Compression of point cloud geometry through a single projection

Dion E. O. Tzamarias, Kevin Chow, Ian Blanes, and Joan Serra-Sagristà

Universitat Autònoma de Barcelona

Department of Information and Communications Engineering (DEIC)

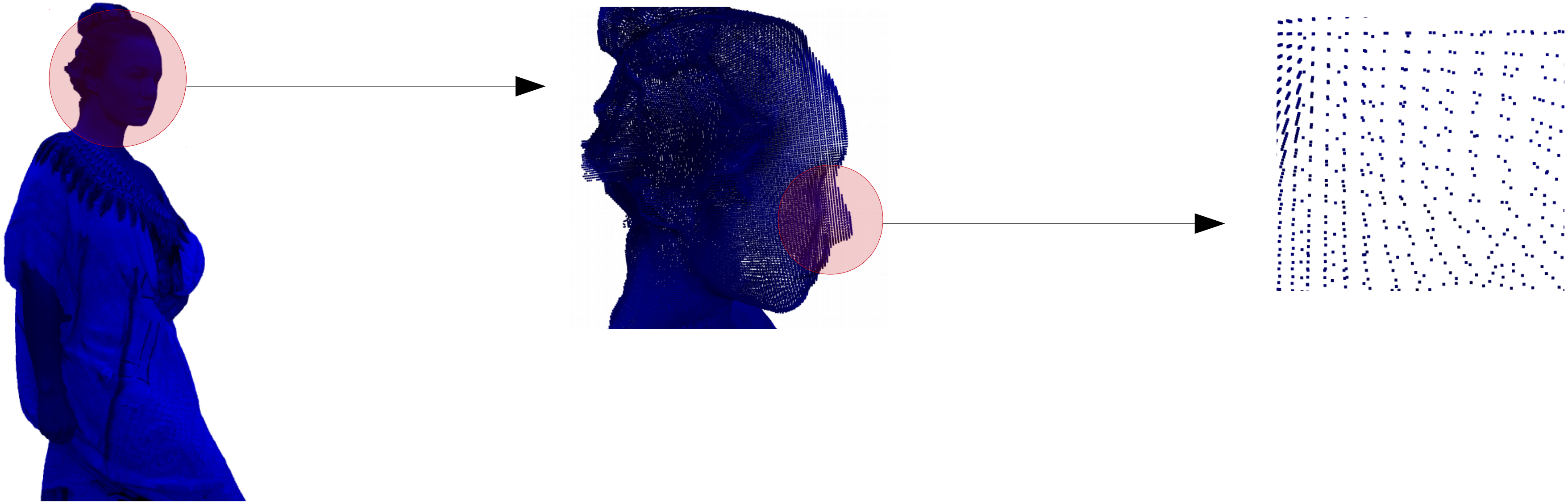
Group on Interactive Coding of Images (GICi)

DCC 2021

Snowbird (UT), USA – March 24 to 25, 2021

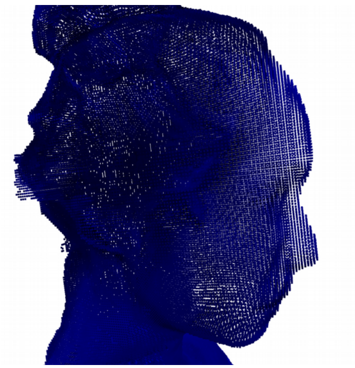
Table of Contents

Introduction-Point clouds

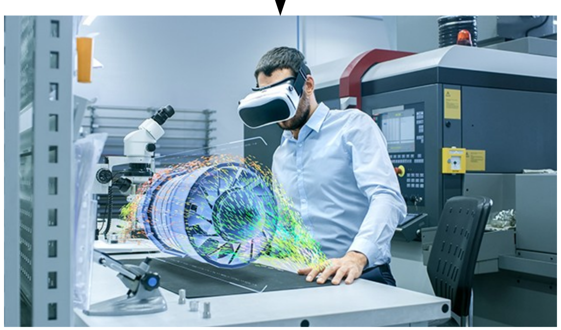


Point clouds are 3D digital representations of objects formed by a collection of points.

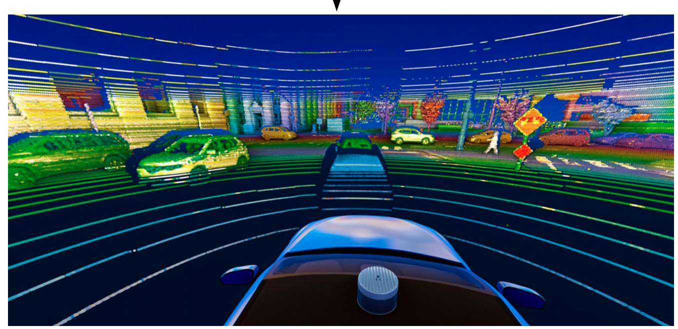
Introduction-Point clouds



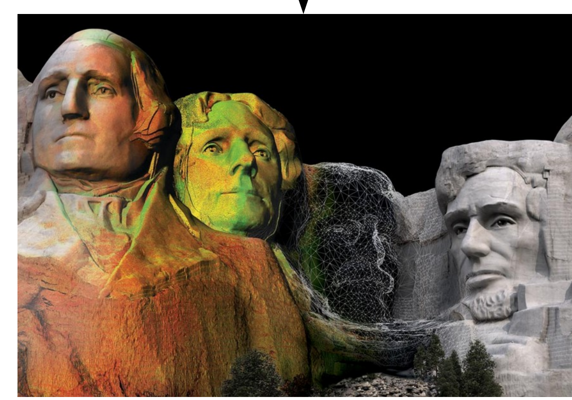
Many applications of point clouds in areas such as:



Virtual/Augmented reality

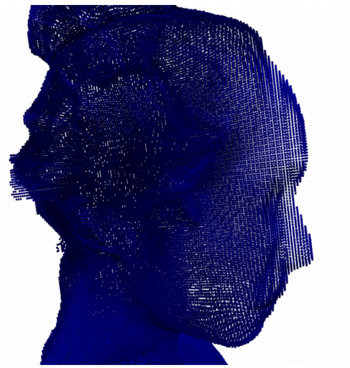


Automated driving

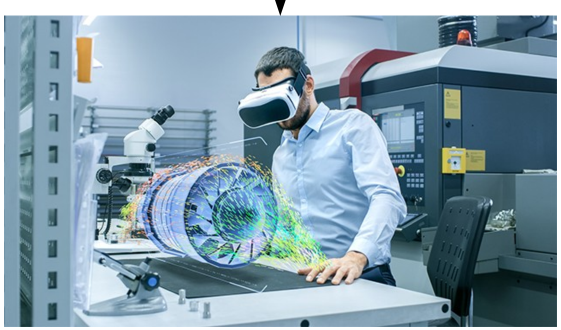


World heritage

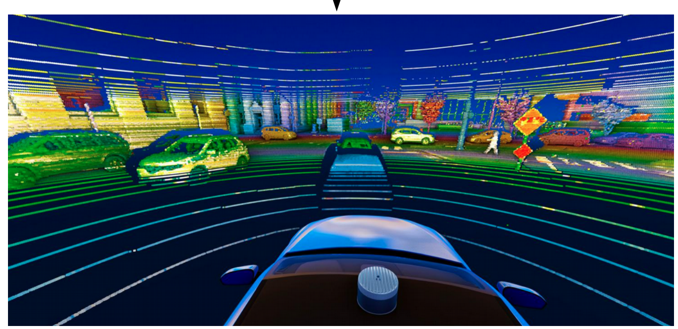
Introduction-Point clouds



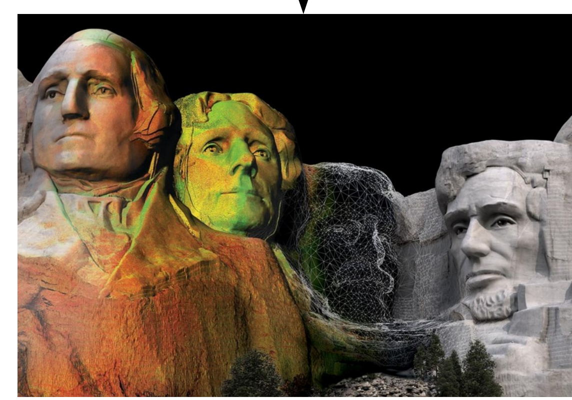
Many applications of point clouds in areas such as:



Virtual/Augmented reality



Automated driving



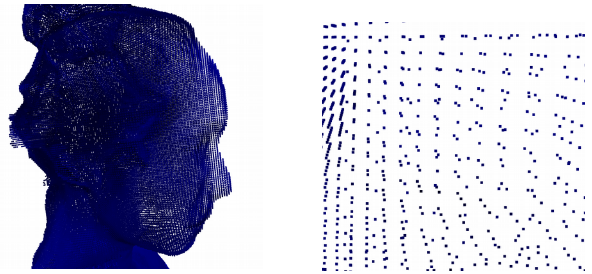
World heritage

Increasing necessity of powerful lossy or lossless compression algorithms.

Introduction-Point clouds

Point clouds

Attributes:
parameters such as color, normals, reflectibility of points.



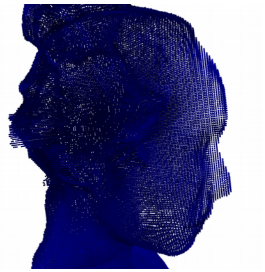
Geometry:
parameters describe the location of points.

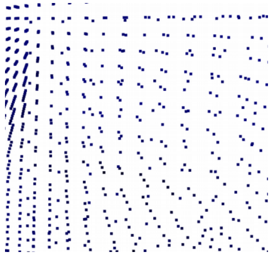
Introduction-Point clouds

Point clouds

Attributes:
parameters such as color, normals, reflectibility of points.

Attribute compression.

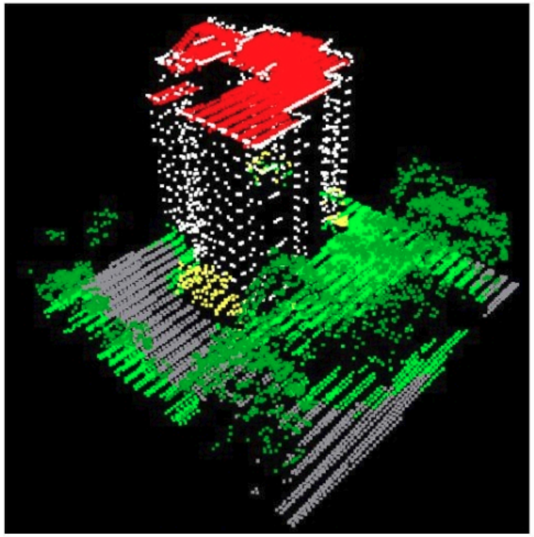




Geometry:
parameters describe the location of points.

Geometry compression.

Introduction-Point clouds

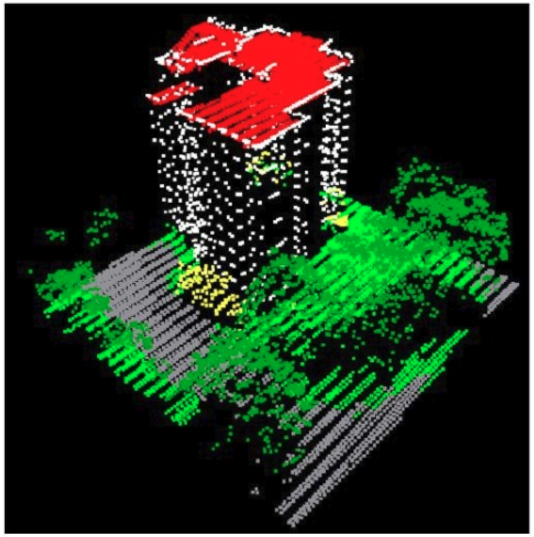


Original

Voxelized

Point cloud geometry is usually pre-processed through voxelization, a type quantization of the coordinates.

Introduction-Point clouds



Original

Voxelized

Point cloud geometry is usually pre-processed through voxelization, a type quantization of the coordinates.

We propose a novel lossless compression algorithm for the geometry compression of voxelized point clouds.

State of the art on geometry compression

Octree representation: [1,2,3]

Iteratively partitioning a cubic voxel space into 8 identically sized smaller cubes.

The partitioning of a subcube occurs under the condition that at least one occupied voxel is located within it.

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.

State of the art on geometry compression

Octree representation: [1,2,3]

Methods often pair entropy encoders for further compression gain.

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.

State of the art on geometry compression

Octree representation: [1,2,3]
Cellular automata: [4,5]

Inherits octree representation but models voxel volumes as cellular automata.

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.
[4] Simone Milani, “Fast point cloud compression via reversible cellular automata block transform,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 4013–4017
[5] Milani, Simone, Enrico Polo, and Simone Limuti. "A Transform Coding Strategy for Dynamic Point Clouds." IEEE Transactions on Image Processing 29 (2020): 8213-8225.

State of the art on geometry compression

Octree representation: [1,2,3]
Cellular automata: [4,5]
Binary tree partitioning: [6]

A non-empty voxel space is iteratively partitioned into two rectangular cuboids that contain the same number of occupied voxels.

Encoding residual distances between points that is minimized through a traveling salesperson problem.

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.
[4] Simone Milani, “Fast point cloud compression via reversible cellular automata block transform,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 4013–4017
[5] Milani, Simone, Enrico Polo, and Simone Limuti. "A Transform Coding Strategy for Dynamic Point Clouds." IEEE Transactions on Image Processing 29 (2020): 8213-8225.
[6] Wenjie Zhu, Yiling Xu, Li Li, and Zhu Li, “Lossless point cloud geometry compression via binary tree partition and intra prediction,” in 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2017, pp. 1–6.

State of the art on geometry compression

Octree representation: [1,2,3]
Cellular automata: [4,5]
Binary tree partitioning: [6]
Classical image compression techniques: [7]

Point clouds are projected onto 2D surfaces.

Classical image encoding techniques are applied on the 2D projections.

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.
[4] Simone Milani, “Fast point cloud compression via reversible cellular automata block transform,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 4013–4017
[5] Milani, Simone, Enrico Polo, and Simone Limuti. "A Transform Coding Strategy for Dynamic Point Clouds." IEEE Transactions on Image Processing 29 (2020): 8213-8225.
[6] Wenjie Zhu, Yiling Xu, Li Li, and Zhu Li, “Lossless point cloud geometry compression via binary tree partition and intra prediction,” in 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2017, pp. 1–6.
[7] D Graziosi, O Nakagami, S Kuma, A Zaghetto, T Suzuki, and A Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc),” APSIPA Transactions on Signal and Information Processing, vol. 9, 2020.

State of the art on geometry compression

Octree representation: [1,2,3]
Cellular automata: [4,5]
Binary tree partitioning: [6]
Classical image compression techniques: [7]
Occupancy array representation: [8,9]

A 3D boolean array G(x,y,z) represents the entire voxelized space.

- =1 *for occupied voxels.*
- =0 *for non-occupied voxels.*

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.
[4] Simone Milani, “Fast point cloud compression via reversible cellular automata block transform,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 4013–4017
[5] Milani, Simone, Enrico Polo, and Simone Limuti. "A Transform Coding Strategy for Dynamic Point Clouds." IEEE Transactions on Image Processing 29 (2020): 8213-8225.
[6] Wenjie Zhu, Yiling Xu, Li Li, and Zhu Li, “Lossless point cloud geometry compression via binary tree partition and intra prediction,” in 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2017, pp. 1–6.
[7] D Graziosi, O Nakagami, S Kuma, A Zaghetto, T Suzuki, and A Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc),” APSIPA Transactions on Signal and Information Processing, vol. 9, 2020.
[8] Eduardo Peixoto, “Intra-frame compression of point cloud geometry using dyadic decomposition,” IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020.
[9] Eduardo Peixoto, Edil Medeiros, and Evaristo Ramalho, “Silhouette 4d: An inter-frame lossless geometry coder of dynamic voxelized point clouds,” in 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020, pp. 2691–2695.

State of the art on geometry compression

Octree representation: [1,2,3]
Cellular automata: [4,5]
Binary tree partitioning: [6]
Classical image compression techniques: [7]
Occupancy array representation: [8,9]

Inspired our proposed method that falls under the occupancy array representation category.

Motivation: Occupancy array is linked to the most competitive results

[1] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.
[2] Diogo C Garcia and Ricardo L de Queiroz, “Intra-frame context-based octree coding for point-cloud geometry,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 1807–1811.
[3] Ricardo L de Queiroz, Diogo C Garcia, Philip A Chou, and Dinei A Florencio, “Distance-based probability model for octree coding,” IEEE Signal Processing Letters, vol. 25, no. 6, pp. 739–742, 2018.
[4] Simone Milani, “Fast point cloud compression via reversible cellular automata block transform,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 4013–4017
[5] Milani, Simone, Enrico Polo, and Simone Limuti. "A Transform Coding Strategy for Dynamic Point Clouds." IEEE Transactions on Image Processing 29 (2020): 8213-8225.
[6] Wenjie Zhu, Yiling Xu, Li Li, and Zhu Li, “Lossless point cloud geometry compression via binary tree partition and intra prediction,” in 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2017, pp. 1–6.
[7] D Graziosi, O Nakagami, S Kuma, A Zaghetto, T Suzuki, and A Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc),” APSIPA Transactions on Signal and Information Processing, vol. 9, 2020.
[8] Eduardo Peixoto, “Intra-frame compression of point cloud geometry using dyadic decomposition,” IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020.
[9] Eduardo Peixoto, Edil Medeiros, and Evaristo Ramalho, “Silhouette 4d: An inter-frame lossless geometry coder of dynamic voxelized point clouds,” in 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020, pp. 2691–2695.

Single-projection geometry compression

We present a competitive lossless intra-frame geometry compression algorithm.

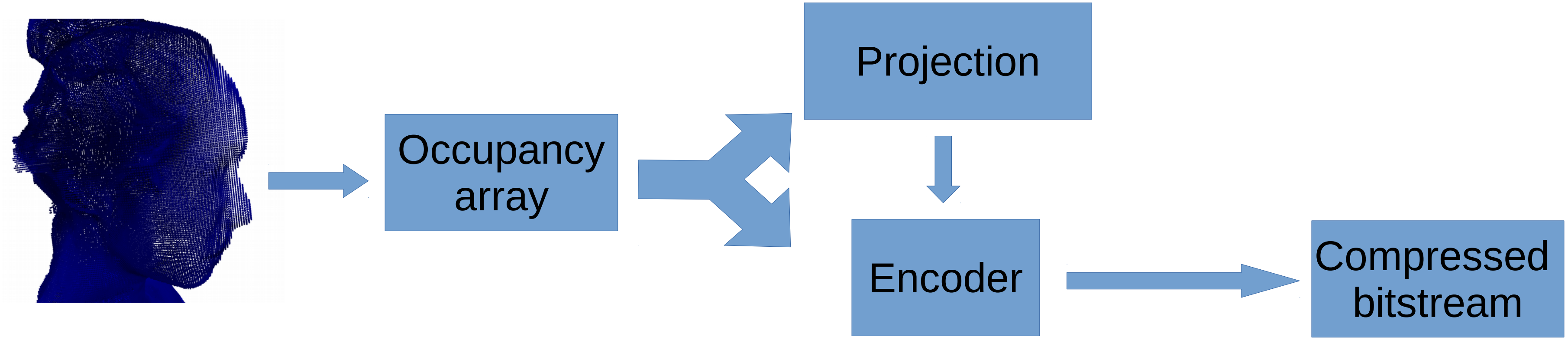
Our proposal is based on the single mode encoder of method [8]

Differences

	<u>Our proposal</u>	<u>Method [8]</u>
Limiting transmission of non-occupied voxels	Single projection	Dyadic decomposition
Axis selection	Cost efficient, non optimal, based on empirical observations	Computationally expensive selection of optimal decomposition axis

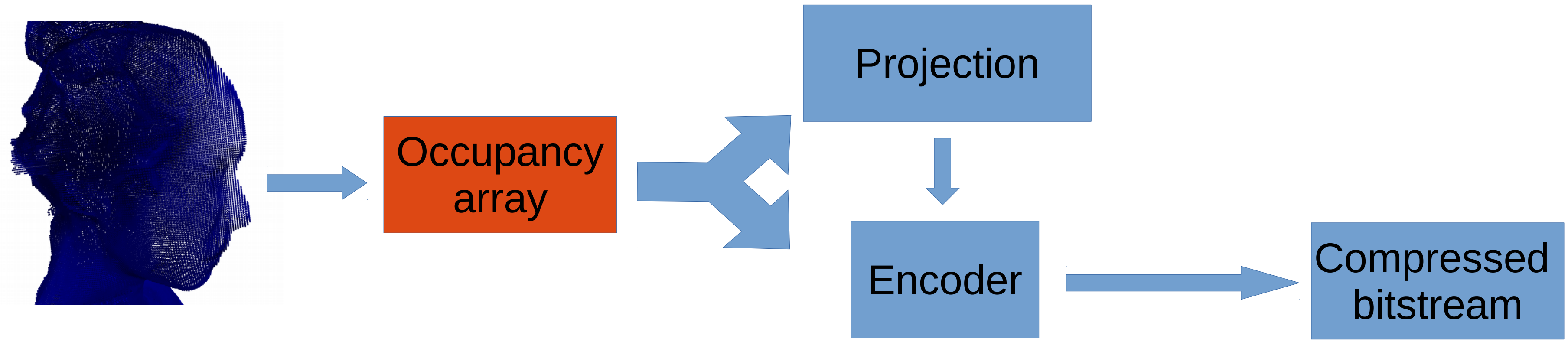
[8] Eduardo Peixoto, "Intra-frame compression of point cloud geometry using dyadic decomposition," IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020

Single-projection geometry compression



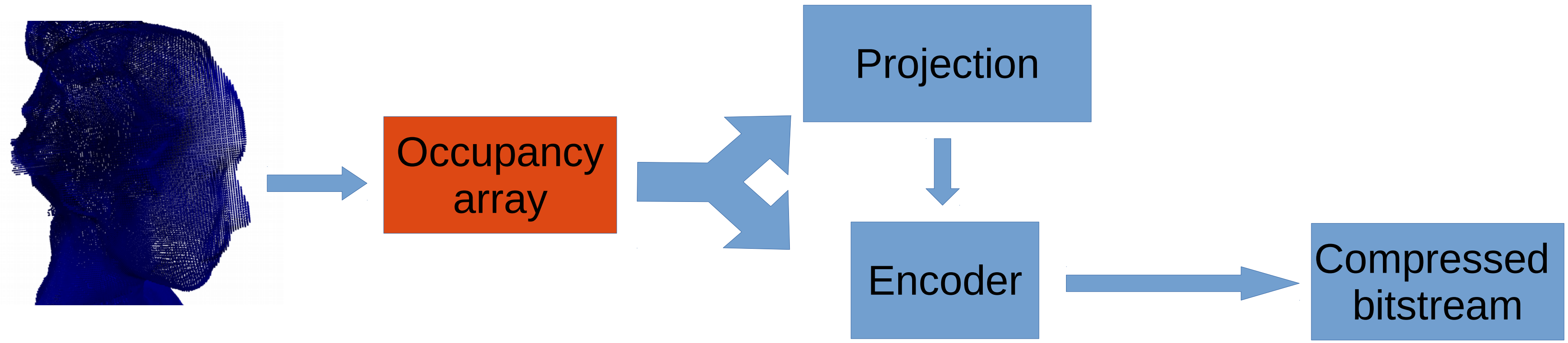
Flow chart of proposed compression scheme.

Single-projection geometry compression

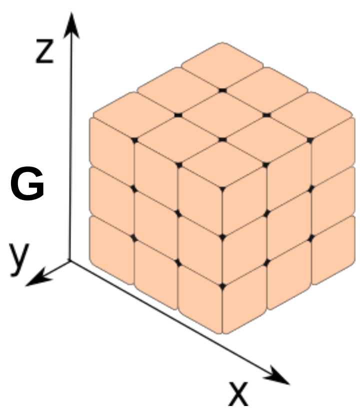


Occupancy array.

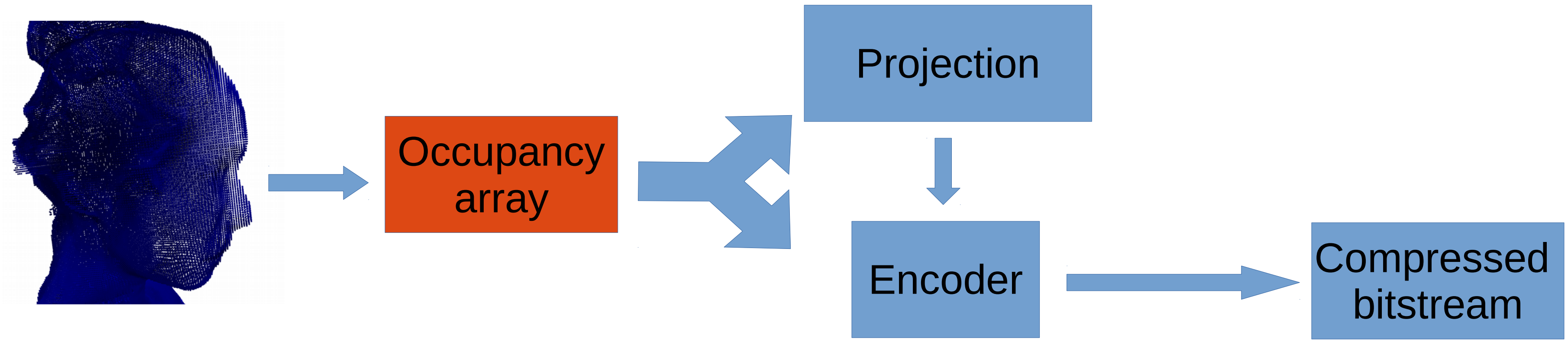
Single-projection geometry compression



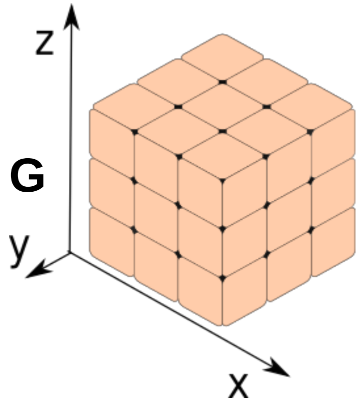
A 3D boolean occupancy array **G** is created, representing the geometry of the point cloud.



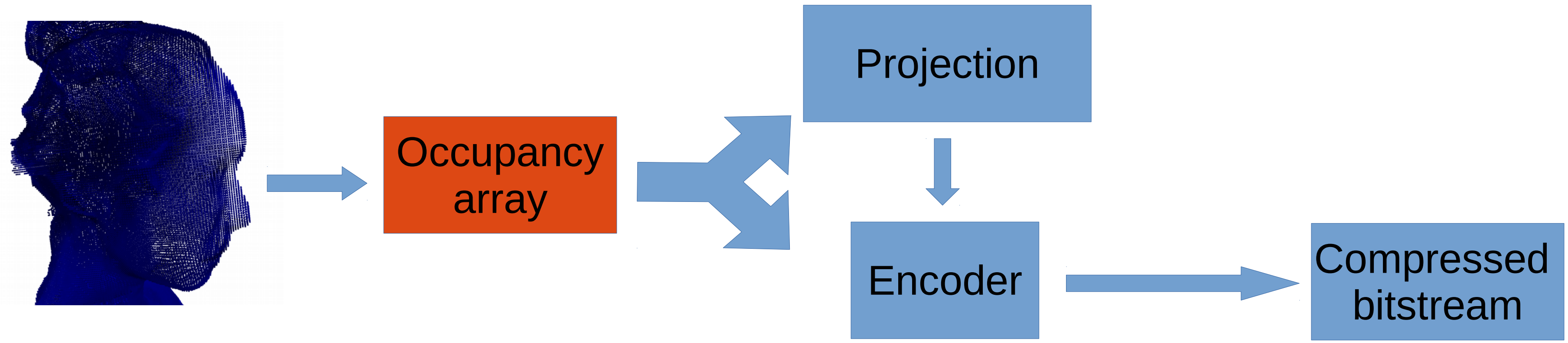
Single-projection geometry compression



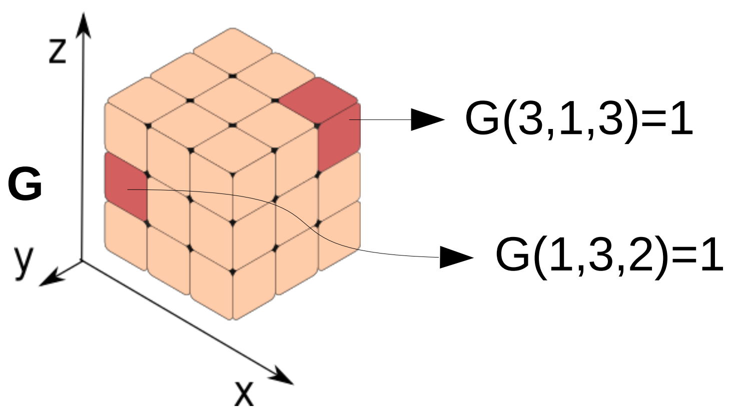
To compress the geometry of a point cloud we need to compress and encode the occupancy array **G.**



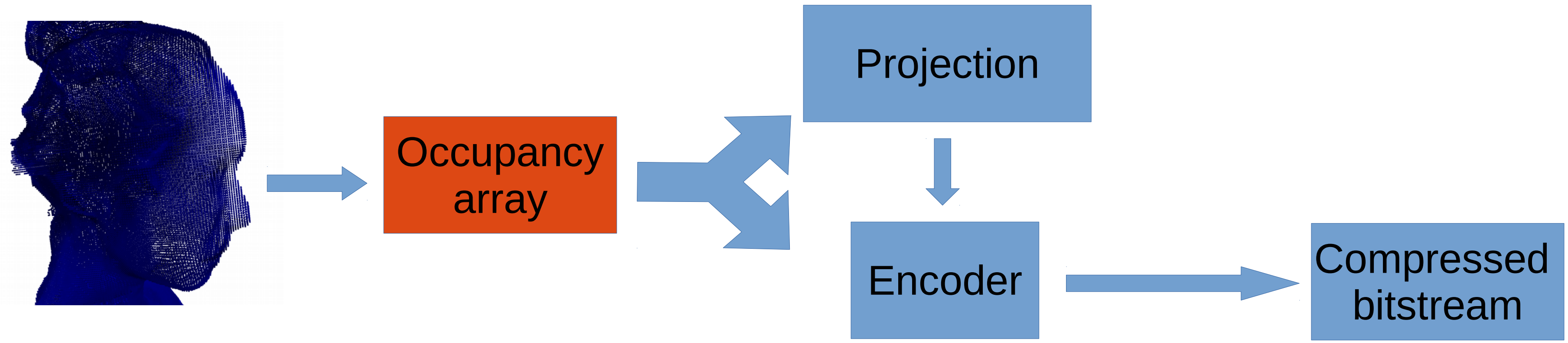
Single-projection geometry compression



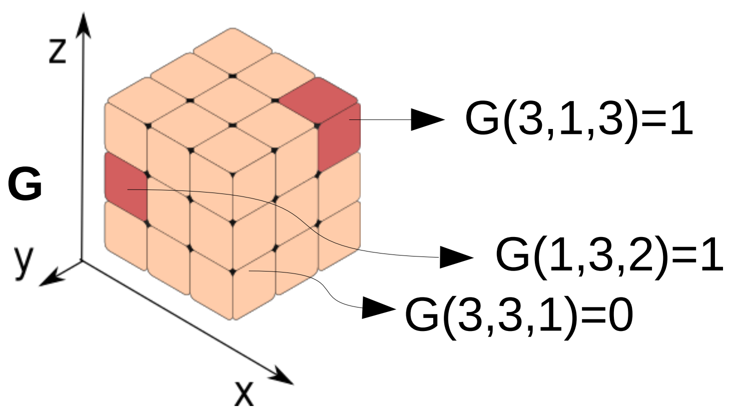
An element of **G** located in coordinates (x_a,y_b,z_c) is equal to 1 if there exists an occupied voxel at the same coordinates.



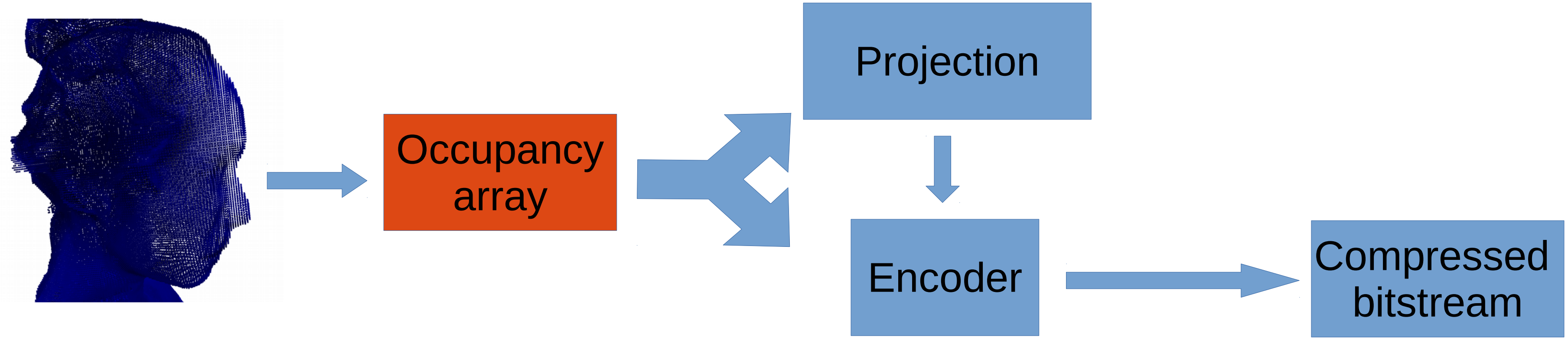
Single-projection geometry compression



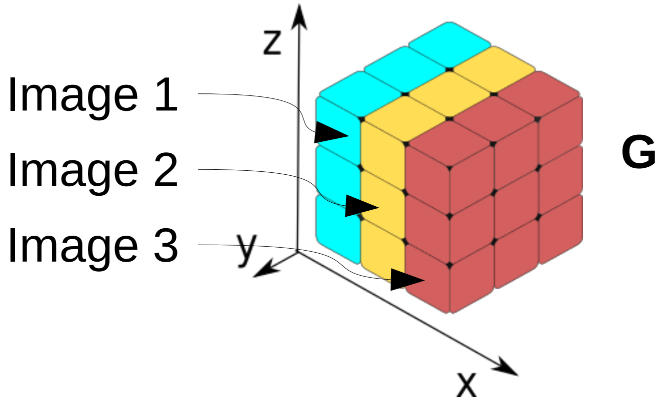
An element of **G** located in coordinates (x_a,y_b,z_c) is equal to 0 if there does not exists an occupied voxel at the same coordinates.



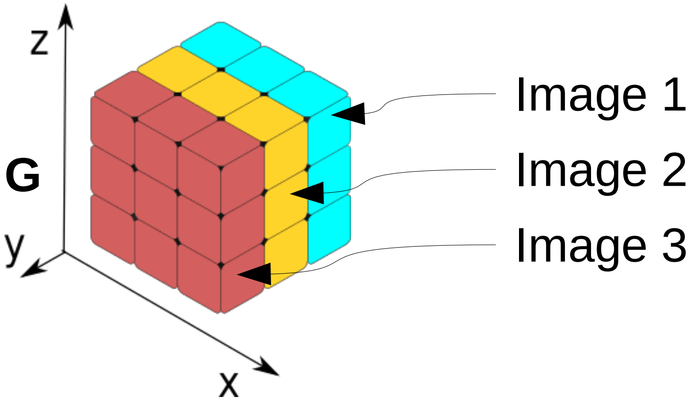
Single-projection geometry compression



By the term image, we call a single 2D array formed by all the elements of the occupancy matrix G, that are all located on the same plane that is perpendicular to the x, y or z axis.

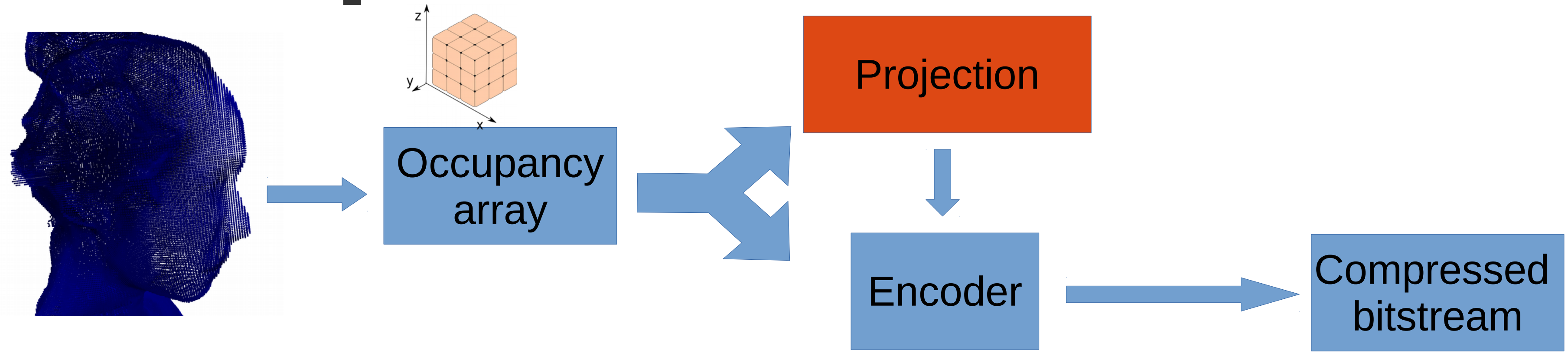


Images perpendicular to the x-axis.



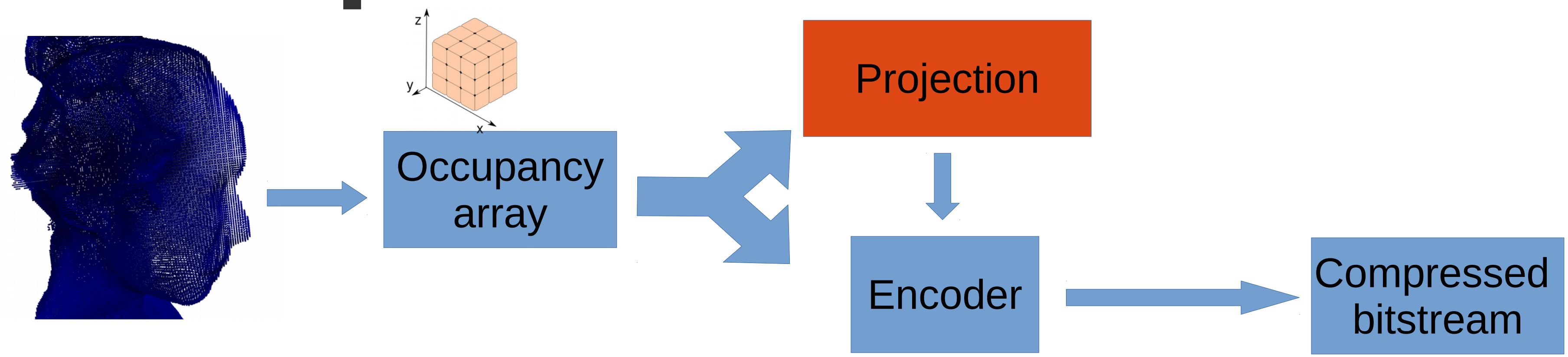
Images perpendicular to the y-axis.

Single-projection geometry compression



Single projection.

Single-projection geometry compression

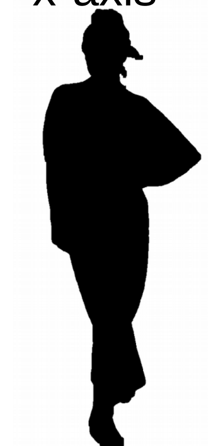


Initially an axis is chosen.

Through a logical OR operation across all images perpendicular to the chosen axis we compute a single 2D array called the projection matrix.

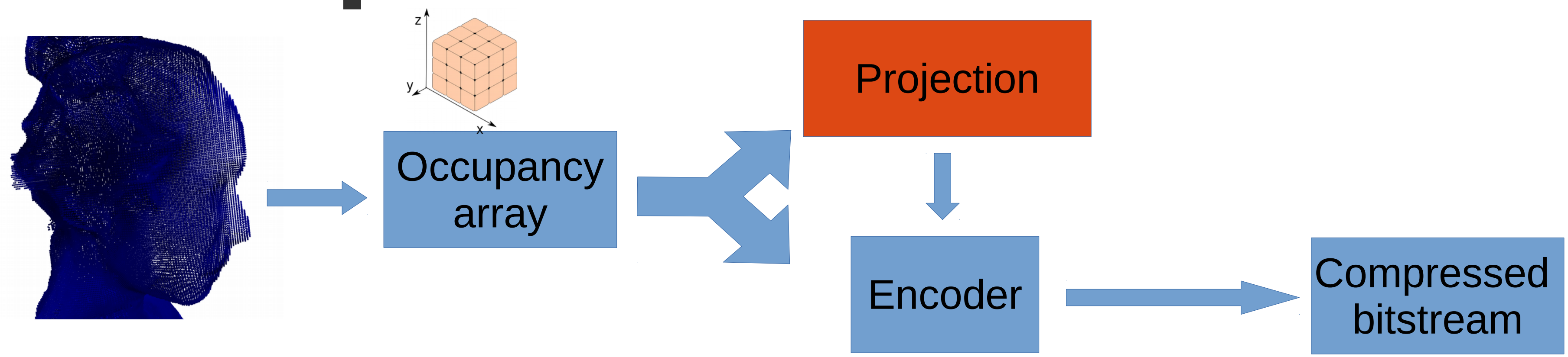
Point cloud → 2D projection on the plane perpendicular to the :

x-axis



y-axis

Single-projection geometry compression

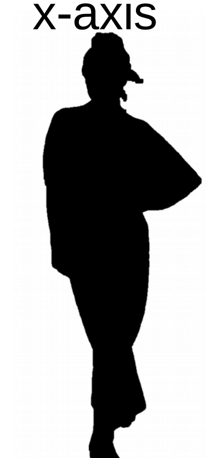


The projection matrix helps indicate the area of 2D locations of each image that need to be encoded.

Elements of each image outside these locations are guaranteed to be 0 (non occupied voxels).

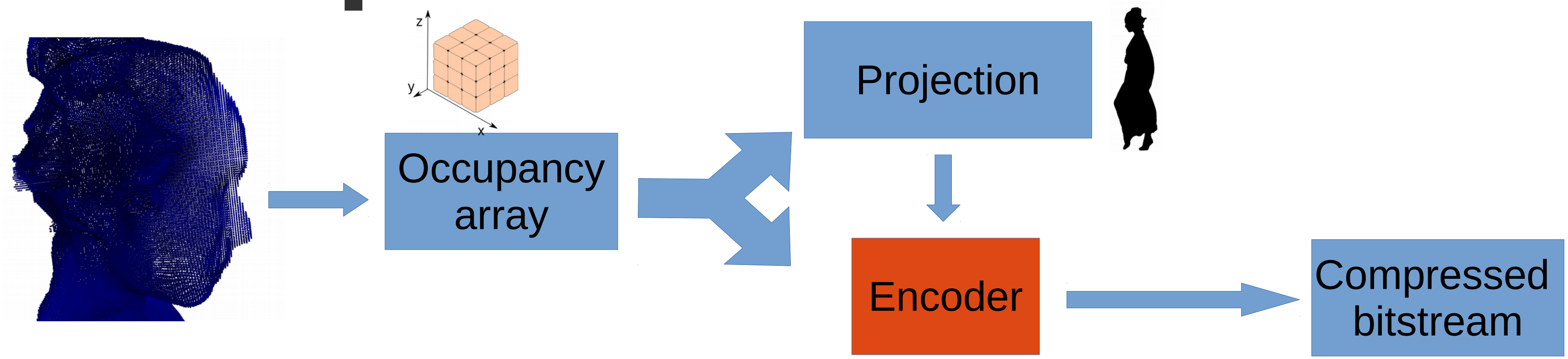
Point cloud → 2D projection on the plane perpendicular to the :

x-axis



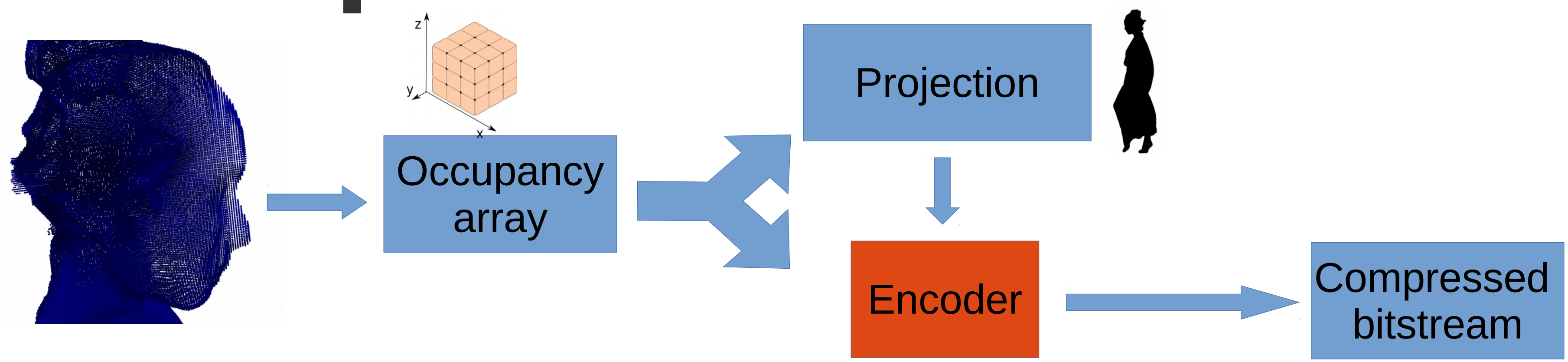
y-axis

Single-projection geometry compression



Encoder.

Single-projection geometry compression

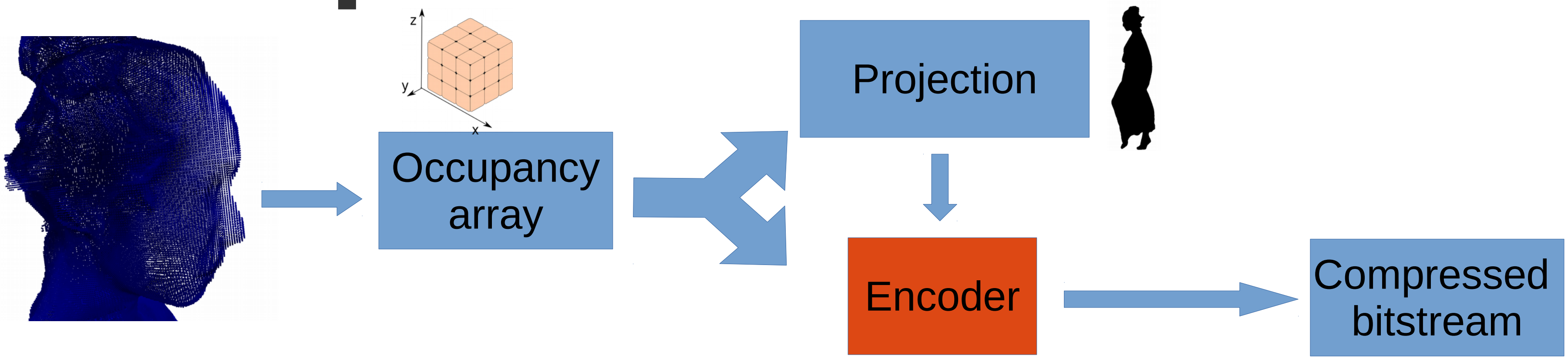


The encoder used is a Context Adaptive Binary Arithmetic Coder (CABAC).

Each image is encoded one after the other, following the same scanning pattern.

For each image, only the elements that are indicated by the projection matrix are encoded.

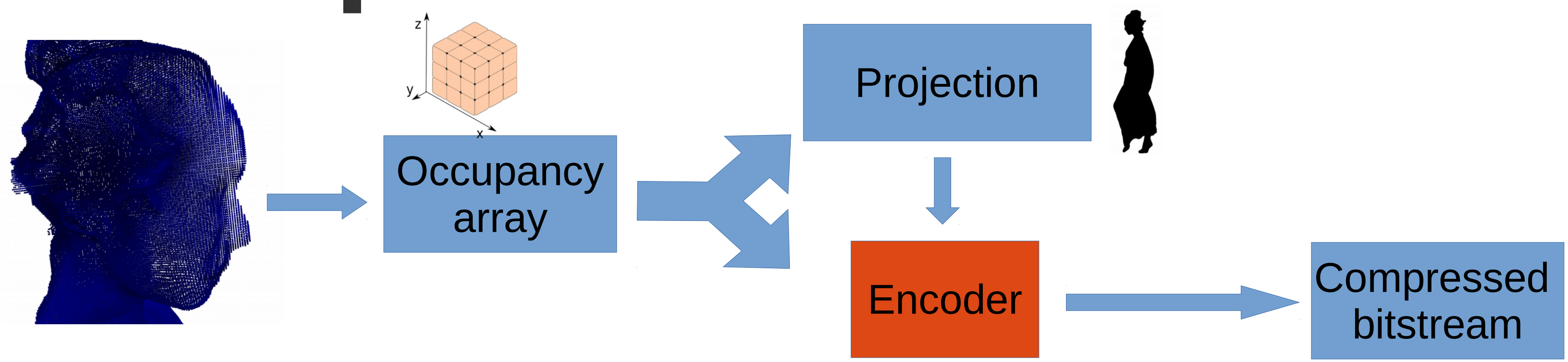
Single-projection geometry compression



The contexts used in the CABAC help us predict a pixel value of an image based on its previously encoded neighbors.

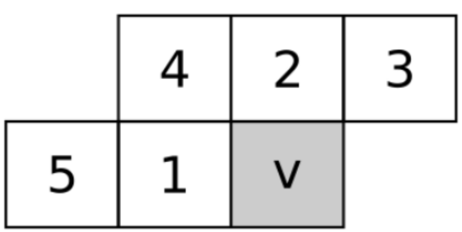
The possible values of the image elements are 1 for occupied and 0 for non-occupied voxels.

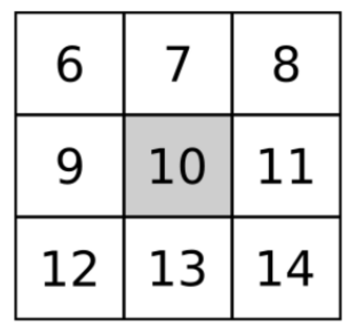
Single-projection geometry compression



Contexts that are used in the CABAC are of two types.

Contexts used to predict value of element v of an image :

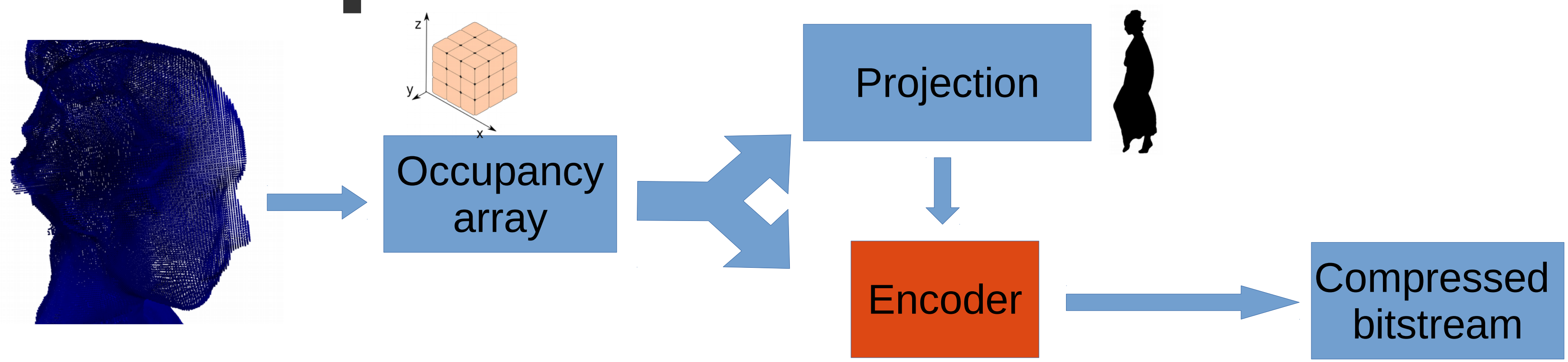




5 pixels used as contexts
from the same image.

9 pixels used as contexts
from the previously encoded image

Single-projection geometry compression



Projection matrix is encoded through a CABAC with 10 contexts:

	9	6	10	
7	4	2	3	8
5	1	v		

Results

Results have been computed using the first 100 frames of sequences from the:

Full body JPEG Pleno database [1].

Microsoft upper body database [2].

[1] E. d’Eon, B. Harrison, T. Myers, and P. A. Chou, “8i voxelized full bodies - a voxelized point cloud dataset,” ISO/IEC JTC1/SC29/WG1 input document M74006 and ISO/IEC JTC1/SC29/WG11 input document m40059, Geneva, January 2017.

[2] C. Loop, Q. Cai, S. Orts Escolano, and P.A. Chou, “Microsoft voxelized upper bodies - a voxelized point cloud dataset,” ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012, Geneva, May 2016.

Results

We compare our results against state of the art geometry compression methods:

1) MPEG G-PCC v7.0 variant [1].

2) S-3D [2].

3) S-4D [3].

4) Plain octree compression [4].

[1] 3DG, G-PCC codec description v4, Tech. Rep., ISO/IEC JTC 1/SC 29/WG 11 input document w18673, 2019.
[2] Eduardo Peixoto, “Intra-frame compression of point cloud geometry using dyadic decomposition,” IEEE Signal Processing Letters, vol. 27, pp. 246–250, 2020.
[3] Eduardo Peixoto, Edil Medeiros, and Evaristo Ramalho, “Silhouette 4d: An interframe lossless geometry coder of dynamic voxelized point clouds,” in 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020, pp. 2691–2695.
[4] Donald Meagher, “Geometric modeling using octree encoding,” Computer graphics and image processing, vol. 19, no. 2, pp. 129–147, 1982.

Results

We show our results for 2 different choices of projection axis:

1) Pz Using images perpendicular to the z-axis.

2) Py Using images perpendicular to the y-axis.

Results

We show our results for 2 different choices of projection axis:

1) Pz Using images perpendicular to the z-axis.

2) Py Using images perpendicular to the y-axis.

The choice of projection axis is based on an empirical observation:

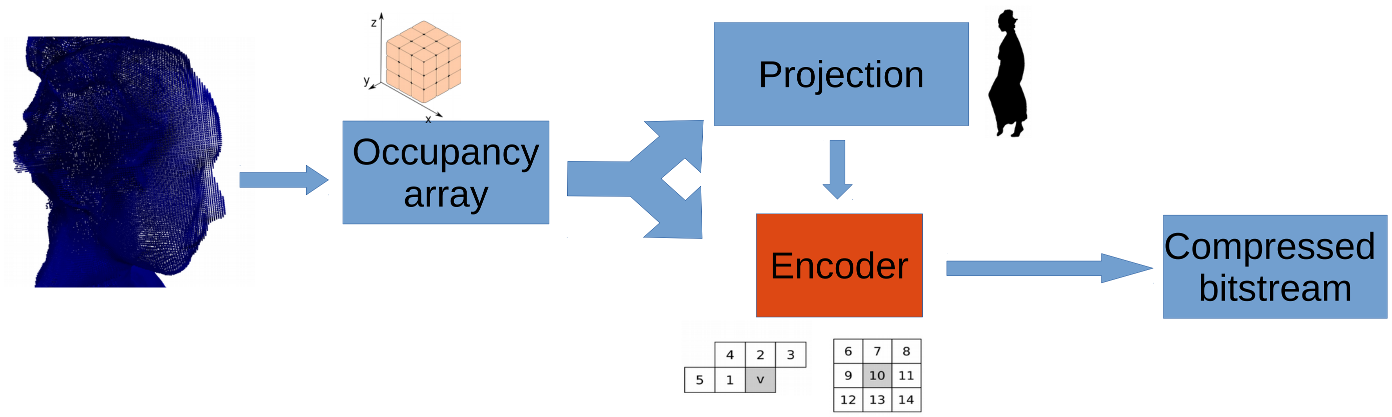
In the cases where the variance of the points along the y-axis is sufficiently larger than along the z-axis, we project the occupancy array along the former.

Results

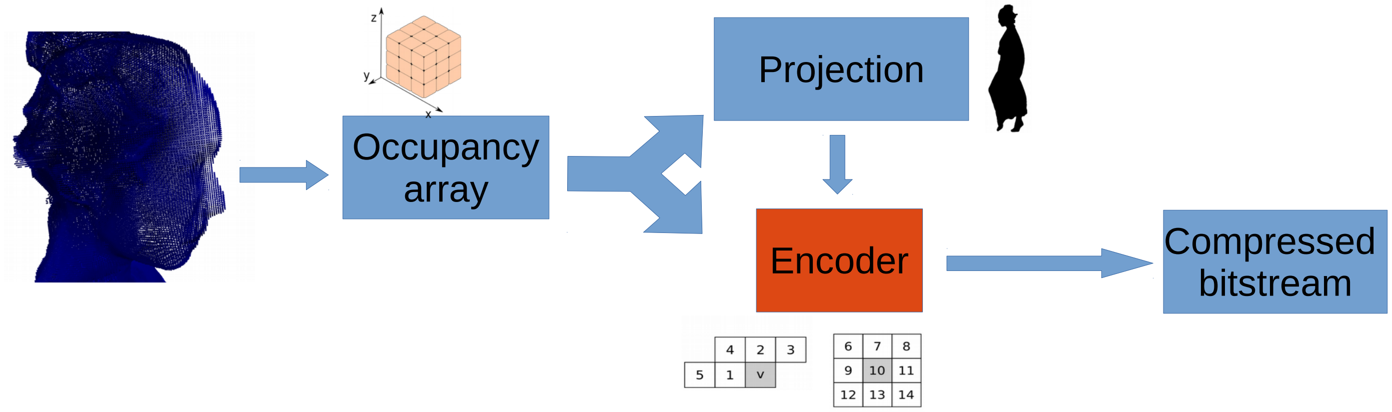
Sequence	Intra Coders: Octree	G-PCC	S-3D	Proposed Pz	Proposed Py	Inter Coder: S-4D	Gains of proposed approach over: G-PCC	S-3D	S-4D
andrew9	2.58	1.14	1.12	**1.02**	1.05	1.08	10.53%	8.93%	5.56%
david9	2.62	1.07	1.06	**0.96**	1.02	1.05	10.28%	9.43%	8.57%
phil9	2.64	1.18	1.14	**1.03**	1.05	1.13	12.71%	9.65%	8.85%
ricardo9	2.59	1.10	1.04	**0.96**	0.97	1.02	12.73%	7.69%	5.88%
sarah9	2.61	1.08	1.07	**0.97**	1.01	1.04	10.19%	9.35%	6.73%
Average	2.61	1.11	1.09	**0.99**	1.02	1.06	11.29%	9.01%	7.12%
longdress	2.99	1.03	0.95	0.94	**0.89**	0.95	13.59%	6.32%	6.32%
loot	2.98	0.97	0.92	0.91	**0.85**	0.91	12.37%	7.61%	6.59%
redandblack	3.00	1.10	1.02	1.00	**0.95**	1.02	13.64%	6.86%	6.86%
soldier	3.00	1.04	0.96	0.93	0.90	**0.81**	13.46%	6.25%	-11.11%
Average	2.99	1.04	0.96	0.95	**0.90**	0.92	13.27%	6.76%	2.17%

Units of rate measured in bpov.

Semi-adaptive approach



Semi-adaptive approach

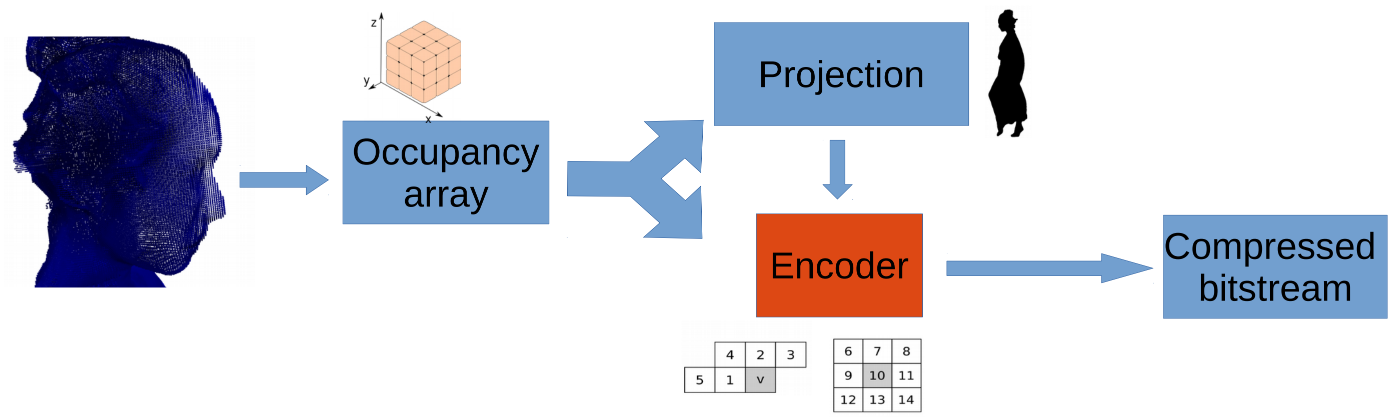


It is possible that a ***fully-adaptive arithmetic*** encoder is not the optimal choice.

Benefit: Absence of side information

Drawback: Slow convergence of estimated probabilities towards true ones.

Semi-adaptive approach

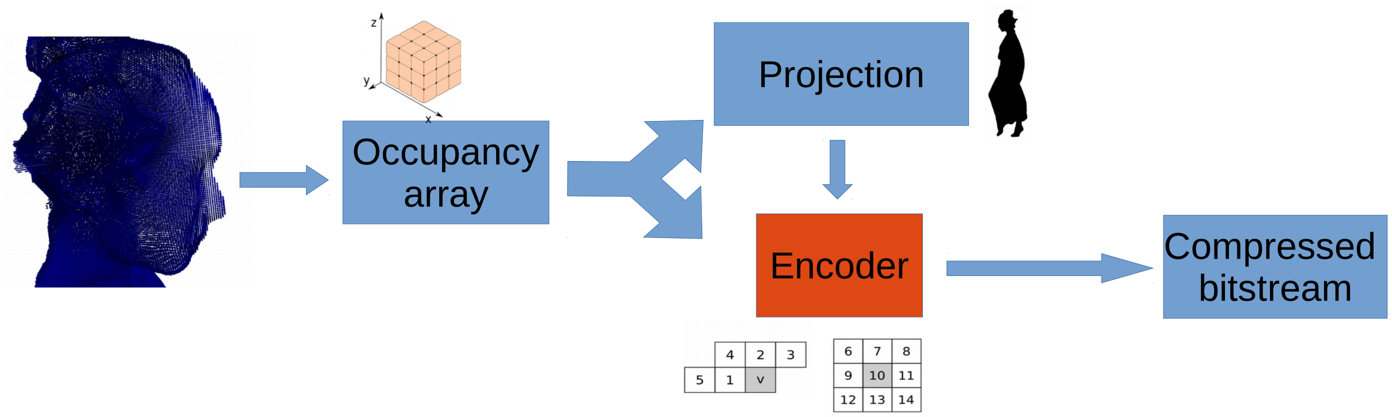


It is possible that a ***fully-adaptive arithmetic*** encoder is not the optimal choice.

Drawback: Slow convergence of estimated probabilities towards true ones.

This might occur when dealing with redundant number of contexts.

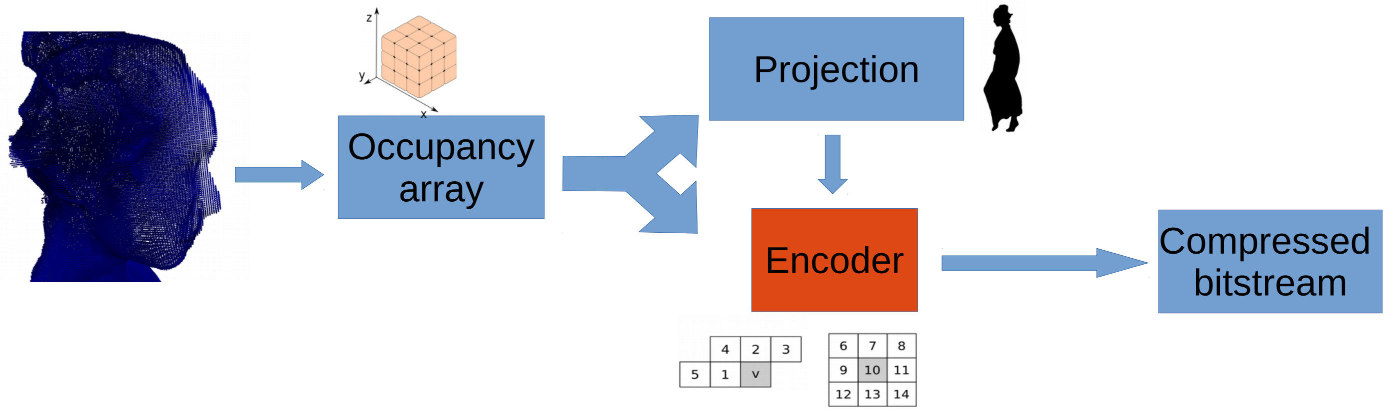
Semi-adaptive approach



It is possible that a ***fully-adaptive arithmetic*** encoder is not the optimal choice.

By transmitting a partial frequency table, a semi-adaptive approach might seem more advantageous.

Semi-adaptive approach



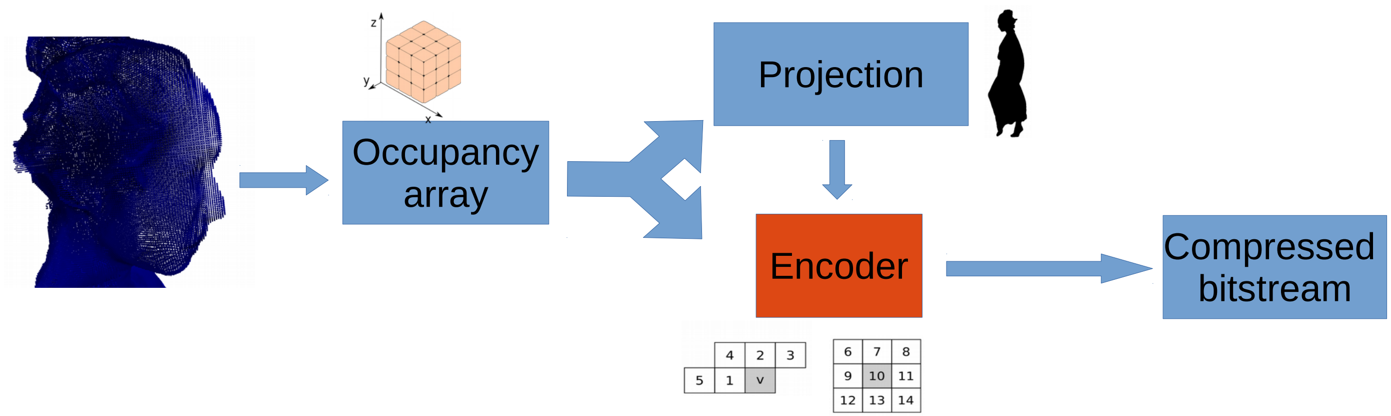
$$I = -\frac{\sum_{c\in\mathcal{C}}(\sum_{v_o\in\mathcal{V}_o} log_2(p(v_o|c)) + \sum_{v_n\in\mathcal{V}_n} log_2(p(v_n|c)))}{NOV}$$

NOV Number of occupied voxels

$p(v_o|c)$ Conditional probability of voxel being occupied given context c

$p(v_n|c)$ Conditional probability of voxel being non-occupied given context c

Semi-adaptive approach



$$I = -\frac{\sum_{c\in\mathcal{C}}(\sum_{v_o\in\mathcal{V}_o} log_2(p(v_o|c)) + \sum_{v_n\in\mathcal{V}_n} log_2(p(v_n|c)))}{NOV}$$

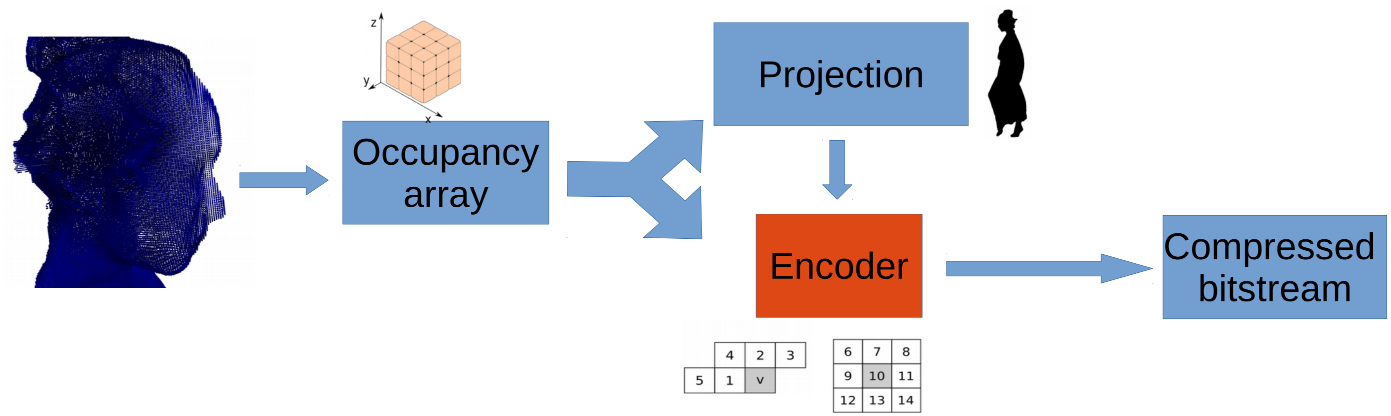
True probabilities:

I = bitrate theoretical lower bound of context non-adaptive entropy encoder.

Progressively updated probabilities:

I = bitrate theoretical lower bound of context adaptive entropy encoder.

Semi-adaptive approach



$$I = -\frac{\sum_{c\in\mathcal{C}}\left(\sum_{v_o\in\mathcal{V}_o} log_2(p(v_o|c)) + \sum_{v_n\in\mathcal{V}_n} log_2(p(v_n|c))\right)}{NOV}$$

True probabilities:

I = bitrate theoretical lower bound of context non-adaptive entropy encoder.

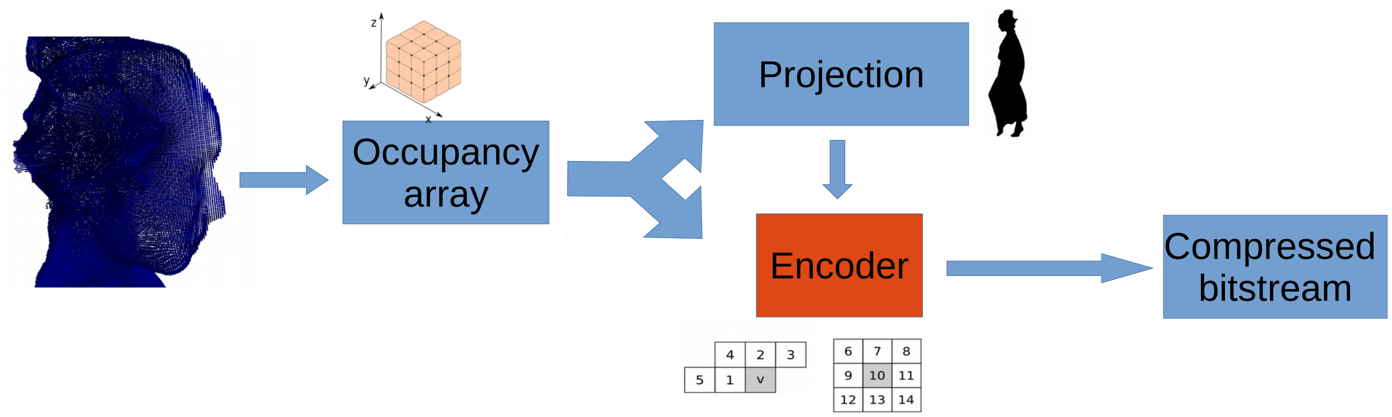
Noted as ***LB***

Progressively updated probabilities:

I = bitrate theoretical lower bound of context adaptive entropy encoder.

Noted as ***ALB***

Semi-adaptive approach

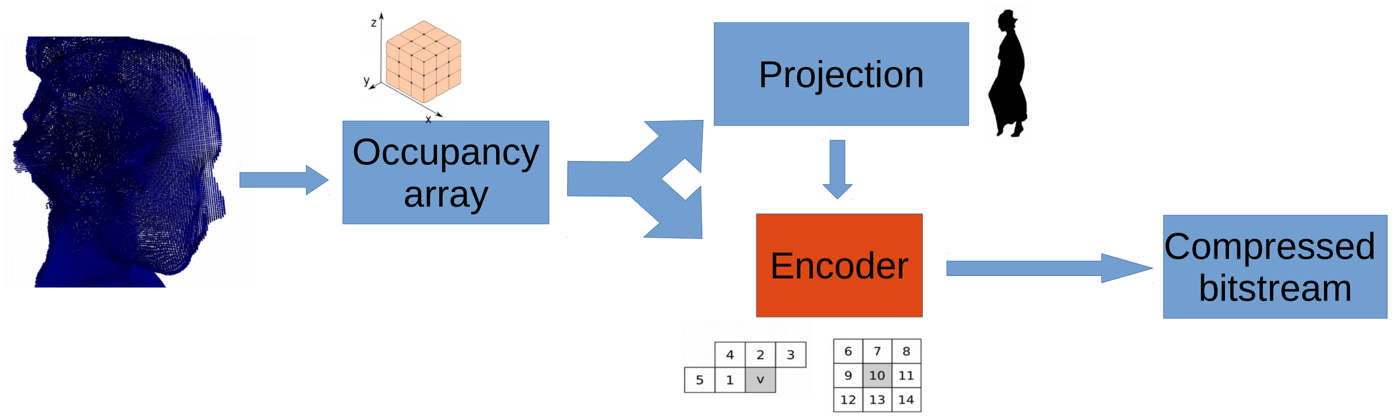


$$I = -\frac{\sum_{c\in\mathcal{C}}(\sum_{v_o\in\mathcal{V}_o} log_2(p(v_o|c)) + \sum_{v_n\in\mathcal{V}_n} log_2(p(v_n|c)))}{NOV}$$

SI: the estimated side information generated by encoding the probabilities corresponding to the 4% of the total number of contexts.

AR: The true rate of the encoded images using a fully-adaptive approach.

Semi-adaptive approach



$$I = -\frac{\sum_{c \in \mathcal{C}} (\sum_{v_o \in \mathcal{V}_o} log_2(p(v_o|c)) + \sum_{v_n \in \mathcal{V}_n} log_2(p(v_n|c)))}{NOV}$$

A theoretical test is conducted as an initial comparison between the context semi-adaptive and fully-adaptive entropy encoders by comparing ***AR***, ***SI***, ***LB*** and ***ALB***.

Semi-adaptive approach

Sequence	AR	ALB	LB	SI	Gains of $SI + LB$ over AR
Andrew9	1.01	1.00	0.88	0.07	5.94%
David9	0.94	0.93	0.83	0.06	5.32%
Phil9	1.02	1.00	0.90	0.06	5.88%
Ricardo9	0.94	0.93	0.81	0.08	5.32%
Sarah9	0.95	0.94	0.82	0.07	6.32%
Loot10	0.90	0.88	0.82	0.03	5.56%
Longdress10	0.94	0.92	0.86	0.02	6.38%
RedAndBlack	0.99	0.98	0.90	0.03	6.06%
Soldier	0.92	0.90	0.85	0.02	5.43%

Rates in bpov related to the encoding cost of images.

Semi-adaptive approach

Sequence	AR	ALB	LB	SI	Gains of $SI + LB$ over AR
Andrew9	1.01	1.00	0.88	0.07	5.94%
David9	0.94	0.93	0.83	0.06	5.32%
Phil9	1.02	1.00	0.90	0.06	5.88%
Ricardo9	0.94	0.93	0.81	0.08	5.32%
Sarah9	0.95	0.94	0.82	0.07	6.32%
Loot10	0.90	0.88	0.82	0.03	5.56%
Longdress10	0.94	0.92	0.86	0.02	6.38%
RedAndBlack	0.99	0.98	0.90	0.03	6.06%
Soldier	0.92	0.90	0.85	0.02	5.43%

We assume that by using the top 4% of contexts, the theoretical lower bound of the semi-adaptive approach yields similar results to the non-adaptive method (LB).

If true, one could expect gains of up to 6.3% over the fully-adaptive approach that is currently used.

Semi-adaptive approach

Sequence	AR	ALB	LB
Andrew9	0.02	0.02	0.01
David9	0.02	0.02	0.02
Phil9	0.02	0.02	0.02
Ricardo9	0.02	0.02	0.01
Sarah9	0.02	0.02	0.02
Loot10	0.01	0.01	0.01
Longdress10	0.01	0.01	0.01
RedAndBlack	0.01	0.01	0.01
Soldier	0.01	0.01	0.01

Rates in bpov related to the encoding of the projection array.

Conclusions

- We propose a novel geometry compression algorithm for point cloud data.
- Our proposal is inspired by Peixoto’s ‘Intra-frame compression of point cloud geometry using dyadic decomposition’, ‘simplifying’ the design by replacing the dyadic decomposition with a single projection.
- Our results show the proposed method outperforms the state-of-the-art inter and intra-frame geometry compression algorithms.
- Best coding performance published so far.
- An empyrical rule is provided for the axis selection of the projection procedure.
- We show that replacing the context adaptive binary arithmetic encoder with a semi-adpative variant might lead to further gains.