CLAIMS

[Claim 1]

A method for compressing a point cloud, wherein the point cloud is composed of a plurality of points in a three-dimensional (3D) space, comprising steps:

acquiring the point cloud with a sensor, wherein each point is associated with a 3D coordinate and at least one attribute;

partitioning the point cloud into an array of 3D blocks of elements, wherein some of the elements in the 3D blocks have missing points;

predicting, for each 3D block by using multivariate nearest-neighbor interpolation and extrapolation, attribute values for the 3D block by projecting the attribute values of neighboring 3D blocks to the adjacent edge plane of the 3D blocks, and obtaining a 3D prediction residual block indicative of a difference between the predicted attribute values for the 3D block and the acquired attribute values, associated with each point of the point cloud, in the 3D block;

applying a 3D transform to each 3D prediction residual block using locations of occupied elements to produce transform coefficients, wherein the transform coefficients have a magnitude and sign;

entropy encoding the transform coefficients according the magnitudes and sign bits to produce a bitstream;

wherein the 3D transform is a shape-adaptive discrete cosine transform (SA-DCT) designed for 3D point cloud attribute compression

defining a contour of points as a region, wherein the region encompasses non-empty positions; and

shifting the points in the regions along each direction toward a border of the block so that there are no empty positions in the block along that border, wherein

the steps are performed in a processor.

[Claim 2]

The method of claim 1, further comprising:

converting the point cloud to an octree of voxels arranged on a grid that is uniform, and wherein the partitioning is repeated until the voxels have a minimal predefined resolution.

[Claim 3]

The method of claim 1, wherein each leaf node in the octree corresponds to a point output by the partitioning, and the position of the point is set to a geometric center of the leaf node, and the attribute value associated with the point is set to an average attribute value of one or more points in the leaf node.

[Claim 4]

The method of claim 1, wherein the partitioning is according to a block edge size.

[Claim 5]

The method of claim 1, wherein the prediction for a current block is from the points contained in non-empty adjacent blocks.

[Claim 6]

The method of claim 5, wherein theprediction selects a prediction direction that yields a least distortion.

[Claim 7]

The method of claim 1, wherein the 3D transform applies a graph transform to each block.

[Claim 8]

The method of claim 7, wherein the graph transform produces two DC coefficients and two corresponding sets of AC coefficients.

[Claim 9]

The method of claim 1, wherein each point of the point cloud is associated with at least one attribute.

[Claim 10]

The method of claim 9, wherein the attribute is color information.

[Claim 11]

The method of claim 9, wherein the attribute is reflectivity information.

[Claim 12]

The method of claim 9, wherein the attribute is a normal vector.

[Claim 13]

The method of claim 1, further comprising:

entropy decoding the bitstream to obtain transform coefficients and point locations;

applying an inverse 3D transform to the transform coefficients to produce a 3D prediction residual block;

arranging the elements in the 3D prediction residual block according to the point locations of occupied elements;

predicting, for each 3D prediction residual block, attribute values for the 3D block based on the attribute values of neighboring 3D blocks, and obtaining a 3D prediction block;

combining the 3D prediction block to the 3D prediction residual block to obtain a 3D reconstructed block;

concatenating the 3D reconstructed block to previously-reconstructed 3D blocks to form an array of 3D reconstructed blocks; and

outputting the array of 3D reconstructed blocks as a reconstructed 3D point cloud.

[Claim 14]

The method of claim 13, wherein the arranging of elements according to the locations of the occupied elements is performed before the inverse 3D transform is applied.

[Claim 15]

The method of claim 1, wherein all missing elements in a 3D block are replaced with predetermined values, and wherein all transforms applied in same direction during the shape-adaptive discrete cosine transform process have same lengths, equal to a number of missing and non-missing elements in the 3D block along that direction.