Fractional motion estimation for point cloud compression

SESSION 11: Friday 11:20am

Haoran Hong[†], Eduardo Pavez[†], Antonio Ortega[†], Ryosuke Watanabe*, Keisuke Nonaka*

[†]University of Southern California, Los Angeles, US
*KDDI Research, Inc., Japan

Data Compression Conference (DCC), 2022

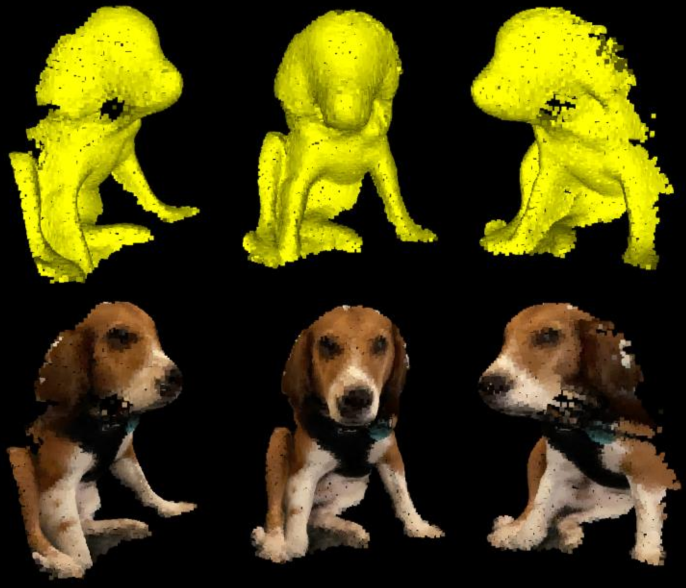
Outline

1. Introduction, motivation, related works
2. Fractional motion estimation for dynamic 3D point clouds
3. Compression experiments
4. Conclusion

Dynamic 3D point clouds

- Visual representations of people, objects, animals
  - Static (single frame)
  - Dynamic (multiple frames)
- Acquisition technology
  - Low cost 3d sensing (lidar, RGB + depth)
  - Large scale and high resolution
  - Real time
- Need for compression

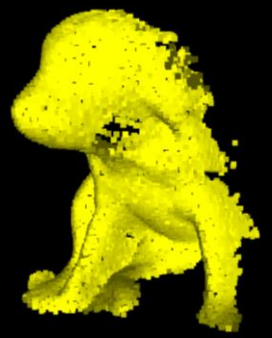
Multiple views of single frame



Dynamic point cloud sequences (same view) 8iVFBv2

- Schwarz, Sebastian, et al. "Emerging MPEG standards for point cloud compression." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2018)
- Graziosi, D., et al. "An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC)." *APSIPA Transactions on Signal and Information Processing* 9 (2020).

Dynamic 3D point clouds (D3DPC)



- List of points (voxels) in 3D space (geometry)

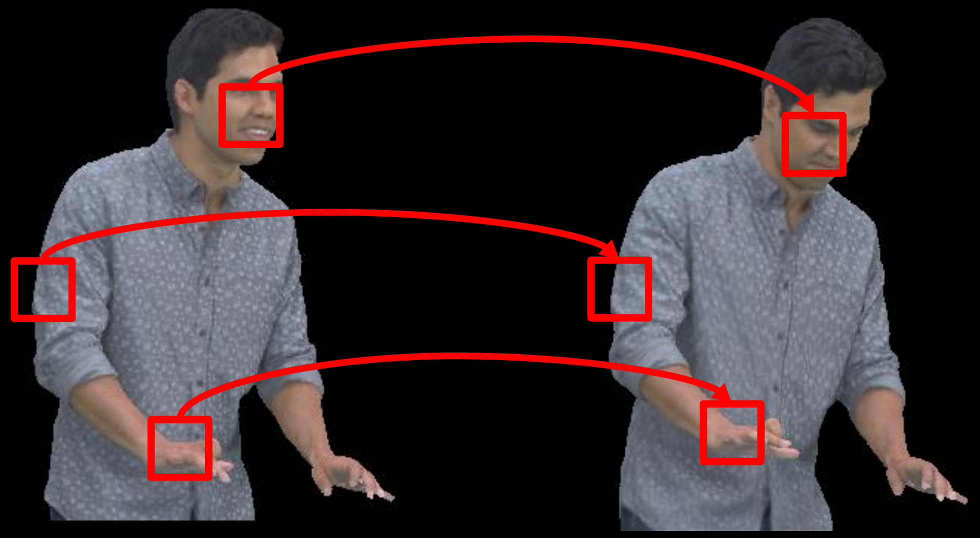
$$\boldsymbol{V} = [\boldsymbol{v}_i], \quad \text{with } \boldsymbol{v}_i = (x_i, y_i, z_i) \in \mathbb{N}^3$$

- Points have attributes (color, transparency, material properties)

$$\boldsymbol{A} = [a_i]$$

- Dynamic 3D point cloud: sequence of 3D point clouds

Inter frame (temporal) predictive coding



$t-1$

Decoded color $\hat{C}_{t-1}$

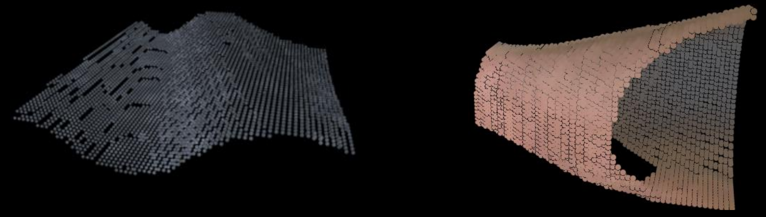
t

Current color C_t

- Remove temporal redundancy using previous frame and code the residual

$$residual_t = C_t - Predic(\hat{C}_{t-1})$$

- Challenge: Integer point coordinates (integer motion)
- Challenge: Irregular point distribution within each frame



- Challenge: Voxels are inconsistent across frames, i.e., both the number of voxels and their distribution in space are different from frame to frame.

Related work

- Voxel-based ME[1]
  - Motion vector is estimated for each voxel using graph smoothness.
- Patch-based ME[2]
  - Predicted frames are partitioned into patches through by K-means clustering.
  - MV is estimated by iterative closest point (ICP) for each patch.
- Block-based ME[3][4]
  - Predicted frames are partitioned into blocks.
  - MV can be estimated by ICP or a hybrid matching criterion.

All these are methods use Integer Resolution Motion Vectors

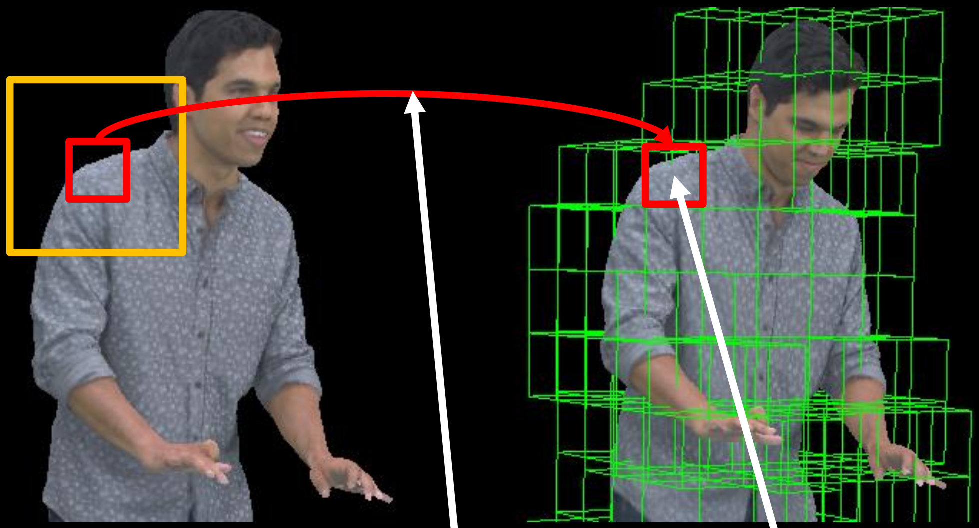
[1]. Dorina Thanou, Philip A. Chou, and Pascal Frossard, “Graph-based compression of dynamic 3D point cloud sequences,” IEEE Transactions on Image Processing, vol. 25, no. 4, pp. 1765–1778, 2016.
[2]. Yiqun Xu, Wei Hu, Shanshe Wang, Xinfeng Zhang, Shiqi Wang, Siwei Ma, Zongming Guo, and Wen Gao, “Predictive generalized graph Fourier transform for attribute compression of dynamic point clouds,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 5, pp. 1968–1982, 2021.
[3]. Rufael Mekuria, Kees Blom, and Pablo Cesar, “Design, implementation, and evaluation of a point cloud codec for tele-immersive video,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 828–842, 2017.
[4]. Ricardo L. de Queiroz and Philip A. Chou, “Motion-compensated compression of dynamic voxelized point clouds,” IEEE Transactions on Image Processing, vol. 26, no.8, pp. 3886–3895, 2017.

Fractional resolution motion for point clouds

- **Motivation:**
  - Motion estimation with sub-pixel accuracy is an essential tool for modern video coding.
  - All existing works for point cloud compression are based on input integer voxel motion.
  - Fractional resolution has potential for point cloud compression.
- **Challenges:**
  - Spatial irregularity: regions have changing geometry with varying number of points
  - Temporal irregularity: number of points and their distribution, geometry, and color change over time.

Contribution summary

- A: Fractional voxel interpolation
- B: Fractional-voxel motion estimation
- C: Motion-compensated prediction/compression

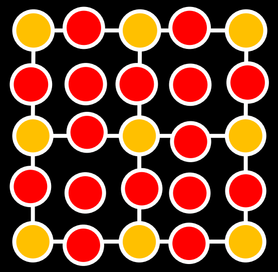


A: Fractional voxel interpolation

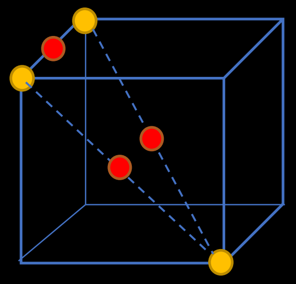
- **More accurate motion-compensated prediction**
  - Real motion, and color are continuous in time/space.
  - Fractional resolution provides better signal approximation
- **Noise removal**
  - Distortion due to lossy coding of reference frame
  - Camera noise, lighting change in the capture environment, object movements, etc.
- **Challenge**
  - Unlike images, 3D voxel grid is not fully occupied
  - Add fractional voxels consistent with object surfaces

● Integer pixel/voxel grid

● Fractional pixel/voxel grid



image



Point cloud

A: Fractional voxel interpolation

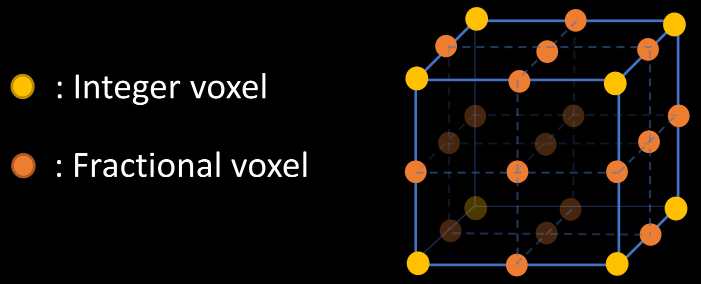
- The fractional voxel v_i is created at the mid point between two neighboring integer voxels, v_j and v_k.

$$v_i = \frac{1}{2}(v_j + v_k) \text{ for each pair of integer voxels with } dist(v_j, v_k) \leq \rho$$

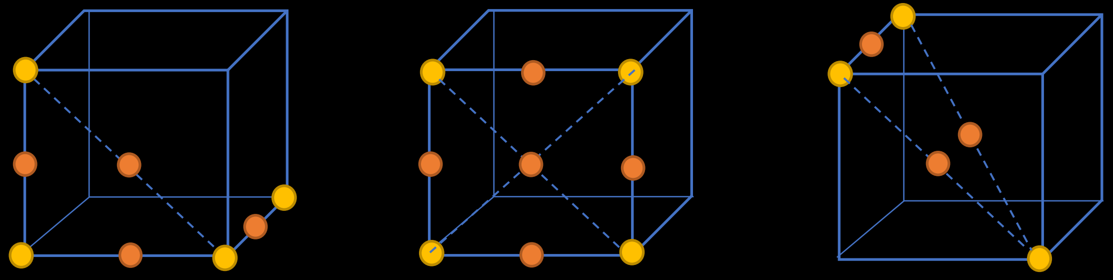
- Color interpolation by averaging

$$C(v_i) = \frac{1}{2}(C(v_j) + C(v_k))$$

- Repeated fractional voxels at the same position are merged by averaging.

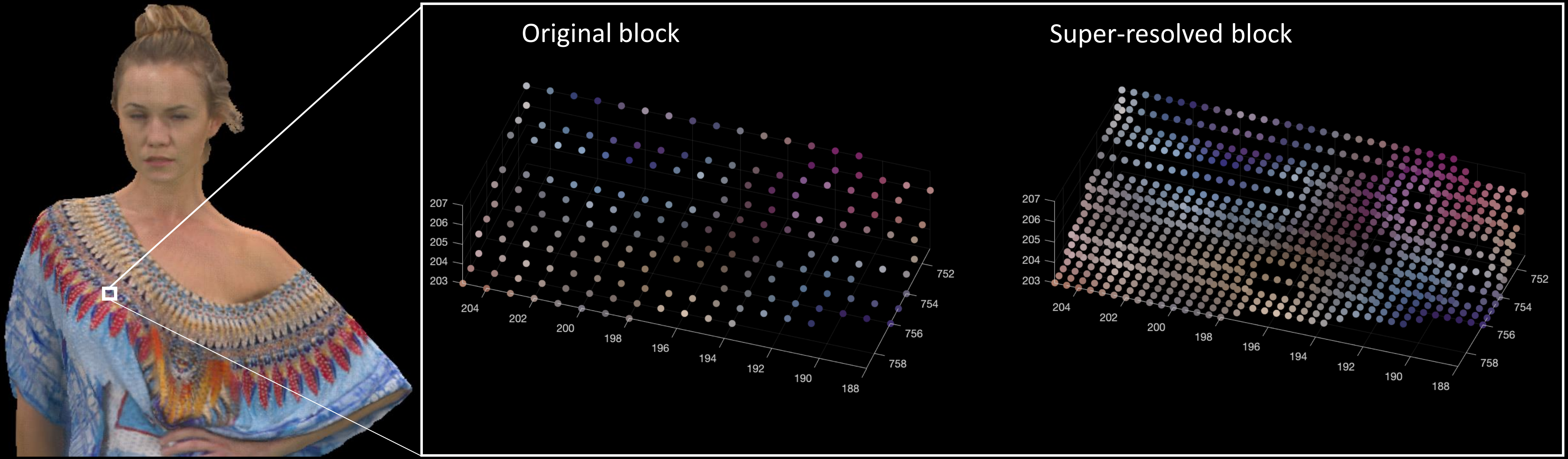


Examples:



A: Fractional voxel interpolation

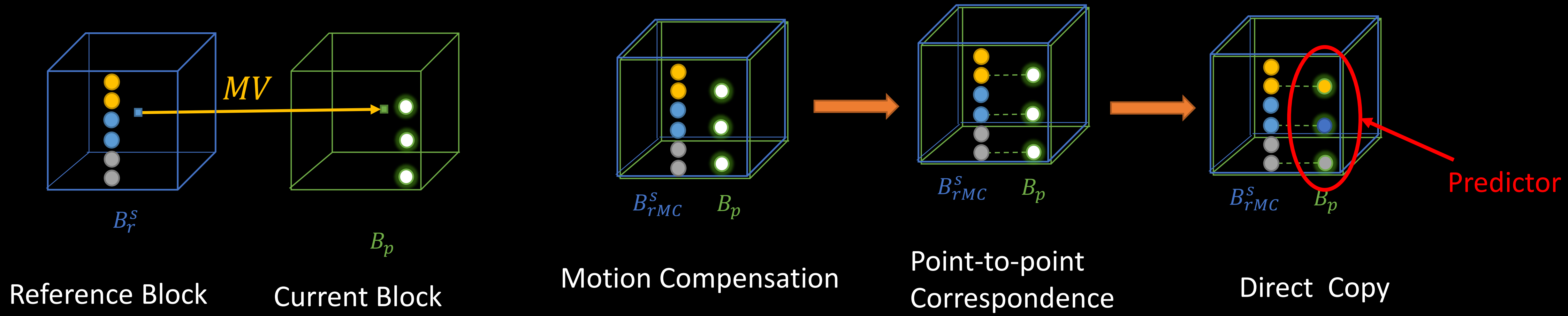
- Geometric and color continuity among voxels are significantly increased underlying surfaces.
- The interpolated voxel locations that are consistent with object surfaces, even though those surfaces are not explicitly known.



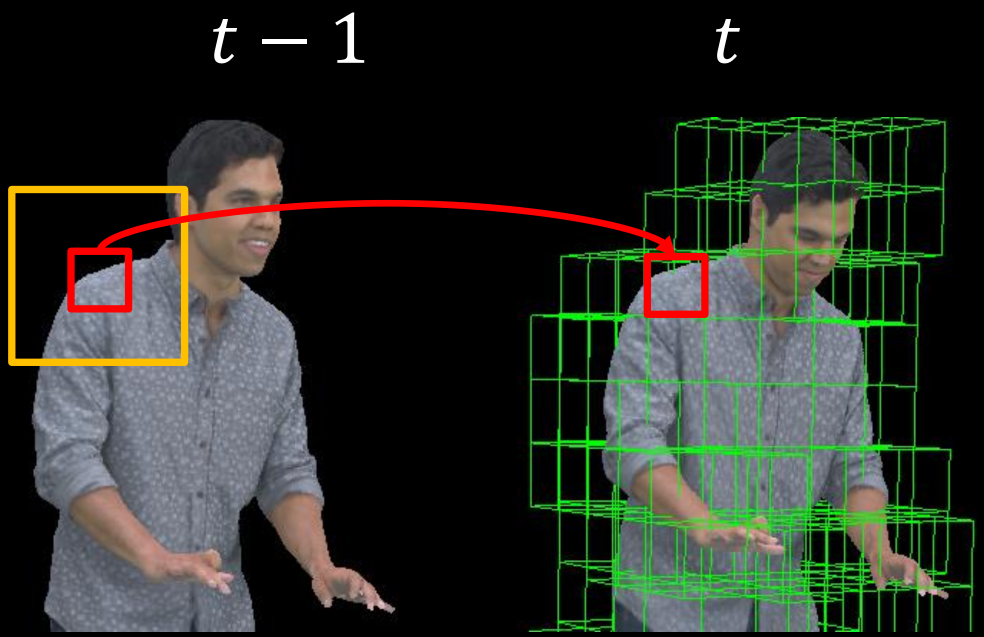
Longdress sequence from 8i database

C: Motion-compensated prediction

- A given motion vector MV induces a reference block.
- Reference block is motion compensated
- For each point in current block, find nearest neighbor (using distances between coordinates).
- Form predictor by copying color attribute from nearest neighbor



B: Fractional motion estimation



- Initial motion using database of motion vectors (DM) that optimizes hybrid geometry/color distance within region

$$\delta_g + \alpha\delta_c$$

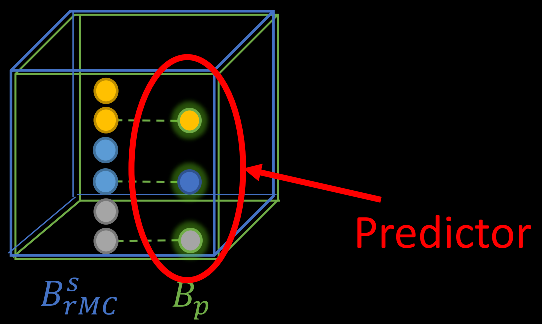
- Super resolve reference region
- Integer motion $MV = DM + \Delta$, with $\Delta \in \{-1,0,1\}^3$

$$MV_i = DM + \arg\min_{\Delta} MotCompColorError(DM + \Delta)$$

- Fractional resolution motion $MV^{opt} = MV_i + \Delta$

$$MV^{opt} = MV_i + \arg\min_{\Delta} MotCompColorError(MV_i + \Delta)$$

Optimizes fractional displacement $\Delta \in \left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}^3$



A 3D Motion Vector Database for Dynamic Point Clouds. André L. Souto, Ricardo L. de Queiroz, Camilo Dorea. 9th Aug 2020.

Overheads

For integer-voxel motion vector (15 bits/block)

Since the searching range of database motion is [-15,15] in each axis, we need to use 5 bits to represent each x,y,z components of MV_i .

A MV_i representation:

5 bits 5 bits 5 bits

$MV_i 1$: 00011111 00001100 00011101

8 bits 8 bits 8 bits

Bitstream: $[MV_i 1, MV_i 2, MV_i 3, \ldots]$

For fractional-voxel motion vector (5 bits/block)

Given a MV_i , there are 27 possible positions in total that a MV_f needs to signal. We need 5 bits to signal a FvMV for each block.

A MV_f representation:

5 bits

$MV_f 1$: 00010001

8 bits

Bitstream: $[MV_f 1, MV_f 2, MV_f 3, \ldots]$

e.g., 00010001 00001100 00011011 ……

$MV_f 1$ $MV_f 2$ $MV_f 3$

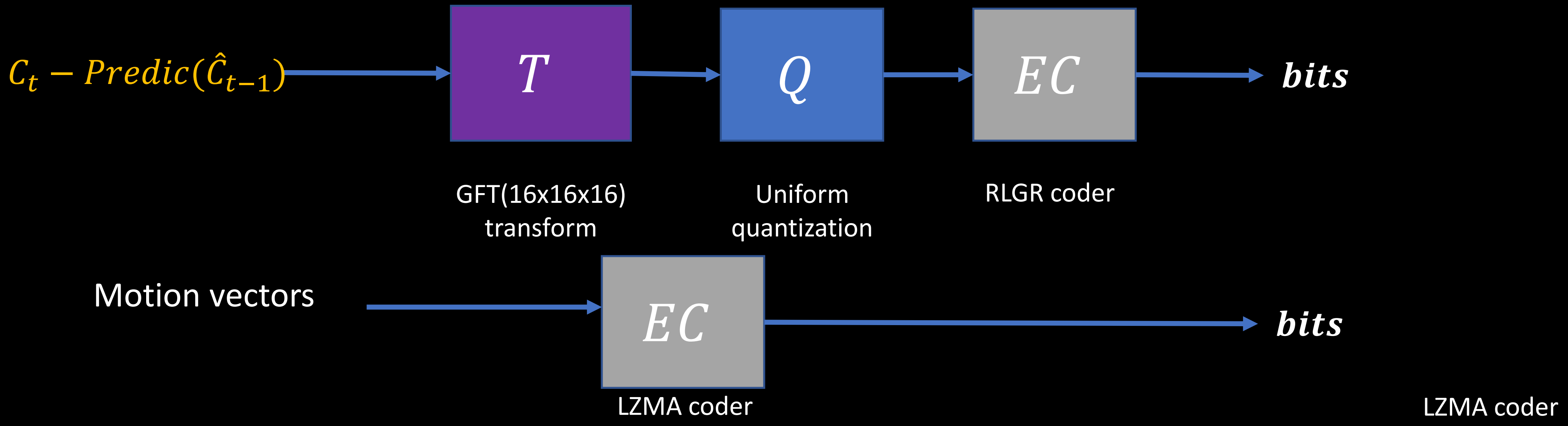
Overheads are entropy coded by Lempel–Ziv–Markov chain algorithm.

I. Pavlov, “7z format,” https://www.7-zip.org/7z.html.

Database

- 8iVFBv2 dynamic point cloud dataset
  - 4 sequences, 300 frames/seq., ~1M points/frame
  - Attributes: color in YUV domain
- A 3D motion vector database for dynamic point clouds
  - The criterion of MV: $\delta = \delta_g + 0.35 * \delta_c$
  - MV for Block size: 16x16x16

loot redandblack longdress soldier



d'Eon et.al., "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset," ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Geneva, 2017
A 3D Motion Vector Database for Dynamic Point Clouds. André L. Souto, Ricardo L. de Queiroz, Camilo Dorea. 9th Aug 2020.
H. S. Malvar, "Adaptive run-length/Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics," Data Compression Conference (DCC'06), 2006

Experimental settings

GOP = 32 frames

1st	2nd	16th	32th

DM	RA-GFT	Database integer motion + GFT16
DM+RF	RA-GFT	Database integer motion + local refinement + GFT16
DM+RF+SR	RA-GFT	Database integer motion + local refinement + super-resolution + GFT16
Proposed FvME	RA-GFT	Database motion + local refinement + super-resolution + fractional motion + GFT16
RA-GFT	RA-GFT16	RA-GFT16
RAHT	RAHT	RAHT

Integer resolution inter coding

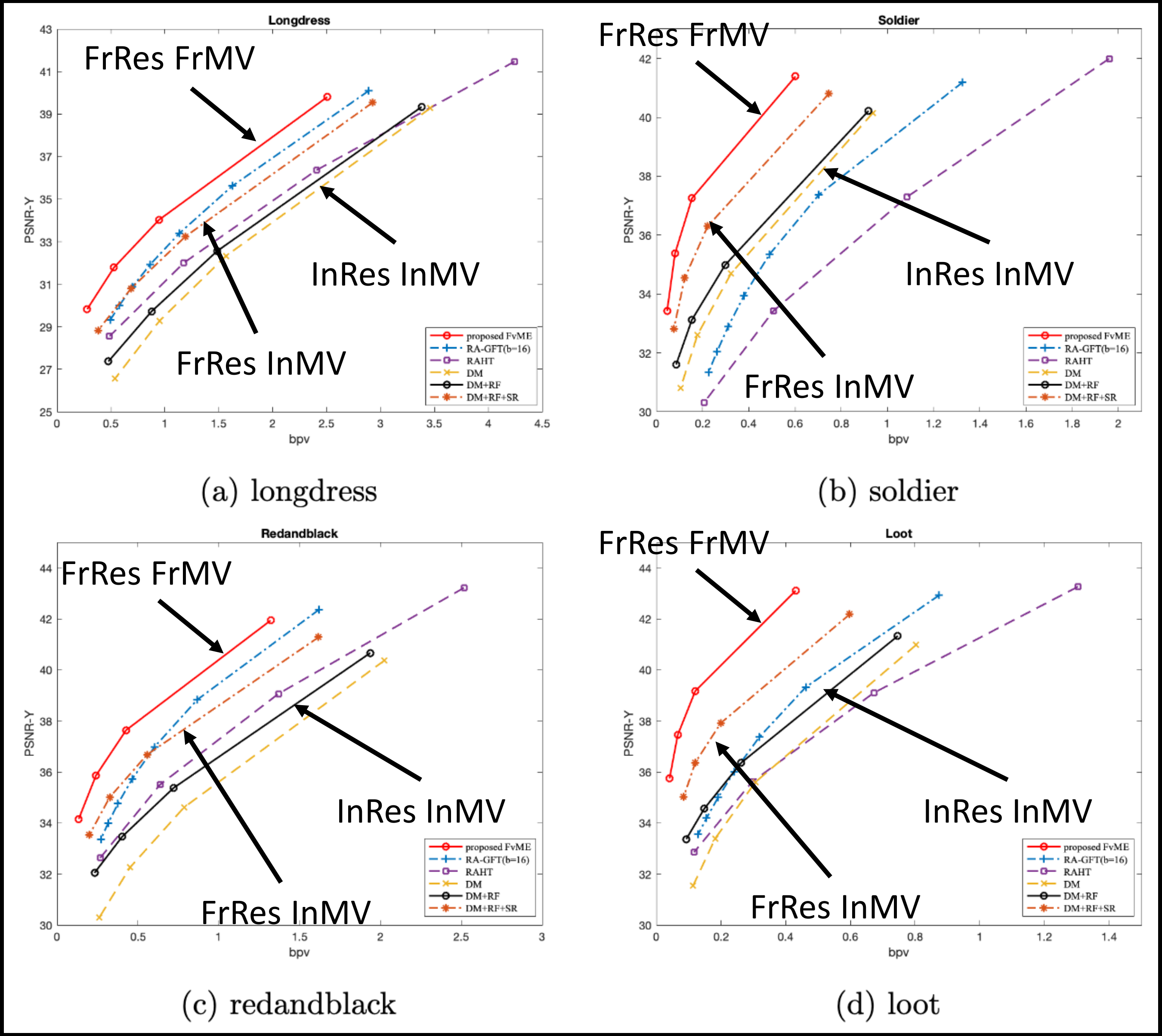
Transform coding (no prediction)

E. Pavez, B. Girault, A. Ortega and P. A. Chou, "Region Adaptive Graph Fourier Transform for 3D Point Clouds," 2020 ICIP.
De Queiroz, Ricardo L., and Philip A. Chou. "Compression of 3D point clouds using a region-adaptive hierarchical transform." *IEEE TIP 2016*
Zhang, Cha, Dinei Florencio, and Charles Loop. "Point cloud attribute compression with graph transform." *2014 ICIP*

Performance analysis

For “DM+RF+SR”
- Achieve 41% average bitrate reduction over IvME (DM+RF).
- Achieve 48% and 21% bitrate reduction on average over RAHT and RA-GFT respectively.
- Outperform IvME (DM+RF) and RAHT in all sequences

For “Proposed FvME”
- Achieve 57% average bitrate reduction over **IvME (DM+RF)**.
- Achieve 61% and 43% bitrate reduction on average over **RAHT** and **RA-GFT** respectively.
- Outperform **IvME (DM+RF)**, **RAHT** and **RAGFT** in all sequences



(a) longdress (b) soldier (c) redandblack (d) loot

Conclusion

- Proposed motion compensated compression of point cloud attributes that exploits fractional voxels (super resolution), and fractional motion.
- Consistently outperform alternative inter and transform coding schemes

Thank you