Trends in efficient representation of 3D point clouds

Sugimoto, K.; Cohen, R.A.; Tian, D.; Vetro, A.

TR2017-189 December 2017

Abstract

In this paper, the current trends of technologies to represent 3D point clouds and its use cases are overviewed. The use of 3D point clouds have been increasing, and efficient representation of 3D point clouds are needed. Mesh coding after surface fitting has been one of the major approaches for point cloud compression, and compression exploiting the correlation after simplifying the original point cloud is also becoming a major approach in recent years. A standardization activity for 3D point cloud compression is ongoing under MPEG. A call for proposals on new coding tools for 3D point cloud compression has been issued in January, 2017, and it is planned to finalize the standardization in April, 2019.

APSIPA ASC

Trends in efficient representation of 3D point clouds

Kazuo Sugimoto*, Robert A. Cohen†, Dong Tian† and Anthony Vetro†
*Mitsubishi Electric Corporation, Kamakura, Kanagawa, Japan
E-mail: Sugimoto.Kazuo@ak.MitsubishiElectric.co.jp Tel: +81-467-41-2463
†Mitsubishi Electric Research Laboratories, Cambridge, MA, USA
E-mail: {cohen, tian, avetro}@merl.com Tel: +1-617-621-7503

Abstract— **In this paper, the current trends of technologies to represent 3D point clouds and its use cases are overviewed. The use of 3D point clouds have been increasing, and efficient representation of 3D point clouds are needed. Mesh coding after surface fitting has been one of the major approaches for point cloud compression, and compression exploiting the correlation after simplifying the original point cloud is also becoming a major approach in recent years. A standardization activity for 3D point cloud compression is ongoing under MPEG. A call for proposals on new coding tools for 3D point cloud compression has been issued in January, 2017, and it is planned to finalize the standardization in April, 2019.**

I. INTRODUCTION

With the increased proliferation of scenarios using 3D technologies for applications such as virtual and augmented reality, mobile mapping, scanning of historical artifacts, and 3D visualization, representing these kinds of data as 3D point clouds has become a popular method for storing and conveying the data independently of how it was captured. A point cloud consists of a set of coordinates indicating the location of each point, along with one or more attributes such as color associated with each point. 3D point clouds can be captured in a variety of ways, such as using multiple cameras and depth sensors. These point clouds may be made up of thousands or even billions of points in order to represent realistically reconstructed objects or scenes; thus efficient representation of point clouds are needed to store or transmit these data. Compared with image compression, 3D point cloud compression is much more difficult, because no predefined orders or local topologies exist, in contrast to an image, which is defined as a uniform 2D grid of pixels. Additionally, each point in a point cloud may have not only a 3D position information (x, y, z) but also a color information (R, G, B) and possibly other attributes such like transparency, time of acquisition, reflectance of laser or material property, etc.

In Section II of this paper, we show some use cases for point cloud compression. In Section III, the current trends of technologies used to represent 3D point clouds are overviewed. In Section IV, the current direction of some standardization activities for point cloud compression is introduced, followed by conclusions in Section V.

II. USE CASES FOR POINT CLOUD COMPRESSION

The use of 3D point clouds have been increasing in recent years, and the recent advances of technologies in capturing and rendering 3D points have realized novel applications in the area of tele-presence, virtual reality and even large-scale dynamic 3D maps [1]. In these applications, compression is needed for efficiently transmitting or storing 3D point clouds. In this section, we introduce several use cases for 3D point cloud compression.

A. Real-Time 3D immersive telepresence

Highly realistic 3D representations can be achieved by using 3D point clouds. One suitable application is real-time 3D immersive tele-presence such as 8i [2], Microsoft holo-portation [3], and in ongoing industry oriented research projects [4]. Compared to 3D mesh reconstruction where connectivity among vertices is specified, reconstruction of non-meshed 3D point clouds by just obtaining a representation of the input point cloud is much simpler, making them amenable for real-time applications. In a typical tele-immersive 3D video conference system, point clouds are generated by processing the data from multiple color plus depth cameras or other sensors. These point clouds are then compressed and transmitted over networks to the receiver, which in turn decompresses and reconstructs the 3D point cloud. Finally, the 3D point clouds are rendered to create the virtual world. Key requirements for this application are:

a) Lossy compression with bit-rate control
b) Low complexity and/or support for real-time encoding/decoding
c) Error resiliency
d) Color attributes coding

B. VR Content Viewing with Interactive Parallax

In recent years, the head mounted display (HMD) is rapidly spreading as a device for experiencing virtual and augmented reality (VR). One major improvement for visual comfort relates to the viewing of content with interactive parallax, where the rendering viewport is updated for each new position of the end-user head. In this case, real-time encoding for 3D point clouds is not required, however, real-time decoding is needed. Since only the 3D point clouds from a narrow range of viewport positions are required to be rendered, typically corresponding to a bounding box around the average position of the end-user's head, the 3D point clouds to be rendered may be a part only of the total content transmitted to the end-user. Key requirements for this application are similar to the ones for the previous use case except that real-time encoding is not necessary.

C. *3D Free viewpoint Sport Replays Broadcasting*

3D Point clouds can be used for free viewpoint playback of sports and interaction on mobile devices and TV. This use case requires a compression method such that the point cloud data can be streamed or stored in an interoperable manner. Multiple cameras are used to reconstruct 3D point clouds with color attributes, which are subsequently compressed, transmitted and stored at the end-user's device. The end-user can virtually watch the scene from any angle by decoding and rendering the relevant stored data. The industry has already moved in this direction, and an example is Replay Technology [5].

D. *Geographic Information Systems*

Geographic information is typically captured using Lidar, SAR or other techniques and is often represented as point clouds. These data are often stored in servers providing a service such as remote rendering or querying based on specific geographic information. The amount of data contained in these 3D point clouds, however, are often too massive to be handled efficiently in the system. 3D point cloud compression can reduce both the storage requirements at the server and the traffic exchanged between the client and the server. In this use case, region selectivity is important to query subsets of the point cloud representing a certain geographic area. Lossless compression may also be required depending on the application. Examples of applications that could benefit from a standardized solution are Oracle spatial and graphical databases [6], ArcGIS [7], and PointScene [8].

E. *Cultural heritage*

Over recent years 3D scanning has become part of a coherent and non-contact approach to archive objects representing cultural heritage. 3D point clouds of cultural sites and objects allow us to preserve them in a virtual world forever. These objects can be viewed from any angle, and 3D point cloud compression can make these objects available to the wider public. Lossless compression is important to enable the best representation of the objects, and generic attributes coding such as for material properties may be required. An example is the project Culture 3D Cloud [9], where pictures taken by smart phones of cultural heritage objects are transformed into 3D point clouds that can be viewed from any angle.

F. *Large-scale 3D maps for autonomous navigation*

Large-scale 3D maps of indoor and outdoor environments can be created using devices that provide localization combined with depth and color measurements of the surrounding environment [10]. Localization can be achieved using GPS devices, inertial measurement units (IMU), cameras, or combinations of these and other devices, while the depth measurements could be achieved with time-of-flight, radar or laser scanning systems. Example mapping systems are already commercially available and come in various forms such as the high-end mobile mapping system [11]. Irrespective of the specific platform that is used to acquire the measurements, it is possible to generate a 3D map by combining the depth measurements, such as the high-density laser-scanned point cloud in Fig. 1, with camera images. This combination of point cloud data with camera images to generate a 3D map is illustrated in Fig. 2. These maps can further be combined with road markings such as lane information and road signs to create maps to enable autonomous navigation of vehicles around a city as shown in Fig. 3[12]. Multiple map layers will be stored and exchanged across the network, including static maps that do not change very frequently and dynamic maps that include real-time information about dynamic objects in the scene such as vehicles or pedestrians. Finally, only necessary information, such as road markings and traffic signs, is extracted from 3D point clouds and camera data measured and collected by the mobile mapping system to create precise 3D vector maps to be provided to the end-users. 3D point clouds are also compared in order to extract changes from the past data. To maintain the original 3D map and update the dynamic 3D map, massive amounts of 3D point cloud data are transmitted, stored and processed. 3D point cloud compression reduces the storage requirements and the amount of traffic.

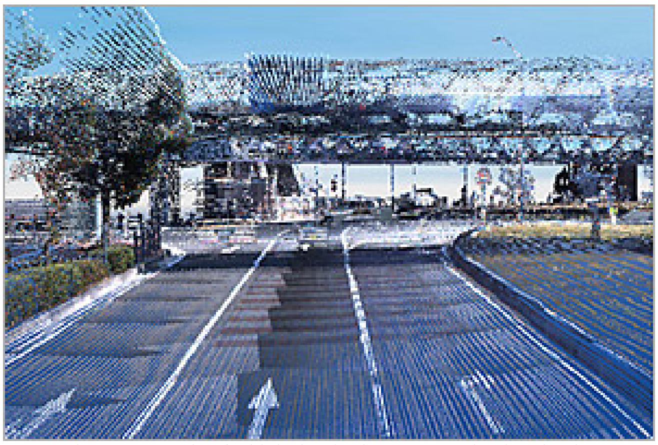

Fig. 1 A sample of a high-density laser-scanned point clouds

Fig. 2 Map creation where images are projected by superimposing laser point cloud data onto camera images

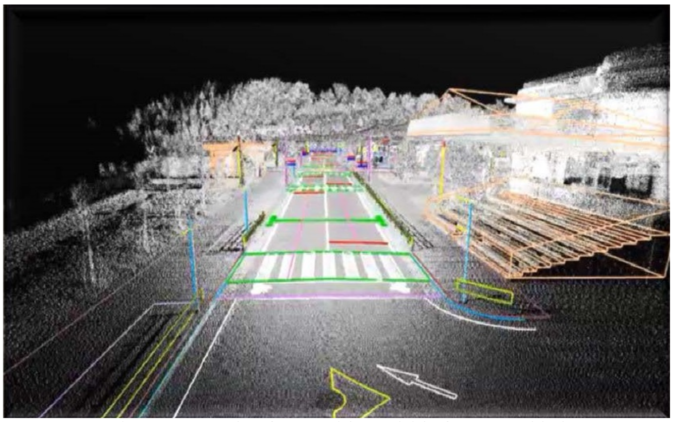

Fig. 3 A sample of map with roadside features added

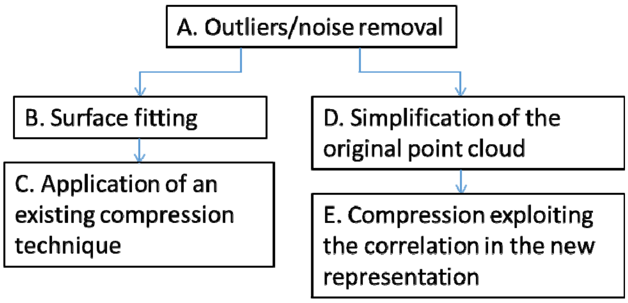

Fig. 4 Overview of the general procedure of point cloud compression

III. TRENDS OF POINT CLOUD REPRESENTATION TECHNOLOGIES

Much of the work on point cloud compression uses techniques that come from the computer graphics community, e.g. generating or modeling shapes using triangular meshes, estimating surfaces on shapes, etc. On the other hand, conventional data compression or some more recent techniques that come from the signal processing community are also investigated. The diagram in Fig. 4 shows an overview of the general procedure that is taken by most existing methods for point cloud compression. We overview trends of technologies for each procedure.

A. Outlier/noise removal

There exists points that are not consistent with the other points in 3D point clouds depending on the acquisition system. These outliers or noise exhibited in the position or attributes can significantly degrade compression efficiency. Removing these outliers or noise can contribute to high compression efficiency especially for the lossy compression case. If the input point clouds have outliers or noise, a surface fitting through the points can be too irregular for efficient compression. To solve this problem, defining a smooth target surface for reconstruction is needed. To choose a suitable smooth surface that resembles the original surface, the moving least squares surfaces (MLS) is used in [12]

B. Surface fitting

Fitting a surface to the original point cloud, e.g., mesh or implicit surface, etc., provides a compact representation of the surface to apply compression techniques in later stages. Much of the earlier work in reducing the size of point clouds has come from the computer graphics community. Many of those approaches achieved compression reducing the number of vertices in triangular or polygonal meshes, for example by fitting surfaces or splines to the meshes. Surveys of many of these methods can be found in [14], [15], and [16].

C. Application of an existing compression technique

Existing compression techniques can be specific to the chosen type of surface representation. Mesh compression techniques have been available for many years, and it is suitable for compression if mesh surface fitting is used . Compression techniques such as [17] and [18] use splat representations that sample and approximate the surface geometry.

D. Simplification of the original point cloud

Simplifying the original point cloud representation into a form that is more easily compressible is another way of providing a compact representation. Resampling can also be effective in creating a more regular point distribution, which often involves down-sampling. Point clustering methods such as splitting the point cloud into a number of subsets, each of which is replaced by one representative sample can be also used. This technique often leads to regular resampling as well. In [19], point clouds were downsampled to a uniform grid, which in turn was partitioned into blocks so that a transform could be directly applied. Some approaches employ special data structures such as octrees and k-d trees. A k-d tree approach uses a tree structure for recursively splitting and subdividing the point cloud [20], [21]. The octree structure provides a multi resolution decomposition of the point cloud, and this approach is especially effective on areas having a large amount of free space. Octree data structures for point cloud compression are explored in [22] and [23]. In [24] and [25] the octree structure for progressive encoding of point clouds is used. A prediction scheme to further improve the octree approach is used in [26] and [27]. More recent work on coding unstructured point clouds in real time is described in [28].

E. Compression exploiting the correlation in the new representation

In this procedure, techniques borrowed from other fields such as statistics or traditional signal/image processing, or just classic data compression methods including prediction, entropy coding, etc., are often used. Significant progress has been made over the past several decades on compressing images and video. Popular standards such as JPEG [29], H.264/AVC [30] and HEVC [31] are in widespread use today. These image and video coding standards also utilize block-based and/or hierarchical methods for coding pixels. Concepts from these image and video coders have also been used to compress point clouds. For example, H.264/AVC was used to compress point clouds in a tele-operation environment [32]. Significant improvements in coding efficiency were reported in [33], in which the PCL-based implementation of [28] was extended to use JPEG to code blocks of attribute values. As mentioned in the previously cited surveys, signal processing based approaches such as wavelets have also been used to compress point clouds. Another recent work that reports significant improvement over the PCL-based implementation of [28] is described in [34], in which a graph transform is applied to blocks of point cloud attributes. In [35], inter-frame compression is introduced to compress dynamic point clouds. Both [34] and [35] use the same codec, applied either to the colors directly or to their prediction residuals, based on an orthogonal graph transform and arithmetic coding of carefully modeled coefficients. The graph transform is a natural choice for the spatial transform of the color signal due to the irregular domain of definition of the signal. Unfortunately, the graph transform requires repeated eigen-decompositions of many and/or large graph Laplacians, rendering the approach infeasible for real-time processing. Other approaches to transforming signals over irregular domains of definition include using shape-adaptive transforms, e.g., [36], or padding the signal to a regular domain and then using an ordinary block transform, e.g., [37].

IV. STANDARDIZATION OF 3D POINT CLOUD COMPRESSION

Advanced 3D representations of the world are enabling more immersive forms of interaction and communication, and they allow machines to understand, interpret and navigate our world. 3D point clouds have emerged as an enabling representation of such information. Compression technologies are needed to reduce the amount of data required to represent a point cloud. We have identified some use cases in the second section of the paper,. To make these applications wide spread, a common method for compression is necessary for interaction. Under ISO/IEC JTC1/SC29/WG11 known as the Moving Picture Experts Group or MPEG, the necessity of standardization of 3D point cloud compression was firstly discussed in October, 2014 [38]. The first target application of 3D point cloud compression was 3D Tele-Immersion, and the group then expanded the potential applications to wider scope including 3D free viewpoint sport replay broadcasting, geographic information systems, cultural heritage preservation and large-scale 3D dynamic maps for autonomous navigation [1]. After more than two years of investigating new coding tools for static and dynamic 3D point clouds, evidence has shown that improving coding efficiency with respect to existing solutions is possible, so MPEG issued a call for proposals on new coding tools for 3D point cloud compression in January, 2017 [39]. Companies and organizations were invited to submit proposals in response to this Call for Proposals by October, 2017.

A. Test Categories and Conditions

There are three test categories and three test conditions planned to be evaluated for the responses to this CfP:

- Test categories
  - ➢ Category 1: Static Objects and Scenes
    Static 3D point clouds
  - ➢ Category 2: Dynamic Objects
    Dynamically changing objects
  - ➢ Category 3: Dynamic Acquisition
    Dynamically acquired scenes

- Test conditions
  - ➢ Lossless geometry and no attributes
    The decoded compressed geometry values are numerically identical to the uncompressed values, and there no attribute values are compressed. This test condition applies to some parts of categories 1 and 3.

  - ➢ Lossless geometry and lossy attributes
    The decoded compressed geometry values are numerically identical to the uncompressed values, and the decoded compressed attributes values are not necessarily numerically identical to the uncompressed values. This test condition applies to some parts of categories 1 and 3.

  - ➢ Lossy geometry and lossy attributes
    The decoded compressed geometry and attribute values are not necessarily numerically identical to the uncompressed values. This test condition applies to category 2 and some parts of categories 1 and 3.

B. Evaluation Method

Evaluation of the proposals is one of the essential parts of this standardization activity. Different evaluation methods are to be used depending on the target parameters.

1) Geometry
   The distance from the position of a point in the decoded 3D point clouds to the position of the corresponding point in the original 3D point clouds is used to calculate PSNR. While point-to-point metrics

capture the geometric distances between paired points, this approach neglects the fact that the points often represent surfaces of structures. As such, these distances have limited interpretations where a reduced set of points are to be reconstructed. Although point-to-surface distances employ surface structures to represent the errors between point clouds, they can often fail when it is hard to construct a surface. Additionally, such metrics tend to rely heavily on the specific surface reconstruction method. To overcome these drawbacks, point-to-plane distance is additionally used for evaluation. The point-to-plane distance of a given point b_i in a decoded 3D point cloud, to the corresponding point a_j in the original 3D point cloud, is calculated by the following process as depicted in Fig. 5. The error vector E(i, j) is projected along the normal direction N_j to get a new error vector $\tilde{E}(i, j)$. The point-to-plane distance is calculated as the magnitude of $\tilde{E}(i, j)$.

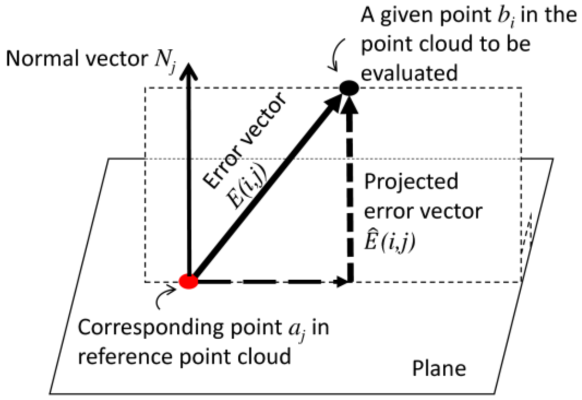

Fig. 5 Illustration of point-to-plane distance

2) Color attributes
Color quality is evaluated using both PSNR and subjective testing.

3) Other attiributes
Reflectance of the laser sensor is evaluated using PSNR.

After evaluating all proposals in October, 2017, the first working draft of the standard is planned to be created in January, 2018, and then additional proposals and evaluations will be discussed. A committee draft of the standard is planned to be issued in July, 2018, and a standard is planned to be finalized in April, 2019.

V. CONCLUSION

In this paper, use cases for 3D point clouds compression and current trends for compression are overviewed. MPEG has started a standardization activity on 3D point cloud compression, which is planned to be finalized in April, 2019. Technologies for acquisition, compression and rendering of 3D point clouds have advanced rapidly in these years, and the market of applications using 3D point clouds is growing quickly. In the near future, more applications such as remote surgery or virtual travel service, etc., can be realized using 3D point cloud compression.

REFERENCES

[1] C. Tulvan, R. Mekuria, Z. Li, and S. Laserre, "Use Cases for Point Cloud Compression (PCC)," ISO/IEC JTC1/SC29/WG11 *Coding of Moving Pictures and Audio*, Geneva, Switzerland, document N16331, Jun. 2016.

[2] "Making virtual Reality Human" Create Volumetric video of real people http://8i.com/

[3] "Microsoft Holoportation" http://research.microsoft.com/en-us/projects/holoportation/

[4] "Real and Virtual engagement in Realistic Virtual Environments" www.reveriefp7.com

[5] Replay technologies http://replay-technologies.com/

[6] Oracle Spatial and Graph https://www.oracle.com/database/spatial/index.html

[7] ArcGis https://www.arcgis.com/features/

[8] PointScene https://pointscene.com/

[9] Culture 3D Cloud http://c3dc.fr/

[10] R.Cohen, D. Tian, M. Krivokuca, K. Sugimoto, A Vetro, K. Wakimoto, and S. Sekiguchi, "Representation and Coding of Large-Scale 3D Dynamic Maps", Proc. SPIE. 9971, Applications of Digital Image Processing XXXIX, pp. 99710T-99710T-8., August 2016.

[11] Mitsubishi Electric Mobile Mapping Systems (MMS), http://www.mitsubishielectric.com/bu/mms/

[12] Automated Driving Systems, October 27, 2015, http://www.sipadus.jp/workshop/program/speaker/profile/dm/koyama.pdf

[13] J. Landa and D. Prochazka, "Automatic road inventory usingS. Fleishman, D. Cohen-Or, M. Alexa, and C. Silva, "Progressive point set surfaces," ACM Trans. On Graphics, Vol. 22, No. 4, pp. 997-1011, 2003.

[14] J. Peng, C.-S. Kim, and C.-C. Jay Kuo, "Technologies for 3D mesh compression: A survey," *J. Vis. Comun. Image Represent.*, vol. 16, no. 6, pp. 688–733, Dec. 2005.

[15] P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes," in *In Advances in Multiresolution for Geometric Modelling*. plus 0.5em minus 0.4emSpringer-Verlag, 2003, pp. 3–26.

[16] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot, "3D mesh compression: Survey, comparisons, and emerging trends," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 44:1–44:41, Feb. 2015.

[17] J.-Y. Sim and S.-U. Lee, "Compression of 3-D point visual data using vector quantization and rate-distortion optimization," IEEE Transactions on Multimedia, vol. 10, no. 3, pp. 305-315, April 2008.

[18] J.-Y. Sim, C.-S. Kim, and S.-U. Lee, "Lossless compression of 3-D point data in QSplat representation," IEEE Transactions on Multimedia, vol. 7, no. 6, pp. 1191-1195, Dec. 2005.

[19] Robert A. Cohen, Dong Tian, and Anthony Vetro, "Point cloud attribute compression using 3-D intra prediction and shape-adaptive transforms," in Proc. 2016 Data Compression Conference, DCC 2016, Snowbird, UT, USA, March 29-April 1 2016.

[20] O. Devillers and P. M. Gandoin, “Geometric compression for interactive transmission,” in IEEE Visualization, 2000, pp. 319 – 326.

[21] P.-M. Gandoin and O. Devillers, “Progressive lossless compression of arbitrary simplicial complexes,” ACM Trans. Graph., vol. 21, no. 3, pp. 372 – 379, July 2002.

[22] J. Elseberg, D. Borrmann, and A. N¨uchter, “One Billion Points in the Cloud – An Octree for Efficient Processing of 3D Laser Scans,” ISPRS Journal of Photogrammetry and Remote Sensing, pp. 76 – 88, 2013.

[23] A. Hornung, K. M. Wurn, M. Bennewitz, C. Stachniss, and W. Burgard,“OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” in Autonomous Robots, 2013.

[24] J. Peng and C.-C. J. Kuo, “Octree-based progressive geometry encoder,” in In Internet Multimedia Management Systems IV. Edited by Smith, John R.; Panchanathan, Sethuraman; Zhang, Tong. Proceedings of the SPIE, 2003, pp. 301 – 311.

[25] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, “Octree-based progressive geometry coding of point clouds,” in Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics, 2006, pp. 103 – 110.

[26] R. Schnabel and R. Klein, “Octree-based point-cloud compression,” in Symposium on Point-Based Graphics 2006, July 2006.

[27] J. Peng and C.-C. J. Kuo, “Geometry-guided progressive lossless 3d mesh coding with octree (ot) decomposition,” ACM Trans. Graph., vol. 24, no. 3, pp. 609 – 616, July 2005.

[28] J. Kammerl, N. Blodow, R. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, “Real-time compression of point cloud streams,” in Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012, pp. 778–785.

[29] ITU-T and ISO/IEC, *Digital compression and coding of continuous-tone still images*. plus 0.5em minus 0.4emITU-T Rec. T.81 and ISO/IEC 10918-1 (JPEG), Sept. 1992.

[30] ITU-T and ISO/IEC JTC1, *Advanced Video Coding for Generic Audio-Visual Services*. plus 0.5em minus 0.4emITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), May 2003 (and subsequent editions).

[31] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[32] X. Xu, B. Cizmeci, and E. Steinbach, “Point-cloud-based model-mediated teleoperation,” in *Proc. of IEEE Int. Symposium on Haptic Audio-Visual Environments and Games (HAVE)*, Istanbul, Turkey, Oct. 2013.

[33] ISO/IEC JTC1/SC29/WG11, “Point cloud encoder, decoder and comparative results,” ISO/IEC JTC1/SC29/WG11 *Coding of Moving Pictures and Audio*, Geneva, Switzerland, document m35910, Feb. 2015.

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

[35] D. Thanou, P. A. Chou, and P. Frossard, “Graph-based motion estimation and compensation for dynamic 3D point cloud compression,” in Proc. IEEE Int. Conf. Image Process., Quebec City, QC, Canada, Sep. 2015, pp. 3235–3239.

[36] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, “3-D wavelet coding of video with arbitrary regions of support,” IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 9, pp. 1063–1068, Sep. 2001.

[37] R. L. de Queiroz, “On data filling algorithms for MRC layers,” in Proc. IEEE Int. Conf. Image Process., Sep. 2000, pp. 586–589.

[38] ISO/IEC JTC1/SC29/WG11, “MPEG-4 Requirements, version 20 (Sapporo Revision),” ISO/IEC JTC1/SC29/WG11 *Coding of Moving Pictures and Audio*, Sapporo, Japan, document N14662, Oct. 2014.

[39] ISO/IEC JTC1/SC29/WG11, “Call for Proposals for Point Cloud Compression,” ISO/IEC JTC1/SC29/WG11 *Coding of Moving Pictures and Audio*, Sapporo, Japan, document N16732, Jan. 2017.