3D Point Cloud Compression: A Survey

Chao Cao
Télécom SudParis
Institut Mines-Télécom
chao.cao@telecom-sudparis.eu

Marius Preda
Télécom SudParis
Institut Mines-Télécom
marius.preda@telecom-sudparis.eu

Titus Zaharia
Télécom SudParis
Institut Mines-Télécom
titus.zaharia@telecom-sudparis.eu

ABSTRACT

In recent years, 3D point clouds have enjoyed a great popularity for representing both static and dynamic 3D objects. When compared to 3D meshes, they offer the advantage of providing a simpler, denser and more close-to-reality representation. However, point clouds always carry a huge amount of data. For a typical example of a point cloud with 0.7 million points per 3D frame at 30 fps, the point cloud raw video needs a bandwidth around 500MB/s. Thus, efficient compression methods are mandatory for ensuring the storage/transmission of such data, which include both geometry and attribute information. In the last years, the issue of 3D point cloud compression (3D-PCC) has emerged as a new field of research. In addition, an ISO/MPEG standardization process on 3D-PCC is currently on-going. In this paper, a comprehensive overview of the 3D-PCC state-of-the-art methods is proposed. Different families of approaches are identified, described in details and summarized, including 1D traversal compression, 2D-oriented techniques, which take leverage of existing 2D image/video compression technologies and finally purely 3D approaches, based on a direct analysis of the 3D data.

CCS CONCEPTS

• **Computing methodologies → Point-based models**; • **Information systems → Data compression**; • **General and reference → Surveys and overviews**.

KEYWORDS

3D point cloud, compression, survey

ACM Reference Format:
Chao Cao, Marius Preda, and Titus Zaharia. 2019. 3D Point Cloud Compression: A Survey. In *Web3D '19: The 24th International Conference on 3D Web Technology (Web3D '19), July 26–28, 2019, Los Angeles, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3329714.3338130

1 INTRODUCTION

Today, 3D point clouds are offering highly realistic representations of a large variety of 3D objects and scenes. They are able to represent objects with various scales as small as a blood vessel or as large as a car, a landmark or even an entire city. With the rapid advancement of 3D point cloud acquisition technologies, such as LIDAR (LIght Detection And Ranging) sensors, high precision point cloud representations become affordable. Moreover, the recent boost in GPU power makes it possible to obtain an effective, real-time rendering and visualization of dense 3D point clouds.

Point clouds are today widely used in numerous application fields, including 3D scanning and modeling, environmental monitoring, agricultural and forestry, bio-medical imagery, CAD, autonomous driving and so on. Nevertheless, there are still a lot of efforts to be made to compress the point clouds in a sufficient ratio and make them suitable for real-time, portable applications such as autonomous navigation and Virtual /Augmented Reality (VR/AR).

In recent years, various families of 3D point cloud compression methods have been proposed. In this paper, we notably propose an overview of the existing techniques, with main principles, advantages and limitations. In contrast with 3D mesh representations (see [Peng et al. 2005], [Alliez and Gotsman 2005] and [Maglo et al. 2015] for representative surveys of 3D mesh compression), point clouds complete lack of topological/connectivity information is the main difficulty to overcome. To the very best of our knowledge, this is a first peer reviewed attempt to structure and review the research in the field.

The rest of the paper is organized as follows. Section 2 introduces the basic concepts involved in 3D point cloud representations. Then, Section 3 reviews the state-of-the-art of 3D point compression methods. Three main families of approaches are here identified with conclusive tables, including 1D traversal compression techniques, 2D projection-based methods and finally direct 3D decorrelation techniques. Section 4 presents ongoing work within the ISO/MPEG standardization where a point cloud compression standard [3DG 2015] is currently under development. Finally, section 5 concludes the paper and proposes a discussion on the emerging trends in the field.

2 PRELIMINARIES

By definition, a *3D point cloud* is a set of points $\{P_i\}_{i=1}^{n}$, embedded in the 3D space and carrying both geometry and attribute information (i.e., photometric properties). The term "cloud" reflects the unorganized nature of the set [Otepka et al. 2013]. Point clouds can be both static and dynamic. In the case of dynamic content, a different point cloud is considered at each frame, representing the continuous variation of a point cloud. Let us underline that the number of points can vary from a frame to another and that there is no point-to-point correspondence available between clouds in successive frames.

The geometry information refers to the point position in a given Cartesian coordinate system, expressed in (X, Y, Z) coordinates. The attribute information describes the visual appearance of each point and can take different forms, depending on the use cases

Web3D '19, July 26–28, 2019, Los Angeles, CA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6798-1/19/07…$15.00
https://doi.org/10.1145/3329714.3338130

considered. The most commonly used attributes (Figure 1) are color values (R, G, B) and normal vectors (n_x, n_y, n_z). In addition, reflectance information can be considered as well.

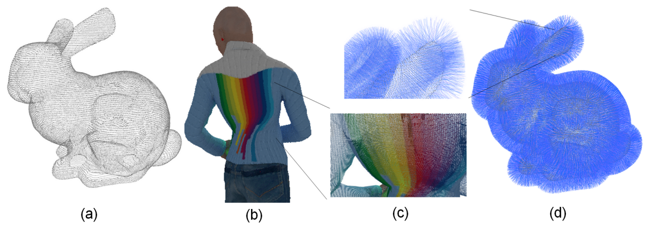
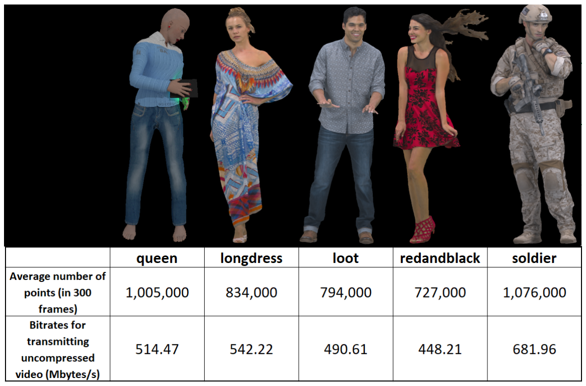


Figure 1: (a) Point cloud "Stanford Bunny" with geometry information, (b) *queen* dataset produced by Technicolor (https://www.technicolor.com/fr) [Creative Common Zero Universal 1.0 license ("CC0")] with the color information and (c) Zoomed detail for pictures in (b) and (d). (d) "Stanford Bunny" with normal information denoted with blue line. (http://graphics.stanford.edu/data/3Dscanrep/).

In the general case, the geometric coordinates are expressed in floating point values. However, for some specific use cases (e.g., real-time application, large scale dense point cloud compression), it is useful to consider an integer representation of the coordinates, which helps saving CPU computation time and optimizing memory usage. This comes to perform a quantization of the geometry with respect to a regular grid. It is observed that the integer representation adds additional constraints on the compression process, since solely integer-to-integer transforms can be used.

In order to achieve visually pleasant and realistic 3D representations, it is necessary to consider high density point clouds. Figure 2 illustrates some examples of video point clouds, so-called *longdress, loot, redandblack, soldier* from [8i Labs 2017] and *queen*, with some representative frames, average number of points over time and corresponding bitrates (in MBytes/s) of the raw, uncompressed data.



	queen	longdress	loot	redandblack	soldier
Average number of points (in 300 frames)	1,005,000	834,000	794,000	727,000	1,076,000
Bitrates for transmitting uncompressed video (Mbytes/s)	514.47	542.22	490.61	448.21	681.96

Figure 2: Example of dynamic point cloud and the requested bandwidth for uncompressed data.

The corresponding bitrates, which are around 500MB/s show the interest of disposing of efficient compression methods for such complex, high bandwidth data. As a result, 3D Point Cloud Compression (3DPCC) has raised an increasing interest in recent years.

3 3DPCC METHODS

When compressing point clouds, both lossless and lossy techniques can be considered. Lossless compression makes data more compact by identifying and eliminating the statistical redundancy while completely preserving the original information. On the other side, lossy compression reduces the data size by removing unnecessary, visually useless information through a quantization process. Whatever the approach considered, the mandatory stage to be specified concerns the data decorrelation.

Figure 3 illustrates, under the form of a *Universe Map*, the various 3D point cloud compression of the state of the art. The *Universe Map* provides a multi-dimensional representation and involves three main levels.

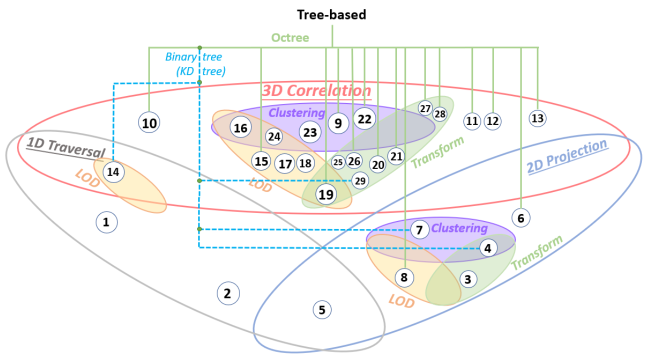


Figure 3: "*Universe Map*" for 3D point cloud compression methods.©Chao CAO

The highest level describes the dimension among which the redundancy is reduced. Three different modes can be here considered: 1D traversal, 2D projection, and 3D decorrelation. The second level presents how the geometry and attribute information is compressed. Three different approaches can be considered, which correspond to decomposition into LODs (*Levels of Details*), clustering methods and transform-based techniques. Interleaving with both the first two levels, the third one concerns the data structure involved processing the geometry information. Various types of tree-based representations can be considered here, including octrees, binary trees and kd-trees.

For the rest of our study, we have considered a categorization of methods among the first dimension (i.e., redundancy reduction), which minimizes the interleaving between the various families of approaches. Let us first describe the 1D traversal compression approaches.

3.1 1D traversal compression

Unlike in the mesh compression field, there is no connectivity information available for point clouds that can provide the geometry correlation between points to help predicting neighboring points.

In the case of 1D prediction methods, the underlying principle consists of constructing a tree-based connectivity, by exploiting the neighborhood relations induced by the native geometric distances between the points in the cloud. Various traversal orders of the resulting trees can then be used to convert the geometry data into a 1D, prediction-adapted signal.

In [Gumhold et al. 2005]① (Circles used in this article are to help the illustration of Figure 3), a prediction tree is constructed, based on a greedy strategy which consists of determining the branches that minimize the prediction residuals, with respect to a breadth first traversal order of the tree. The resulting residuals are finally binary coded with the help of a valence-based arithmetic encoder.

The single-resolution method introduced in [Merry et al. 2006]② aims at compressing the geometry information. Motivated by the work presented in [Gumhold et al. 2005]① and [Taubin and Rossignac 1998], the technique uses multiple geometry predictors in order to take into account the various correlations between neighboring points. Four different symbols ($B - base, L - left, R - right and F - forward$) indicate the direction of the predictors of the child nodes starting from a root node. An additional T (Terminal) code terminates the list of child predictors and implicitly specifies the valence of the children so that the points starting from the root node can be reordered and predicted using such symbols. An example of a resulting spanning tree is illustrated in Figure 4.

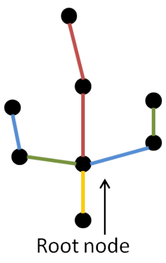


Figure 4: The proposed spanning tree example using different predictors ($B(yellow), L(green), R(blue), F(red)$).

Table 1 summarizes the various 1D traversal compression approaches. Here, we have included two additional approaches, originally intended for 3D mesh compression but which can be also directly applied for 3D point cloud geometry compression. The first one is the reference kd-tree method in [Gandoin and Devillers 2002], while the second concerns the generic predictive coder in [Waschbüsch et al. 2004].

The compression rates reported in Table 1 are expressed in bits per point (bpp). It should be underlined that the term "lossless support" reveals the reported condition in each reference without considering the potential of the methods. It is here applied to the situation where the input data may be voxelized (or, equivalently, represented with integer values). More importantly, it can be recovered identically by the decoder. We can observe that the 3D point cloud-dedicated methods outperform the previous mesh-based techniques.

Although the tree-based traversal methods offer the advantage of a relatively simple implementation, the compression performances are quite limited because such algorithms do not fully take into account the 3D spatial correlations. Thus, exploiting higher dimensional correlations becomes necessary.

Table 1: Summary of 1D traversal compression techniques.©Chao CAO

Summary of 1D traversal compression techniques					
Algorithm	Compression rates (bpp) for geometry	Compression rates (bpp) for attributes	Type of input (datatype)	Lossless support	Remarks
Progressive lossless coder [Gandoin and Devillers 2002]	16.12 (in lossy mode)	-	Static (float)	Yes	Lossless data required domain such as medical research; Progressive; Kd-tree based; Support lossless coding
Generic progressive coder [Waschbüsch et al.2004]	15.51	2 for simple color; 11 for normal	Static (float)	No	Progressive decoding; preditive coding for position, color and normal
Prediction-tree coder [Gumhold et al.2005] ①	10.47	-	Static (float)	No	Streaming; Efficient greedy prediction; Non-guaranteed point order
Linear&lateral predictor coder [Merry et al.2006]②	11.08 (in Axial mode)	-	Static (float)	No	Laser range scanning; streaming; Suitable for real-time applications; Poor on non-regularly sampled or sparse point sets

3.2 Projection and mapping onto 2D planar domains

In this case, the principle consists of converting the 3D point cloud into 2D images/videos by projection or mapping. Then, existing image/video coding technologies are used for compression purposes.

In [Ochotta and Saupe 2004]③, the point cloud is projected onto a regular grid after being partitioned into a set of patches, also denoted as point clusters. Each cluster is then parameterized as a height field, over the corresponding base plane. The use of an atlas of height fields and the heuristic split-and-merge partition of the point set into patches is similar to the approach introduced in [Pauly and Gross 2001]④.

The height fields are projected onto the base plane to obtain a 2D image, which is compressed with the wavelet-based shape-adaptive image coding technique described in [Li and Li 2000]. Finally, with a rate-distortion optimization step arranging an optimal bit allocation, progressive rates and quality can be achieved according to the network bandwidth.

Another mapping-based method is introduced in [Daribo et al. 2012]⑤ with the help of a physical projection. Here, a 2D grid pattern consisting of horizontal and vertical lines is projected onto the 3D model, according to a set of viewpoints. For each viewpoint, this procedure yields a set of 3D curves.Considering the resulting 3D curves as coding units, compression is achieved by predictive coding using a spanning tree of the points' positions.

Several other approaches also adopt a multi-view representation strategy. The principle consists of representing the point cloud as a set of depth images, corresponding to projection according to various angles of view. Based on the multi-view video coding algorithms introduced in [Merkle et al. 2007], which exploit both temporal and inter-view statistical dependencies under a prediction framework, several coding schemes targeting real-time compression, are proposed in [Mekuria et al. 2017]⑥ and [Lien et al. 2009]⑦.

A combination of octree and projection-based method is presented in [Ainala et al. 2016]⑧. Here, the residuals obtained from planar projections are exploited as an enhancement layer within the framework of a coarse octree coding scheme. The points in the enhancement layer are projected onto a 2D plane, determined with the help of a PCA (*Principle Component Analysis*) for each cell of the considered octree. Then, the geometry and color attributes are predictively coded. In addition, the resulting prediction residuals are entropy encoded with the help of a learning-based technique, which leads to improvements in both compression performance and computational complexity. Authors show that the method achieves significant gains in compression ratio when compared to the JPEG-based compression approach introduced in [Mekuria et al. 2017]⑥.

Combination of the octree-based geometry compression and image-coding based color has become in the last years an emerging research direction. A patch-based method, so called Video-based Point Cloud Compression (V-PCC) has been introduced in [Schwarz et al. 2019]⑨. The main philosophy behind V-PCC is to leverage existing video codecs for compressing the geometry and texture information of a dynamic point cloud. The underlying workflow is illustrated in Figure 5(a), while Figure 5(b) presents an example of the geometry and texture images that are generated.

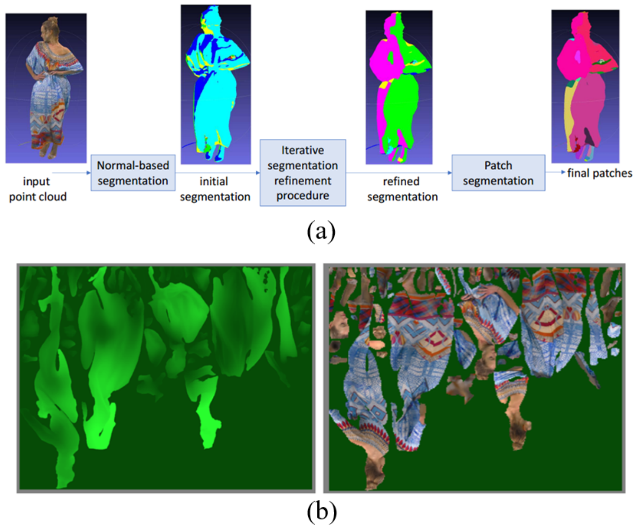


Figure 5: (a) Overview of the V-PCC patch generation process. (b) Example of geometry (left) and texture (right) images. Figures from [Schwarz et al. 2019]⑨

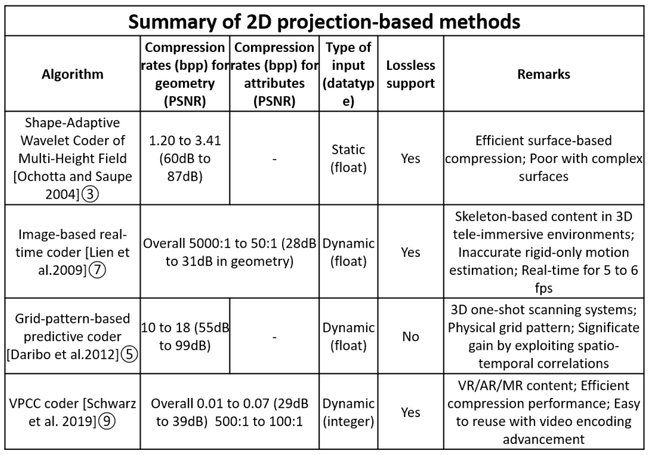
V-PCC decomposes the input point cloud into a set of patches. First, a normal vector is associated to each point, with the help of a local planar surface approximation. Then, an initial clustering of the points is determined by grouping neighboring points with similar orientations (with respect to a given threshold), within a given search radius. The clustering is then refined iteratively by updating the neighboring points in a certain range which have similar normal vectors, in order to obtain smooth cluster borders and to minimize the associated mapping distortion. As the points gathered in the resulting clusters have similar normal vectors, a reference orientation can be associated to each patch, which corresponds to the mean value of the individual normal vectors.

Once the partition of the point cloud into patches accomplished, the set of patches are orthogonally projected onto a 2D plane corresponding to the reference normal vector. The projection plane is discretized according to a regular sampling grid at a given resolution in order to obtain a pixelized projection image representation. Both depth and attribute (color) information are retained in the resulting projection image.

Finally, the geometry and attribute images are generated by assembling the orthogonally projected 2D patches within a global image. The patches are padded into the image in descending order of the size. The pixels will be noted as "occupied" once filled with a patch. The bounding box information of the 2D patch is used to avoid intersection between patches while minimizing the empty space between patches in the global image. The padding process makes it possible to obtain a smooth image with more spatial and temporal consistency which is better suited for video coding.

Table 2 gives an overview of the projection-based approaches. The various techniques discussed here show that the efficiency of 3D point cloud compression can be greatly improved by applying existing 2D image/video compression technologies.

Table 2: Summary of 2D projection-based methods.©Chao CAO



Summary of 2D projection-based methods

Algorithm	Compression rates (bpp) for geometry (PSNR)	Compression rates (bpp) for attributes (PSNR)	Type of input (datatype)	Lossless support	Remarks
Shape-Adaptive Wavelet Coder of Multi-Height Field [Ochotta and Saupe 2004]③	1.20 to 3.41 (60dB to 87dB)	-	Static (float)	Yes	Efficient surface-based compression; Poor with complex surfaces
Image-based real-time coder [Lien et al.2009]⑦	Overall 5000:1 to 50:1 (28dB to 31dB in geometry)		Dynamic (float)	Yes	Skeleton-based content in 3D tele-immersive environments; Inaccurate rigid-only motion estimation; Real-time for 5 to 6 fps
Grid-pattern-based predictive coder [Daribo et al.2012]⑤	10 to 18 (55dB to 99dB)	-	Dynamic (float)	No	3D one-shot scanning systems; Physical grid pattern; Significate gain by exploiting spatio-temporal correlations
VPCC coder [Schwarz et al. 2019]⑨	Overall 0.01 to 0.07 (29dB to 39dB) 500:1 to 100:1		Dynamic (integer)	Yes	VR/AR/MR content; Efficient compression performance; Easy to reuse with video encoding advancement

In addition, such approaches can benefit from the expected evolution of image/video codecs. In particular, the VPCC coder provides state of the art performances, outperforming the other approaches, with compression rates of up to 500:1.

A third family of approaches, described in the following section, notably attempt to directly exploit such 3D correlations.

3.3 Direct exploitation of 3D correlations

Within this framework, let us first discuss the octree-based techniques, which have become highly popular for yielding progressive 3D representations. An octree is a 3D data structure where each node represents a 3D bounding box that is recursively decomposed into eight children leaves.

A neighborhood-based predictor for the point cloud geometry is introduced in [Huang et al. 2006]⑩. An octree is built according to the pre-defined depth of the tree structure. It starts with one root node representing the entire space containing the point cloud. At each level of the octree, 8 leaf nodes are generated as children. They correspond to a uniform subdivision of the parent 3D cell. To each child node is assigned a binary occupancy code, with a value of 1 for occupied cells and 0 for void ones. The binary occupancy code can be entropy encoded directly in single-rate compression or used to find the varying points in a progressive coding scheme. Each level of the octree can be interpreted as a LOD with further subdivision of the space. The octree construction process is terminated when reaching a user-defined, maximum depth level, which is equivalent to specifying a number of quantization bits for the geometry representation. In addition, the octree model is providing a stable structure of the 3D space, that can be notably exploited for motion estimation/compensation purposes in sequential frames.

A representative octree-based lossless intra-frame compression method for point-cloud geometry is presented in [Garcia and de Queiroz 2018]⑪. The usded octree representation is completely described by the *octree sequence* **t**, which defines the binary pattern of the octree decomposition, starting from the root node (depth traversal of the tree).

Two additional sequences are used and exploited for generating a set of contexts:

- the *parent value* **v**, representing the corresponding binary value of its parent illustrating the occupancy.
- the *bit position* **p**, defining the corresponding ordered octee position of the point.

The **t** sequence is finally encoded conditionally with respect to the contexts generated by the **v** and **p** sequences, with both an LZW [Sayood 2017] and an arithmetic encoder. Let us also note that the contexts can be further exploited for temporal geometry prediction.

The experimental results reported demonstrate that the use of contexts makes it possible to achieve a 2% (resp. 5%) gains when compared to the same arithmetic encoder (resp. LZW encoder), directly applied to the **t** sequence.

In [Kammerl et al. 2012]⑫, the octree-based representation is adapted for streaming applications. At each frame, the point clouds are decomposed into octrees. Then, the difference between the octree data structures of successive frames is computed and encoded.

The octree structures of two consecutive frames are built separately and serialized into binary sequences representing the occupancy code. Then they are compared by using exclusive disjunction operator to detect the changing part of the octree. Finally, the resulting difference vector is entropy encoded. The method offers the advantage of a differential octree representation that saves computation and reduces memory requirements.

The lossless inter-frame compression method introduced in [Garcia and de Queiroz 2017]⑬ also takes into account the temporal aspects of the 3D geometry sequence. The principle consists of re-ordering the points in the octree structure in the current 3D frame based on the prediction of the previous frame, prior to entropy encoding. In this way, it becomes possible to improve the temporal consistency of the points, notably for those that have similar properties and exhibit a relatively low motion. The experimental results reported show an average rate reduction of 30% when compared to the uncompressed octree representation. In addition, the approach yields average rate reductions of 24% and 22% when compared to the entropy-encoded octree using GZIP [Sayood 2017] and to the method presented in [Kammerl et al. 2012]⑫, respectively.

Other tree-based methods exploit kd-trees [Lien et al. 2009]⑦ or combinations of binary trees and quadtrees [Kathariya et al. 2018]⑭. The method introduced in [Bentley 1975] is a type of binary tree that is widely used for efficient neighbor search for the points' geometry because of its low complexity of computation. In contrast with the octree-based approach, which builds a predictive-adapted structure along time, the kd-tree splits the space into half at each internal node. This induces a high dependency on the time-varying point cloud bounding box. Thus, it is considered particularly efficient for the intra coding mode. In the case of inter-frame coding, the kd-tree is computationally expensive to build.

In [Kathariya et al. 2018]⑭, authors introduce a so-called binary tree quadtree (*BTQT*) structure. It uses a binary-tree to split the point cloud after building a full octree to detect empty blocks and avoid encoding the full depth octree. Binary-tree blocks contain the local surfaces of the point cloud. Any local surface that exhibit flat characteristics is then projected onto the 2D plane and encoded with the quadtree. Other surfaces that are not flat enough are encoded with the full octree. The proposed scalable coding solution can efficiently compress point cloud data at variable rates which leads to multiple LODs (*Levels Of Detail*), which correspond to gradually increasing complexities of the 3D representation. This brings us into the discussion the importance of disposing of multiple LODs since adaptive bitrate and quality are today highly demanded requirements for data transmission.

LOD (*Level Of Detail*) - based methods are usually based on, or combined with a tree-structure decomposition. An early study [Huang et al. 2008]⑮ introduces a generic point cloud encoder, which creates an octree-based LOD structure. Here, the geometric center and the statistical averages of the normals and colors in each nonempty cell of the octree are estimated and represent the part of the model within each cell. Each level of the octree represents a LOD of the original 3D model. A novel adaptive color frame determination method is introduced to better exploit the color coherency. Using PCA, the coordinates in the RGB color space are transformed into a new frame where the correlation of the colors is more concentrated and is aligned with the eigenvectors from PCA. Therefore, adaptive color quantization is achieved with the *Generalized Lloyd Algorithm* (GLA). Similarly, GLA is also applied for clustering the points and generate a coarser LOD in [Fan et al. 2013]⑯.

Another LOD-based method, based on adaptive subsampling of the points in the cloud is presented in [Kitago and Gopi 2006]⑰. A cost function, which integrates information about the redundancy and non-uniformity of the points, is defined and used to prioritize the points. The set of points is sub-sampled based on the dominant geometric features and on the local sampling density of the model. The problem is then converted to a minimization of CSRBF (*Compactly Supported Radial Basis Function*) [Wendland 1995] coefficients. The points position and the relative normal information are used to estimate a surface along which the points are resampled and scattered in a way that minimizes the distortion of the reconstructed points. By selecting different cost functions, multiple LODs can be achieved with the help of a subsampling process, which takes into account various precisions of the reconstructed points.

A voxel-based, real-time, adaptive LOD approach is introduced in [Golla and Klein 2015]⑱. The method is able to take into account the network condition as well as application-related parameters. Thus, the required responding time from the application and the network available bandwidth are used to compute on-the-fly a target compression ratio. The point cloud is decomposed into 2D images with height map as in [Ochotta and Saupe 2004]③ and then compressed by applying JPEG image coding. The voxels are resized into a higher or lower resolution in a way that ensures a trade-off between quality and compression ratio.

The sparse voxel arrays are described as voxelized point clouds in [de Queiroz and Chou 2017a]⑲. Voxels are acquired by decomposing the point cloud space into blocks, as illustrated in Figure 6, corresponding to a discretization of the 3D space on a regular grid, defined by a given step size (i.e., the cell size).

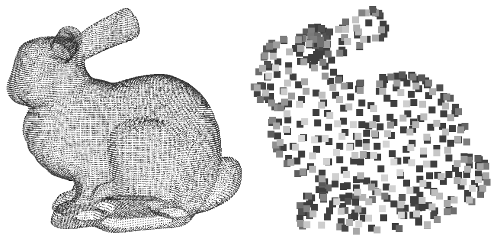


Figure 6: Point cloud object bunny (left) and voxelized object bunny (right).

Let us underline that the voxelization process leads to an integer representation of the geometric coordinates of the cloud points. By varying the step size chosen for the voxelization process, multiple LODs can be generated.

Leveraging on the graph-based motion estimation approach introduced in [Thanou et al. 2015]⑳, the motion-compensation compression technique proposed in [de Queiroz and Chou 2017a]⑲ shows high efficiency and ensures real-time processing of voxelized point clouds. Similar results are also presented in [Loop et al. 2013].

The principle consists of extending the block-based, 2D video motion compensation techniques, widely used in various video compression standards, to 3D point clouds. In this case, 2D blocks are replaced by 3D cubes which are considered as "voxels". The matching procedure for consecutive frames determines the correspondence between voxels by minimizing the distance for both geometry and color information. The geometry residuals are considered as motion vectors and are associated with each voxel or voxel cube. Color residuals are considered as color prediction errors.

Compared to the results presented in [de Queiroz and Chou 2016]㉑, where intra coding is used, the overall rate for geometry and color is about 3.7 bit per voxel for both cases. The approach in [de Queiroz and Chou 2017a]⑲ produces better color estimation but introduces some noise caused by geometry shifts. Optimization can be further investigated to select a fair tradeoff between geometry and color information.

Clustering-based methods come into vision in recent years since the dense 3D point cloud can be segmented/clustered into different parts, based on color or motion information.

In [Zhang et al. 2018]㉒, authors propose a dedicated point cloud color compression scheme based on hierarchical segmentation. A global color segmentation is first performed, in order to cluster all the point according to their color. To this purpose, the high-dimensional mean shift clustering algorithm [Georgescu et al. 2003] is here used. Then, a local segmentation is performed in the geometric space in order to guarantee the color consistency within the resulting patches. This property is notably useful for efficient intra prediction.

A similar method for color attribute clustering and encoding is proposed in [Lien et al. 2009]⑦. Here, a skeleton-based motion estimation approach is used for deriving the kinematic parameters of a human body. The skeletons are generated offline with the method in [Cheung et al. 2003]. The point cloud is clustered into point sets associated with the skeletons, in order to obtain a more accurate motion estimation.

Here, the rigid-body motion estimation is based on the *Iterative Closest Points* (ICP) algorithm [Paul J. Besl 1992], which has been also used for similar purpose in [Mekuria et al. 2017]⑥. The motion compensation residuals are stored within an atlas of residuals which is generated by projecting the residuals onto a regular 2-D grid embedded in a cylindrical coordinate system defined around the skeleton. They are finally compressed with the H.264 video codec.

The advantage of using a skeleton is that it can be used to interact with the objects in the virtual environment or to analyze the subject's movement. However, since the 3D point clouds are often noisy, the correlations between the points are not sufficient enough to support a unique skeleton registration in the 3D space. This issue can significantly penalize the compression process, since inaccurate motion compensation can lead to 2D residuals that are inappropriate for efficient video coding.

Based on the geometric driven approach introduced in [Morell et al. 2014]㉓ a compression clustering-based technique is introduced in [Navarrete et al. 2018]㉔. Both geometric and color features are considered for clustering the point cloud into a set of planar patches, by employing the method introduced in [Viejo and Cazorla 2014]. Then, a dedicated compression scheme is designed, based on the *Gaussian Mixture Models* (GMMs). The principle consists of replacing the clustered points with each planar patch by the corresponding GMM approximation. Multiple LODs can be obtained by modifying the configuration of the model.

Transform-based methods attempt to extend the traditional signal/image transforms to point clouds. The difficulty here comes from the lack of topological information of the point cloud representation. In [Wiman and Qin 2009]㉕, a wavelet-based framework is proposed for LIDAR-like point cloud compression, called WALZ (*WAvelet Lidar Zipper*). Each coordinate component, X, Y, and Z, is independently wavelet-transformed as a 1D vector. In this way, highly correlated points, such as consecutively recorded points, will generate small and similar detail coefficients, which can be efficiently compressed.

A graph transform method is presented in [Zhang et al. 2014]㉖. Assuming that the point cloud is octree-decomposed, each occupied leaf node can be interpreted as a representative voxel. Exploiting the implicit topological structure of the voxel grid, the neighboring points are connected. This makes it possible to construct a set of graphs on small neighborhoods of the point cloud. The graph transform [Zhang and Florencio 2013], which is equivalent to *Karhunen-Loeve Transform* (KLT), is finally applied in order to decorrelate the geometry signal. However, the transform performance highly depends on the topological information of the resulting graphs. Moreover, the method is poorly adapted for inter-frame coding, since the graphs can vary significantly from one frame to another.

The so-called RAHT (*Region-Adaptive Hierarchical Transform*) approach introduced in [de Queiroz and Chou 2016]㉑ aims at improving the color compression. A hierarchical, octree-based transform is here used. The voxels are grouped into a lower level of the octree in order to predict the colors of the nodes in the successive level. The process can be reduced to the Haar transform in the case where the prediction weights are uniform. The main advantage of the method is its low computational complexity, which makes it appropriate for real-time transmission.

The *Graph Fourier Transform* (GFT) proposed in [Thanou et al. 2016]㉗ is based on their previous work [Thanou et al. 2015]⑳ and exploits both the spatial correlation and the temporal redundancy of the color and geometry sequences. As in [Zhang et al. 2014]㉖, a graph is constructed from the initial 3D point cloud, with the help of an octree decomposition. A spectral representation of the resulting geometry and color signals is obtained with the help of the GFT which is defined through the graph Laplacian operator.

The *Gaussian Process Transforms* (GPT) introduced in [de Queiroz and Chou 2017b]㉘ describes an approach to model the statistics of the point colors by assuming that they are samples of a stationary Gaussian process defined over an infinite 3D space. The covariance function of the Gaussian process is used to derive the covariance matrix of the colors, which in turn is used to obtain the KLT of the color signal.

In [Shao et al. 2017]㉙, an optimized graph transform scheme for attribute compression is introduced. A kd-tree partition is here used to generate the graph. The Laplacian sparsity for graph transform is optimized with the help of an offline training stage. A Lagrangian, RDO-based quantization procedure is also considered. The experimental results show a significant gain on both transform efficiency and R-D performance, for static point clouds. The applicability of the method to dynamic point clouds, under an inter prediction mode, requires further studies.

Table 3 summarizes the various 3D approaches discussed in this section.In a general manner, the fully 3D approaches offer useful solutions. In a majority of cases, they require integer/voxelized input representations.

However, in a majority of cases, such approaches suffer from a relatively high computational complexity, since they require the creation of a topological information. This issue is particularly critical in the case of dynamic 3D point clouds. Moreover, in such cases, the additional 3D motion estimation/compensation procedure increases the computational burden.

Let us observe that such approaches fail to achieve the compression performances offered by the 2D-based V-PCC approach, which fully takes advantage, in a pragmatic manner, of the efficiency of HEVC 2D video codec. However, the motion estimation stage, performed in the 3D space, is much more precise than in the case of 2D, video-based approaches. It would be interesting in the future to investigate how to take advantage of the accuracy of the 3D motion estimation and the efficiency of 2D video-based techniques.

Table 3: Summary of fully 3D approaches.©Chao CAO

Summary of direct 3D decorrelation techniques

Algorithm	Compression rates (bpp) for geometry (PSNR)	Compression rates (bpp) for attributes (PSNR)	Type of input (datatype)	Lossless support	Remarks
Generic progressive octree-based coder [Huang et al.2008]⑮	0.32 to 6.1 (53dB to 70dB)	0.63 to 7.9 (18dB to 26dB) for normal and 0.41 to 7.2 (23dB to 39dB) for color Y	Static (integer)	No	Potential use for lossless encoding
Hierarchical clustering coder [Fan et al.2013]⑯	0.36 to 6 (49dB to 61dB)	0.36 to 6 (19dB to 27dB) for normal	Static (integer)	No	Low bandwidth network; Better reconstructed quality at low bitrates
Graph transform attribute coder [Zhang et al.2014] ㉖	?	0.16 to 5.36 (28dB to 52dB) for color Y	Static (integer) Voxelized	No	Efficient than Octree-based methods
Region-Adaptive Hierarchical Transform (RAHT) coder [de Queiroz and Chou 2016] ㉑	?	0.85 to 2.5 (31.7dB to 39.8dB) for color Y	Static (integer) Voxelized	No	High computational efficiency; Real-time 3D video rate of 30 fps
Gaussian Process Transforms (GPTs) coder [de Queiroz and Chou 2017b] ㉙	?	0.54 to 2.11 (34.2dB to 41.8dB) for color Y	Static (integer) Voxelized	No	Gaussian processes model; Superior compression performance than RAHT
Hierarchical segmentation coder [Zhang et al.2018] ㉒	?	Overall 2.0 (51dB) for color Y	Static (integer)	Yes	Hierarchical segmentation based on attributes; Intra-cluster prediction method for lossless compression
Context-based intra coder [Garcia and de Queiroz 2018]⑪	Overall 1.95 (?)		Static (integer) Voxelized	Yes	Optimized contexts employed on octree for lossless coding
Diffiential octree coder [Kammerl et al.2012]⑫	11:1 to 40:1 with precision from 0.5mm to 9mm	?	Dynamic (float)	No	Online compression; Streaming; Optimized for computation and memory requirements
Image-based real time coder [Golla and Klein 2015]⑱	0.1 to 3.2 (65dB to 86dB) overall 930:1 with 5mm precision	?	Dynamic (float)	No	Online robotics applications; Efficient in storage and computational cost; Using JPEG2000
Graph-based motion estimation coder [Thanou et al.2015]⑳	?	0.08 to 1.85 (34dB to 44.5dB) for color Y	Dynamic (integer) Voxelized	No	Accurate motion estimation for color compression performance; Graph transform
Optimized motion compensation coder [de Queiroz and Chou 2017a] ⑲	Overall 3.7 (?)		Dynamic (integer) Voxelized	No	Low bit rates targeted; Rate-distortion optimized motion compensation

To accelerate the advance of 3D point cloud compression technologies, standardization is carried out to gather experts and researchers from different fields to discuss the state of the art which is conducive to the inspiration of innovation.

4 STANDARDIZATION OF 3D POINT CLOUD COMPRESSION

The *Moving Picture Experts Group* (MPEG), is a working group of ISO/IEC whose mission is to develop standards for coded representation of digital audio and video and related data.

After identifying the increasing need for 3D point cloud compression technologies int the field of consumer electronics industry, a dedicated activity has been launched by the MPEG-3DG (3D Graphics group).

Within this framework, there are currently two different compression frameworks that are under development:

1. *Video-based Point Cloud Compression* (V-PCC).
2. *Geometry-based Point Cloud Compression* (G-PCC).

V-PCC is a toolbox of video-based compression methods and aims at providing low complexity decoding capabilities for applications that require real-time decoding, such as virtual/augmented reality, immersive communications... V-PCC is leveraging the existing and future video compression technologies, as well as the video eco-system in general (hardware acceleration, transmission services, and infrastructure) while enabling new kinds of application [3DG 2018]. The reference model encoder implementation from the 124th MPEG meeting shows compression rates of 125:1 while achieving good perceptual quality. In October 2018, the V-PCC standard has been promoted to the Committee Draft stage.

G-PCC is considered to provide efficient lossless and lossy compression for deployment of autonomous driving, 3D maps, and other applications that exploit LiDAR generated point clouds (or similar content). Several geometric-driven approaches are included in the G-PCC pack. G-PCC standard has been promoted to the Committee Draft stage in March 2019.

5 CONCLUSION AND PROSPECT

This survey explores the state-of-the-art 3D point cloud compression methods. Several families of approaches have been identified and discussed, including 1D traversal techniques, 2D approaches applying projection and mapping and fully 3D methods, with octree-based, LOD-based, clustering-based and transform-based representations.

The 1D traversal approaches present the advantage of simplicity. More often, they rely on various predictive schemes. However, they fail to take into account the intrinsic nature of the 3D geometric correlations.

The 2D approaches are based on multi-view representations, achieved by projecting the 3D point cloud onto various images. Their main advantage comes from the possibility of exploiting existing, optimized image/video codecs.

Furthermore, such approaches seem to be today the most well-suited to realize real-time mobile media applications. However, it is difficult to ensure a lossless representation with 2D techniques, because of the inherent 3D/2D projection stage. This penalizes them for applications that require a high precision, such as autonomous driving, robotics, and medical assistance.

In this case, the solution can rely on fully 3D approaches, which offer a generic framework for both lossy and lossless 3D point cloud compression.

However, determining both spatial and temporal correlation within unstructured point cloud data is still an arduous task. Future studies may continue by improving traditional methods like prediction, estimation and compensation to overcome the constraints of this storage demanding 3D data representation.

A recent study on mesh animation compression [Luo et al. 2019] shows the potential of spatio-temporal segmentation techniques. It would be interesting to explore how such techniques can be adapted and applied for motion-based 3D point cloud compression.

The emerging deep learning technologies, that have known recently a world-wide success in the field of artificial intelligence, can also be a promising axis of research. Their first applications to point cloud content concern the multiple object classification [Qi et al. 2017]. However, such techniques can be *a priori* used for the optimization of the various stages involved in the compression schemes.

REFERENCES

3DG. 2015. Current Status on Point Cloud Compression. ISO/IEC JTC1/SC29/WG11, Geneva, CH.

3DG. 2018. Liaison on MPEG-I Point Cloud Compression. ISO/IEC JTC1/SC29/WG11, Macau, CN.

Inc. 8i Labs. 2017. 8i Voxelized Full Bodies, version 2 âĂŞ A Voxelized Point Cloud Dataset. ISO/IEC JTC1/SC29/WG11 m40059 ISO/IEC JTC1/SC29/WG1 M74006, Geneva, CH.

Khartik Ainala, Rufael N. Mekuria, Birendra Khathariya, Zhu Li, Ye-Kui Wang, and Rajan Joshi. 2016. An improved enhancement layer for octree based point cloud compression with plane projection approximation. In *Applications of Digital Image Processing XXXIX*, Andrew G. Tescher (Ed.). SPIE. https://doi.org/10.1117/12.2237753

Pierre Alliez and Craig Gotsman. 2005. Recent advances in compression of 3D meshes. In *Advances in multiresolution for geometric modelling*. Springer, 3–26.

Jon Louis Bentley. 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* 18, 9 (Sept. 1975), 509–517. https://doi.org/10.1145/361002.361007

K. M. G. Cheung, S. Baker, and T. Kanade. 2003. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 1. I–I. https://doi.org/10.1109/CVPR.2003.1211340

Ismael Daribo, Ryo Furukawa, Ryusuke Sagawa, Hiroshi Kawasaki, Shinsaku Hiura, and Naoki Asada. 2012. Efficient rate-distortion compression of dynamic point cloud for grid-pattern-based 3D scanning systems. *3D Research* 3, 1 (jan 2012). https://doi.org/10.1007/3dres.01(2012)2

R. L. de Queiroz and P. A. Chou. 2016. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. *IEEE Transactions on Image Processing* 25, 8 (Aug 2016), 3947–3956. https://doi.org/10.1109/TIP.2016.2575005

R. L. de Queiroz and P. A. Chou. 2017a. Motion-Compensated Compression of Dynamic Voxelized Point Clouds. *IEEE Transactions on Image Processing* 26, 8 (Aug 2017), 3886–3895. https://doi.org/10.1109/TIP.2017.2707807

R. L. de Queiroz and P. A. Chou. 2017b. Transform Coding for Point Clouds Using a Gaussian Process Model. *IEEE Transactions on Image Processing* 26, 7 (July 2017), 3507–3517. https://doi.org/10.1109/TIP.2017.2699922

Y. Fan, Y. Huang, and J. Peng. 2013. Point cloud compression based on hierarchical point clustering. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. 1–7. https://doi.org/10.1109/APSIPA.2013.6694334

Pierre-Marie Gandoin and Olivier Devillers. 2002. Progressive Lossless Compression of Arbitrary Simplicial Complexes. *ACM Trans. Graph.* 21, 3 (July 2002), 372–379. https://doi.org/10.1145/566654.566591

D. C. Garcia and R. L. de Queiroz. 2017. Context-based octree coding for point-cloud video. In *2017 IEEE International Conference on Image Processing (ICIP)*. 1412–1416. https://doi.org/10.1109/ICIP.2017.8296514

D. C. Garcia and R. L. de Queiroz. 2018. Intra-Frame Context-Based Octree Coding for Point-Cloud Geometry. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. 1807–1811. https://doi.org/10.1109/ICIP.2018.8451802

Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. 2003. Mean shift based clustering in high dimensions: a texture classification example. In *Proceedings Ninth IEEE International Conference on Computer Vision*. 456–463 vol.1. https://doi.org/10.1109/ICCV.2003.1238382

T. Golla and R. Klein. 2015. Real-time point cloud compression. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5087–5092. https://doi.org/10.1109/IROS.2015.7354093

Stefan Gumhold, Zachi Kami, Martin Isenburg, and Hans-Peter Seidel. 2005. Predictive Point-cloud Compression. In *ACM SIGGRAPH 2005 Sketches (SIGGRAPH '05)*. ACM, New York, NY, USA, Article 137. https://doi.org/10.1145/1187112.1187277

Y. Huang, J. Peng, C. . J. Kuo, and M. Gopi. 2008. A Generic Scheme for Progressive Point Cloud Coding. *IEEE Transactions on Visualization and Computer Graphics* 14, 2 (March 2008), 440–453. https://doi.org/10.1109/TVCG.2007.70441

Yan Huang, Jingliang Peng, C.-C. Jay Kuo, and M. Gopi. 2006. Octree-based Progressive Geometry Coding of Point Clouds. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics (SPBG'06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 103–110. https://doi.org/10.2312/SPBG/SPBG06/103-110

J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. 2012. Real-time compression of point cloud streams. In *2012 IEEE International Conference on Robotics and Automation*. 778–785. https://doi.org/10.1109/ICRA.2012.6224647

B. Kathariya, L. Li, Z. Li, J. Alvarez, and J. Chen. 2018. Scalable Point Cloud Geometry Coding with Binary Tree Embedded Quadtree. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. https://doi.org/10.1109/ICME.2018.8486481

Masaki Kitago and M. Gopi. 2006. Efficient and Prioritized Point Subsampling for CSRBF Compression. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics (SPBG'06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 121–129. https://doi.org/10.2312/SPBG/SPBG06/121-128

Shipeng Li and Weiping Li. 2000. Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding. *Circuits and Systems for Video Technology, IEEE Transactions on* 10 (09 2000), 725 – 743. https://doi.org/10.1109/76.856450

Jyh-Ming Lien, Gregorij Kurillo, and Ruzena Bajcsy. 2009. Multi-camera tele-immersion system with real-time model driven data compression. *The Visual Computer* 26, 1 (may 2009), 3–15. https://doi.org/10.1007/s00371-009-0367-8

Charles Loop, Cha Zhang, and Zhengyou Zhang. 2013. Real-time High-resolution Sparse Voxelization with Application to Image-based Modeling. In *Proceedings of the 5th High-Performance Graphics Conference (HPG '13)*. ACM, New York, NY, USA, 73–79. https://doi.org/10.1145/2492045.2492053

Guoliang Luo, Zhigang Deng, Xiaogag Jin, Xin Zhao, Wei Zeng, Wenqiang Xie, and Hyewon Seo. 2019. 3D Mesh Animation Compression based on Adaptive Spatio-temporal Segmentation. (2019).

Adrien Maglo, Guillaume Lavoué, Florent Dupont, and Céline Hudelot. 2015. 3D Mesh Compression: Survey, Comparisons, and Emerging Trends. *ACM Comput. Surv.* 47, 3, Article 44 (Feb. 2015), 41 pages. https://doi.org/10.1145/2693443

Rufael Mekuria, Kees Blom, and Pablo Cesar. 2017. Design, implementation, and evaluation of a point cloud codec for tele-immersive video. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 4 (2017), 828–842.

Philipp Merkle, Aljoscha Smolic, Karsten Muller, and Thomas Wiegand. 2007. Multi-view video plus depth representation and coding. In *2007 IEEE International Conference on Image Processing*, Vol. 1. IEEE, I–201.

Bruce Merry, Patrick Marais, and James Gain. 2006. Compression of Dense and Regular Point Clouds. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH '06)*. ACM, New York, NY, USA, 15–20. https://doi.org/10.1145/1108590.1108593

Vicente Morell, Sergio Orts, Miguel Cazorla, and Jose Garcia-Rodriguez. 2014. Geometric 3D point cloud compression. *Pattern Recognition Letters* 50 (2014), 55–62.

Javier Navarrete, Diego Viejo, and Miguel Cazorla. 2018. Compression and registration of 3D point clouds using GMMs. *Pattern Recognition Letters* 110 (2018), 8–15.

Tilo Ochotta and Dietmar Saupe. 2004. *Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields.*

Johannes Otepka, Sajid Ghuffar, Christoph Waldhauser, Ronald Hochreiter, and Norbert Pfeifer. 2013. Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. *ISPRS International Journal of Geo-Information* 2, 4 (oct 2013), 1038–1065. https://doi.org/10.3390/ijgi2041038

Neil D. McKay Paul J. Besl. 1992. Method for registration of 3-D shapes. https://doi.org/10.1117/12.57955

Mark Pauly and Markus Gross. 2001. Spectral Processing of Point-sampled Geometry. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 379–386. https://doi.org/10.1145/383259.383301

Jingliang Peng, Chang-Su Kim, and C.-C. Jay Kuo. 2005. Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation* 16, 6 (dec 2005), 688–733. https://doi.org/10.1016/j.jvcir.2005.03.001

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Khalid Sayood. 2017. *Introduction to data compression.* Morgan Kaufmann.

S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. KrivokuÄĞa, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko. 2019. Emerging MPEG Standards for Point Cloud Compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (March 2019), 133–148. https://doi.org/10.1109/JETCAS.2018.2885981

Yiting Shao, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li. 2017. Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform. In *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 1–4.

Gabriel Taubin and Jarek Rossignac. 1998. Geometric Compression Through Topological Surgery. *ACM Trans. Graph.* 17, 2 (April 1998), 84–115. https://doi.org/10.1145/274363.274365

D. Thanou, P. A. Chou, and P. Frossard. 2015. Graph-based motion estimation and compensation for dynamic 3D point cloud compression. In *2015 IEEE International Conference on Image Processing (ICIP)*. 3235–3239. https://doi.org/10.1109/ICIP.2015.7351401

D. Thanou, P. A. Chou, and P. Frossard. 2016. Graph-Based Compression of Dynamic 3D Point Cloud Sequences. *IEEE Transactions on Image Processing* 25, 4 (April 2016), 1765–1778. https://doi.org/10.1109/TIP.2016.2529506

Diego Viejo and Miguel Cazorla. 2014. A robust and fast method for 6DoF motion estimation from generalized 3D data. *Autonomous Robots* 36, 4 (2014), 295–308.

Michael Waschbüsch, Markus H Gross, Felix Eberhard, Edouard Lamboray, and Stephan Würmlin. 2004. Progressive Compression of Point-Sampled Models.. In *SPBG*. Citeseer, 95–102.

Holger Wendland. 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics* 4, 1 (01 Dec 1995), 389–396. https://doi.org/10.1007/BF02123482

Hakan Wiman and Yuchu Qin. 2009. Fast compression and access of LiDAR point clouds using wavelets. In *2009 Joint Urban Remote Sensing Event*. 1–6. https://doi.org/10.1109/URS.2009.5137589

C. Zhang and D. Florencio. 2013. Analyzing the Optimality of Predictive Transform Coding Using Graph-Based Models. *IEEE Signal Processing Letters* 20, 1 (Jan 2013), 106–109. https://doi.org/10.1109/LSP.2012.2230165

C. Zhang, D. FlorÃłncio, and C. Loop. 2014. Point cloud attribute compression with graph transform. In *2014 IEEE International Conference on Image Processing (ICIP)*. 2066–2070. https://doi.org/10.1109/ICIP.2014.7025414

K. Zhang, W. Zhu, and Y. Xu. 2018. Hierarchical Segmentation Based Point Cloud Attribute Compression. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3131–3135. https://doi.org/10.1109/ICASSP.2018.8461644