

MOTION ESTIMATION AND FILTERED PREDICTION FOR DYNAMIC POINT CLOUD ATTRIBUTE COMPRESSION

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

^{*}University of Southern California, Los Angeles CA. 90089 USA

[†]KDDI Research, Inc., Japan

ABSTRACT

In point cloud compression, exploiting temporal redundancy for inter predictive coding is challenging because of the irregular geometry. This paper proposes an efficient block-based inter-coding scheme for color attribute compression. The scheme includes integer-precision motion estimation and an adaptive graph based in-loop filtering scheme for improved attribute prediction. The proposed block-based motion estimation scheme consists of an initial motion search that exploits geometric and color attributes, followed by a motion refinement that only minimizes color prediction error. To further improve color prediction, we propose a vertex-domain low-pass graph filtering scheme that can adaptively remove noise from predictors computed from motion estimation with different accuracy. Our experiments demonstrate significant coding gain over state-of-the-art coding methods.

Index Terms— inter prediction, geometry-based point cloud coding, motion estimation, low-pass filter, graph filter.

1. INTRODUCTION

In current video coding systems, motion estimation (ME) and motion compensated prediction (MCP) are two essential techniques to exploit temporal redundancy between frames, so that inter-predictive coding techniques can be applied. In video-based point cloud compression (V-PCC) [1] a direct extension of these motion-based techniques to dynamic point cloud compression (PCC) is straightforward. However, for geometry-based PCC systems, where a point cloud (PC) is encoded directly in 3D space, ME becomes challenging. To see why, note that within a block in a video frame *all* pixel positions contain color information. Thus, both the reference and predictor blocks have the same number of pixels. In contrast, since PCs represent the surface of objects within a volume. Each 3D block in general has *a different geometry and a different number of occupied voxels*. Due to this irregular spatial structure, ME for geometry-based PC coding has to rely on PC registration (PCR) techniques [2], which match point sets of different sizes and geometries.

While PCR techniques have been widely studied and play a critical role in many computer vision applications, they need to be modified for ME and MCP, where the goal is to achieve better temporal prediction of PC attributes such as color. ME can be viewed as the estimation of a 3D transformation between two PCs, which seeks to align each element in a set of non-overlapping units in the predicted PC (e.g., voxels, blocks or clusters) to a corresponding unit in the reference PC. As an example, each block in the predicted PC is

matched to a corresponding block in the reference and the transformation that best aligns the blocks is the estimated motion. Existing geometry-based ME techniques are based on iterative closest point (ICP) [3, 4, 5], graph matching [6] and block matching [7, 8, 9].

ICP-based methods are the most commonly used due to their good performance and efficiency. They work particularly well when the two PCs to be matched are nearly aligned to begin with, which is usually the case for consecutive frames in dynamic PCs captured at high frame rates (e.g., 30 frames per second). In contrast, other PCR methods for ME are more complex and cannot be easily scaled to large PCs. Thus, our proposed ME is based on ICP.

MCP for geometry-based inter predictive coding can be accomplished by simply replacing the voxels in a predicted block with voxels in its best matching reference block [3, 7], with mode decision determining whether intra coding should be used instead if a sufficiently good approximation is not given by the copied block. As an alternative, in [4, 10] the color of each voxel in the current frame is predicted by the color of its corresponding voxel in the reference frame and a residual signal is sent. Our proposed method uses blocks instead of clusters as the basic unit for ME (as [3, 7]) but generates residues (as [6, 11, 4]). In addition, we propose a vertex-domain low pass graph filter that leads to significant improvements in prediction.

Our work addresses several limitations in recent work. First, existing block-based ME schemes [3, 7, 8, 9, 5] mimic video coding approaches by matching blocks of *same size* in the predicted and reference frames. Instead, as in [4] we directly optimize the alignment of a block in the current frame to a *larger* bounding box in the reference frame. With this approach, the closest points matched in the reference frame are not restricted to be within a block of same size as the block in the current frame. Note that using blocks is more efficient than using clusters, as in [4], since no clustering operations are needed, and blocks can be efficiently obtained using the octree structure. Also, different from recent ME schemes [4, 3, 5] we use a modified color-ICP [12] without rotation transformation (to avoid sending rotation information as overhead), with a hybrid matching metric that combines geometry and color, which leads to better inter prediction performance.

Even though our proposed block-based ME is based on a criterion that combines color and geometry (using color-ICP [12]) there is no guarantee that it is optimal for predicting PC color attributes. Thus, after the initial block matching, we propose two novel techniques not considered in the geometric point cloud coding literature. First, we use a *refined local search* around the color-ICP based motion to generate a refined motion that minimizes the color prediction error. By minimizing residual energy we improve prediction, while preserving geometric consistency because we only search locally around the initial motion found through color-ICP. Second, for MCP, we propose a vertex-domain low-pass graph filter. Unlike pre-

This work was funded in part by KDDI Research, Inc. and by the National Science Foundation (NSF CNS-1956190).

vious approaches [4], the proposed adaptive filtering scheme has low computational complexity since it can be implemented with sparse matrix vector products. The proposed adaptive filtering scheme can eliminate high frequency components in predictors and therefore improve prediction when there is inaccurate motion. Our method can outperform transform coding baselines, [13] and [14], with average gains of $3.21dB$ and $4.95dB$, respectively, and outperform the state-of-the-art inter coding scheme [4] with average gains of $2.15dB$.

2. BLOCK-BASED MOTION ESTIMATION

2.1. Notation

A 3D point cloud P consists of a list of points in \mathbb{R}^3 with their coordinates, along with a list of point attributes, such as color. Each point/attribute can be denoted by $\mathbf{p} = (\mathbf{v}, \mathbf{c})$, where $\mathbf{v} = (x, y, z)$ are the Cartesian coordinates, and $\mathbf{c} = (Y, U, V)$ are their corresponding color attributes in YUV format. A dynamic PC is a sequence of PCs denoted by $\{P_1, \dots, P_T\}$. We also denote by \mathbf{V}_t and \mathbf{C}_t , the $N_t \times 3$ matrices containing the point coordinates and color attributes at time t , respectively.

2.2. Motion compensated compression of dynamic point clouds

MC prediction for inter coding of 3D PC color attributes has been proposed with various types of motion vectors [4, 10, 15]. There is a fundamental difference between MC prediction in video and 3D PCs. Because predicted and reference video blocks have the same number of pixels and geometry, there is a 1 to 1 mapping between reference and predicted pixels. In contrast, because 3D PC geometry is irregular, and blocks may have different number of points, such correspondences are not readily available. We next describe the process of motion compensated matching between the target and reference blocks in more detail.

Let $P_{t,i} \subset P_t$ be the i th block of a PC at time t , with points $\mathbf{V}_{t,i}$ and colors $\mathbf{C}_{t,i}$, and let $\mathbf{m}_{t,i}$ be its corresponding motion vector. To encode the color attributes $\mathbf{C}_{t,i}$ of the i th block, given the motion vector $\mathbf{m}_{t,i}$ and the decoded attributes $\tilde{\mathbf{C}}_{t-1}$ of P_{t-1} , we define a predictor by finding for each $\mathbf{p}_j = [\mathbf{v}_j, \mathbf{c}_j] \in P_{t,i}$ its motion-compensated nearest neighbor in the reference frame:

$$\tilde{\mathbf{p}}_j = [\tilde{\mathbf{v}}_j, \tilde{\mathbf{c}}_j] = \arg \min_{\mathbf{q}=[\mathbf{v}, \mathbf{c}] \in P_{t-1}} \|\mathbf{v}_j + \mathbf{m}_{t,i} - \mathbf{v}\|^2. \quad (1)$$

Note that given $\mathbf{m}_{t,i}$, which is transmitted as overhead, the decoder can use (1) to find the nearest neighbor in P_{t-1} of any point in $P_{t,i}$ (PC geometry for P_t is encoded without loss and available at the decoder). The predictor for $\mathbf{C}_{t,i}$ is the matrix $\tilde{\mathbf{C}}_{t,i} = [\tilde{\mathbf{c}}_j]$. The residual $\mathbf{C}_{t,i} - \tilde{\mathbf{C}}_{t,i}$ can then be transformed, quantized, and entropy coded. In Section 2.3 and Section 2.4 we describe our proposed ME algorithm to generate $\mathbf{m}_{t,i}$. Later in Section 3 we introduce an in-loop filter for enhancing the MC predictor $\tilde{\mathbf{C}}_{t,i}$.

2.3. Block based color-ICP for initial ME

The current frame P_t is partitioned into non-overlapping blocks (e.g., using the octree structure). The i th block is denoted by $P_{t,i} \subseteq P_t$, as in Section 2.2. ICP algorithms iterate two steps until convergence: 1) point matching, and 2) transformation. Because our goal is color compression, we consider the hybrid metric between points with undistorted original colors $\mathbf{p}_1 = [\mathbf{v}_1, \mathbf{c}_1]$, and $\mathbf{p}_2 = [\mathbf{v}_2, \mathbf{c}_2]$, given by

$$\delta(\mathbf{p}_1, \mathbf{p}_2) = \alpha \|\mathbf{v}_1 - \mathbf{v}_2\|^2 + (1 - \alpha) \|\mathbf{c}_1 - \mathbf{c}_2\|^2, \quad (2)$$

where $\alpha \in [0, 1]$, so that both geometry and color are used for matching. While in traditional ICP, transformations consist of both rotation and translations, we only allow translations in integer precision as part of our transformations, to reduce the overhead of transmitting motion vectors. This leads to the following iterative steps.

Matching: For each $\mathbf{p}_j \in P_{t,i}$, we find the best matching point in the reference frame by solving

$$\mathbf{p}'_j = [\mathbf{v}'_j, \mathbf{c}'_j] = \arg \min_{\mathbf{q} \in P_{t-1}} \delta(\mathbf{p}_j, \mathbf{q}), \quad (3)$$

where δ is defined in (2).

Translation: The optimal translation $\mathbf{t}_{t,i}$ is the one that minimizes the average geometric distance between the points in $P_{t,i}$ and their matches in the reference, thus

$$\mathbf{t}_{t,i} = \arg \min_{\mathbf{t}} \sum_{j: \mathbf{p}_j \in P_{t,i}} \|\mathbf{v}_j + \mathbf{t} - \mathbf{v}'_j\|^2, \quad (4)$$

where \mathbf{v}'_j is computed using (3). It can be shown that $\mathbf{t}_{t,i} = (1/|P_{t,i}|) \sum_{j=1}^{|P_{t,i}|} (\mathbf{v}'_j - \mathbf{v}_j)$ [16]. After the optimal translation is found with (4), for each point $\mathbf{p}_j = [\mathbf{v}_j, \mathbf{c}_j] \in P_{t,i}$, we shift its coordinates towards the reference, that is, $\mathbf{v}_j \leftarrow \mathbf{v}_j + \mathbf{t}_{t,i}$, and then the **matching** and **translation** steps are repeated until convergence. Note that at the end of the ℓ th iteration, the translation step outputs a translation vector which we store and denote by $\mathbf{t}_{t,i}^\ell$. The final color ICP motion for each block is computed by adding $\mathbf{t}_{t,i}^\ell$ over all iterations and rounding up to integer precision resulting in

$$\mathbf{t}_{t,i}^{ICP} = \text{round} \left(\sum_{\ell \geq 1} \mathbf{t}_{t,i}^\ell \right). \quad (5)$$

2.4. Locally refined motion search

Even if the color-ICP motion vector preserves original color and geometric consistency, it is not guaranteed to be optimal for predicting (and coding) color attributes, especially since the reconstructed reference is distorted by lossy coding. We propose to refine the motion vector based on distorted reconstructed colors to further minimize the prediction error of the MC predictor from Section 2.2. For each block $P_{t,i}$, we take $\mathbf{t}_{t,i}^{ICP}$ as the initial motion and apply exhaustive search within a small search range to locally refine it. The coordinates of candidate displacement vectors Δ are restricted to be integers in $[-\beta, \beta]$, resulting in $(2\beta + 1)^3$ possible MVs. For each point $\mathbf{p}_i \in P_{t,i}$ in the current block and candidate motion vector $\mathbf{t}_{t,i}^{ICP} + \Delta$ we compute the MC predictor using (1) resulting in $\tilde{\mathbf{c}}_j(\mathbf{t}_{t,i}^{ICP} + \Delta)$. Then the optimal displacement is the one that minimizes color distance for all voxels in the predicted block,

$$\Delta_{t,i} = \arg \min_{\Delta} \sum_{j: \mathbf{p}_j \in P_{t,i}} \|\mathbf{c}_j - \tilde{\mathbf{c}}_j(\mathbf{t}_{t,i}^{ICP} + \Delta)\|^2. \quad (6)$$

The optimal motion vector for block $P_{t,i}$ is

$$\mathbf{m}_{t,i} = \mathbf{t}_{t,i}^{ICP} + \Delta_{t,i}. \quad (7)$$

3. IN-LOOP GRAPH FILTERING

Following Section 2.2, for a given motion vector $\mathbf{m}_{t,i}$ for the i th block, we can obtain the MC predictor for the block attributes $\tilde{\mathbf{C}}_{t,i}$, which can be computed using (1). In video coding, it is well known that significant coding gains can be obtained by removing noise from

predictors with some form of in-loop filtering. Since PCs have an irregular spatial structure, we propose to use graph filters for this purpose (refer to [17] for an overview of graph filters for image processing). We now describe a novel in-loop graph filtering technique to refine color attribute prediction.

3.1. Graph based in-loop filtering

A graph G is defined as $G = (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. The a -th vertex is denoted $v_a \in V$ and the edge connecting v_a and v_b is denoted as $e_{ab} \in E$. The weight of an edge is denoted as $W_{a,b}$. We can represent the graph connectivity information using a $n \times n$ adjacency matrix \mathbf{W} , which has zero diagonal elements and $\mathbf{W}_{a,b} = W_{a,b}$ for off-diagonal terms. A degree matrix \mathbf{D} is a $n \times n$ diagonal matrix with $D_{a,a} = \sum_b W_{a,b}$. A combinatorial Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. A random walk Laplacian matrix is defined as $\mathbf{L}_{RW} = \mathbf{D}^{-1}\mathbf{L}$, which can be eigen-decomposed as $\mathbf{L}_{RW} = \mathbf{U}\Lambda\mathbf{U}^{-1}$ [18, 19].

For 3D PCs, we construct a graph with random walk Laplacian $\mathbf{L}_{RW}^{(t,i)}$ for each block $P_{t,i}$, where the edge weights $W_{a,b}$ are inversely proportional to the geometric distance $\|\mathbf{v}_a - \mathbf{v}_b\|^2$ (see experiments for more details), while the color signal $\mathbf{C}_{t,i}$, and its MC predictor $\tilde{\mathbf{C}}_{t,i}$ are the graph signals.

3.2. Proposed graph filter

Let \mathbf{x} and \mathbf{y} be the graph signals before and after filtering, respectively. We define a one-hop weighted average filter as

$$y(a) = \frac{1}{2D_{a,a}}(D_{a,a}x(a) + \sum_{b \in \mathcal{N}(a,1)} W_{a,b}x(b)), \quad (8)$$

which can be written in vector form as:

$$\mathbf{y} = \left(\mathbf{I} - \frac{1}{2}\mathbf{L}_{RW} \right) \mathbf{x} = h(\mathbf{L}_{RW})\mathbf{x}, \quad (9)$$

where $h(\lambda) = 1 - \frac{1}{2}\lambda$ and $h(\mathbf{L}_{RW}) = \mathbf{U}h(\Lambda)\mathbf{U}^{-1}$. Since (8) can be computed locally when the graph is sparse, this filter can be computed efficiently. Because $\lambda \in [0, 2]$, the filter $h(\lambda)$ takes values in $[0, 1]$ thus acting as a low-pass filter that preserves the DC component, i.e., $\mathbf{y} = \mathbf{1}$ whenever $\mathbf{x} = \mathbf{1}$. For the i th block in the PC at time t , we propose using the filtered predictor

$$\tilde{\mathbf{C}}_{t,i}^{(1)} = h(\mathbf{L}_{RW}^{(t,i)})\tilde{\mathbf{C}}_{t,i}. \quad (10)$$

3.3. Adaptive filtering for improved prediction

Since motion estimation accuracy, compression error and noise may vary from block to block, we consider a block-level adaptive filtering approach where the filter from (10) is applied k times, resulting in the refined predictor

$$\tilde{\mathbf{C}}_{t,i}^{(k)} = h^k(\mathbf{L}_{RW}^{(t,i)})\tilde{\mathbf{C}}_{t,i}. \quad (11)$$

As k increases the filter (11) has a sharper frequency response, as shown in Figure 1, thus providing additional flexibility for removing high frequency noise. For each block, we choose k to minimize the color error, that is,

$$k_{t,i}^* = \arg \min_{k \geq 0} \|\mathbf{C}_{t,i} - \tilde{\mathbf{C}}_{t,i}^{(k)}\|^2. \quad (12)$$

When $k_{t,i}^* = 0$, we recover the MC predictor from Section 2.2. The sequence of $k_{t,i}^*$ for all blocks is encoded and sent as overhead.

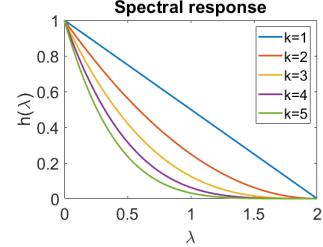


Fig. 1: Spectral response of the proposed filter.

4. EXPERIMENTS

4.1. Dataset and parameter settings

We evaluate the proposed integer motion estimation scheme for compression of color attributes of the 8iVFBv2 dataset [20], which consists of four sequences: `longdress`, `redandblack`, `loot` and `soldier`. We use the first 8 frames of each sequence. We implement a conventional inter-coding system, where previously decoded frames are used as reference. The group of frames (GOF) is set to be 8, wherein the first frame is coded in intra mode, and the rest are coded in inter mode. The prediction and transform units are both $16 \times 16 \times 16$. We use block graph Fourier transform (block-GFT) [21] to transform residues and choose $\alpha = 0.1^1$ in (2). To reduce the complexity of nearest point search in Section 2.3 we match each predicted block to a larger region (a block of size $61 \times 61 \times 61$) centered around the zero motion point in the reference frame. In Section 2.4 we set $\beta = 1$, so that 27 candidate motion vectors are evaluated during the refinement stage. For filtering, we construct a threshold graph with weight equal to the reciprocal of the geometric distance, and voxels within a block are connected to each other if their Euclidean distance is less than or equal to 1. The constructed graph is also used for block-GFT. In (12) we choose integer values $k \in [0, 5]$. For all schemes (proposed and baselines) the quantized coefficients are entropy coded by the adaptive run-length Golomb-Rice algorithm (RLGR) [22] while the overheads of $m_{t,i}$ s and $k_{t,i}^*$ s computed for blocks are separately entropy coded with the Lempel-Ziv-Markov chain algorithm of [23]. Other experimental settings are the same as in [15]. Rate is measured in bits per voxel (bpv), while decoded quality is measured in average peak signal-to-noise ratio over Y component (PSNR-Y),

$$PSNR_Y = -10 \log_{10} \left(\frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{Y}_t - \hat{\mathbf{Y}}_t\|_2^2}{255^2 N_t} \right), \quad bpv = \frac{\sum_{t=1}^T b_t}{\sum_{t=1}^T N_t},$$

where \mathbf{Y}_t and $\hat{\mathbf{Y}}_t$ represent original and reconstructed luma signals of the t -th frame, respectively, and T is the total number of frames. b_t is the total bitrate required to encode YUV components and overhead (when necessary) of t -th frame. N_t is the total number of occupied voxels in t -th frame.

4.2. Analysis of the proposed schemes

We evaluate our proposed ME and adaptive filtering method and show how each component contributes to the overall inter coding performance. We considered the following schemes: ME using geometric ICP and color-ICP of the hybrid criterion are denoted as `Geo` and `Geo+Col`, where added local refinement search is denoted

¹Performance was robust to a very wide range of α and very minor changes were observed in the results with $\alpha \in [0.1, 0.5]$.

as Geo+Col+RF. All the above schemes do not use any filtering scheme. Then adding a low pass filtering scheme with fixed $k = 1$ is denoted by Geo+Col+RF+LP, while the version with adaptive filtering is denoted Geo+Col+RF+LP(k). Except Geo, all the above schemes that integrate color in ME, include coded MVs in the overhead. For Geo+Col+RF+LP(k), the optimal k_i^* chosen for each block is coded as overhead. For Geo+Col+RF+LP(k), the optimal k_i^* chosen for each block is coded as overhead.

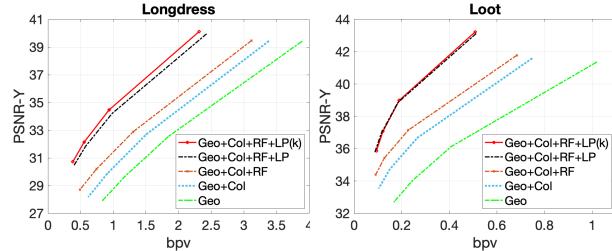


Fig. 2: Evaluation of proposed ME and in-loop filters.

From Figure 2, comparing Geo+Col and Geo+Col+RF to Geo we can see that both the color-ICP (Section 2.3) and local refined search (Section 2.4) lead to performance improvements. The largest boost in performance is obtained with the low pass filtered prediction of Section 3.2 (see Geo+Col+RF+LP), while adaptive filtering can provide additional gains for some sequences.

4.3. Effect of in-loop filtering

In this section we further evaluate the proposed in-loop filtering approaches when combined with other ME techniques. We considered zero motion vectors (ZMV) [10] and the database motion (DM) [24] for $16 \times 16 \times 16$ blocks, which are both combined with our proposed adaptive filtering, leading to ZMV+LP(k) and DM+LP(k). To illustrate how the proposed adaptive filtered prediction scheme can improve performance when combined with other methods, including methods that generate inaccurate motion, we exclude the mode decision function proposed in [10]. After prediction, the residues are transformed by block-GFT [21] with block size equal to 16. The intra coding baseline using block-GFT scheme denoted as block-GFT($b=16$) is also included in the experiment. To further demonstrate the performance of our ME scheme, we consider the above schemes without filtering as anchors denoted as ZMV and DM respectively. Our results are shown in Figure 3.

Combined with the proposed adaptive low pass filtering scheme, both ZMV and DM schemes are improved significantly. ZMV+LP(k) can even outperform intra coding method, block-GFT($b=16$), in some sequences without using mode decision to select between intra and inter coding. This is because our adaptive filtering scheme can aggressively eliminate AC components of predictors while preserving DC components. When the available motion is inaccurate, a filtered prediction with large k is more likely to be selected. In this case, MCP will mostly eliminate DC components of predicted signals while AC components remain unchanged. ZMV+LP(k) can be viewed as being reduced to transform scheme RA-GFT. From this perspective, our adaptive filtered prediction scheme plays a role similar to mode decision for ZMV. The combination of our efficient filtering scheme and ZMV, ZMV+LP(k), can be a promising scheme for real-time applications, since it avoids high computational cost ME and mode decision. Due to the local refined search, Geo+Col+RF can always outperform DM, which illustrates that the motion computed by our proposed ME scheme is more accurate than DM for color prediction.

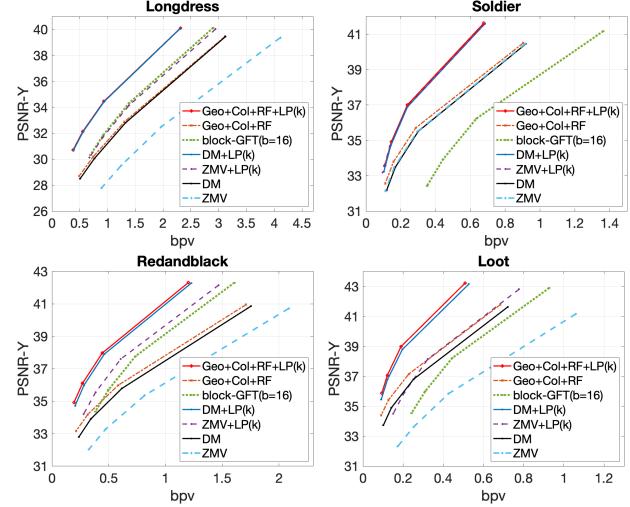


Fig. 3: Rate distortion curves of the combination of the proposed filtering scheme and the existing ME schemes.

4.4. Comparison with state-of-the-art methods

We compare our method against our implementation of the patch-based inter-coding scheme of [4] denoted as predictive NWGFT, which includes K-means clustering, geometric ICP-based ME, optimal inter-prediction, predictive transform and GFT based on the normal-weight graph (NWGFT). As discussed previously, for simplicity, none of our implementations, including that of [4], makes use of RD optimized mode decision. We leave this for future work. We use the same parameter settings in the paper [4] except the ϵ^2 in NWGFT, which we set to $\epsilon^2 = 3$ instead of $\epsilon^2 = 50^2$. Since the ME in [4] only depends on the geometry, no overhead is required to transmit the MVs. Additionally, to make our performance evaluation more comprehensive, we include two state-of-the-art non-predictive anchor solutions, namely, the region adaptive graph Fourier transform (RA-GFT) with block size 16 [13], and the region adaptive Haar transform (RAHT) [14].

We can see from Figure 4 that our inter coding scheme can consistently and significantly outperform transform schemes such as RA-GFT($b=16$) and RAHT, with average gains of $3.21dB$ and $4.95dB$, respectively, even without using a mode decision function or a rate-distortion optimization technique. We attribute our superior performance to an effective estimation of motion to exploit temporal redundancy. Our method can also outperform competitive predictive NWGFT with average gains of $2.15dB$ because of the more accurate motion by the integration of color attributes during ME and the proposed filtered prediction scheme.

5. CONCLUSION

In this paper, we propose an inter predictive approach for compression of dynamic point cloud attributes. The proposed encoder consists of an efficient motion estimation scheme and an in-loop low-pass filtering scheme for improved motion compensated prediction. Our motion estimation scheme uses color-ICP combined with a refined local search scheme to search for the motion that minimizes color error. We also propose an efficient vertex domain graph-based low-pass filter to remove noise from reconstructed reference signals.

²We observe that a smaller ϵ leads to significantly better performance.

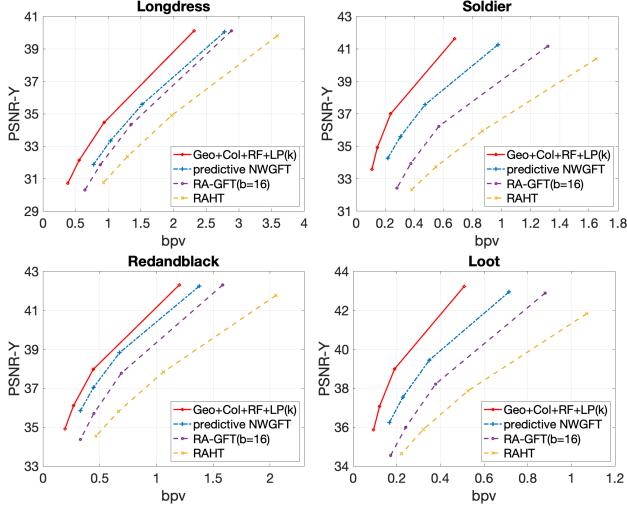


Fig. 4: Rate distortion curves of the proposed scheme and state of the art schemes.

The proposed filtered prediction scheme that combines the aforementioned techniques, significantly outperforms state of the art approaches for attribute compression in dynamic 3D point clouds.

6. REFERENCES

[1] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “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*, vol. 9, p. e13, 2020.

[2] X. Huang, G. Mei, J. Zhang, and R. Abbas, “A comprehensive survey on point cloud registration,” *ArXiv*, vol. abs/2103.02690, 2021.

[3] R. Mekuria, K. Blom, and P. 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] Y. Xu, W. Hu, S. Wang, X. Zhang, S. Wang, S. Ma, Z. Guo, and W. 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.

[5] C. Santos, M. Gonçalves, G. Corrêa, and M. Porto, “Block-based inter-frame prediction for dynamic point cloud compression,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3388–3392, 2021.

[6] D. Thanou, P. A. Chou, and P. Frossard, “Graph-based compression of dynamic 3D point cloud sequences,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.

[7] R. L. de Queiroz and P. A. Chou, “Motion-compensated compression of dynamic voxelized point clouds,” *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, 2017.

[8] C. Dorea and R. L. de Queiroz, “Block-based motion estimation speedup for dynamic voxelized point clouds,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2964–2968, 2018.

[9] C. Dorea, E. M. Hung, and R. L. de Queiroz, “Local texture and geometry descriptors for fast block-based motion estimation of dynamic voxelized point clouds,” in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3721–3725, 2019.

[10] A. L. Souto and R. L. de Queiroz, “On predictive RAHT for dynamic point cloud coding,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2701–2705, 2020.

[11] E. Pavez, P. A. Chou, R. L. De Queiroz, and A. Ortega, “Dynamic polygon clouds: representation and compression for vr/ar,” *APSIPA Transactions on Signal and Information Processing*, vol. 7, 2018.

[12] A. E. Johnson and S. Bing Kang, “Registration and integration of textured 3D data,” *Image and Vision Computing*, vol. 17, no. 2, pp. 135–147, 1999.

[13] E. Pavez, B. Girault, A. Ortega, and P. A. Chou, “Region adaptive graph Fourier transform for 3D point clouds,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2726–2730, 2020.

[14] R. L. de Queiroz and P. A. Chou, “Compression of 3D point clouds using a region-adaptive hierarchical transform,” *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016.

[15] H. Hong, E. Pavez, A. Ortega, R. Watanabe, and K. Nonaka, “Fractional motion estimation for point cloud compression,” in *2022 Data Compression Conference (DCC)*, pp. 369–378, 2022.

[16] P. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[17] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, “Graph spectral image processing,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.

[18] A. Ortega, *Introduction to graph signal processing*. Cambridge University Press, 2022.

[19] B. Girault, A. Ortega, and S. S. Narayanan, “Irregularity-aware graph fourier transforms,” *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5746–5761, 2018.

[20] T. M. Eugene d’Eon, Bob Harrison and P. A. Chou, “8i voxelized full bodies - a voxelized point cloud dataset,” *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, January 2017.

[21] C. Zhang, D. Florencio, and C. Loop, “Point cloud attribute compression with graph transform,” in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2066–2070, IEEE, 2014.

[22] H. Malvar, “Adaptive run-length/Golomb-Rice encoding of quantized generalized gaussian sources with unknown statistics,” in *Data Compression Conference (DCC’06)*, pp. 23–32, 2006.

[23] I. Pavlov, “7z format,” <https://www.7-zip.org/7z.html>.

[24] A. L. Souto, R. L. de Queiroz, and C. Dorea, “A 3D motion vector database for dynamic point clouds,” 2020.