Tree-Structured Point-Lattice Vector Quantization for 3-D Point Cloud Geometry Compression

Amira Filali, Vincent Ricordel, Nicolas Normand

Titre :

Tree-Structured Point-Lattice Vector Quantization for 3-D Point Cloud Geometry Compression

Auteurs :

Amira Filali, Vincent Ricordel, Nicolas Normand

LS2N Laboratoire, Polytech Nantes, rue Christian Pauc BP 50609, 44306 Nantes Cedex, France

{ amira.filali, vincent.ricordel, nicolas.normand }@univ-nantes.fr

Résumé (*100-200 mots*) :

This paper deals with the current trends of technologies which aim at representing efficiently 3D point cloud, they have become a popular challenge for processing, storing and conveying the data independently of how it was captured. Compressing attributes of 3D point clouds such as geometry, colours or normal directions remain challenging problem, since these signals are unstructured. We propose an adaptive Tree-Structured Point-Lattice Vector Quantization (TSPLVQ), resulting in hierarchically structured 3D content, to improve compression performance for static point cloud.The novelty of the proposed approach lies in adaptively selecting the optimal quantization scheme for the point cloud geometric data, such that its intrinsic correlations can be better exploited.Two quantization modes are dedicated to project recursively the 3D point clouds into a series of embedded truncated cubic lattices. At each step of the process, the optimal quantization mode is chosen according to a rate-distortion criterion in order to achieve the best trade-off between coding rate and geometry distortion, such that the compression flexibility and performance can be greatly improved. To achieve rate-distortion optimization, the core of the proposed method relies on an accurate estimation of the distortion inside each cubic voronoï, between fitting plane and cloud points, taking into account the visual rendering of the 3D object. We experimentally evaluated the interest of the proposed method for geometry compression and visualization tasks.

Mots-clefs (*3 à 5 max.*) :

3D point cloud, Lattice Vector Quantization, Rate-Distortion Optimization, Compression

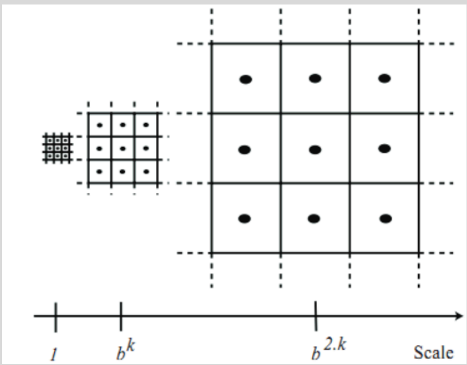
Contexte et état de l'art :

Advances in 360-degree video and virtual/augmented reality fields, provide many fast and reliable cameras, laser scanners, etc. which are rapidly spreading as devices for capturing the 3D data in different ways providing high definition 3D content. The highly increased quality and quantity of the captured data with additional reflectance and color produces very dense 3-D point cloud (PC) data to realistically represent the real world enabling immersive forms of interaction, navigation and communication. The common processing such as storage, transmission from the source data to the diverse platforms, or visualization, etc. become complex, or just impossible. In many industrial domains, it is desirable to transmit this data via restricted bandwidth networks. PCs provide a significant barrier for mass market applications. Thus, MPEG has issued a Call for Proposal (January 2017) for PC compression for its intended applications. Multiple methods of compression have been developed. In [1], Morell et al. have proposed an approach based on plane extraction which represents the points of each scene plane as a Delaunay triangulation and a set of points/area information. The concave hull of a plane is computed to find the borders and get a polygon representing it, an edge reduction is then performed to reduce the number of points needed. Many methods (available in the PCL [2]) are based on Octree decomposition to reduce the amount of data content. The principle is to build a large cuboid to contain all the PC data in a 3-D space. Then the algorithm splits the large cuboid into smaller 3D voxels of same size. For each voxel, the algorithm calculates the mean value of the 3-D coordinates of all the points. The mean value is used to represent all the points that lie in that voxel. In this way, the point cloud can be compressed while keeping the geometric details. In the context of classical images and video encoding applications, Ricordel et al. [4] have proposed a new vector quantizer (TSLVQ) based on truncated lattice embedding. They have investigated its complete design with: the lattice truncation, the multi-stage procedure of quantization, the unbalanced tree-structured codebook design according to a distortion v.s. rate trade-off. The purpose of our method is to reduce the amount of data of a

3D points set but trying to preserve as much information as possible. To the best of our knowledge, we are the first, in the context of 3-D point clouds compression, to propose applying adaptive TSPLVQ with two quantization schemes (2x2x2 or 3x3x3 Voronoï splitting) depending on a rate-distortion criterion which takes into consideration the distortion between points and reconstructed planes associated to points in each cubic Voronoï taking into account the visual rendering of the 3D object and the tree coding cost.

Travail proposé (*Décrire l'objectif du travail et indiquer clairement le problème/défi technique étudié*) :

Background: Tree-Structured LVQ (TSLVQ) is a gathering of many quantization approaches where the quantization is processed through a decision tree. Its benefits are reduced computation with the use of simpler sub-codebooks, and a structure adapted to progressive representation.



TSLVQ [4] aims at using a hierarchical set of embedded lattices which is obtained such as it is possible to embed a lower scale truncated lattice into a cell of the next higher scale truncated lattice. So a scaling factor ***b*** between successive truncated lattices of the hierarchy has to be set (see Fig. 1). The use of the cubic lattice permits optimal embedding and fast quantization. A rate-distortion optimization is used in order to choose which node to split in the iterative tree design.

Fig.1. Hierarchy principle corresponding to the cubic lattice.
Here in 2D, the scaling factor equals b=3 such as a cubic Voronoï is splitted in (3x3) [4].

Proposed Work: The point cloud is highly complex and unstructured, such that an organization strategy of the point cloud can directly affect the coding performance. The proposed approach, based on the embedding of truncated lattices, permits hierarchical description of the 3-D point cloud through an adaptive tree-structure codebook simplifying its representation and compression. One single splitting mode, such as octree structure (where a cubic voronoï is splitted in 2x2x2), is insufficient to account for the characteristics of all kinds of point cloud geometries. Instead of a single splitting mode, in this work two slitting modes (2x2x2 versus 3x3x3) are employed to better map the point cloud to 3-D grids. We also have to scale and truncate the lattices suitably providing the best rate-distortion trade-off.

The basic quantization procedure used for our method TSPLVQ based on TSLVQ is illustrated in Fig. 2:

(0): Initialization: The points set is first normalized to fit inside the root voronoï cell by centering the points and then by scaling data with the factor: $F = \frac{1}{\sqrt{E_{max}}}$, where E_{max} is the maximal euclidean distance between a point and the center.

(1) and (2): An input vector *x* (its components are 3D point coordinates) inside a given cubic voronoï is scaled by the *b* factor and then shifted, the shifting *U* and factor *b* set up the splitting of the cube either in 2x2x2 (b=2, U=(1/2,1/2,1/2)) or in 3x3x3 (b=3, U=(0,0,0)).

(3): The fast quantization algorithm is then used to produce by rounding the corresponding reproduction vector *Y*.

(4): The output vector is centered to permit the next quantization level.

So to initialise the construction of the tree-strutured codebook that will represent the 3D point cloud geometry, all the cloud points are projected into a first cubic Voronoï (namely the tree root). Next, the basic quantization structure is repeated iteratively from each point of the cloud. At each loop of the greedy process, the cube to split and the adapted splitting method, have to be chosen according to a rate-distortion criterion. Exactly a Lagrangian optimization is then employed at this level, the multiplier associated to each Voronoï (namely tree node) is computed, such as when a node is splitted, the increase in rate is calculated in terms of tree coding cost (here: the entropic cost of the nodes is considered), and the decrease in distortion in terms of geometric distortion. To assess objectively this distortion, we put in competition 2 metrics:

- Point-to-point metric, as the square distance between the lattice point Y_i and all assigned cloud points *X* inside the corresponding Voronoï cell C_i: $D_{P2Point}(X, Y_i) = \sum_{X \in C_i} d(X, Y_i)$

- Point-to-plane metric [3,5], as the square distance between the lattice point Y_i to normals $\overrightarrow{N}_{X_i}$ to the plane associated to the cloud points *X* inside the corresponding cell C_i: $D_{P2Plane}(X, Y_i) = \sum_{X \in C_i} || \overrightarrow{V_d}(X, Y_i) \cdot \overrightarrow{N}_{X_i} ||$

For a given loop of the tree growing process, the rate-distortion optimization is performed locally. It aims at choosing among all the possible splitted nodes, the one which offers the maximal decrease in distortion, and the minimal increase in rate. The codebook process is stopped for instance when the bit budget is reached, and the final tree-structure is then unbalanced. Each occupied leaf node corresponds to a point output. The location of the output representant is set to the average value of the PC points contained within the leaf node.

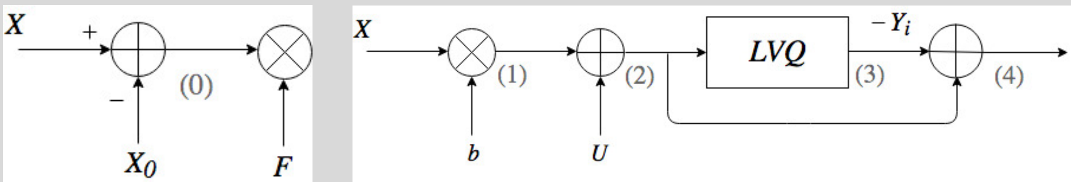


Fig.2. Initialization (left), basic procedure of TSPLVQ where X is the input vector, Y the reproduction vector, b the scaling factor (right).

Résultats

For the experimental results, we compared different voronoï splitting strategies: for the first case only 2x2x2 splitting mode (octree) is used, for the second case only 3x3x3 mode, and for the third case the 2 splitting modes (2x2x2 and 3x3x3) have been put in competition. The performance comparisons between the 3 different schemes are shown in Fig.3. Note that the distortion is assessed according to the 2 metrics that are considered: point-to-point or point-to-plane. For a fair comparison, we set up a parameter to our method, such as the rate in function of leaves number. It can be observed that the splitting strategies leads to different rate-distortion performances, for example, the 2x2x2 splitting scheme is more progressive and requiring more loops for the same number of obtained leaves compared to the two other schemes. The slopes of the curve is a key-point because when it becomes flat, it indicates that further splittings are not necessary. We can note that the slopes behaviors depend on the splitting method and of the used metric, and again the 2x2x2 method seems more interesting than the 3x3x3 method. When we put in competition the 2 splitting methods (called adaptive method), the metric choice is important: with P2Ppoint the 2 splitting methods are combined and the slopes are not flat, with P2Plane no combination are made. At this level of the work, it also seems that the P2Plane metric does not show advantage in term of objective quality evaluation. But it is also important to consider the rendering of the encoded point cloud, Fig.4 presents the rendering coded 3D point cloud in order to analyze and compare their visual quality. Our adaptive method preserves more visual details in the complex areas (for instance the head of the dragon) compared to the two other splitting schemes, whereas it produces less points to represent smooth surfaces (for instance the dragon body surface). Hence our proposed adaptive method using point-to-plane distortion seems more linked with the subjective quality of the rendered decoded object.

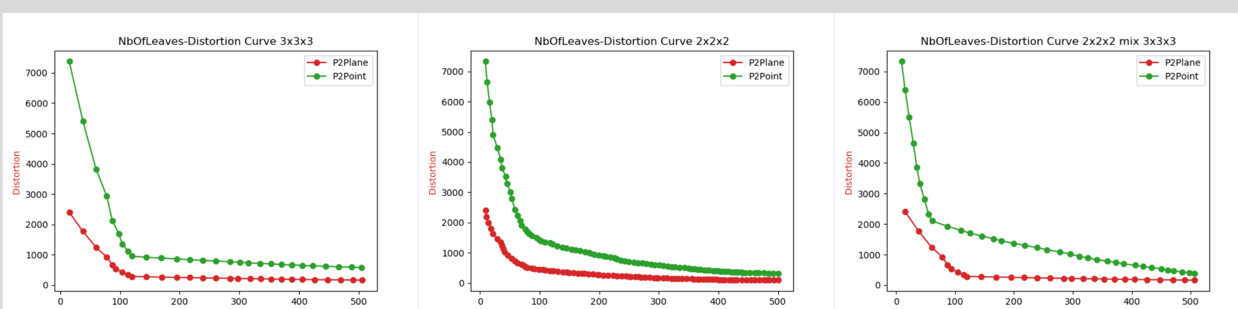


Fig.3. TSPLVQ method using P2Point vs. P2Plane distortion metric Of « Dragon.ply ».

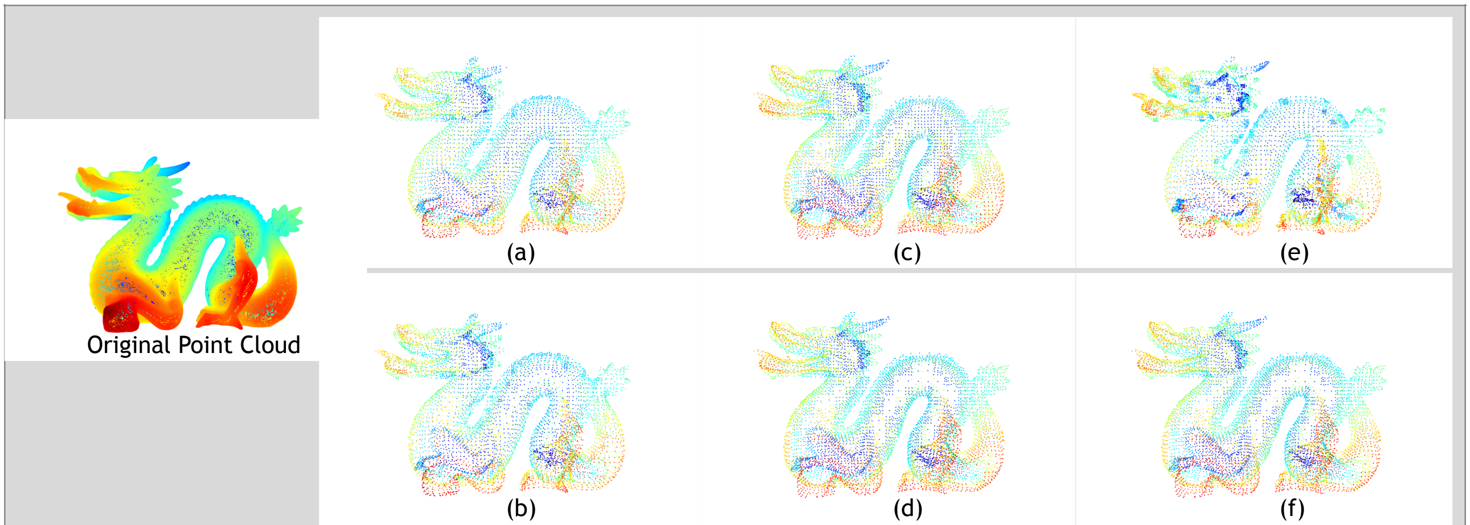


Fig.4. Visual rendering result of Dragon point cloud: (a), (b) only 2x2x2 splitting using P2Point and P2Plane distortion respectively; (c), (d) only 3x3x3 splitting using P2Point and P2Plane distortion respectively; (e), (f) 2x2x2 vs.3x3x3 splitting using P2Point and P2Plane distortion respectively.

Conclusion et perspectives :

Our main research investigated into a compression method of the geometric data of PCs. We have proposed a lossy compression method based on TSPLVQ. Our method innovates because it introduces three splitting strategies (2x2x2 versus 3x3x3) for the cubic voronoï, and by the use of two metrics (P2Point vs. P2Plane). Experimental results show the different behaviours of the rate-distortion curves and different qualities of the rendered contents, their analysis explains the TPSLVQ coding improvement in this context. As future work, we would like to include more accurate geometric primitives and other attributes (as colour) to better characterise the content of a voronoï cell, in order to better render the decoded 3D objects. We also plan to improve the rate computation for the coding of the tree-structure.

Références (*5 max.*):

1. Morell, V., Orts, et al., “Geometric 3D point cloud compression”, Pattern Recognition Letters (2014).
2. Rusu, R. B., Cousins, S., “3D is here: Point Cloud Library (PCL)”, International Conference on Robotics and Automation (2011), Shanghai, China.
3. Podulka, P.: selection of reference plane by the least squares fitting methods. Advances in Science and Technology Research Journal, 10(30), 164-175 (2016).
4. Ricordel, V. and Labit, C., “Vector Quantization by packing of embedded truncated lattices”, IEEE International Conference on Image Processing (ICIP) (1995), Washington.
5. Alireza, J., et al., “Subjective and Objective Quality Evaluation of Compressed Point Clouds”, 19th International Workshop on Multimedia Signal Processing (MMSP) (2017), St. Petersburg, FL, USA.