# EECE5698
# Networked XR Systems

Some of the slides credits: MPEG & Groot paper

# Lecture Outline for Today

- Homework2
- Point Cloud Compression
  - MPEG GPCC
  - MPEG VPCC

# Point Clouds

- A point cloud is a discrete set of data points in space.
- Or a set of 3D independent points
- Each Point (X, Y, Z) + Attributes
- Attributes: Color, Alpha, Reflectance

# Point Cloud

- Representation
  - Each Point is a floating-point number – 32 bits
  - <X, Y, Z> : 96 bits
  - RGB: 3 channels: 24 bits
  - Also, has other attributes sometimes (light related)

  - Each point: 96 + 24 bits or 15 bytes

- Typically, a point cloud has thousands to millions of points – guess the data rate numbers

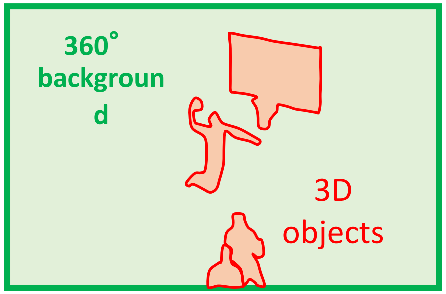
# Point Cloud

Sample data numbers

| | 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 |

# Example Applications



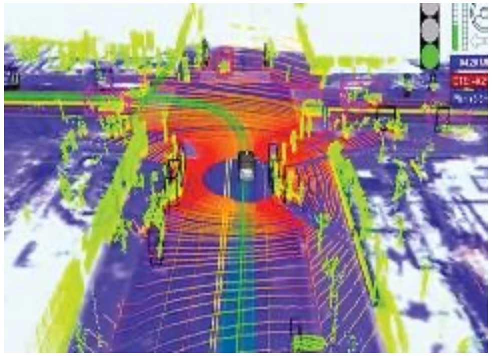


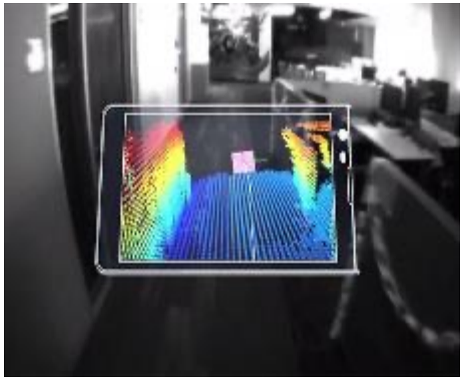

1-3 Gbps per object

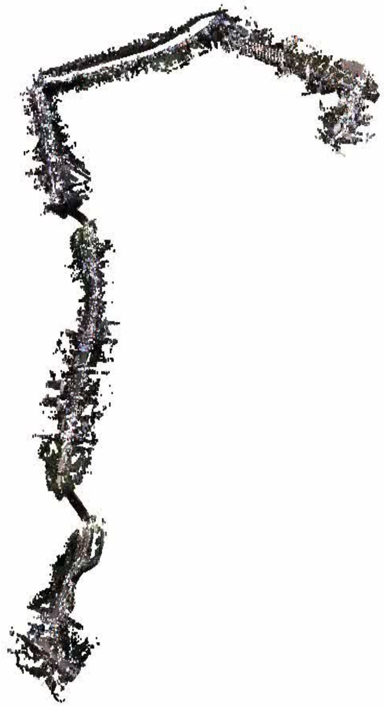
# Example Applications

# Example Applications

- ~20 million points
  - 2,020,734,515 bytes

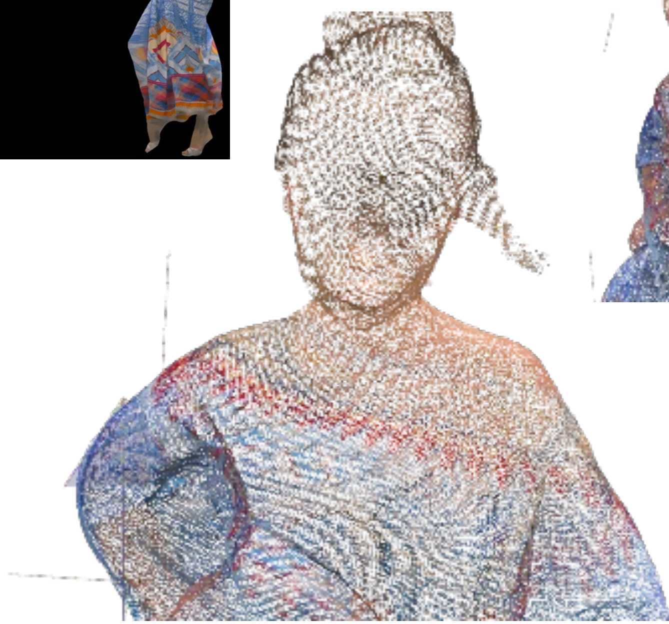


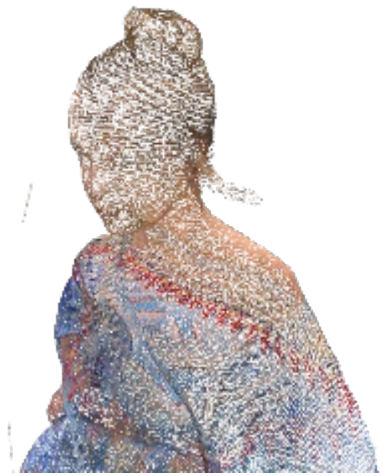




# Point Cloud Compression

800,000 points -> 1 000 Mbps (uncompressed)





Compression is required in order to make PC useful

# Point Cloud Compression

- Can we use similar block based intra and inter frame prediction and transform coding for point clouds?
  - E.g., is it possible to divide the point cloud into 3D blocks and apply similar techniques that we used in case of 2D videos (block matching algorithms etc.)

# MPEG Point Cloud Compression

V-PCC 01/2020

G-PCC 4/2020

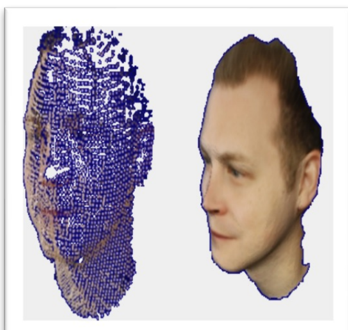
2014 2015 2016 2017 2018 2019 2020

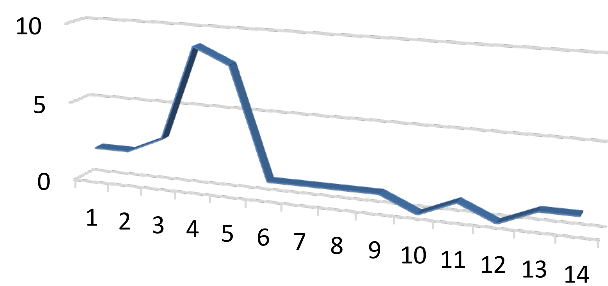
MPEG initiated the work on PCC

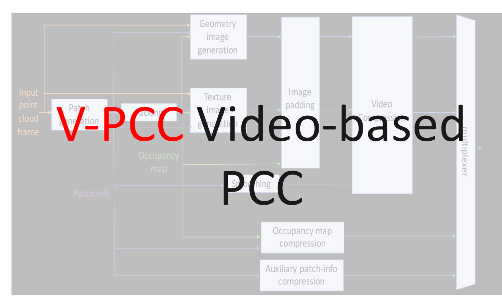
In April 2017 MPEG issued a Call for Proposals

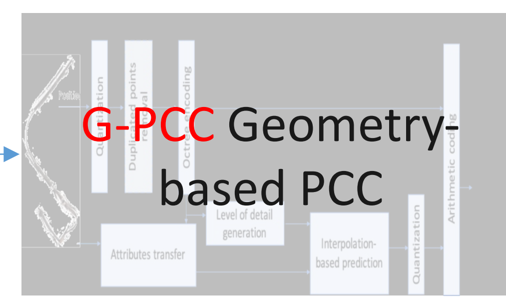
First Committee Draft issued in October 2018

9 technology leading companies responded and MPEG evaluated them in October 2017





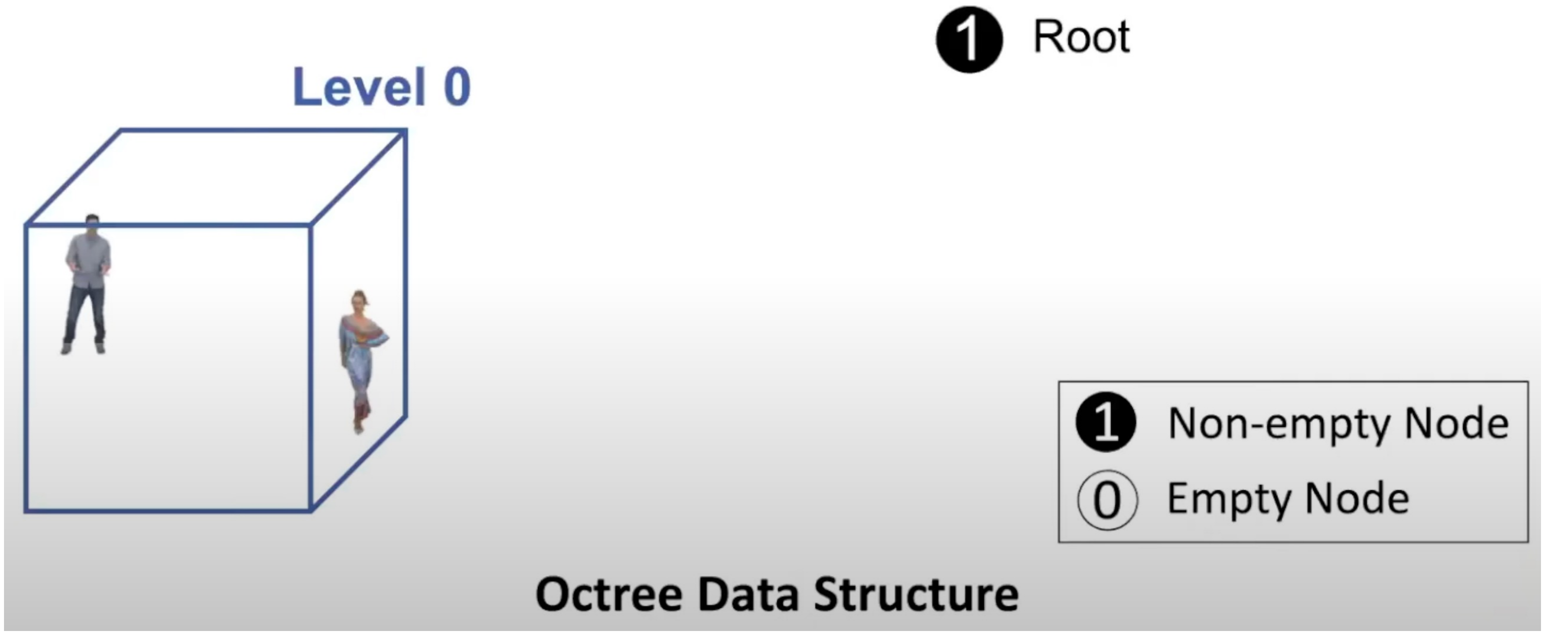



# MPEG GPCC

- Geometry based point cloud compression
  - 3D tree data structures (Octree or KD-tree)

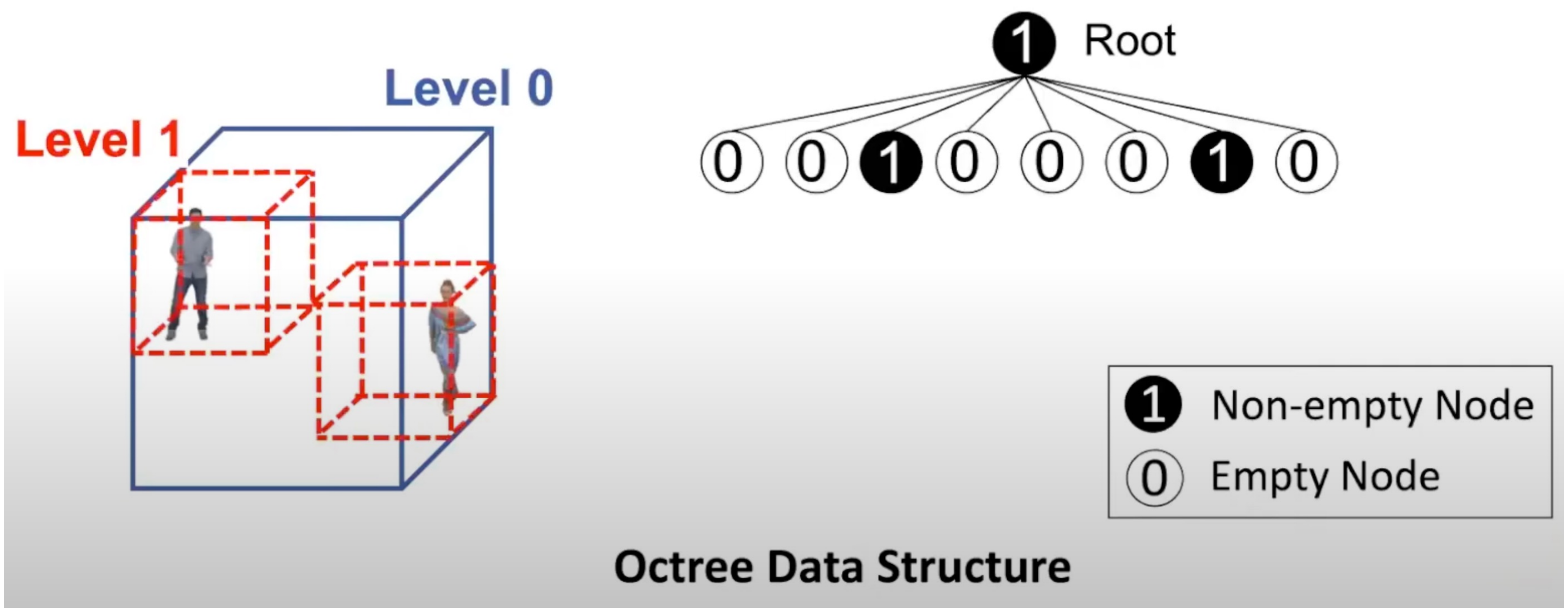
# MPEG GPCC

- Geometry based point cloud compression
  - 3D tree data structures (Octree or KD-tree)



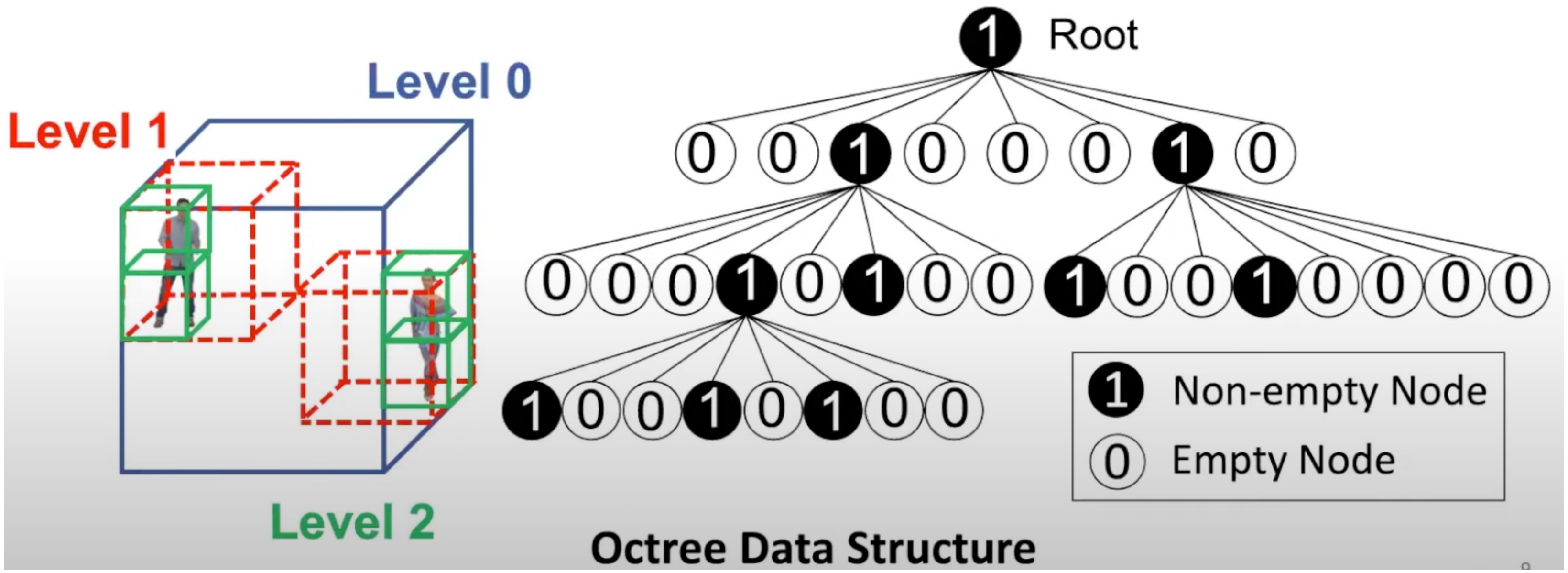
# MPEG GPCC

- Geometry based point cloud compression
  - 3D tree data structures (Octree or KD-tree)

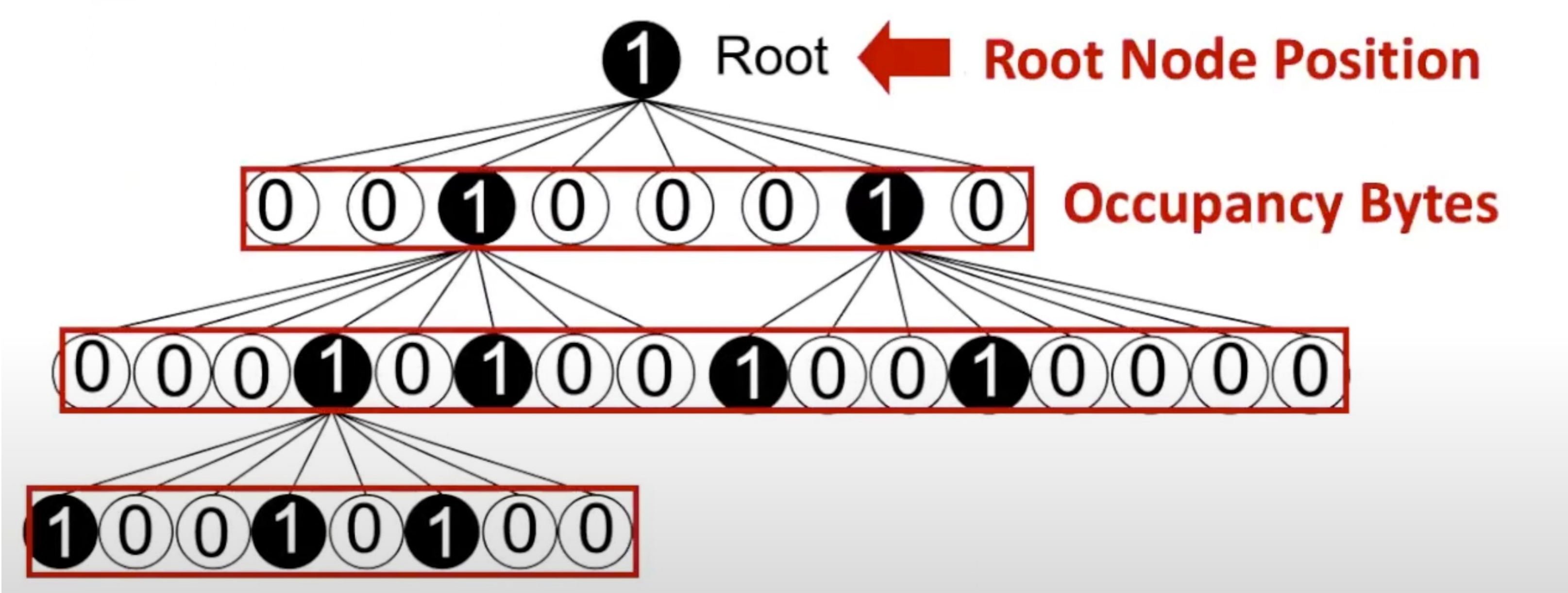


# MPEG GPCC

- Geometry based point cloud compression
  - 3D tree data structures (Octree or KD-tree)

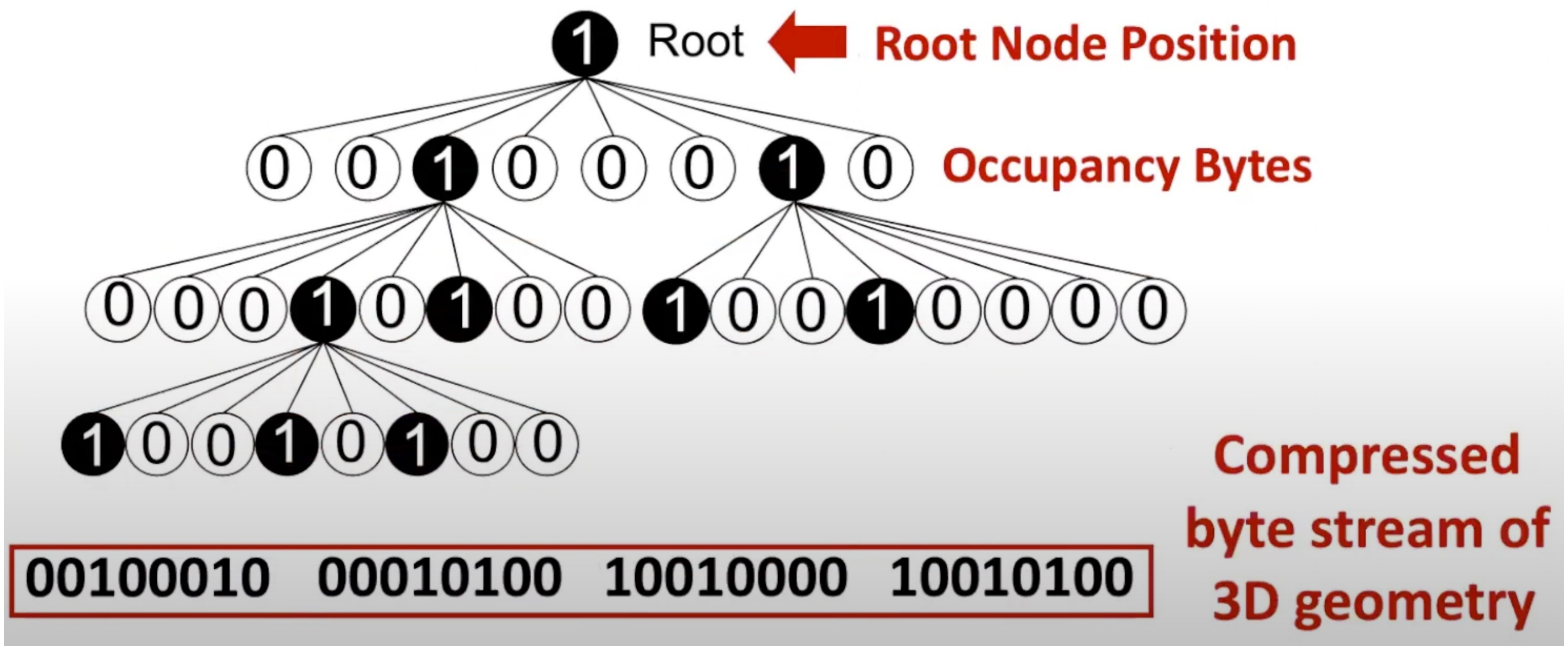


# MPEG GPCC

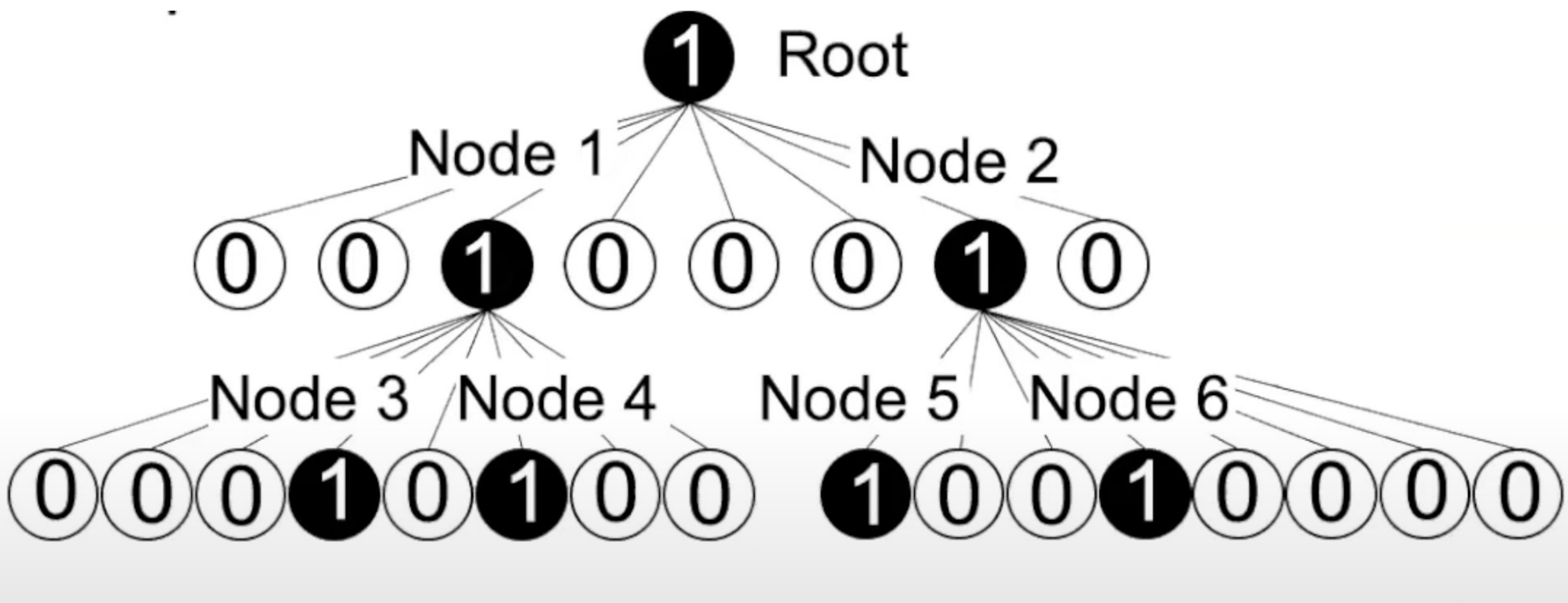


1
Root
Root Node Position
0 0 1 0 0 0 1 0
Occupancy Bytes
0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0
1 0 0 1 0 1 0 0

# MPEG GPCC

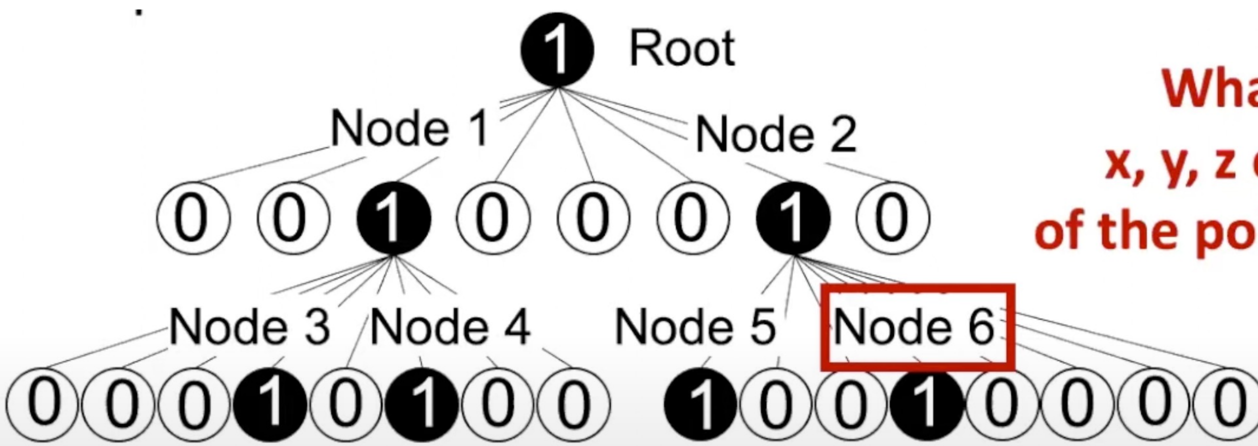

1
Root
Root Node Position
0 0 1 0 0 0 1 0
Occupancy Bytes
0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0
1 0 0 1 0 1 0 0
Compressed byte stream of 3D geometry
00100010 00010100 10010000 10010100

# MPEG GPCC



1
Root
Node 1
Node 2
0 0 1 0 0 0 1 0
Node 3
Node 4
Node 5
Node 6
0 0 0 1 0 1 0 0
1 0 0 1 0 0 0 0

# MPEG GPCC

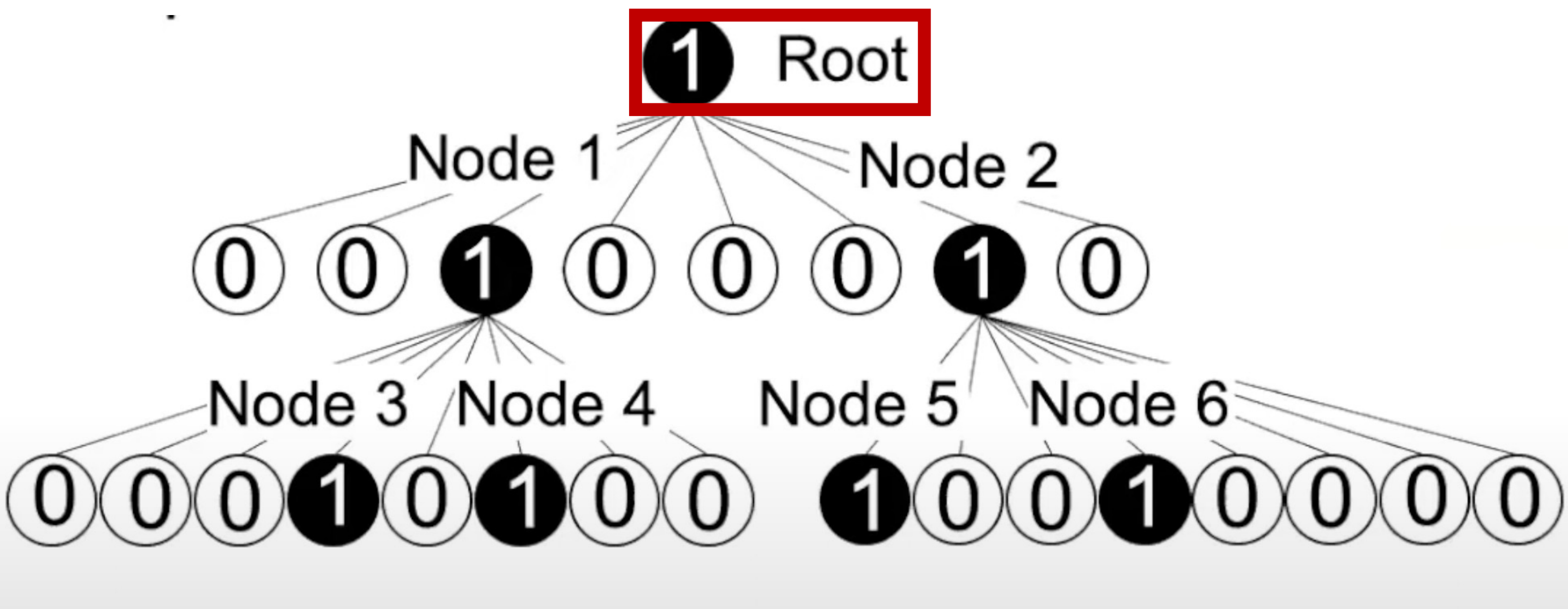



**What are the x, y, z coordinates of the points in Node 6?**

| | | | | | | |
|---|---|---|---|---|---|---|
| 00100010 | 00100010 | 10010000 | 00010100 | 0001110 | 10000100 | 01110100 |

Compressed Byte Stream

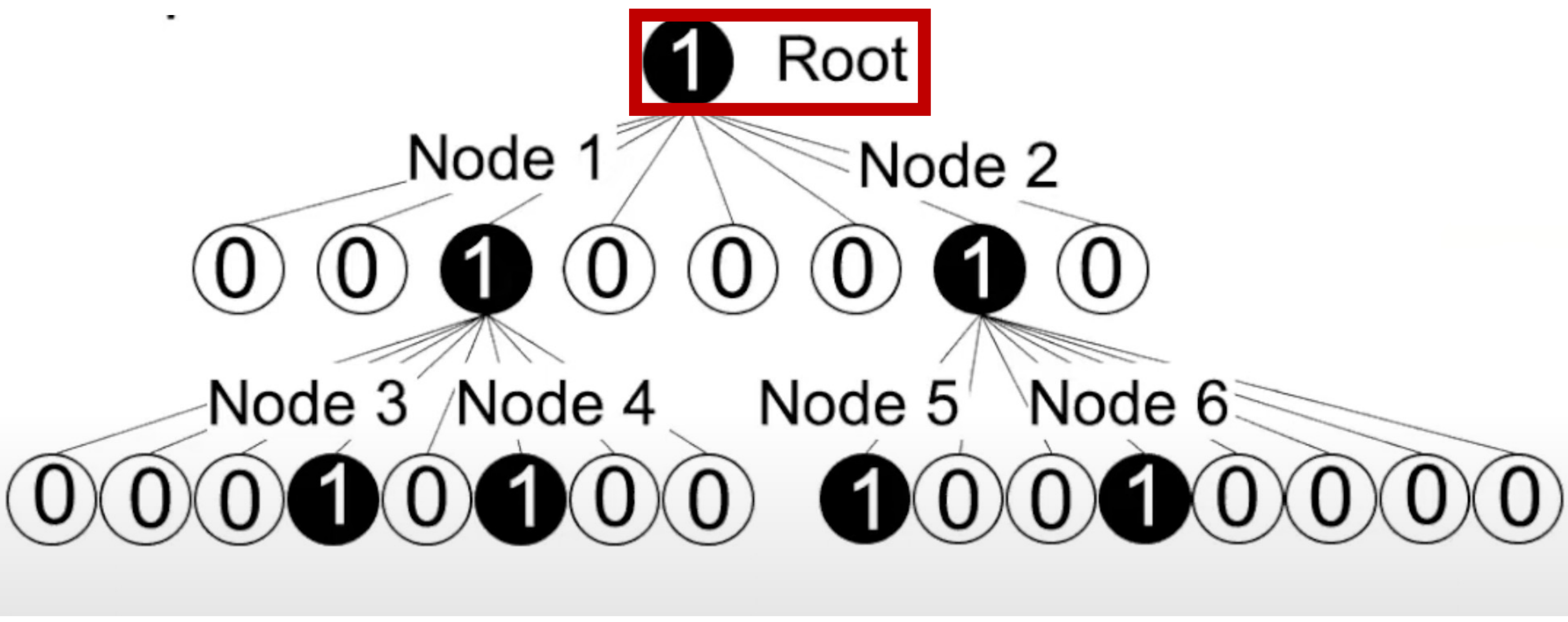
# MPEG GPCC




| Root | Node1 | Node2 | | | | |
|---|---|---|---|---|---|---|
| 00100010 | 00100010 | 10010000 | 00010100 | 0001110 | 10000100 | 01110100 |

Compressed Byte Stream

# MPEG GPCC

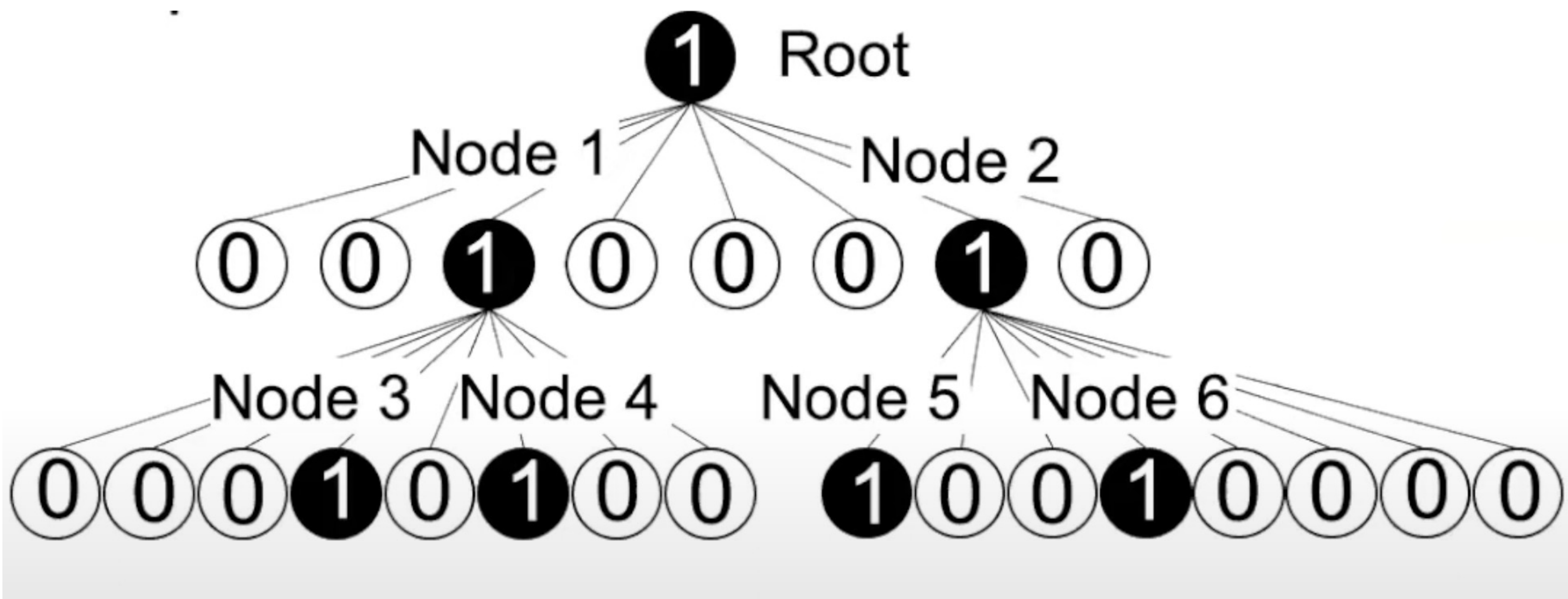



Which one is Node6?

| Root | Node1 | Node2 | | | | |
|---|---|---|---|---|---|---|
| 00100010 | 00100010 | 10010000 | 00010100 | 0001110 | 10000100 | 01110100 |

Compressed Byte Stream

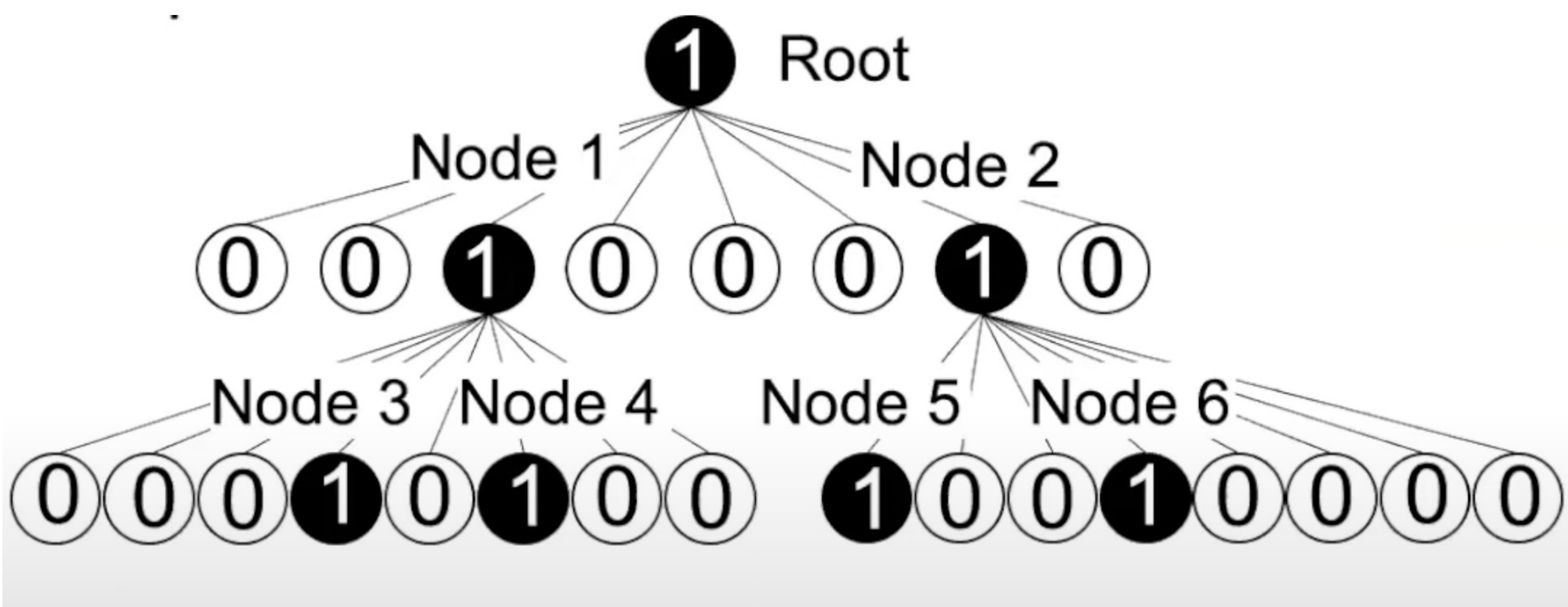
# MPEG GPCC




| Root | Node1 | Node2 | Node3 | Node4 | | |
|---|---|---|---|---|---|---|
| 00100010 | 00100010 | 10010000 | 00010100 | 0001110 | 10000100 | 01110100 |

Compressed Byte Stream

# MPEG GPCC




| Root | Node1 | Node2 | Node3 | Node4 | Node5 | Node6 |
| --- | --- | --- | --- | --- | --- | --- |
| 00100010 | 00100010 | 10010000 | 00010100 | 0001110 | 10000100 | 01110100 |

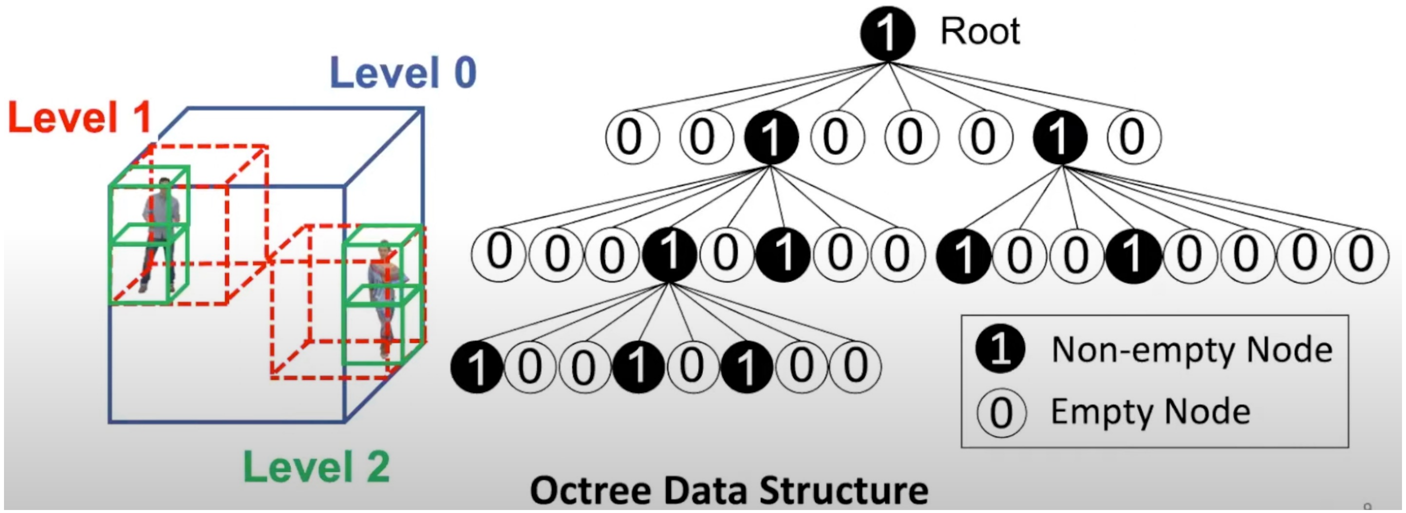
Compressed Byte Stream

# MPEG GPCC

- Problem
  - Generates a dependency between the points – makes parallel processing difficult
  - Computationally expensive

# MPEG GPCC

- Problem
  - Points jump from one branch of the tree to another even with small motion or due to sensor noise
  - Great for static point cloud frames, but problematic temporally



# Point Cloud Error Metrics

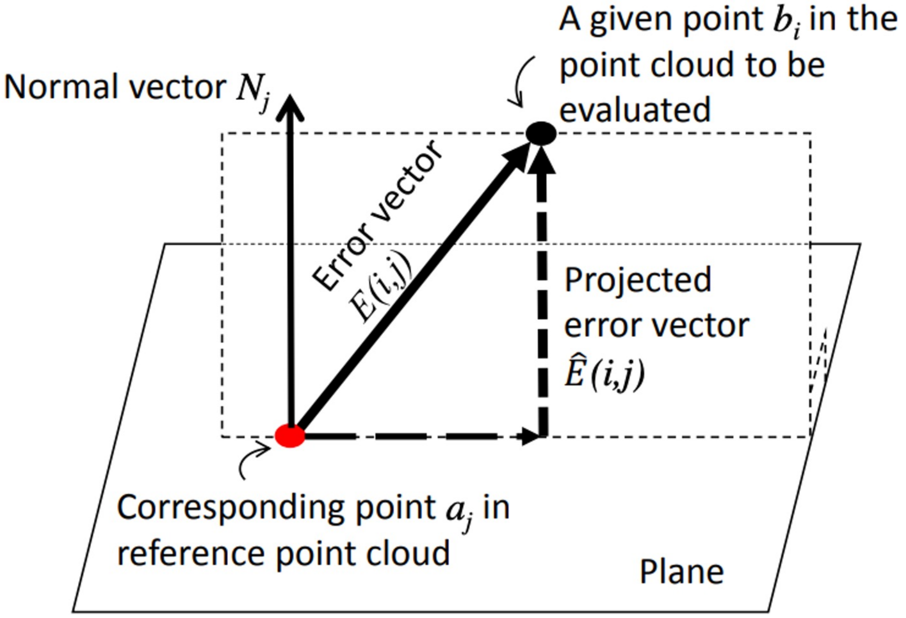
- Point-to-Point
  - Error between nearest neighbor points

$$\frac{1}{N_A} \sum_{\forall a_j \in \mathbf{A}} \|E(i,j)\|_2^2$$

GEOMETRIC DISTORTION METRICS FOR POINT CLOUD COMPRESSION: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8296925

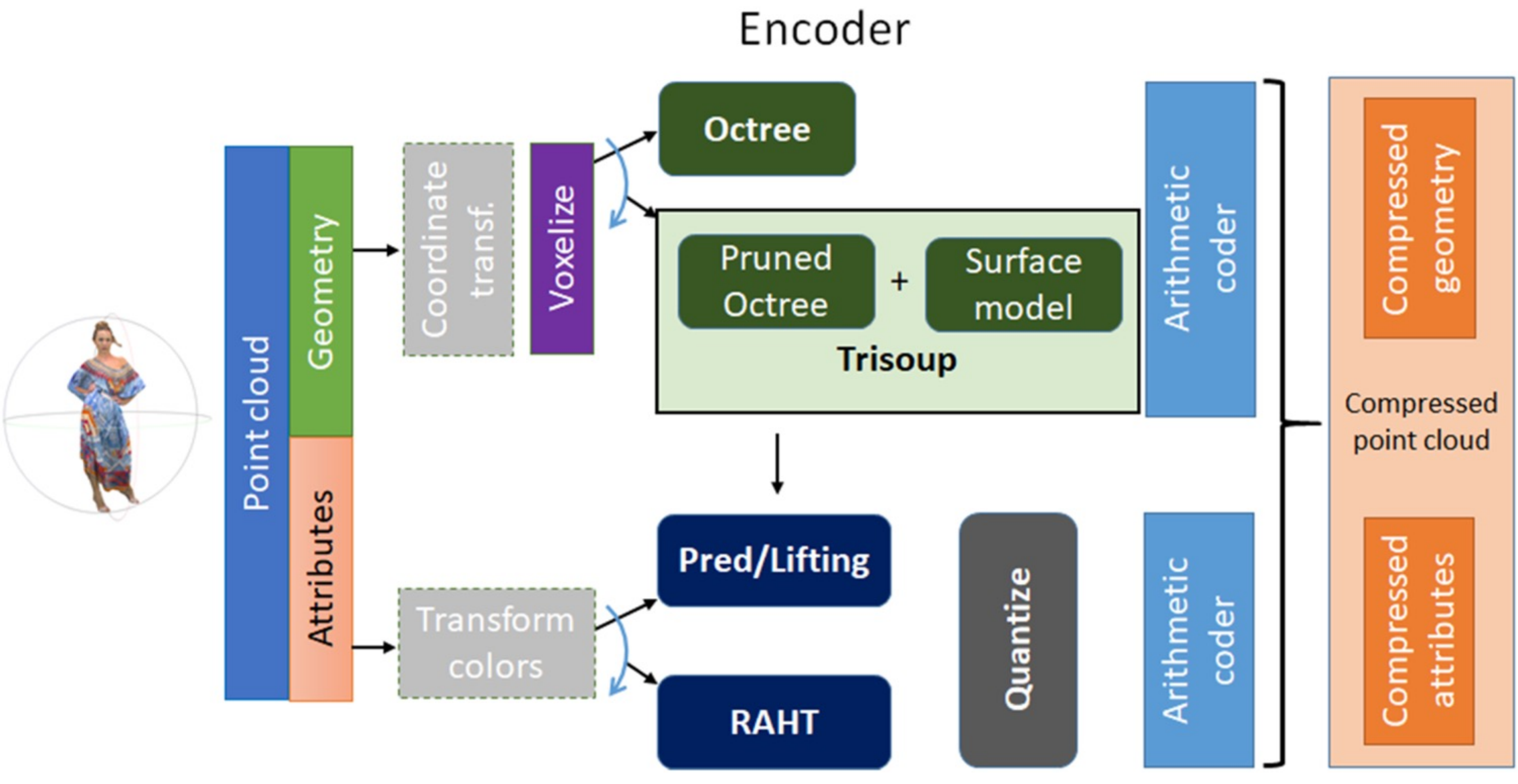
# Point Cloud Error Metrics

- Point-to-Plane
  - Measures error along normal directions
  - More penalty on error that are away from surface




GEOMETRIC DISTORTION METRICS FOR POINT CLOUD COMPRESSION: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8296925

# MPEG GPCC

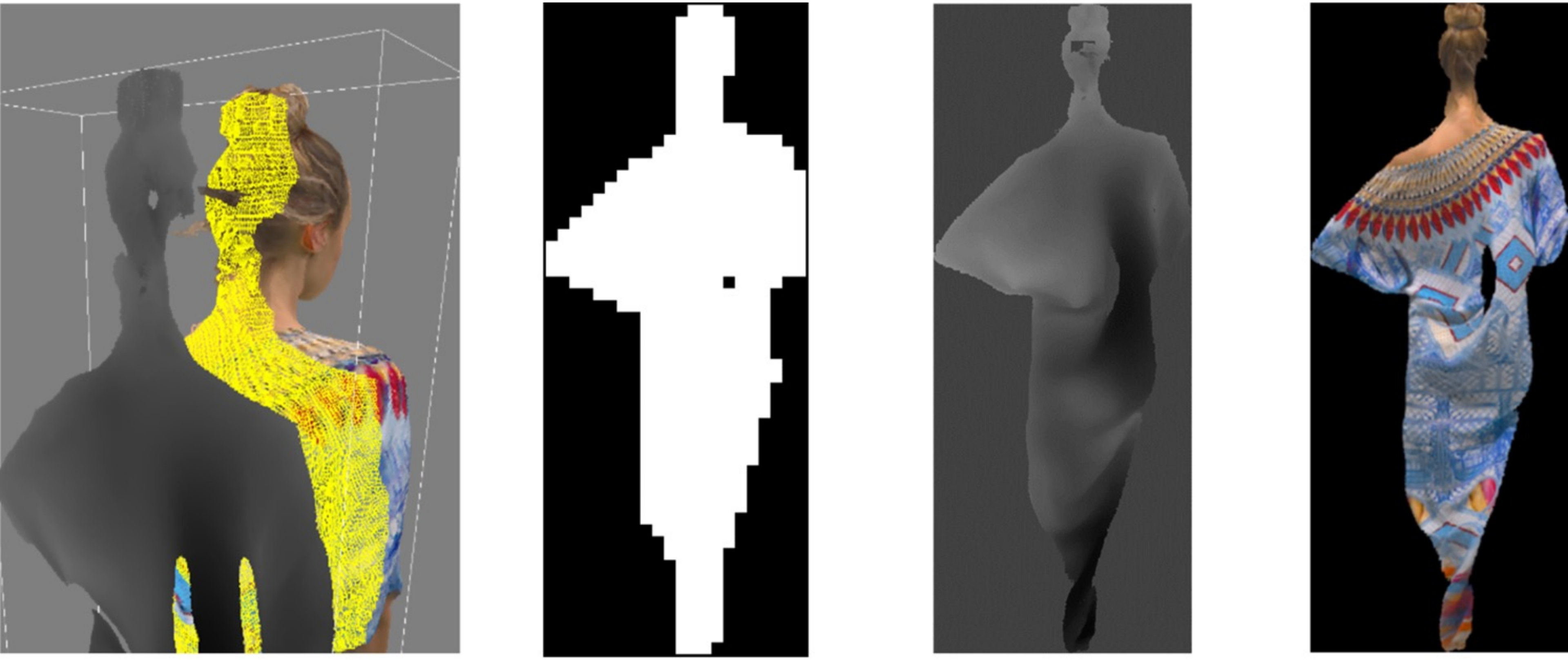



High level architecture

# MPEG VPCC

- Video based point cloud compression
  - Projection based coding – from 3D to 2D
  - Idea: Take advantage of existing video codecs to compress 2D projections

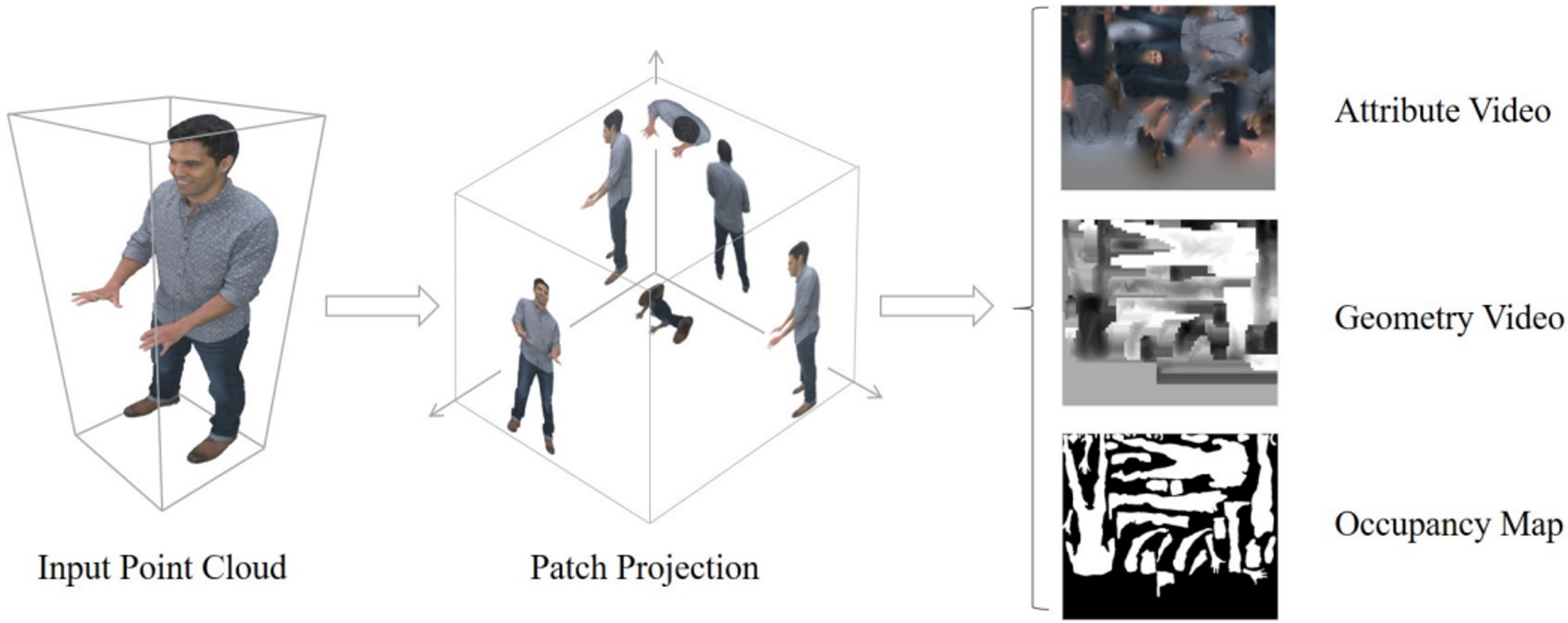
# MPEG VPCC



(a) (b) (c) (d)

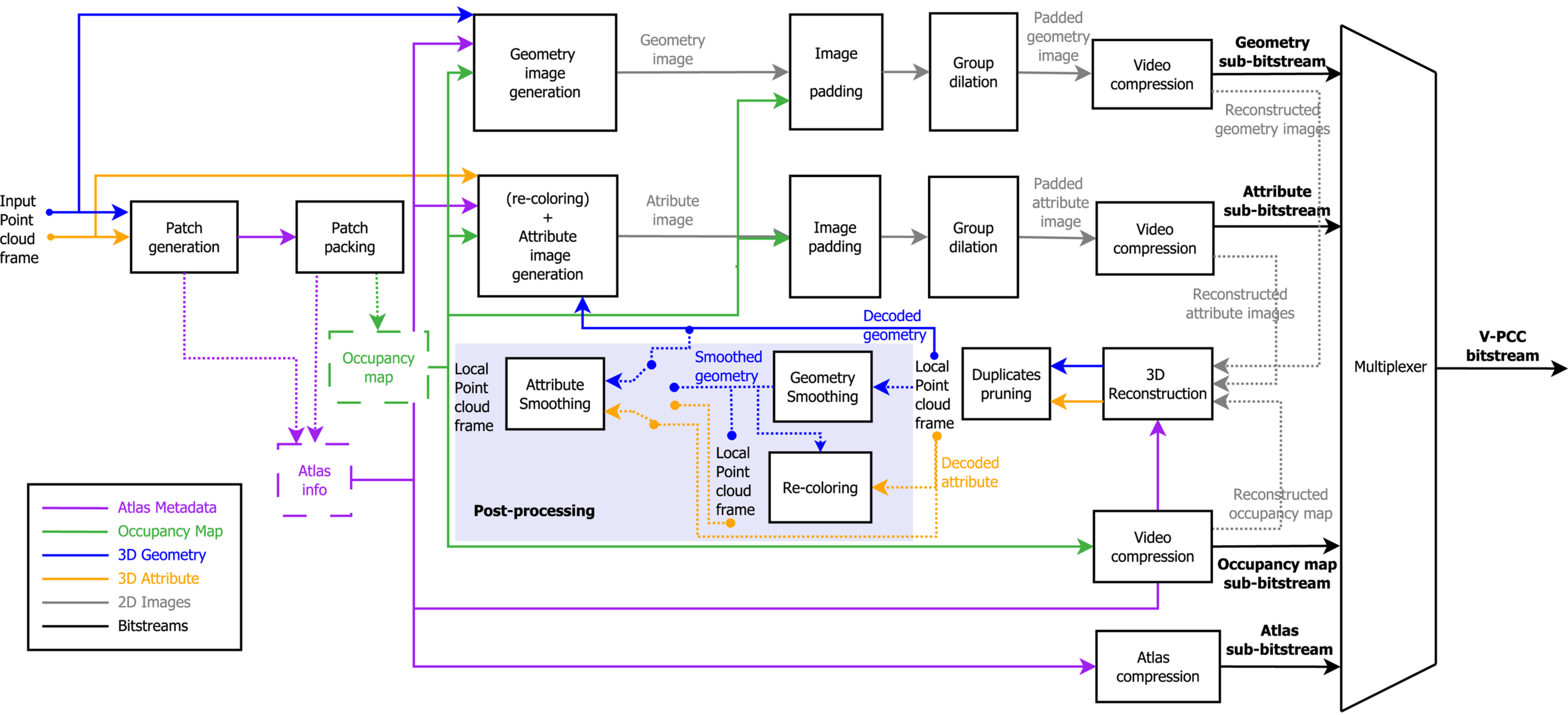
3D Patch projection and respective occupancy map, geometry, and attribute 2D images, (a) 3D patch, (b) 3D Patch Occupancy Map, (c) 3D Patch Geometry Image, (d) 3D Patch Texture Image.

# MPEG VPCC




3 Video streams + 1 additional meta data stream

# MPEG VPCC



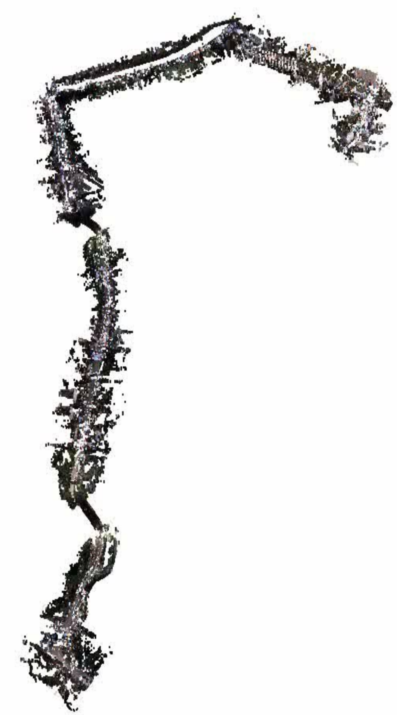

High level architecture

# MPEG VPCC

- Problem
    - Computationally expensive – Patch generation, packing, video generation, compression (4 streams)

# Compressing Large Scale Point Clouds

- Both GPCC and VPCC suffer
  - Need to rely on the other forms of data structures for efficiency



# Summary of the Lecture

- MPEG GPCC
- MPEG VPCC

- Both are computationally expensive – unusable at this point for practical purpose

Next up: Mesh Compression