Implicit Geometry Partition for Point Cloud Compression

Xiang Zhang, Wen Gao, Shan Liu

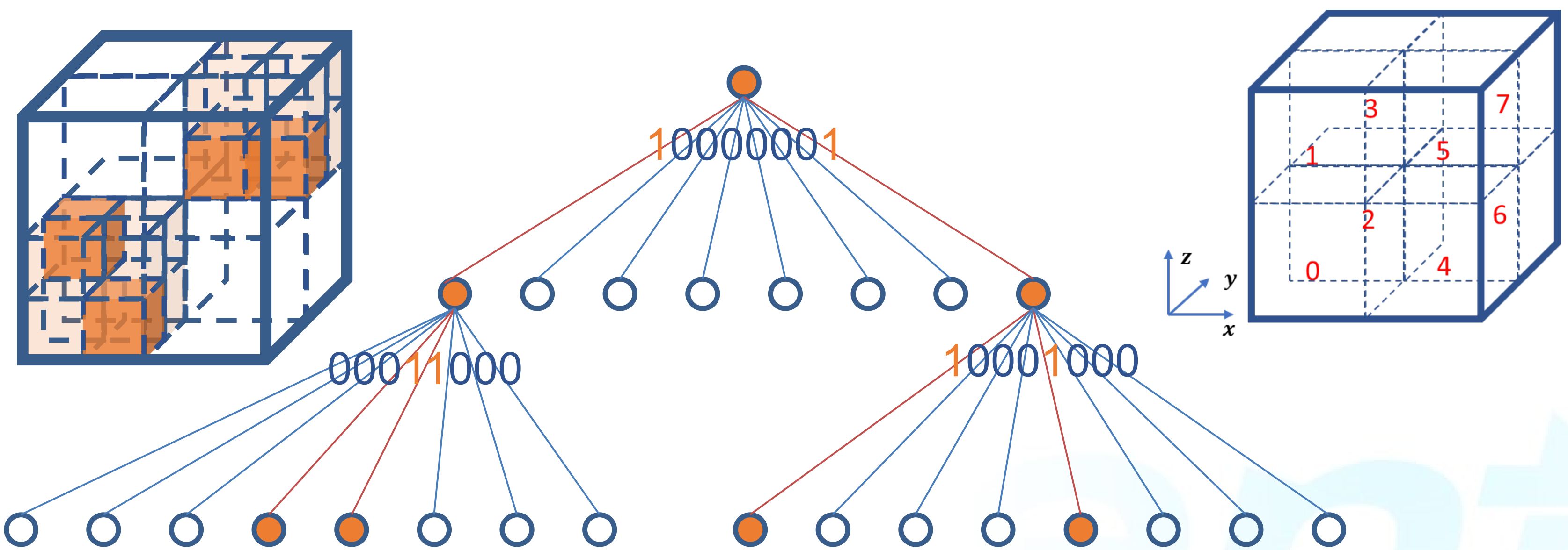
Tencent

03/29/2020

Tencent 腾讯

Background

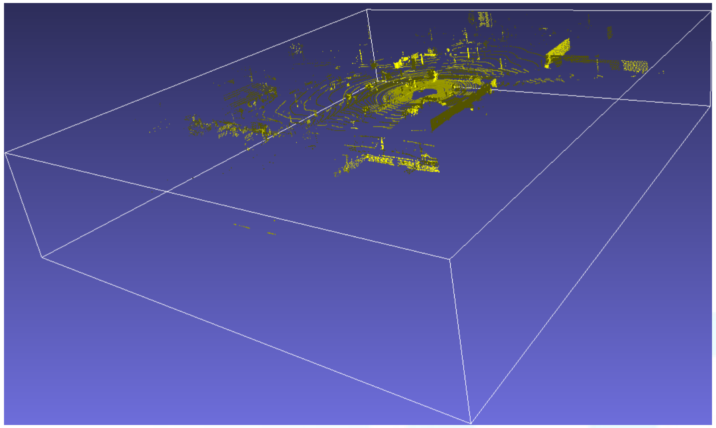
- Octree partition in point cloud compression
  - Start from a $(2^d, 2^d, 2^d)$ cube bounding box
  - Recursively divide until reaching leaf nodes (1,1,1)



[1] TMC13 is the reference software of MPEG G-PCC standard

Motivations

- Symmetric geometry partition may not be the most efficient because of the asymmetric shape of the 3D scene
- Borrow experience from video coding, increasing the partition modes can boost the coding performance



Introducing QT and BT partitions

4 bins can be skipped for QT
6 bins can be skipped for BT

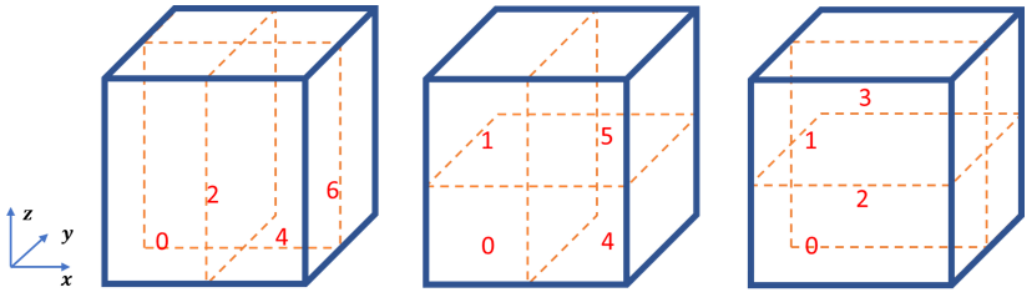


Figure 3: Quad-tree partition of a 3D cube, along x-y, x-z, y-z axes, respectively.

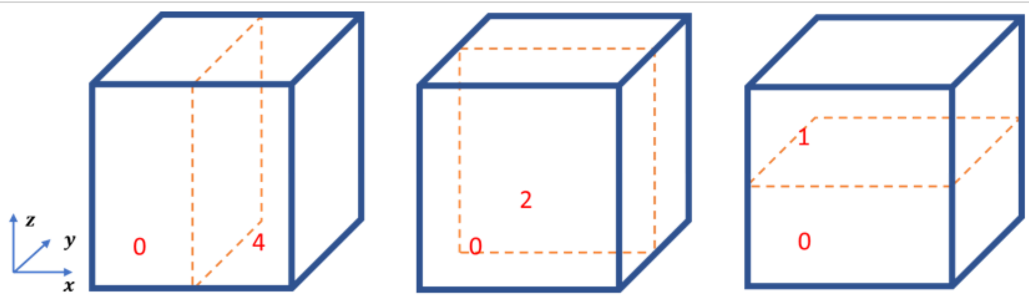


Figure 4: Binary-tree partition of a 3D cube, along x, y, z axis, respectively.

Implicit QT and BT

Introducing 2 parameters K and M

- K $(0 \leq K \leq \max(d_x, d_y, d_z) - \min(d_x, d_y, d_z))$
  - maximum times of QT&BT before OT
- M $(0 \leq M \leq \min(d_x, d_y, d_z))$
  - minimal size of QT&BT, NO QT&BT if $d_x, d_y, d_z \leq M$
- BT is performed before QT (when $d_x \neq d_y \neq d_z$)

Table 1: Conditions of implicit geometry partition for the first K partition depths.

QT along x-y axes	QT along x-z axes	QT along y-z axes
$d_z < d_x = d_y$	$d_y < d_x = d_z$	$d_x < d_y = d_z$
BT along x axis	BT along y axis	BT along z axis
$d_y < d_x$ & $d_z < d_x$	$d_x < d_y$ & $d_z < d_y$	$d_x < d_z$ & $d_y < d_z$

Table 2: Conditions of implicit geometry partition after the first K partition depths.

QT along x-y axes	QT along x-z axes	QT along y-z axes
$d_z = M < d_x = d_y$	$d_y = M < d_x = d_z$	$d_x = M < d_y = d_z$
BT along x axis	BT along y axis	BT along z axis
$d_y = M \leq d_z < d_x$	$d_x = M \leq d_z < d_y$	$d_x = M \leq d_y < d_z$
$d_z = M \leq d_y < d_x$	$d_z = M \leq d_x < d_y$	$d_y = M \leq d_x < d_z$

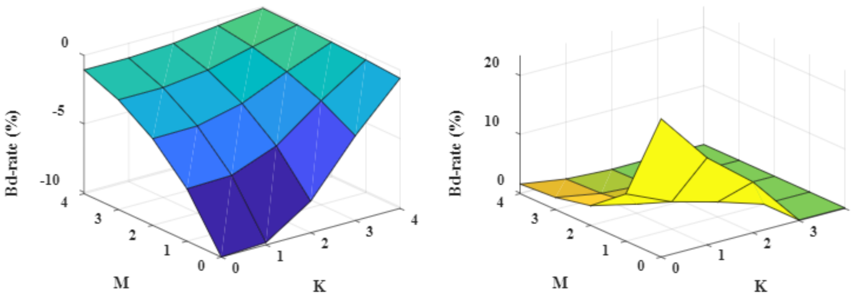
Tencent 腾讯

Examples: $B = (6,5,4)$

- $K = 0, M = 0$: OT -> BT/QT
  - (6,5,4)->(5,4,3)->…->(2,1,0)->(1,1,0)->(0,0,0)
- $K = 2, M = 0$: BT/QT -> OT
  - (6,5,4)->(5,5,4)->(4,4,4)->(3,3,3)->…->(0,0,0)
- $K = 1, M = 0$: BT/QT -> OT -> BT/QT
  - (6,5,4)->(5,5,4)->(4,4,3)->…->(1,1,0)->(0,0,0)
- $K = 1, M = 1$: BT/QT -> OT -> BT/QT -> OT
  - (6,5,4)->(5,5,4)->(4,4,3)->…->(2,2,1)->(1,1,1)->(0,0,0)

Impact of K and M

- Optimal K and M vary in terms of characteristics of point clouds
- Simple decision based on point cloud density
  - Lidar scenes: K=0, M=0
  - VR contents: K=4, M=0



(a) qnxadas-navigating-bends (b) longdress_viewdep_vox12

Figure 5: BD-rate as functions of K and M for two representative sequences.

Table 3: Optimal K and M achieving maximum coding gains on C2 condition.

Sequences	K	M	BD-rate
ulb_unicorn_vox13	4	0	-0.6%
landscape_00014_vox14	0	1	-3.0%
ulb_unicorn_vox20	0	0	-2.0%
citytunnel_q1mm	0	4	-0.9%
overpass_q1mm	0	3	-1.3%
tollbooth_q1mm	0	2	-4.0%

Results

- Anchor: TMC13v7 [1]
- Test Conditions [2]:
  - C2: lossy geometry coding
  - CW: lossless geometry coding
- Test Sequences [2]
  - 7 Lidar sequences in Cat3-frame category

[1] TMC13 is the reference software of MPEG G-PCC standard, https://github.com/MPEGGroup/mpeg-pcc-tmc13
[2] S. Schwarz and D. Flynn, “Common test conditions for point cloud compression,” ISO/IEC JTC1/SC29/WG11 output document N18665, Sep. 2019.

Tencent 腾讯

Lossy Geometry Coding

C2 Sequences	BD-TotGeomRate D1	BD-TotGeomRate D2	BD-TotalRate Reflectance
ford_01_q1mm	-6.6%	-6.6%	-6.3%
ford_02_q1mm	-6.7%	-6.7%	-6.4%
ford_03_q1mm	-7.3%	-7.3%	-7.0%
qnxadas-junction-approach	-8.7%	-8.7%	-8.0%
qnxadas-junction-exit	-8.8%	-8.8%	-8.4%
qnxadas-motorway-join	-9.8%	-9.8%	-9.1%
qnxadas-navigating-bends	-10.1%	-10.0%	-8.9%
Average	**-8.3%**	**-8.3%**	**-7.7%**
Enc Time	99%		
Dec Time	106%		

Lossless Geometry Coding

CW Sequences	Bitrate Ratio Total	Bitrate Ratio Geometry
ford_01_q1mm	95.3%	94.2%
ford_02_q1mm	95.1%	93.9%
ford_03_q1mm	95.6%	94.6%
qnxadas-junction-approach	96.4%	95.8%
qnxadas-junction-exit	96.4%	96.0%
qnxadas-motorway-join	96.3%	95.9%
qnxadas-navigating-bends	97.2%	96.8%
Average	**95.9%**	**95.2%**
Enc Time	101%	
Dec Time	104%	

Tencent 腾讯

Conclusion

- Introducing Quad-tree and Binary-tree partition structure into Octree based geometry coding in PCC
- Bring two parameters to apply implicit QT and BT partitions
- For Lidar data, -8.3% coding gains for lossy geometry and -4.8% bitrate saving for lossless geometry without much complexities
- This method has been adopted to MPEG G-PCC standard in July, 2019 [1,2]

[1] X. Zhang, W. Gao, Y. Sehoon, and S. Liu, "Implicit geometry partition for point cloud coding," ISO/IEC JTC1/SC29/WG11 input document M49231, March 2019
[2] X. Zhang, W. Gao, and S. Liu, "[G-PCC] CE13.22 report on implicit QTBT partition," ISO/IEC JTC1/SC29/WG11 input document M50921, July 2019.

Thanks