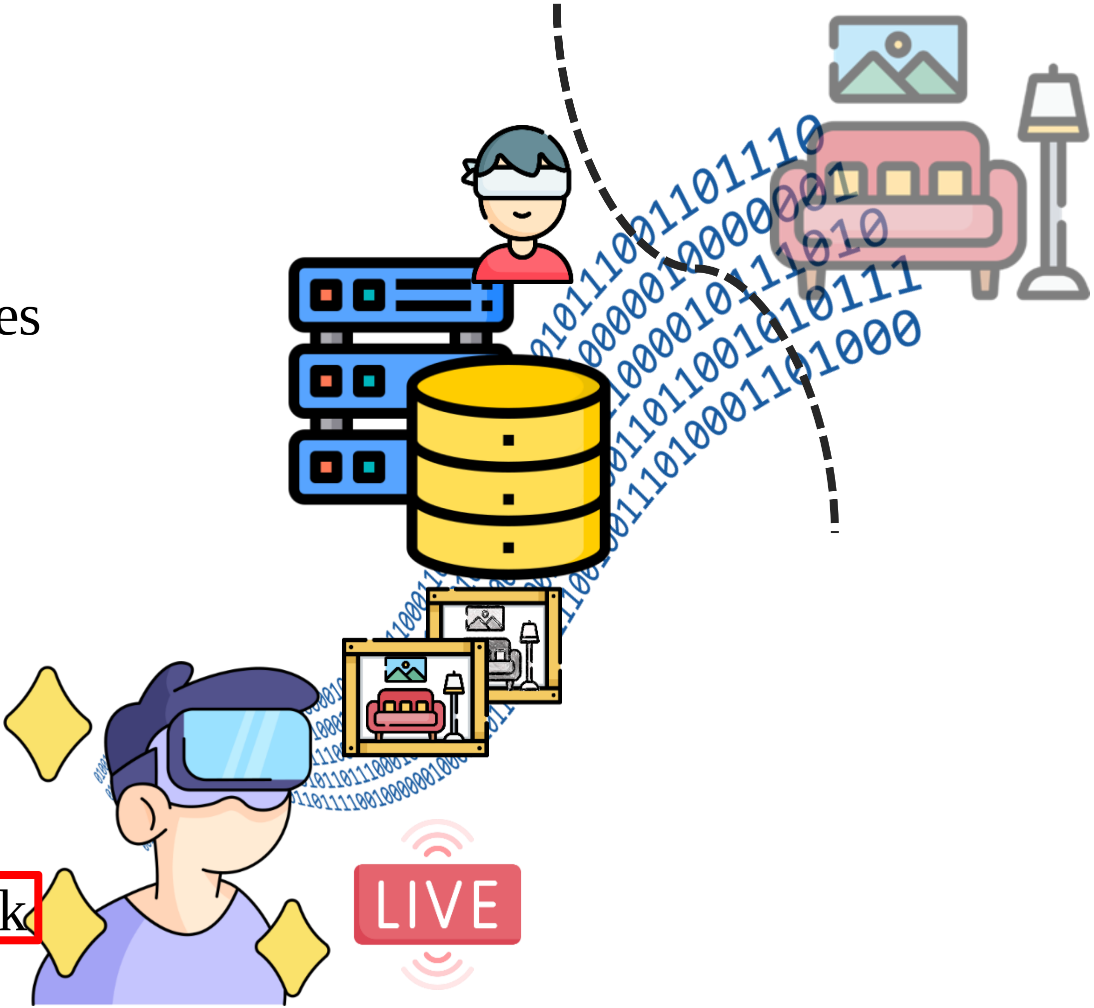
3DFC

Pipeline Q

No Loss | 2DFC | Zeng et al. | Akhtar et al. | Pipeline Q

Outline

- Inspiration
- Related Work & Challenges
- Design Framework
- Algorithms
  - Matching
  - Motion Estimation
  - Prediction
  - Pre- and Post-processing
- Objective Experiments
- User Study
- Conclusion & Future Work



Conclusion

- Proposed the very first pipeline framework for concealing distorted dynamic point clouds
  - https://github.com/Huang-I-Chun/error_concealment_pipeline
- Extensive objective experiments and subjective tests :
  - improved at most 5.32 dB in GPSNR, 2.22 dB PSNR, and 11.67 in VMAF compared to 3DFC
  - achieved better temporal smoothness than 3DFC
  - achieved a 100% winning rate on preference over learning-based algorithms
- Recommended pipelines under different scenarios:

Requirement / Motion Variance	Minor	Medium	Significant
High Quality	Q	Q	3DFC
Low Overhead	F	Q+	3DFC

Residual Challenges

- Users aware of the irregular surface of the error-concealed frames
  - Too large motion between frames
  - Error propagation of the proposed pipeline stages

Ill-shape basketball

Squeezed face

Bending gun

Future Research Direction

- Adaptive pipeline selection based on the motion among frames
- Handle more severe motion variance between frames
- Further improve spatial smoothness by better algorithms
- Code optimization for real-time

Publications and Collaberators

Publications:

- **I-Chun Huang**, Yuang Shi, Wei Tsang Ooi, Chun-Ying Huang, and Cheng-Hsin Hsu, "Composing Error Concealment Pipelines for Dynamic 3D Point Cloud Streaming," ACM Transactions on Multimedia Computing, Communications, and Applications, under review
- **I-Chun Huang**, Jiyan Wu, and Wei Tsang Ooi, "Embed Image into Video: A Robust and Blind Video Watermark Method in DT CWT Domain," IEEE Transactions on Circuits and Systems for Video Technology, under review
- Yuan-Chun Sun, **I-Chun Huang**, Yuang Shi, Wei Tsang Ooi, Chun-Ying Huang, and Cheng-Hsin Hsu, "A Dynamic 3D Point Cloud Dataset for Immersive Applications," in Proc. of the ACM Multimedia Systems Conference (MMSys '23), Vancouver, Canada, June 2023
- Tzu-Kuan Hung, **I-Chun Huang**, Sam Cox, Wei Tsang Ooi, and Cheng-Hsin Hsu, "Error Concealment of Dynamic 3D Point Cloud Streaming," in Proc. of the ACM International Conference on Multimedia (MM'22), Lisbon, Portugal, October 2022

- Tzu-Kuan Hung, *Phison*
- Yuan-Chun Sun, *National Tsing Hua University*
- Yuang Shi, *National University of Singapore*
- Sam Cox, *National University of Singapore*
- Wei Tsang Ooi, *National University of Singapore*
- Chun-Ying Huang, *National Yang Ming Chiao Tung University*

Thank you for listening

Q&A

I-Chun Huang (ichun0627@gmail.com)
Special thanks for the help of the committees and all lab mates.

Number of Continuous See-through Pixels: Mean (Max)

Seq. / Algo.	Que.	Loo.	Red.	Sol.	Lon.	Bas.	Dan.
Pipeline Q	1 (167)	15 (576)	53 (793)	4 (335)	38 (937)	161 (2123)	85 (1495)
Akhtar.	12 (1204)	119 (3708)	97 (1301)	43 (2317)	190 (2703)	548 (3870)	326 (3165)
Zeng.	18 (764)	55 (1118)	89 (1417)	56 (1010)	79 (1086)	1325 (29097)	286 (5321)