# INTRA-FRAME CONTEXT-BASED OCTREE CODING FOR POINT-CLOUD GEOMETRY

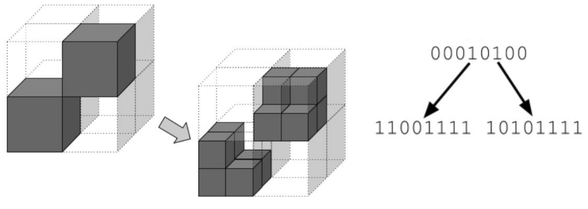
*Diogo C. Garcia and Ricardo L. de Queiroz*
Universidade de Brasilia, Brasilia, Brasil
diogogarcia@unb.br and queiroz@ieee.org

## ABSTRACT

- 3D and free-viewpoint video using point-clouds;
- Lossless intra-frame compression method for point-cloud geometry;
- Derive better contexts for entropy coding from the octree structure;
- Results show that the proposed solution offers state-of-the-art performance, with an average rate reduction of 29% compared to the octree representation.

## INTRODUCTION

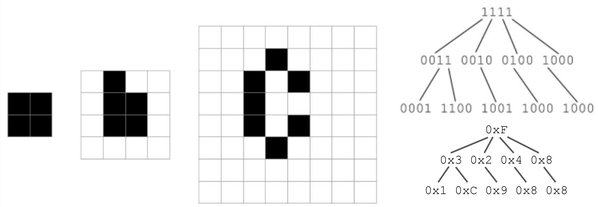
- AR/VR applications ⇒ greater interest in 3D signals (capturing, processing and rendering).
- No established standards.
- We focus on signals captured using a set of RGBD cameras: *voxelized point clouds.*
- Geometry representation: octrees.
  - Data compression;
  - Fast search;
  - Spatial scalability and mixed-resolution scenarios.




- Lossless intra-frame compression method for point-cloud geometry:
  - Entropy-code each octant in the octree according to its father octant.

## OCTREE SCANNING AND CODING

- Recursive division of 3D space in fixed-size cubes.
- Coding through bitwise indication of filled octants, followed by entropy coding.



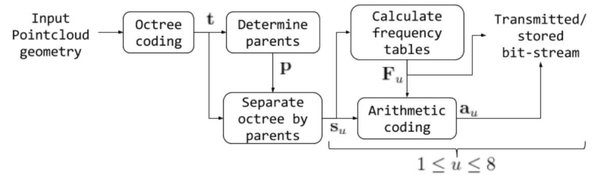

- *Example:* frame 149 in point-cloud sequence *Man*, 27 bpov (bits per occupied voxel).
  - Octree representation: 70994 bytes, 3.13 bpov.
  - LZW-based encoding of octree: 58266 bytes, 2.57 bpov.

## CONTEXT-BASED OCTREE INTRA-FRAME CODING

- Better contexts for the octree **t**'s entropy coding: octree parent values (**v**) and positions (**p**).


$$\mathbf{t} = [0\text{xF } 0\text{x3 } 0\text{x2 } 0\text{x4 } 0\text{x8 } 0\text{x1 } 0\text{xC } 0\text{x9 } 0\text{x8 } 0\text{x8}]$$
$$\mathbf{v} = [0\text{x0 } 0\text{xF } 0\text{xF } 0\text{xF } 0\text{xF } 0\text{x3 } 0\text{x3 } 0\text{x2 } 0\text{x4 } 0\text{x8}]$$
$$\mathbf{p} = [0\text{x0 } 0\text{x1 } 0\text{x2 } 0\text{x3 } 0\text{x4 } 0\text{x3 } 0\text{x4 } 0\text{x3 } 0\text{x2 } 0\text{x1}]$$

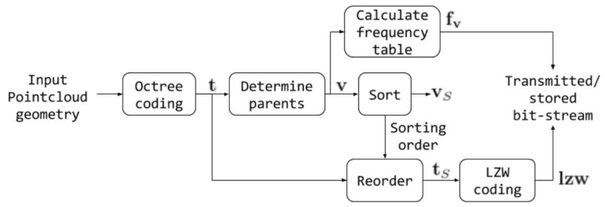
## ARITHMETIC CODING




$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{s}_1 = [0\text{x3 } 0\text{x8}] \qquad \mathbf{s}_2 = [0\text{x2 } 0\text{x8}]$$
$$\mathbf{s}_3 = [0\text{x4 } 0\text{x1 } 0\text{x9}] \qquad \mathbf{s}_4 = [0\text{x8 } 0\text{xC}]$$

## LZW CODING




$$\mathbf{v}_S = [0\text{x0 } 0\text{x2 } 0\text{x3 } 0\text{x3 } 0\text{x4 } 0\text{x8 } 0\text{xF } 0\text{xF } 0\text{xF } 0\text{xF}]$$
$$\mathbf{t}_S = [0\text{xF } 0\text{x9 } 0\text{x1 } 0\text{xC } 0\text{x8 } 0\text{x8 } 0\text{x3 } 0\text{x2 } 0\text{x4 } 0\text{x8}]$$
$$\mathbf{f}_\mathbf{v} = [0\ 1\ 2\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 4].$$

( $\mathbf{v}_S$ sorting order increases symbol repetition in $\mathbf{t}_S$ )

## EXPERIMENTAL RESULTS

**Table I**. Proposed testing scenarios.

| | |
|---|---|
| **MPEG** | MPEG anchor code [13] |
| **Draco** | Draco open-source library [15] |
| **OR** | Octree representation |
| **AC** | Arithmetically-coded octree representation |
| **P(AC)** | Arithmetic-code-based proposed method |
| **LZW** | LZW-coded octree representation |
| **P(LZW)** | LZW-based proposed method |

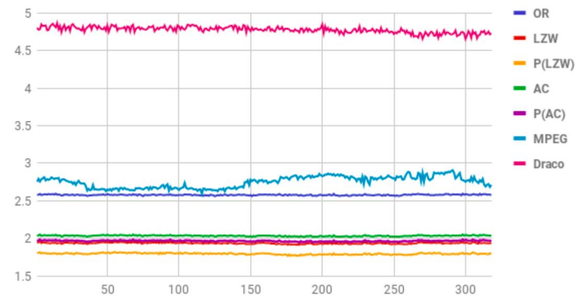
(**LZW** ⇒ GZIP from Keka application, version 1.0.7)

**Table II**. Average rate in bits per occupied voxel for the scenarios proposed in Table I.

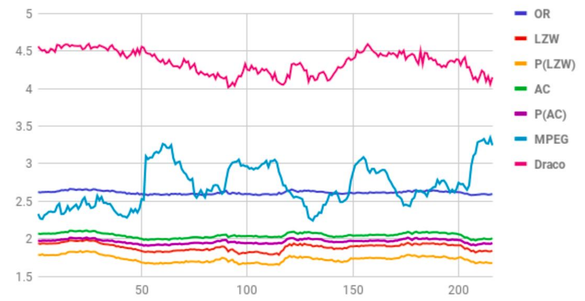
| Sequence | MPEG | Draco | OR | AC | P(AC) | LZW | P(LZW) |
|---|---|---|---|---|---|---|---|
| *Andrew9* | 2.75 | 4.77 | 2.58 | 2.04 | 1.97 | 1.94 | 1.80 |
| *David9* | 2.71 | 4.35 | 2.62 | 2.05 | 1.97 | 1.89 | 1.73 |
| *Man* | 4.49 | 5.04 | 3.17 | 2.66 | 2.60 | 2.52 | 2.32 |
| *Phil9* | 2.89 | 4.54 | 2.64 | 2.10 | 2.03 | 2.00 | 1.84 |
| *Ricardo* | 3.53 | 4.82 | 2.92 | 2.43 | 2.37 | 2.37 | 2.27 |
| *Sarah9* | 3.10 | 4.86 | 2.61 | 2.04 | 1.97 | 1.89 | 1.75 |
| *Average* | 3.25 | 4.73 | 2.76 | 2.22 | 2.15 | 2.10 | 1.95 |

**Table III**. Average rate gain over the octree representation for the scenarios proposed in Table I.

| Sequence | MPEG | Draco | AC | P(AC) | LZW | P(LZW) |
|---|---|---|---|---|---|---|
| *Andrew9* | -7% | -85% | +21% | +24% | +25% | +30% |
| *David9* | -3% | -66% | +22% | +25% | +28% | +34% |
| *Man* | -42% | -59% | +16% | +18% | +21% | +27% |
| *Phil9* | -9% | -72% | +21% | +23% | +24% | +30% |
| *Ricardo* | -21% | -65% | +17% | +19% | +19% | +22% |
| *Sarah9* | -19% | -87% | +22% | +25% | +28% | +33% |
| *Average* | -17% | -72% | +20% | +22% | +24% | +29% |




Rate on a frame basis, in bpov, for seq. *Andrew9*




Rate on a frame basis, in bpov, for seq. *David9*

## CONCLUSIONS

- Lossless intra-frame compression method for point-cloud geometry;
- Different contexts for entropy coding, based on the octree structure.
- Results for experiments with six point-cloud sequences showed that the proposed method offers an average rate reduction of 29%, when compared to the octree representation, yielding a 5% gain over entropy encoding the octree.
- We believe the proposed method is the new-state-of-the-art in intra-frame geometry compression for point-clouds.

## ACKNOWLEDGEMENT

- Work partially supported by CNPq under grant 308150/2014-7.