LOCAL TEXTURE AND GEOMETRY DESCRIPTORS FOR FAST BLOCK-BASED MOTION ESTIMATION OF DYNAMIC VOXELIZED POINT CLOUDS

Camilo Dorea[1], *Edson M. Hung*[2] *and Ricardo L. de Queiroz*[1]

[1]Dept. of Computer Science, [2]Dept. of Electrical Engineering
University of Brasilia, DF, Brazil
Emails: {camilodorea, mintsu}@unb.br, queiroz@ieee.org

ABSTRACT

Motion estimation in dynamic point cloud analysis or compression is a computationally intensive procedure generally involving a large search space and often complex voxel matching functions. We present an extension and improvement on prior work to speed up block-based motion estimation between temporally adjacent point clouds. We introduce local, or block-based, texture descriptors as a complement to voxel geometry description. Descriptors are organized in an occupancy map which may be efficiently computed and stored. By consulting the map, a point cloud motion estimator may significantly reduce its search space while maintaining prediction distortion at similar quality levels. The proposed texture-based occupancy maps provide significant speedup, an average of 26.9% for the tested data set, with respect to prior work.

***Index Terms*—** Point clouds, volumetric media, 3D, motion estimation.

1. INTRODUCTION

Among the novel 3D representations for imaging systems, point clouds (PCs) constitute a geometrically simple yet versatile alternative, offering relative freedom to acquisition and rendering procedures. They are a set of points (x, y, z) in 3D space with associated data, such as color. In voxelized clouds, the points assume integer coordinate values on a regular 3D grid. Points within such a grid are called voxels and may be occupied or not. A temporal sequence of clouds, organized in frames, may be used to depict movement of dynamic objects or scenes.

The large amount of data generally involved in this 3D representation requests compression. This is, in fact, current subject of standardization efforts [1]. Previous studies have focused on compression of geometry [2, 3] and color attributes [4, 5] pertaining to static PCs while dynamic clouds have been addressed in [6–8]. In the latter case, motion estimation (ME) between PCs (a process henceforth referred to as PCME) plays a key role in the design of a successful predictive coder. To exploit temporal redundancy, [6] performs PCME by matching features derived from graph-based representations of successive clouds. Block-based partitions are matched to those of temporally adjacent clouds in [7, 8]. The former uses an iterative closest points (ICP) algorithm to locate a correspondence while the latter proposes optimization of a block-matching metric. PCME, nevertheless, may account for significant portions of processing time.

This work concerns the speeding up of block-based PCME. It is an extension and improvement on prior work [9]. Therein, the use of occupancy maps containing local statistics based on geometry, i.e., 3D moment-based shape descriptors, was used to efficiently restrict PCME search space. Here, occupancy maps based on local texture characteristics are introduced to offer further PCME speedups while maintaining provided quality. The proposed framework may be directly applied in codecs such as [7, 8] or in the development of PCME-dependent applications which use a block-based paradigm. Note that assessment of block-based PCME accuracy or its compression efficiency is beyond the scope of the current study. The novel texture-based occupancy maps are used in conjunction with those of [9] and contribute to significant performance gains while respecting constraints imposed for their efficient computation and storage.

Block-based PCME and prior work are briefly reviewed in Sec. 2. The proposed texture-based occupancy map is introduced in Sec. 3. Experimental results and conclusions are discussed in Secs. 4 and 5.

2. PREVIOUS WORK

A PC may be partitioned into blocks, in particular, non-overlapping cubes of dimension $L \times L \times L$. In block-based PCME, for each such cube of a current (or source) frame, a search space of size $S \times S \times S$ is defined in a previous (or target) frame around a co-located cube, as exemplified in Fig. 1. Among the set of L-dimensioned target cubes available within the search space, a best match with respect to the source cube is determined. In a full search scenario, S^3

This work was partially supported by FAPDF and by CNPq grants 301647/2018-6 and 428084/2016-8.

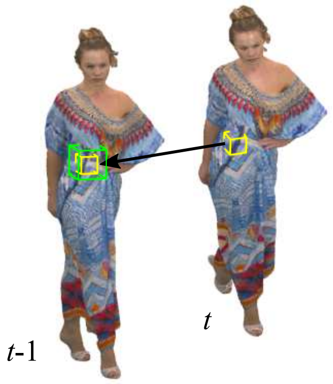

Fig. 1: Examples from the PC sequence *Longdress* [11] shown at different time instants and perspectives. Motion estimation establishes a correspondence between a source cube (in yellow) of current frame and a target cube within a search space (in green) of previous frame.

target cubes are considered in optimization. The motion vector indicating selected target cube position relative to source is used towards compensation, forming a prediction of the current frames's PC. This process presents some differences with respect to ME in conventional 2D video. The 3D grid is often sparsely populated, containing unoccupied voxels. Moreover, the number of occupied voxels may vary from frame to frame. Cube-matching criteria should thus account for these geometry differences. Complexity levels are also aggravated by the increase in search space dimension. Fast matching schemes commonly used in 2D video, e.g., [10], must operate with prior knowledge of geometry for adequate search space subsampling.

Prior work [9] introduced an occupancy map to speedup PCME. The map for a PC is efficiently pre-computed with one-pass update formulas (discussed at the end of this section). The descriptors therein are simple yet discriminative and, as scalar values, may be succinctly stored for multiple referencing. By consulting the map and comparing the occupancy value of the given source cube to that of a candidate target cube, the motion estimator may discard target cubes and avoid evaluation of, generally costly, cube-matching with candidates deemed incompatible. Speedup is measured as the number of discarded candidates with respect to full search. No assumptions are made on the cube-matching criterion; thus, the framework may be generically applied.

The occupancy maps of [9] describe local geometry, or spatial distribution of voxels within cubes, through statistical moments. A 3D moment [12] of order $p+q+r$ is defined as

$$M_{pqr} = \sum_{x=0}^{L-1}\sum_{y=0}^{L-1}\sum_{z=0}^{L-1} x^p y^q z^r v(x,y,z) \tag{1}$$

where $v(x,y,z)$ is the voxel density function, assuming value 1 for occupied and 0 for unoccupied voxels, within a domain of size L^3. The occupancy map $O(x,y,z)$ is formed by the 3D moments of all cube partitions of dimension L and origin (x,y,z) in a voxel space of dimensions N^3.

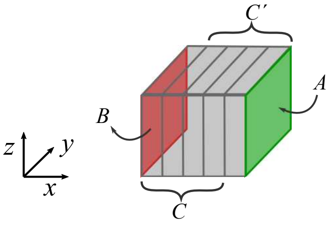

Fig. 2: A displacement of cube C (in x-direction) introduces voxels lying on the frontal plane (set A, in green) and eliminates those from the rear surface (B, in red) forming C'.

3D moments have the added benefit of being computable with one-pass update formulas [13], a particularly useful property when constructing occupancy maps by incrementally scanning through multiple overlapping partitions. Consider, without loss of generality, a cube sliding along the x-direction as illustrated in Fig. 2. Each increment in x introduces a set A and discards a set B of L^2 points. The statistical moments of next cube C' may be updated in terms of the of the previous cube C and those of A and B. For example, the zero-th ordered moment M_{000}, representing cube size or number of occupied voxels, is updated as

$$M_{000}^{C'} = M_{000}^{C} + M_{000}^{A} - M_{000}^{B}. \tag{2}$$

3. TEXTURE-BASED OCCUPANCY MAPS

Besides geometry, we expand the framework by considering texture as a local descriptor. Texture-based occupancy maps are used in conjunction with geometry and, likewise, are computed in the same one-pass update step described previously. We adopt the average luma component of the occupied voxels within a cube as a simple yet discriminative descriptor. Let $Y(x,y,z)$ represent the luma intensity at point (x,y,z) and $Y_S^C(x,y,z)$ the summation of intensities within a cube C of dimension L and origin (x,y,z) such that

$$Y_S^C(x,y,z) = \sum_{x=0}^{L-1}\sum_{y=0}^{L-1}\sum_{z=0}^{L-1} v(x,y,z)Y(x,y,z). \tag{3}$$

The update formula is analogous to (1), i.e.,

$$Y_S^{C'} = Y_S^C + Y_S^A - Y_S^B \tag{4}$$

and the texture-based descriptor for a given cube is defined as $\bar{Y}^C = Y_S^C / M_{000}^C$. The texture-based occupancy map is thus $O_T(x,y,z) = \bar{Y}^C$.

In this work we have selected cube size as our local geometry descriptor due to reported efficiency [9]. The geometry-based occupancy map is thus $O_G(x,y,z) = M_{000}^C$.

The motion estimator compares the local statistic of the current source cube $O_i^{src}(x,y,z)$ to those contained in the map and restricts the search space to target cubes presenting similar statistics, i.e., within a tolerance range defined by a threshold D_i such that $(1 - D_i)O_i^{src}(x, y, z) \leq O_i^{tgt}(x', y', z') \leq (1 + D_i)O_i^{src}(x, y, z), i \in \{G, T\}$.

4. EXPERIMENTAL RESULTS

Our tests were conducted on a variety of publicly available PC data sets, i.e., the upper body sequences *Andrew*, *David*, *Phil*, *Ricardo*, *Sara* [14] and full body human subjects *Longdress*, *Loot*, *Red and Black*, *Soldier* [11]. All have spatial resolution of $512 \times 512 \times 512$ voxels and are furnished with RGB color attributes. For all sets, the 10th frame is chosen as the source and the 9th as target.

Results of our geometry-and-texture (G&T) proposal were compared to those of full search PCME and those of prior work [9] employing geometry-only (GO) occupancy maps, as defined in Sec. 3, in terms of prediction distortion, speedup and motion vector error. In the current PCME implementation, a cube-matching criterion similar to that of [8] is adopted wherein nearest neighbor correspondences are determined between source and target cube voxels. The average Euclidean distance δ_G and average color distance δ_T, in Y-channel, between correspondences are combined in $\delta = \delta_G + 0.35\delta_T$. The final matching distance is symmetric and considers the maximum δ among source-to-target and target-to-source cubes. Cube dimension is chosen as $L = 8$ and search space dimension $S = 15$.

Distortion between predicted and source clouds is assessed through two peak signal-to-noise metrics, which consider both geometry and texture degradations, and via subjective evaluation. The first metric (PSNR-Y-G), similar to that of [8], determines for each voxel of the prediction a nearest neighbor within the source cloud and derives the average Euclidean distance δ_G. Differently from other metrics [15], the average Y-channel color distance δ_T is jointly considered as $\delta = \delta_G + 0.35\delta_T$ and reported in terms of a single measurement PSNR-Y-G $= 10\log(255^2/\delta)$. The second metric (PSNR-P), developed in [16, 17], adopts frontal orthographical projections, i.e., visible PC voxels are projected onto each of 6 projection planes parallel to the voxel surfaces, forming 2D images. From each of the 6 projected image pairs, originating from the predicted and source voxel sets, an overall mean square error (MSE) in the Y-channel color component is determined and PSNR-P $= 10\log(255^2/\text{MSE})$. In order to avoid rendering ambiguities, subjective quality evaluation is performed through visual inspection of frontal orthographical projections.

Speedups are measured as ratios between the full search space size and a reduced search space size, resulting from the elimination of target cube candidates whose occupancy values lie beyond defined tolerance ranges. Excluded from the full search space count is the trivial case with empty target cube. Tolerance thresholds, in percentages, for geometry D_G and texture D_T are selected from the first and second elements, respectively, of the ordered pairs $\{(\infty, \infty)$, $(100, 50)$, $(90, 50)$, $(80, 50)$, $(70, 50)$, $(60, 30)$, $(50, 30)$, $(40, 30)$, $(30, 30)$, $(20, 10)$, $(10, 10)$, $(5, 10)$, $(1, 10)\}$. Given speedup-distortion tradeoffs, we resort to a model similar to the BD metric [18], commonly used in assessing rate-distortion efficiency, to compute average distortion gains and average speedups between the proposal and a reference curve across a range of thresholds. Lastly, direct comparisons between motion vector fields (akin to optical flow), resulting from different PCME configurations, are established through average absolute error in flow endpoints as defined in [19].

The described experimental set up is used in assessing performance of our G&T proposal and GO of [9]. The curves of Fig. 3 and Fig. 4 for *Longdress* depict typical behavior also observed in other data sets. Search space reduction through occupancy map consultation provides significant speedups, initially guaranteeing distortions (PSNR-Y-G and PSNR-P) equivalent to those produced by full search PCME (speedup 1). With further speedup (e.g., 10) G&T outperforms GO in terms predicted distortion as well as direct comparison of motion vector errors with respect to full search, see Fig. 4. In spite of incurring some quality degradation when operating at speedup 10, these are deemed as acceptable, as supported by visual inspection of the images in Fig. 5. In this case, quality of predicted frames (c) and (d) is similar to that of full search (b). Nevertheless, difference images (e) and (f) indicate superior quality of G&T relative to GO.

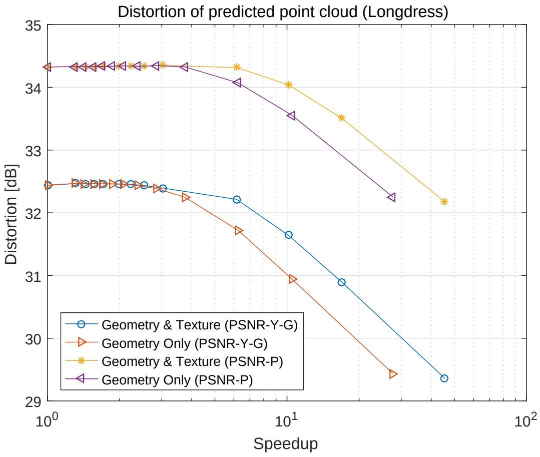

Fig. 3: PSNR-Y-G and PSNR-P distortions as a function of speedup using G&T and GO occupancy maps for *Longdress*.

Objective comparisons between distortion-speedup curves (e.g., Fig. 3) are established with a BD-like model [18] and summarized in Tables 1 and 2 for all sequences. The G&T proposal outperforms GO in most cases. In few exceptions, such as *David*, performance is similar with an overlap among curves. Distortion gains averaged across all sets are 0.47 dB and 0.18 dB, in PSNR-Y-G and PSNR-P, respectively, while speedup increases correspond to 26.9% and 10.2% at equiv-

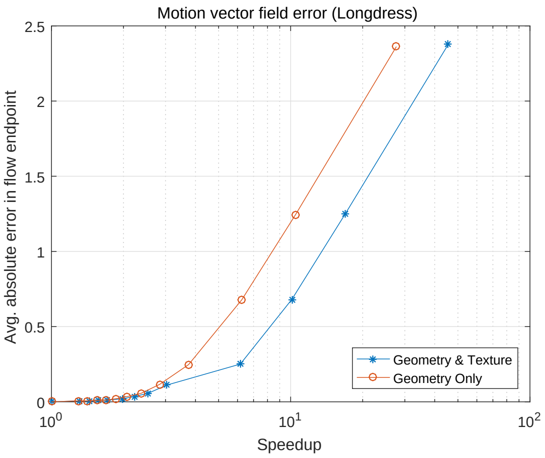

Fig. 4: Motion vector field errors [19], with respect to full search PCME, as a function of speedup using G&T and GO occupancy maps for *Longdress*.

alent prediction quality. More modest gains under PSNR-P reflect the lower sensitivity of this metric to quality degradation. Note that flatter PSNR-P curves may lead to numerical modeling instability reflected in some corresponding speedup increase measurements.

5. CONCLUSIONS

We have presented a work in progress which expands and improves upon prior study by considering the usage of texture descriptors, in addition to geometry, for the speedup of PCME. Texture characteristics of the search space are rapidly computed with a one-pass update formula and succinctly stored as scalar-values for multiple, subsequent referencing within an occupancy map. Results using the proposed G&T maps present significant speedup, up to 10 times over full search and 26.9% on average relative to GO, while maintaining equivalent quality. Future work includes the development of adaptive thresholding and analysis of distortion-speedup operating points for PCME optimization.

Table 1: Average PSNR-Y-G and PSNR-P gains of G&T proposal relative to GO for various data sets.

Data Set	Avg. PSNR-Y-G gain	Avg. PSNR-P gain
Andrew	0.42 dB	0.12 dB
David	-0.01 dB	-0.18 dB
Longdress	0.55 dB	0.43 dB
Loot	0.26 dB	0.16 dB
Phil	0.73 dB	-0.09 dB
Red and Black	0.26 dB	0.12 dB
Ricardo	0.20 dB	0.02 dB
Sarah	0.30 dB	-0.07 dB
Soldier	1.45 dB	1.10 dB

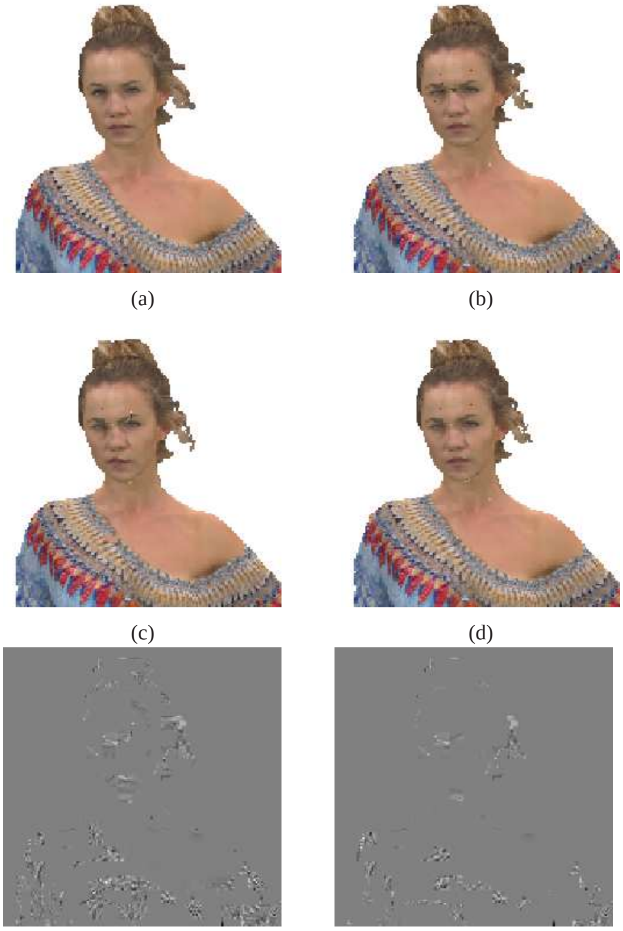
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 5: Detail crops from images of frontal orthographical projections of PCs from (a) source frame, predicted with (b) full PCME, (c) GO, (d) G&T occupancy maps (at speedup 10) and, respectively in (e) and (f), difference images of predicted (c) and (d) with respect to source (a) for *Longdress*, 10th frame.

Table 2: Average speedup increase at equivalent quality (PSNR-Y-G or PSNR-P) of G&T proposal relative to GO for various data sets.

Data Set	Avg. speedup increase (PSNR-Y-G)	Avg. speedup increase (PSNR-P)
Andrew	22.2 %	-8.9 %
David	1.2 %	50.5 %
Longdress	48.7 %	22.1 %
Loot	16.8 %	23.5 %
Phil	51.5 %	29.2 %
Red and Black	18.6 %	-26.6 %
Ricardo	12.6 %	6.8 %
Sarah	32.7 %	1.1 %
Soldier	37.8 %	-5.8 %

6. REFERENCES

[1] S. Schwarz, *et al.*, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Syst.*, vol. Pre-Print, 2018.

[2] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *IEEE Int. Conf. on Robotics and Automation*, May 2012.

[3] C. Loop, C. Zhang, and Z. Zhang, "Real-time high-resolution sparse voxelization with application to image-based modeling," in *Proc. High-Performance Graphics Conf.*, Jul. 2013.

[4] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. IEEE Int. Conf. on Image Process.*, Oct. 2014.

[5] R. L. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, Aug. 2016.

[6] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, Apr. 2016.

[7] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation and evaluation of a point cloud codec for tle-immersive video," *Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, Apr. 2017.

[8] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Trans. Image Process.*, vol. 26, no. 8, Aug. 2017.

[9] C. Dorea and R. L. de Queiroz, "Block-based motion estimation sppedup for dynamic voxelized point clouds," in *Proc. IEEE Int. Conf. on Image Process.*, Oct. 2018.

[10] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," *Proc. SPIE Visual Comm. and Image Process.*, vol. 4671, 2002.

[11] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i voxelized full bodies - a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, Geneva, Switzerland, Jan. 2017.

[12] F. A. Sadjadi and E. L. Hall, "Three-dimensional moment invariants," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, no. 2, Mar. 1980.

[13] P. Pébay, "Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments," Tech. Rep. SAND2008-6212, Sandia National Laboratories, 2008.

[14] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, Geneva, Switzerland, May 2016.

[15] R. Mekuria, Z. Li, C. Tulvan, and P. Chou, "Evaluation criteria for point cloud compression," in *ISO/IEC JTC1/SC29/WG11 MPEG2016 n16332*, Geneva, Switzerland, Feb. 2016.

[16] R. L. de Queiroz, E. Torlig, and T. A. Fonseca, "Objective metrics and subjective tests for quality evaluation of point clouds," in *ISO/IEC JTC1/SC29/WG1 input document M78030*, Rio de Janeiro, Brazil, Jan. 2018.

[17] E. Torlig, E. Alexiou, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Proc. of SPIE, Applic. of Digital Signal Process. XLI*, Aug. 2018.

[18] G. Bjøntegaard, "Improvements of the BD-PSNR model," in *doc VCEG-A111, ITU-T SG16/Q6*, Berlin, Germany, Jul. 2008.

[19] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. Journal of Computer Vision.*, vol. 92, no. 1, Mar. 2011.