



(51) International Patent Classification:
H04N 19/597 (2014.01)

(21) International Application Number:
PCT/JP2018/037496

(22) International Filing Date:
02 October 2018 (02.10.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/572,880 16 October 2017 (16.10.2017) US
15/916,497 09 March 2018 (09.03.2018) US

(71) Applicant: **MITSUBISHI ELECTRIC CORPORATION** [JP/JP]; 7-3, Marunouchi 2-chome, Chiyoda-ku, Tokyo, 1008310 (JP).

(72) Inventors: **COHEN, Robert**; c/o Mitsubishi Electric Research Laboratories, Inc., 201, Broadway, Cambridge,

Massachusetts, 02139 (US). **TIAN, Dong**; c/o Mitsubishi Electric Research Laboratories, Inc., 201, Broadway, Cambridge, Massachusetts, 02139 (US). **OCHIMIZU, Hideaki**; c/o Mitsubishi Electric Research Laboratories, Inc., 201, Broadway, Cambridge, Massachusetts, 02139 (US). **VETRO, Anthony**; c/o Mitsubishi Electric Research Laboratories, Inc., 201, Broadway, Cambridge, Massachusetts, 02139 (US).

(74) Agent: **SOGA, Michiharu** et al.; S. SOGA & CO., 8th Floor, Kokusai Building, 1-1, Marunouchi 3-chome, Chiyoda-ku, Tokyo, 1000005 (JP).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

(54) Title: POINT CLOUD DECODER AND METHOD FOR DECODING POINT CLOUD

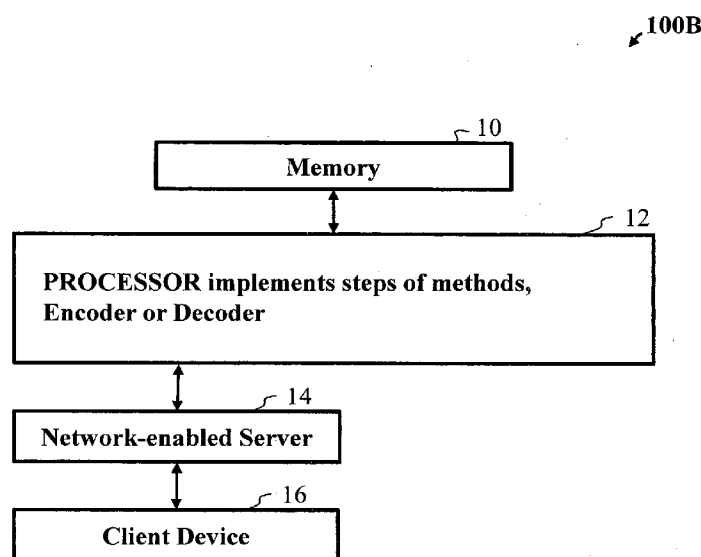


FIG. 1B

(57) Abstract: Systems and methods for a point cloud decoder including a processor to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud. Wherein each block includes a set of points, such that for each block the processor is to decode a set of prediction residuals from a compressed bitstream. Use a predetermined location in the block, and compute for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block. Wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.



OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

[DESCRIPTION]

[Title of Invention]

POINT CLOUD DECODER AND METHOD FOR DECODING POINT
CLOUD

[Technical Field]

[0001]

The present disclosure relates generally to methods, encoders and decoders representing point cloud signals, and more particularly, to compressing, representing and processing point clouds.

[Background Art]

[0002]

A point cloud is generally characterized by a set of points located in three-dimensional (3D) space. For each point in a point cloud, attributes such as color or other data can be associated with points. Given that the amount of data in a point cloud can become quite large, compression is needed in order to store the data to a storage medium such as a disk drive, or in order to transmit the data, such as signaling it to a bit-stream for streaming from a network-enabled server to a client device. While methods for compressing images are well-established, different methods are needed for compressing 3D point clouds, because unlike images which comprise a uniform grid of picture elements or pixels, the points in point clouds can be located anywhere in 3D space.

[0003]

Therefore, there is a need for methods, encoders and decoders for compressing, representing and processing point clouds.

[Summary of Invention]

[0004]

Embodiments of the present disclosure relate methods, encoders and decoders representing point cloud signals, and more particularly, to compressing, representing and processing point clouds. A point cloud can be characterized by a set of points located in three-dimensional space, where attributes or other data can be associated with each point. Aspects of the present disclosure include compressing point clouds since the amount of data in a point cloud can be large, resulting in the need for compression, in order to either store the data to a storage medium, or in order to transmit the data, such as signaling the data to a bit-stream for streaming from a network-enabled server to a client device.

[0005]

Regarding point prediction, some embodiments include signaling the coordinates of each point in a point cloud relative to a predetermined location, by computing the difference between the point position and the predetermined location and signaling it. This difference can be a prediction residual. The predetermined location can be the center of a block containing the points to be processed. The predetermined location can also be a previously-decoded point location, for example the first decoded point.

[0006]

Some embodiments include decoding a difference between a point position and a predetermined location, which can be a prediction residual.

[0007]

Some embodiments include signaling the coordinates of each point in a

point cloud relative to a previously-decoded point. For example, given a point in a sequence of points, the difference between the location of the point and the location of the point that was processed or encoded or decoded immediately prior to the point in sequential order, can be computed. This difference can be a prediction residual.

[0008]

Some embodiments include decoding a difference between a point position of a point and the position of a previously-decoded point. This difference can be a prediction residual. The previously-decoded point can be the point that was decoded sequentially prior to the point.

[0009]

Regarding point reordering and skipping, some embodiments include reordering a sequence of points such that the distance between subsequent points is minimized. Further other embodiments can include reordering a sequence of points such that the number of different distances between subsequent points is minimized.

[0010]

Some embodiments include skipping the signaling of a point if it is collocated or within a threshold distance from a previously-signaled point. Additionally, if a given decoded point is collocated or within a threshold distance from a previously-signaled point, then the reconstruction of the point can be skipped. The signaling or reconstruction of points or decoded points outside a bounding box or minimum and maximum coordinate limits can also be skipped.

[0011]

Regarding data-dependent partitioning, some embodiments include performing data-dependent non-uniform partitioning of a space, based on a scoring function that determines a location of the partitioning across at least one dimension of the space. Wherein, data-dependent non-uniform partitioning of the space can be based on measures or calculations from data representing the object, along with information about the current partitioning structure, such as a minimum or maximum block size, wherein a function, i.e. a scoring function, can specify how or where to split the current block or sub-block being processed.

[0012]

Regarding organizing an unorganized point cloud, some embodiments include organizing an unorganized point cloud by mapping each point to positions in an organizational grid. The mapping process can be implemented with a scanning process, in which the points in the point cloud can be virtually scanned based on their position relative to the scan origin, and in an order based upon a scan direction or angle.

[0013]

Regarding resampling and realigning a point cloud, some embodiments include using a scoring function to identify a subset of points in a point cloud, and subsequent processing of the point cloud is applied to the subset of points.

[0014]

Some embodiments include resampling or aligning the points in a point cloud to a predetermined subset of locations such as a grid. The point locations can be signaled, or an index indicating to where the point was aligned can be signaled.

[0015]

According to an embodiment of the present disclosure, a point cloud decoder including a processor to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud. Wherein each block includes a set of points, such that for each block the processor is to decode a set of prediction residuals from a compressed bitstream. Use a predetermined location in the block, and compute for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block. Wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[0016]

According to another embodiment of the present disclosure, a method for decoding a point cloud, including using a processor connected to a memory, to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud. Wherein each block includes a set of points, such that for each block the processor is for decoding a set of prediction residuals from a compressed bitstream. Using a predetermined location in the block, and computing for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block. Wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[0017]

According to another embodiment of the present disclosure, a point cloud decoder including a memory having data stored including previously decoded

points. A processor to decode a sequence of points from a compressed bitstream, so as to obtain a decoded point cloud. The processor is to decode a sequence of prediction residuals from the compressed bitstream. Compute for each prediction residual in the sequence of prediction residuals, a position of a point by adding the prediction residual to a position of a previously decoded point stored in the memory, so as to obtain a sequence of decoded points. Wherein the sequence of decoded points represents the decoded point cloud.

[0018]

The presently disclosed embodiments will be further explained with reference to the attached drawings. The drawings shown are not necessarily to scale, with emphasis instead generally being placed upon illustrating the principles of the presently disclosed embodiments.

[Brief Description of Drawings]

[0019]

[Fig. 1A]

FIG. 1A is a block diagram illustrating steps of a first point prediction method, according to embodiments of the present disclosure.

[Fig. 1B]

FIG. 1B is a block diagram illustrating some components implementing methods of the present disclosure, according to embodiments of the present disclosure.

[Fig. 2A]

FIG. 2A is a schematic illustrating a data-dependent non-uniform partitioning method, according to embodiments of the present disclosure.

[Fig. 2B]

FIG. 2B is a schematic illustrating a PRIOR ART partitioning method.

[Fig. 3A]

FIG. 3A is a schematic illustrating a process for using data along a line to calculate where to perform the data-dependent non-uniform split, wherein given a block or patch of data, data along the vertical halfway split position is scanned horizontally by a sliding window of data points, which is used to determine where to split the current block horizontally, i.e. by a vertical line, e.g. into a left sub-block and a right sub-block for the case of partitioning a 2-D grid, according to embodiments of the present disclosure.

[Fig. 3B]

FIG. 3B is a schematic illustrating a process for using data along a line to calculate where to perform the data-dependent non-uniform split, wherein given a block or patch, data along the horizontal split position or along a line orthogonal to the vertical split position is scanned vertically by a sliding window of data points, which is used to determine where to split the current block vertically, i.e. by a horizontal line, e.g. into a lower sub-block and an upper sub-block for the case of partitioning a 2-D grid according to embodiments of the present disclosure.

[Fig. 4A]

FIG. 4A is a schematic illustrating a process for using a multidimensional sliding window of data points to calculate where to perform the data-dependent non-uniform split, wherein the data is a region oriented along a horizontal split position, e.g. a vertical rectangle, according to embodiments of the present

disclosure.

[Fig. 4B]

FIG. 4B is a schematic illustrating a process for using a multidimensional sliding window of data points to calculate where to perform the data-dependent non-uniform split, wherein the data is a region oriented along a vertical split position, e.g. a horizontal rectangle, according to embodiments of the present disclosure.

[Fig. 5A]

FIG. 5A is a schematic illustrating a 3D point cloud, according to embodiments of the present disclosure.

[Fig. 5B]

FIG. 5B is a schematic illustrating a range map or depth map which represents the distance of each point from a plane or the elevation or depth of each point is used as data input to the scoring function, according to embodiments of the present disclosure.

[Fig. 5C]

FIG. 5C is a schematic illustrating the output of a scoring function applied to a point cloud or range map, in which higher score values are illustrated using lighter colors, according to embodiments of the present disclosure.

[Fig. 5D]

FIG. 5D is a schematic illustrating an example of PRIOR ART uniform splitting.

[Fig. 5E]

FIG. 5E is a schematic illustrating a method of data-dependent

non-uniform splitting, according to embodiments of the present disclosure.

[Fig. 5F]

FIG. 5F is a schematic illustrating a method of data-dependent non-uniform splitting, according to embodiments of the present disclosure.

[Fig. 6]

FIG. 6 is a flow diagram illustrating a list of some steps for implementing a splitting process, according to embodiments of the present disclosure.

[Fig. 7]

FIG. 7 is a block diagram illustrating some steps for decoding data from a bit-stream, when data-dependent non-uniform splitting is used, according to embodiments of the present disclosure.

[Fig. 8A]

FIG. 8A is a schematic illustrating a method for organizing an unorganized point cloud, according to embodiments of the present disclosure.

[Fig. 8B]

FIG. 8B is a schematic illustrating an organizational grid or a matrix for example in 2D space, with each position in the grid having an index, according to embodiments of the present disclosure.

[Fig. 8C]

FIG. 8C is a schematic illustrating a scanning process for a method for organizing an unorganized point cloud, according to embodiments of the present disclosure.

[Fig. 8D]

FIG. 8D is a schematic illustrating a mapping process for a method for

organizing an unorganized point cloud, according to embodiments of the present disclosure.

[Description of Embodiments]

[0020]

While the above-identified drawings set forth presently disclosed embodiments, other embodiments are also contemplated, as noted in the discussion. This disclosure presents illustrative embodiments by way of representation and not limitation. Numerous other modifications and embodiments can be devised by those skilled in the art which fall within the scope and spirit of the principles of the presently disclosed embodiments.

[0021]

Predicting from center or predetermined location in block

FIG. 1A is a block diagram illustrating steps of a first point prediction method 100A, according to embodiments of the present disclosure. For example, instead of signaling a coordinate of each point relative to an origin or corner of a region, a component-wise difference between each point and a predetermined location in the block can be signaled. For instance, if the point position is (x, y, z) and the predetermined location in the block is defined as the center of the block (x_c, y_c, z_c) , then $(x - x_c, y - y_c, z - z_c)$ is signaled to the bit-stream.

[0022]

Step 110 of FIG. 1A includes decoding a point cloud, using a processor connected to a memory, to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud, wherein each block includes a set of points

[0023]

Step 115 of FIG. 1A includes decoding a set of prediction residuals from a compressed bitstream for each block by a processor.

[0024]

Step 120 of FIG. 1A includes using a predetermined location in the block.

[0025]

Step 125 of FIG. 1A includes computing for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block, wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[0026]

FIG. 1B is a block diagram 100B illustrating some components implementing the methods of the present disclosure, according to embodiments of the present disclosure. For example, a processor 12 can be for methods, encoder or decoders of the present disclosure, that is connected to a memory 10. The processor 12 can be connected to a network-enabled server 14, which is connected to a client device 16.

[0027]

Predicting a chain or sequence of points from previously-decoded points

According to the present disclosure another method can include a second point prediction method. The second method discloses that given a set of point locations (x_n, y_n, z_n) , $n = 1, 2, \dots, N$, the point locations (x_n, y_n, z_n) can be signaled to the bit-stream. If predicting from the center or predetermined location in the block, then for each point location (x_n, y_n, z_n) , a prediction residual

$(x_n - x_c, y_n - y_c, z_n - z_c)$ is signaled to the bit-stream. In some cases, such as when groups of points are clustered near each other, the prediction residual can be reduced by using previously-decoded points as predictors for a currently-decoded point. For example, the first point $n=1$ can be signaled as (x_n, y_n, z_n) or $(x_n - x_c, y_n - y_c, z_n - z_c)$. The second point $n=2$ can be signaled as $(x_2 - x_1, y_2 - y_1, z_2 - z_1)$. The third point can be signaled as $(x_3 - x_2, y_3 - y_2, z_3 - z_2)$, etc. for all n .

[0028]

For example, the second method can be for decoding a point cloud, including using a processor connected to a memory, to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud. Wherein each block includes a set of points, such that for each block the processor is for decoding a set of prediction residuals from a compressed bitstream. Using a predetermined location in the block, and computing for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block. Wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[0029]

According to the present disclosure another method can include a third point prediction method. The third method discloses that the first decoded point (x_1, y_1, z_1) can be used to predict subsequent points, so the prediction residual signaled for point $n=2$ is $(x_2 - x_1, y_2 - y_1, z_2 - z_1)$, and for point $n=3$ $(x_3 - x_1, y_3 - y_1, z_3 - z_1)$, and so on for all n .

[0030]

For example, the third method includes a point cloud decoder including a memory having data stored including previously decoded points. A processor to decode a sequence of points from a compressed bitstream, so as to obtain a decoded point cloud. The processor is to decode a sequence of prediction residuals from the compressed bitstream. Compute for each prediction residual in the sequence of prediction residuals, a position of a point by adding the prediction residual to a position of a previously decoded point stored in the memory, so as to obtain a sequence of decoded points. Wherein the sequence of decoded points represents the decoded point cloud.

[0031]

Reordering of points or sequence of points to reduce magnitude or energy in prediction residual

Given a sequence of points (x_n, y_n, z_n) , $n=1, 2, \dots, N$ in a block or region, the points can be reordered such that the distance between subsequent points is minimized. Doing so reduces the magnitude or energy of the prediction residuals signaled for the case when a chain or sequence of points is signaled.

[0032]

Reordering of points to increase frequency of occurrence of identical residuals

Given a sequence of points (x_n, y_n, z_n) , $n=1, 2, \dots, N$ in a block or region, the points can be reordered such that the number of different distances between subsequent points is minimized. For example, for one coordinate, if the sequence of differences between five successive points is 1, 4, 4, 1, but if the points can be reordered such as the differences between the successive points becomes 3, 3, 3, 3,

then although the total of the differences after reordering is higher than the total before reordering, the number of different distances is reduced. When a coder such as an entropy coder is used to signal these differences or prediction residuals, the residuals generated after reordering points can therefore have a reduced entropy, thus decreasing the number of bits that would need to be signaled to the bit-stream.

[0033]

Skip encoding and/or decoding of a point if it is identical to or within a threshold distance from an already-decoded point

When encoding a set of points and signaling them to a bit-stream, if a given point is collocated or within a threshold distance from a previously-signaled point, then that point is not signaled to the bit-stream.

[0034]

In another embodiment, if a given decoded point is collocated or within a threshold distance from a previously-signaled point, then that point is not reconstructed, i.e. it is not included in the reconstructed point cloud. The total number of reconstructed points in the decoded point cloud can therefore be reduced, thus reducing memory or storage requirements and improving rendering speeds due to the reduced number of points that would need to be rendered.

[0035]

In another embodiment, if a given point is outside a bounding box or minimum and maximum coordinate limits, then it is not signaled to the bit-stream.

[0036]

In another embodiment, if a given decoded point is outside a bounding

box or minimum and maximum coordinate limits, then it is not included in the reconstructed point cloud.

[0037]

Data-dependent non-uniform partitioning

Referring to FIG. 2A, the data-dependent non-uniform partitioning 202 can be performed according to embodiments of this present disclosure. The data-dependent non-uniform partitioning 202 can be based upon measures or calculations based on the data representing the object, along with information about the current partitioning structure, such as a minimum or maximum block size, a function can specify how or where to split the current block or sub-block being processed. For example, in 202 of FIG. 2A, instead of splitting at the halfway positions and breaking the object among different blocks as shown in the prior art method 201 of FIG. 2B, the first split creates four sub-blocks or sub-partitions, but only one of them, the lower-right one, is occupied by object data (see FIG. 2A), according to the present disclosure. Similarly, FIG. 2A shows the second split partitions with the lower-right block into four sub-blocks without splitting the object data among different sub-blocks, according to the present disclosure.

[0038]

If each point in 3D space is mapped or indexed to a position on a 2D grid, then this data-dependent non-uniform partitioning method can be applied to the 2D grid of indices.

[0039]

FIG. 2B is a schematic illustrating a PRIOR ART partitioning method for

partitioning a 3D space as in an octree. In particular, FIG. 2B shows the partitioning in 2D space, with a quadtree as shown in 201. The outside border in 201 denotes the initial block or partitioning of the space. The star is an example of an object represented by the data in the block. A decision process is used to decide whether to split the initial block into 2^N blocks or partitions, where N is the dimension of the space ($N=2$ for 2D space and $N=3$ for 3D space). Each partition is split in half vertically and horizontally into 2^N blocks. This splitting process repeats for each sub-block or sub-partition until a desired stopping criteria is met.

[0040]

FIG. 3A is a schematic illustrating a process for using data along a line to calculate where to perform the data-dependent non-uniform split, wherein given a block or patch 303, data along the vertical halfway split position 305 is scanned horizontally by a sliding window 307 of data points, which means splitting the current block horizontally, i.e. by a vertical line 311, e.g. into a left sub-block and a right sub-block for the case of partitioning a 2-D grid, according to embodiments of the present disclosure. The skip area 313 is on one side of the search area 307, used as input to a scoring function, where skip area 315 is on the other side of the search area 307, such that the width of the skip area 317 can be set to $1/3$ the width of the patch of interest.

[0041]

FIG. 3B is a schematic illustrating a process for using data along a line to calculate where to perform the data-dependent non-uniform split, such that given a block or patch 323, data along the horizontal split position 325 is scanned

vertically by a sliding window 327 of data points, which means splitting the current block vertically, i.e. by a horizontal line 331, e.g. into a lower sub-block and an upper sub-block for the case of partitioning a 2-D grid according to embodiments of the present disclosure.

[0042]

Referring to FIG. 3A and FIG. 3B, are schematics or diagrams of at least one process for using data along a line to calculate where to perform the data-dependent non-uniform split. For example, FIG. 3A shows a block or patch 303, where data along the vertical halfway split position 305 is scanned horizontally by a sliding window 307 of data points. These data points can be data related to the position of the points in the point cloud, such as the distance or depth from a plane, or it can be related to attributes in the data, such as brightness or color. The sliding window contains M points, and data across the vertical half-way position is scanned horizontally (FIG. 3A), i.e. M points are selected and are used as input to a function such as a scoring function. To prevent blocks from becoming too thin, minimum border distance or skip area 315 and 313 can be defined, for example 317, in which the width of a skip area can be set to $\frac{1}{3}$ the width of the patch, which ensures a minimum block size. The sliding window can slide one data point position at a time, or it can skip points, e.g. every other point, to simplify computation. For each window for which a score is computed, the best score, typically a maximum or minimum score, is located, and the position 309 corresponding to that best score is indicated as being the split position for splitting the current block horizontally, i.e. by a vertical line 311, e.g. into a left sub-block and a right sub-block for the case of partitioning a 2-D grid.

[0043]

Referring to FIG. 3B, if it is desired to split the same block in the other direction, e.g. vertically, this process can be repeated in a direction orthogonal to the initial direction. This will result in a vertical split, i.e. a splitting by a horizontal line 331, into an upper block or blocks and lower block or blocks. The output of this process is the position of each split, for example the location 329 of the maximum score resulting from a scoring function, or an index that can be mapped to the position of each split, for example, the position of the split line 331 relative to a corner of the patch. If a 2D block is split both horizontally and vertically, then the block is partitioned into four sub-blocks. If only horizontal or only vertical partitioning is done, then the block is partitioned into two sub-blocks. To prevent blocks from becoming too thin, minimum border distance or skip area 333 and 335 can be defined, for example 337, in which the height of a skip area can be set to $1/3$ the height of the patch, which ensures a minimum block size. The sliding window can slide one data point position at a time, or it can skip points, e.g. every other point, to simplify computation. For each window for which a score is computed, the best score, typically a maximum or minimum score, is located, and the position 329 corresponding to that best score is indicated as being the split position for splitting the current block horizontally, i.e. by a horizontal line 331, e.g. into a top sub-block and a bottom sub-block for the case of partitioning a 2-D grid.

[0044]

FIG. 4A is a schematic illustrating a process for using multidimensional sliding window of data points to calculate where to perform the data-dependent

non-uniform split on a block or patch 403, wherein the data is a region oriented along the horizontal split position 411, e.g. vertical rectangle 419, according to embodiments of the present disclosure.

[0045]

FIG. 4B is a schematic illustrating a process for using multidimensional sliding window of data points to calculate where to perform the data-dependent non-uniform split on a block or patch 423, wherein the data is a region oriented along a vertical split position 431, e.g. a horizontal rectangle 439, according to embodiments of the present disclosure.

[0046]

Referring to FIG. 4A and FIG. 4B, are diagrams of at least one process for using multidimensional data to calculate where to perform the data-dependent non-uniform split. Instead of using a one-dimensional sliding window of data as input to a scoring function as is illustrated in FIG. 3A and FIG. 3B, in FIG. 4A and FIG. 4B, data from a multidimensional region, for example, a 2D region 419 or 439, can be used as input.

[0047]

Regarding FIG. 4A, to prevent blocks from becoming too thin, minimum border distance or skip area 415 and 413 can be defined, for example 417, in which the width of a skip area can be set to $1/3$ the width of the patch, which ensures a minimum block size. The sliding window can slide one data point position at a time, or it can skip points, e.g. every other point, to simplify computation. For each window for which a score is computed, the best score, typically a maximum or minimum score, is located, and the position 409

corresponding to that best score is indicated as being the split position for splitting the current block horizontally, i.e. by a vertical line 411, e.g. into a left sub-block and a right sub-block for the case of partitioning a 2-D grid.

[0048]

Regarding FIG. 4B, to also prevent blocks from becoming too thin, minimum border distance or skip area 433 and 435 can be defined, for example 437, in which the height of a skip area can be set to $1/3$ the height of the patch, which ensures a minimum block size. The sliding window can slide one data point position at a time, or it can skip points, e.g. every other point, to simplify computation. For each window for which a score is computed, the best score, typically a maximum or minimum score, is located, and the position 429 corresponding to that best score is indicated as being the split position for splitting the current block horizontally, i.e. by a horizontal line 431, e.g. into a top sub-block and a bottom sub-block for the case of partitioning a 2-D grid.

[0049]

FIG. 5A is a schematic illustrating a 3D point cloud 510, according to embodiments of the present disclosure.

[0050]

FIG. 5B is a schematic illustrating a range map or depth map 520 which represents the distance of each point from a plane or the elevation or depth of each point is used as data input to the scoring function, according to embodiments of the present disclosure.

[0051]

FIG. 5C is a schematic illustrating the output of a scoring function, for

example a graph signal processing (GSP) score map 530, in which higher score values 531 are illustrated using lighter colors, according to embodiments of the present disclosure. This scoring function, for example, can assign higher score values to areas of the patch, grid, or point cloud that exhibit discontinuities, as can be the case along edges of objects represented in 2D or 3D space.

[0052]

Resampling

Still referring to FIG. 5C, regarding resampling (to reduce the number of points in the block), given a region or block containing $P1$ points, the scoring function 530 can be used to identify a subset of $P2$ points whose score is less than or greater than a threshold. Subsequent processing of the region, block or partition then continues using the $P2$ points instead of the $P1$ points. Computational complexity of the system or the amount of data that is signaled to the bit-stream can be reduced when $P2 < P1$.

[0053]

FIG. 5D is a schematic illustrating a PRIOR ART uniform splitting method 540 having a number of patches at 52.

[0054]

FIG. 5E and FIG. 5F are schematics illustrating a method of data-dependent non-uniform splitting 550 of FIG. 5E and 560 of FIG. 5F, according to embodiments of the present disclosure. Wherein, FIG. 5E shows the locations of non-uniform splitting when using sliding windows 307 of FIG. 3A and 327 of FIG. 3B along lines of data when calculating maximum score points 309 of FIG. 3A and 329 FIG. 3B, resulting in 31 patches. Wherein FIG. 5F shows

the locations of non-uniform splitting when using sliding windows 407 of FIG. 4A and 427 FIG. 4B along regions or rectangles of data when calculating maximum score points 409 of FIG. 4A and 429 of FIG. 4A, resulting in 36 patches. Specifically, the data-dependent non-uniform splitting 550 of FIG. 5E and 560 of FIG. 5F, according to embodiments of the present disclosure show fewer patches, respectively, 31 patches for 550 of FIG. 5E, 36 patches for 560 of FIG. 5F, than that of the PRIOR ART method 540 of FIG. 5D, i.e. 52 patches.

[0055]

Referring to FIG. 5B, the range map or depth map 520 represents the distance of each point from a plane or the elevation or depth of each point is used as data input to the scoring function. In the score map 530, the point locations corresponding to the point cloud 510 or indices corresponding to the point cloud 510, or corresponding to the range map 520, are highlighted or illustrated with a lighter color where the scoring function was best, e.g. maximum. In this case, a graph signal processing (GSP) method was used to compute the scores. The locations at where the best scores occur in this example tend to correspond to edges or discontinuities in the point cloud 510 or range map 520.

[0056]

Referring to FIG. 5E and FIG. 5F, the optimal horizontal and vertical splitting locations 550 of FIG. 5E are shown when the methods of FIG. 3A and FIG. 3B are used. Also, the optimal horizontal and vertical splitting locations 560 of FIG. 5F are shown when the methods of FIG. 4A and FIG. 4B are used. The graph signal processing (GSP) method can be a fast resampling method. For example, when a particular graph filter is applied on the geometric information of

each point in the point cloud, the filtering results or output 530 of FIG. 5C can serve as scores reflecting the importance of a point or set of points input to the filter, in terms of geometric properties. If any other attributes are of interest in the importance evaluation, the same graph filter can be applied on them, and the final filtering score could be a weighted sum of all the filtering results on geometric position and attributes associated with the points.

[0057]

FIG. 6 is a flow diagram illustrating a list of some steps for implementing a splitting process 600, according to embodiments of the present disclosure. Some steps for implementing a splitting process, can include step 610 which illustrates a splitting function, i.e. "*Function Split (r)*", that has as input a block, partition or patch r , that has an initial dimension such as width and height.

[0058]

Referring to steps 615, 620 and 625, step 615 of FIG. 6A, shows that if the height is greater than the width, that this can be expressed as, "*if Height of a patch > Width of a patch*". Then, step 620 illustrates the split direction can be set to horizontal if the height is greater than the width, otherwise, the split direction can be set to vertical, as shown in step 625, i.e. "*choose split direction->Horizontal, else split direction->Vertical*".

[0059]

Step 630 of FIG. 6A, illustrates a scoring function is then computed, for example, a graph signal processing scoring function $GSPscoreScan(conv.width)$, where $conv.width$ indicates the size, or size and dimensions of the search area. Also, $conv.width$ can include other information, such as the size of the skip area.

The scoring function returns *max_score_location*, which is the location, e.g. a horizontal or vertical position or offset, of the location where the score is optimal, i.e. "*max_score_location = GSPscoreScan(conv. width)*". The block, partition or patch or region *r* can be split into two regions *r1* (step 635) and *r2* (step 640).

[0060]

Step 635 of FIG. 6A, illustrates the block, partition or patch or region (*r*) can be split into region 1 (*r1*), wherein *r1* corresponds to where the position in the region is to one side of the split location, i.e. *r1 = set of points in patch whose position with respect to the split direction is less than (<) to the max_score_location*.

[0061]

Step 640 of FIG. 6A, illustrates the block, partition or patch or region (*r*) that can be split into region 2 (*r2*), wherein *r2* corresponds to where the position is on the other side of the split location, i.e. *r2 = set of points in patch whose position with respect to the split direction is greater than or equal (>=) to the max_score_location*.

[0062]

Step 645 of FIG. 6A, illustrates that this splitting process can be repeated on the sub-partitions *r1*, for example, if the sub-partitions satisfy certain criteria, such as a minimum size or dimension, i.e. *dimensions of r1 > specified dimension or dimensions*.

[0063]

Step 650 of FIG. 6A, shows that if the sub-partitions satisfy the criteria that *r1* is greater than the specified dimension, then call *Function Split (r1)*, i.e. if

the dimensions of r1 are greater (>) than a specified dimension or dimensions, then call Function Split (r1).

[0064]

Step 655 of FIG. 6A, shows that if the sub-partitions do not satisfy the criteria, then the r1 splitting process is done, i.e. *done splitting r1*.

[0065]

Step 660 of FIG. 6A, illustrates that this splitting process can be repeated on the sub-partitions r2, for example, if the sub-partitions satisfy certain criteria, such as a minimum size or dimension, i.e. *dimensions of r2 > specified dimension or dimensions*.

[0066]

Step 665 of FIG. 6A, shows that if the sub-partitions satisfy the criteria that r2 is greater than the specified dimension, then call *Function Split (r1)*, i.e. *if the dimensions of r1 are greater (>) than a specified dimension or dimensions, then call Function Split (r1)*.

[0067]

Step 670 of FIG. 6A, shows that if the sub-partitions do not satisfy the criteria, then the r2 splitting process is done, i.e. *done splitting r2*.

[0068]

FIG. 7 is a block diagram illustrating some steps for decoding data from a bit-stream, when data-dependent non-uniform splitting is used, according to embodiments of the present disclosure. For example, FIG. 7 lists the syntax for a bit-stream used for decoding data, when data-dependent non-uniform splitting is used. The initial patch, region, or block size is decoded 701, e.g. a height and

width. A split flag for the region is decoded 702. The presence of a split is determined by determining 705 whether the split flag is true (yes) or false (no). For example, a flag value of 1 can indicate the presence of a split, and a flag value of 0 can indicate that no split is present. If the split flag is true (yes), e.g. 1, then the region will be split into sub-regions. If the split flag is false (no), e.g. 0, then the region will not be split further. If the split flag is 1, then the split location or split index which defines each sub-region is decoded 703. The split location can be an offset, absolute location or coordinate, an integer specifying an offset from a pre-specified position in the current region such as the corner of the region, a horizontal, vertical, or horizontal and vertical offset, or an index to a look-up table, which maps each index to a horizontal, vertical, or horizontal and vertical offset. The set of split flags can be considered as being a specification of a split tree, which defines the split hierarchical for the entire set of partitions or sub-partitions. If the split flag is false (no), e.g. 0, then the region is no longer split, and data associated with the region is decoded 704, then the method is done with decoding syntax for region or sub-region 709. This data can include control points for fitting patches, quantized prediction residuals, point location data, or attribute data. If the split flag is true (yes), e.g. 1, and thus sub-regions are defined, then for each sub-region output by 703, the syntax repeats for each sub-region starting from 702. It is not necessary to go back to 701 to specify a width and height, because the decoder can infer the width and height of each sub-region based upon the split locations.

[0069]

Still referring to FIG. 7, noted is that the split index includes, but is not

limited to:

- (1) An integer specifying an offset from a pre-specified position in the current block or grid;
- (2) A horizontal, vertical, or horizontal and vertical offset;
- (3) An index to a look-up table, which maps each index to a horizontal, vertical, or horizontal and vertical offset.

[0070]

Aligning/resampling to a grid

Given a region, block or partition containing arbitrarily-located points, the points are resampled or aligned to a predetermined subset of locations. For example, if the coordinates of each point comprise floating point numbers between 0.0 and 1.0, the coordinates can be aligned, such as through a quantization or repositioning process, such that all coordinates are multiples of 0.1. In this example, the points would therefore be aligned to a grid of resolution 0.1. This aligning or resampling therefore reduces the number of possible coordinates for each point, which reduces the number of possible different values for coordinates that would need to be signaled to the bit-stream.

[0071]

In another embodiment, the coordinates would not be signaled directly, but an index would be signaled. For example, if the coordinate of each point is greater than or equal 0.0 and is less than 1.0, and if a grid resolution is 0.5, then point coordinates in the region would be realigned to 0.0 or 0.5. In that case, a one-bit flag can be used to indicate whether a coordinate has been aligned to 0.0 or 0.5. If the grid resolution is 1.0, then nothing would need to be signaled to

indicate the realigned point coordinates, as they would be co-aligned with the origin or corner of the region.

[0072]

Method to organize an unorganized point cloud

FIG. 8A is a schematic illustrating a method for organizing an unorganized point cloud, according to embodiments of the present disclosure.

[0073]

FIG. 8B is a schematic illustrating an organizational grid or a matrix for example, in 2D space, with each position in the grid having an index, according to embodiments of the present disclosure;

[0074]

FIG. 8C is a schematic illustrating a scanning process for a method for organizing an unorganized point cloud, according to embodiments of the present disclosure;

[0075]

FIG. 8D is a schematic illustrating a mapping process for a method for organizing an unorganized point cloud, according to embodiments of the present disclosure;

[0076]

For example, the point cloud 801 of FIG. 8A can have N arbitrarily-located points at coordinates (x_n, y_n, z_n) , $n=1, 2, \dots, N$ and is denoted as being an unorganized point cloud. In this example, each point is labeled with a letter, A through G. An organizational grid 802 of FIG. 8B is defined as a matrix or grid, for example, in 2D space, with each position in the grid having an index.

For the 2D grid case, there is a horizontal index u 802 and vertical index v 803 of FIG. 8B. Each point in the point cloud 801 of FIG. 8A will be mapped, thus organized, to a position in the grid 802 of FIG. 8B. Therefore, each point in the point cloud can be indexed via (u,v) . Subsequent partitioning of the (u,v) space 802 therefore partitions the point cloud 801 into subgroups of points where each element in a given partition of (u,v) space corresponds to a subset of points in the point cloud 801.

[0077]

In one embodiment, the mapping of each point in the point cloud 701 to positions in the organizational grid 802 of FIG. 8B is predetermined or known before the point cloud is encoded.

[0078]

In another embodiment, a process maps each point of the point cloud 801 of FIG. 8A to a position in the organizational grid 802 of FIG. 8B. This process can be a scanning process 810 of FIG. 8C. A position such as a fixed position in 3D space is designated as the scan origin 811 of FIG. 8C. The points in the point cloud 801 of FIG. 8A will be virtually scanned based on their position relative to the scan origin, in an order based upon a scan direction 812 of FIG. 8C. The scan direction can be specified for example by an angle θ relative to a fixed direction. There are J scans, with each scan having a radius or distance 813 of FIG. 8C from the scan origin r_j , $j=1, 2, \dots, J$. In the mapping example 820 of FIG. 8D, the point cloud is scanned 830, 832, 834 with radii r_1, r_2, r_3 , respectively. In this case, the number J of radii correspond to the number of elements V 804 of FIG. 8B in the v -direction of the organizational grid. All points in the point cloud within a given

threshold distance from the scan are associated with that scan and then are removed from future consideration for subsequent scans. The range of angles over which a scan operates is divided into U equal partitions where U is the number of elements 803 of FIG. 8B in the organizational grid in the u direction. Given a scan radius r_j , the scan 820 of FIG. 8D associated with that radius can be divided into U partitions. These partitions can be equally spaced with respect to the scan direction 820. For all points in the point cloud that have been associated with scan r_j , each of those points is assigned to an index position u on the organization grid corresponding to the partition of the scan. Larger values of U would correspond to either longer scans or scans having smaller partitions, e.g. finer resolution scans. In 820, points B and D are associated with the first scan, and are placed in the corresponding row $v=1$ of the organizational grid. Points A and C are associated with the second scan, and points E, F, and G are associated with the third scan. The final mapping 821 of FIG. 8D of each point in the point cloud to the organizational grid allows each point in the 3D point cloud to be indexed by a location (u,v) in the organizational grid. Empty positions in the organizational grid are not mapped to a point in the point cloud. By mapping in this way, points that are adjacent to each other in 3D space are likely to be adjacent to each other in the organizational grid. Selecting regions or groups of points in the organizational grid therefore are likely to select related groups of points in 3D space.

[0079]

In another embodiment, U and V are sufficiently large so every point in the point cloud is mapped to a unique position on the organizational grid.

[0080]

In another embodiment, more than one point can be mapped to the same position in the organizational grid. In this case, the set of points that are mapped to a given position in the organizational grid can be represented by a single point, such as the average of the point positions or some other function of the point positions.

[0081]

In another embodiment, the organizational grid may be sparse, i.e. have many empty positions. In this case, the organizational grid can be resampled or down sampled to reduce the number of empty positions. One example method to achieve this is by shifting or compacting non-empty positions toward one edge of the organizational grid. Another method can reduce the number of rows or columns in the organizational grid by grouping points eliminated rows or columns into the corresponding row or column of non-removed rows or columns. The non-removed elements of the organizational grid that contain multiple points can be replaced with single points as described earlier.

[0082]

A method for encoding and decoding a point cloud of representing a scene using an encoder and decoder including a processor in communication with a memory includes steps of performing data-dependent non-uniform partitioning, resampling to reduce the number of points in a block, aligning and resampling points to a grid, predicting points from the center or predetermined location in a block, predicting a chain or sequence of points from previously-decoded points, reordering the points in a sequence of point cloud points such as the distance

between successive points is minimized, reordering the point cloud points in a sequence such that the frequency of occurrence of identical differences between points is maximized, skipping the encoding or decoding of a point if it is identical to or within a specified threshold distance from an already-decoded point, and organizing an unorganized point cloud by mapping each point or groups of points to positional elements on an organizational grid.

[CLAIMS]

[Claim 1]

A point cloud decoder, comprising:

a processor to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud, wherein each block includes a set of points, such that for each block the processor is to

decode a set of prediction residuals from a compressed bitstream;

use a predetermined location in the block;

compute for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block,

wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[Claim 2]

The point cloud decoder of claim 1, wherein a decoded point is excluded from the sequence of decoded points representing the decoded point cloud, if the decoded is identical to, or within a predetermined threshold distance, from a previously decoded point in the sequence of decoded points representing the decoded point cloud.

[Claim 3]

The point cloud decoder of claim 1, wherein the compressed bitstream is produced by a point cloud encoder, such that the encoder includes a processor to encode a block of points from a point cloud, so as to obtain an encoded point cloud, wherein the point cloud includes a set of blocks and each block includes a

set of points, the processor is to use a predetermined location in the block, compute for each point, a difference between a position of the point to the predetermined location, so as to obtain a set of prediction residuals for the set of points in the block, and a transmitter to transmit the set of prediction residuals over a compressed bitstream.

[Claim 4]

The point cloud decoder of claim 3, wherein the points in each block are processed according to an order, in which, the points were acquired by a sensor.

[Claim 5]

The point cloud decoder of claim 3, wherein the encoding of a point is skipped if the point is identical to, or within a predetermined threshold distance, from a previously encoded point.

[Claim 6]

The point cloud decoder of claim 3, wherein the encoding of a point is skipped, if the point when decoded, is identical to, or within, a predetermined threshold distance, from a previously decoded point.

[Claim 7]

The point cloud decoder of claim 3, further comprising:
organize the point cloud by mapping each point from the set of points to positional elements on an organizational grid.

[Claim 8]

The point cloud decoder of claim 7, wherein the set of points are scanned according to an angle and a radius, such that each point in the set of points is mapped to a two-dimension (2D) grid,

wherein a first dimension of the 2D grid is according to the angle, such that as the angle increases, the position on the grid along the first dimension also increases,

wherein a second dimension of the 2D grid is according to the angle, such that as the radius increases, the position on the grid along the second dimension also increases.

[Claim 9]

The point cloud decoder of claim 3, further comprising:

perform data-dependent non-uniform partitioning of a region having a set of points, such that a location of the partitioning across at least one dimension of the region is determined by a scoring function,

wherein an input to the scoring function is a subset of points within a search area, and the location of the point that corresponds to where an output of the scoring function is maximized, corresponds to the location of the partitioning of the region across the dimension.

[Claim 10]

The point cloud decoder of claim 9, wherein the output of the scoring function, which has as an input a subset of points associated with an area of the point cloud exhibiting discontinuities, are score values higher than the output of the scoring function which has as an input a subset of points associated with areas of the point cloud that do not exhibit discontinuities.

[Claim 11]

The point cloud decoder of claim 3, further comprising:

perform resampling of the points in the block of points, wherein the

resampling is to reduce a number of points in the set of points of the block to a subset of points, and is based on a resampling scoring function, such that the resampling scoring function identifies the subset of points, such that if output of the scoring function is higher than a predetermined threshold, then the points are included in the subset.

[Claim 12]

The point cloud decoder of claim 3, further comprising:
align the points in the set of points to a grid resolution, and resample points to a predetermined subset of locations or a grid.

[Claim 13]

A method for decoding a point cloud, comprising:
using a processor connected to a memory, to decode each block in a set of blocks from a point cloud, so as to obtain a decoded point cloud, wherein each block includes a set of points, such that for each block the processor is for
decoding a set of prediction residuals from a compressed bitstream;
using a predetermined location in the block;
computing for each prediction residual in the set of prediction residuals, a position of a point by adding the prediction residual to the predetermined location, so as to obtain a set of decoded points for the block,
wherein the decoded points for the blocks in the set of blocks represent the decoded point cloud.

[Claim 14]

A point cloud decoder, comprising:
a memory having data stored including previously decoded points;

a processor to decode a sequence of points from a compressed bitstream, so as to obtain a decoded point cloud, the processor is to

decode a sequence of prediction residuals from the compressed bitstream;

compute for each prediction residual in the sequence of prediction residuals, a position of a point by adding the prediction residual to a position of a previously decoded point stored in the memory, so as to obtain a sequence of decoded points,

wherein the sequence of decoded points represents the decoded point cloud.

[Claim 15]

The decoder of claim 14, wherein the previously decoded point corresponds to a point in the sequence of points that is decoded immediately before the point in the sequence.

[Claim 16]

The decoder of claim 14, wherein the previously decoded point corresponds to the first point in the sequence of points that is decoded.

[Claim 17]

The decoder of claim 14, wherein a decoded point is excluded from the sequence of decoded points representing the decoded point cloud, if the decoded is identical to, or within a predetermined threshold distance, from a previously decoded point in the sequence of decoded points representing the decoded point cloud.

[Claim 18]

The decoder of claim 14, further comprising:

compute for a first prediction residual in the sequence of prediction residuals, a position of a first decoded point of the sequence of decoded points, by adding the first prediction residual, to a predetermined location in a three dimensional (3D) space, so as to obtain the first decoded point of the sequence of decoded points.

[Claim 19]

The decoder of claim 18, wherein the 3D space is an N dimensional space.

[Claim 20]

The decoder of claim 18, wherein the predetermined location is a center of the block.

[Claim 21]

The point cloud decoder of claim 14, wherein the compressed bitstream is produced by a point cloud encoder, such that the encoder includes a processor to encode a sequence of points from a point cloud, so as to obtain an encoded point cloud, wherein the processor is to compute for each point, a difference between a position of the point to a previously decoded point stored in the memory, so as to obtain a sequence of prediction residuals for the sequence of points, and a transmitter to transmit the sequence of prediction residuals over a compressed bitstream.

[Claim 22]

The decoder of claim 21, further comprising:

reorder the points in a sequence of point cloud points, so that a distance

between successive points is less than that of the point cloud prior to reordering.

[Claim 23]

The decoder of claim 21, further comprising:

reorder the point cloud points in a sequence, such that a frequency of occurrence of identical differences between points is greater than the frequency of occurrence of identical differences between points prior to reordering.

[Claim 24]

The decoder of claim 21, wherein the order of points in the sequence are according to an order, in which, the points were acquired by a sensor.

[Claim 25]

The decoder of claim 21, wherein the encoding of a point is skipped if the point is identical to, or within a predetermined threshold distance, from a previously encoded point.

[Claim 26]

The decoder of claim 21, wherein the encoding of a point is skipped if the point when decoded is identical to, or within a predetermined threshold distance, from a previously decoded point.

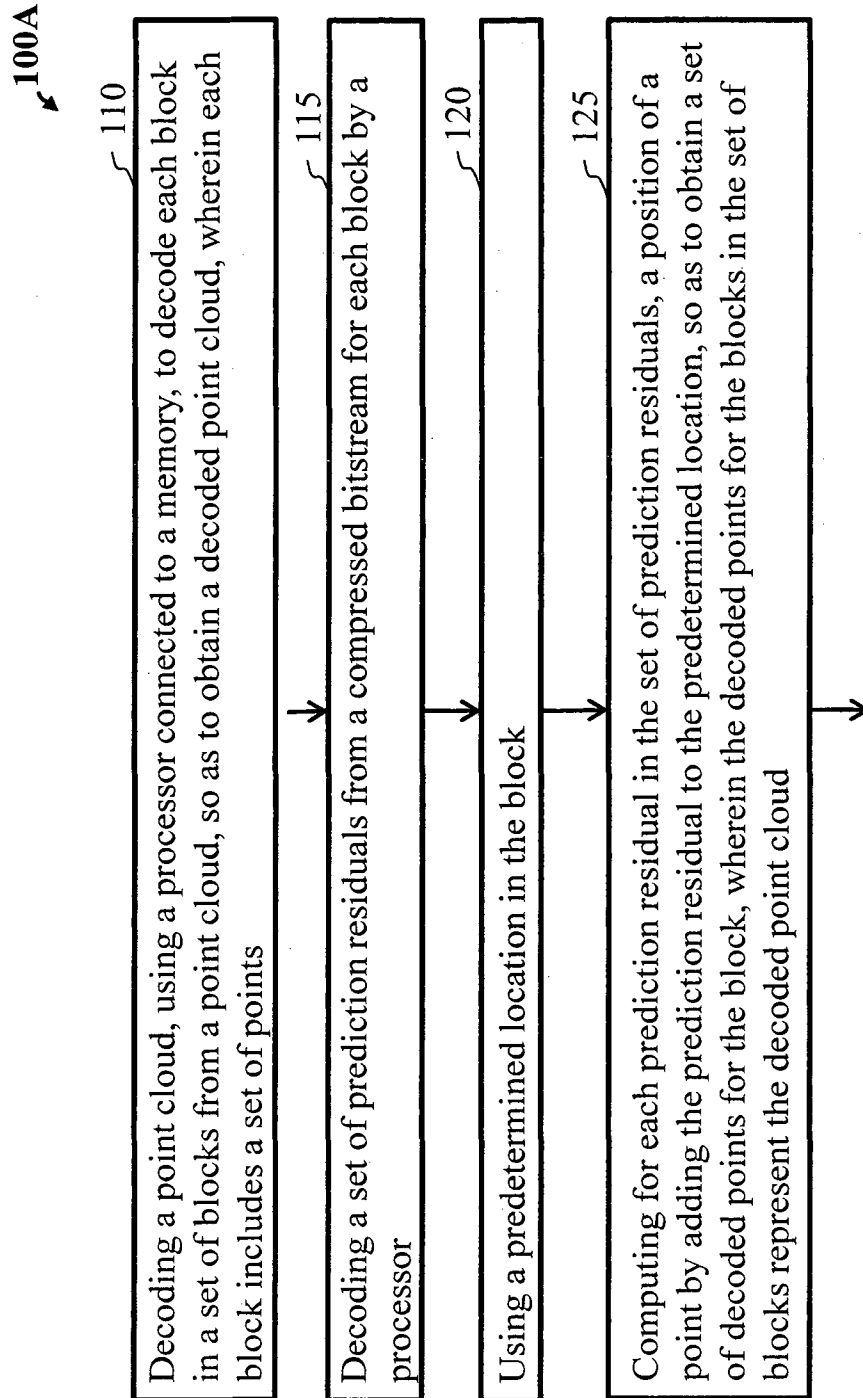


FIG. 1A

100B

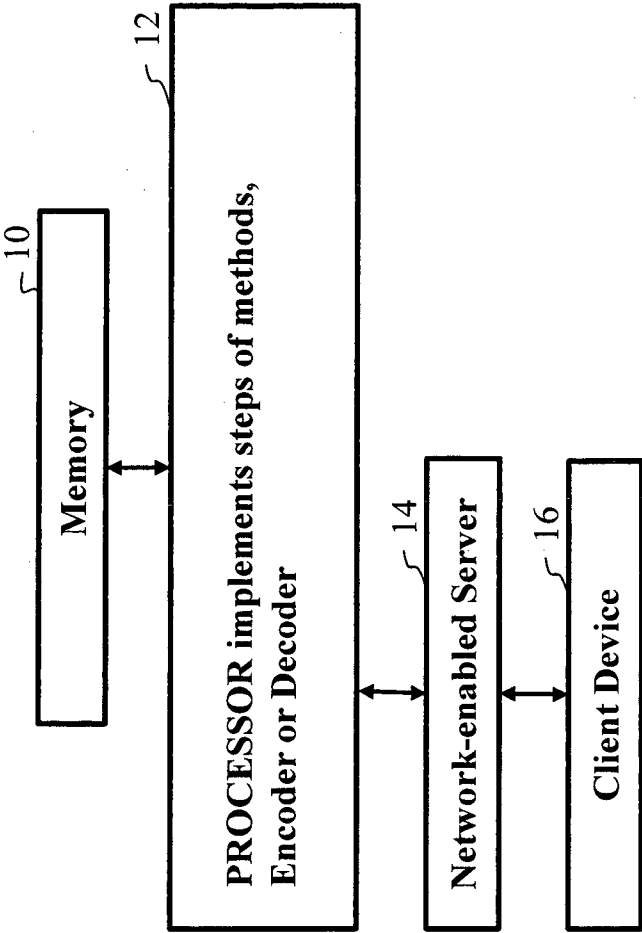
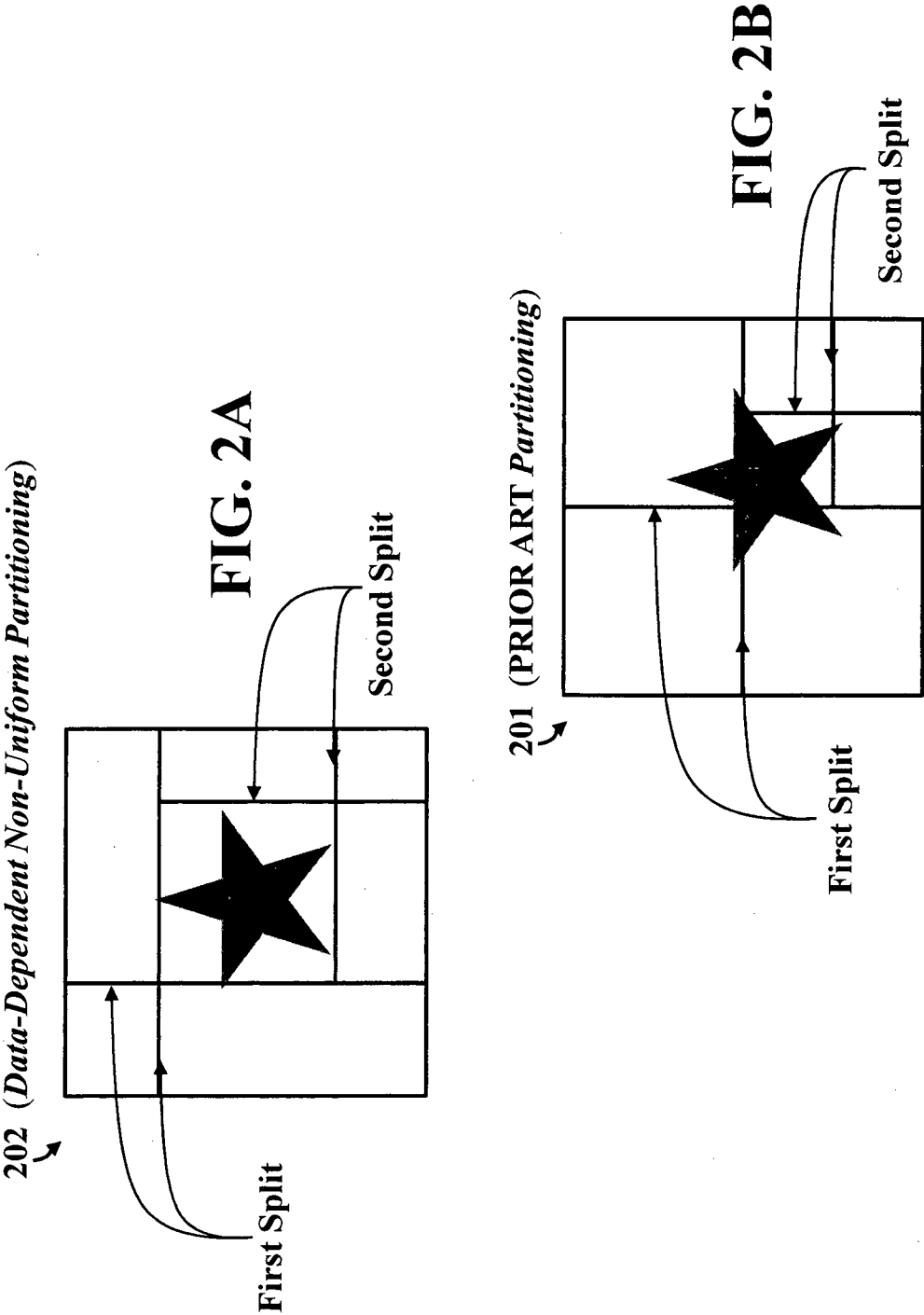


FIG. 1B



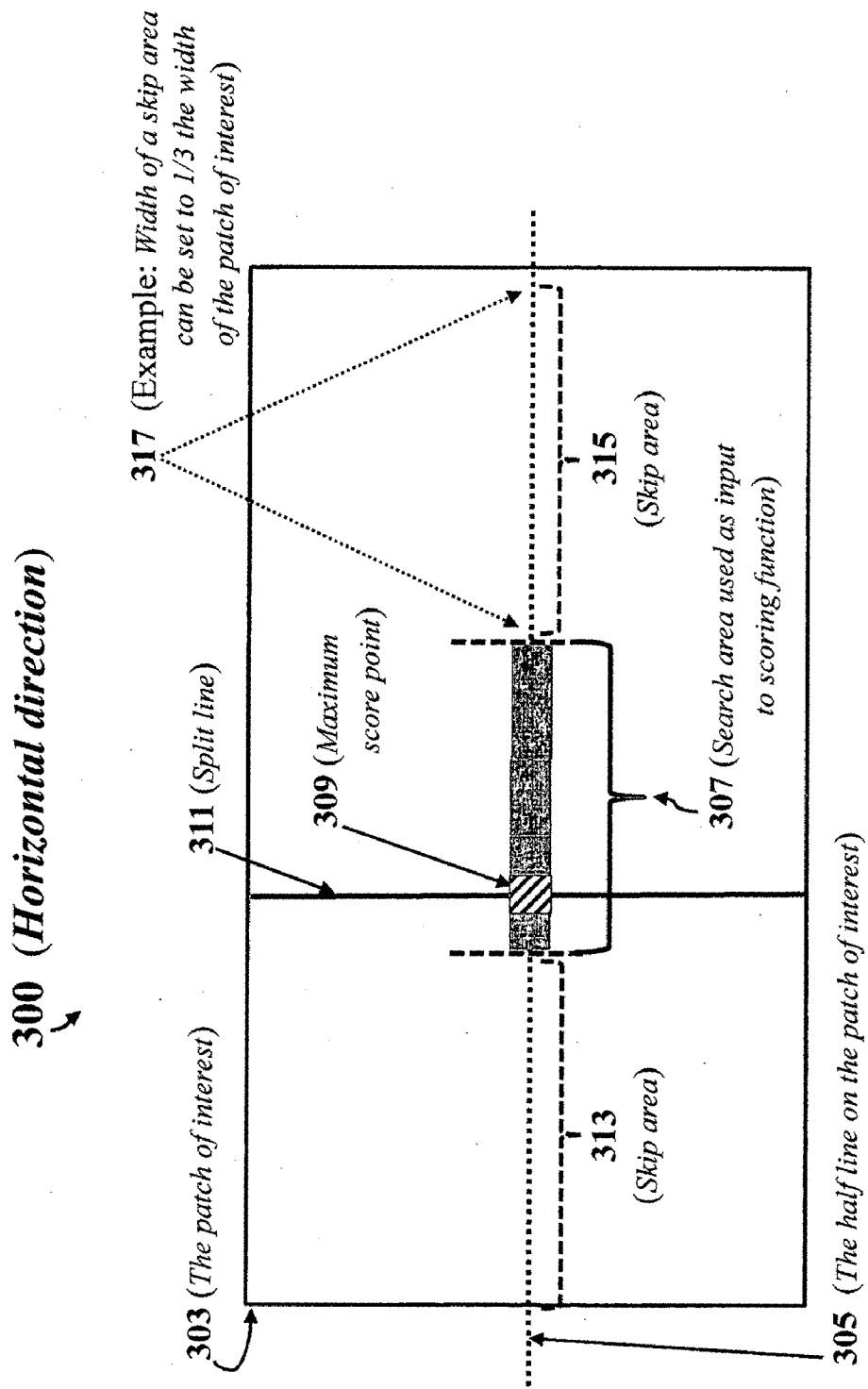


FIG. 3A

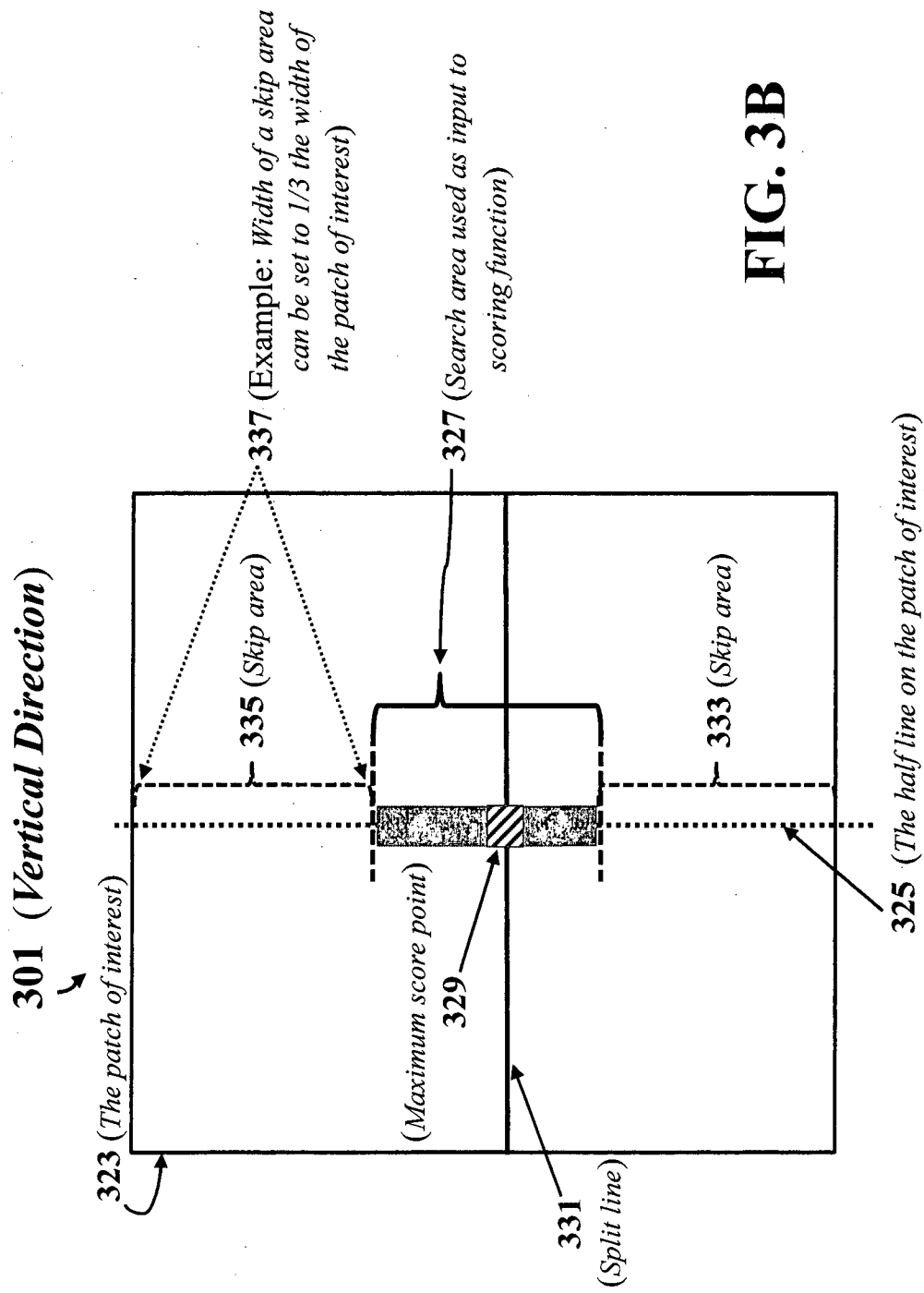


FIG. 3B

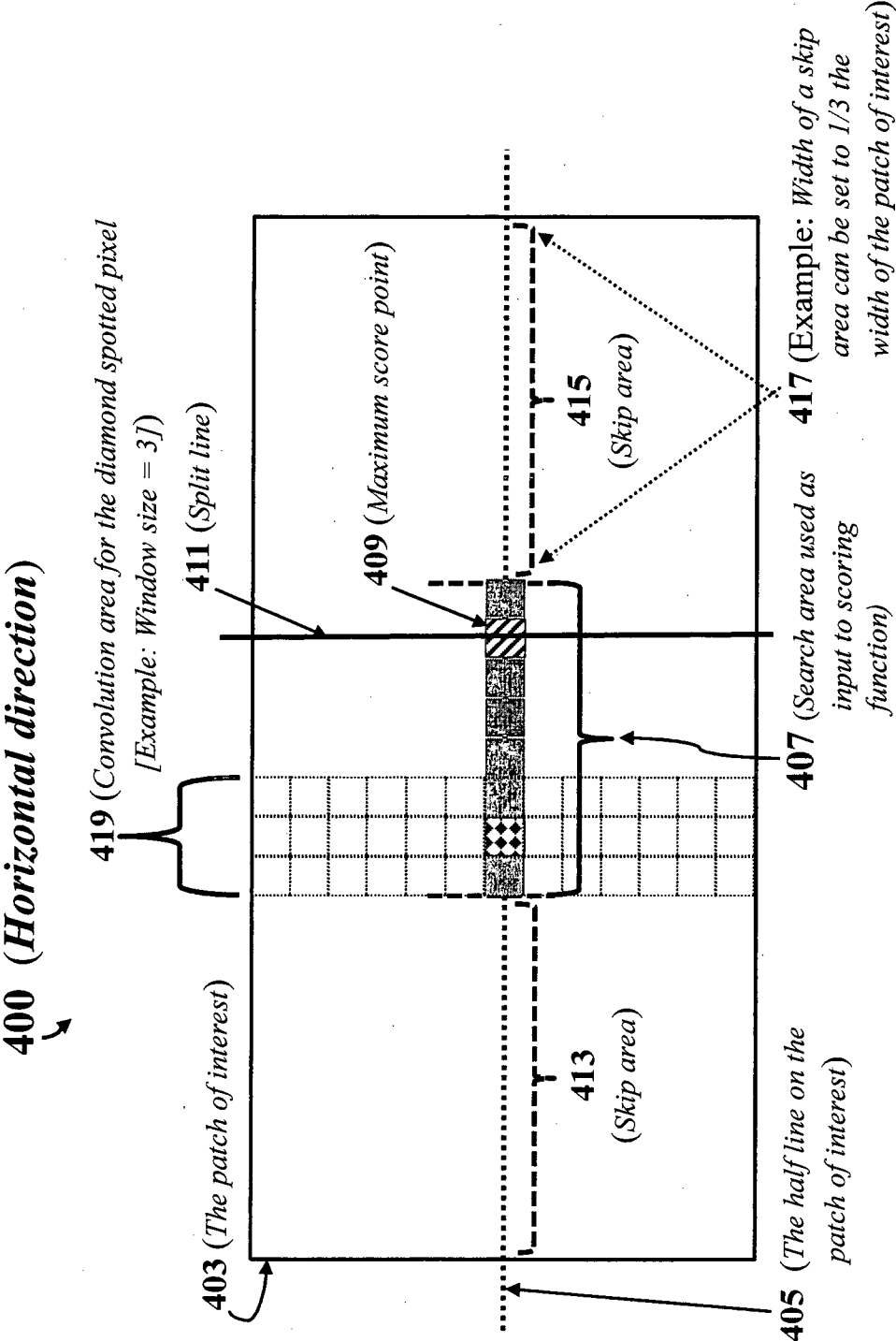


FIG. 4A

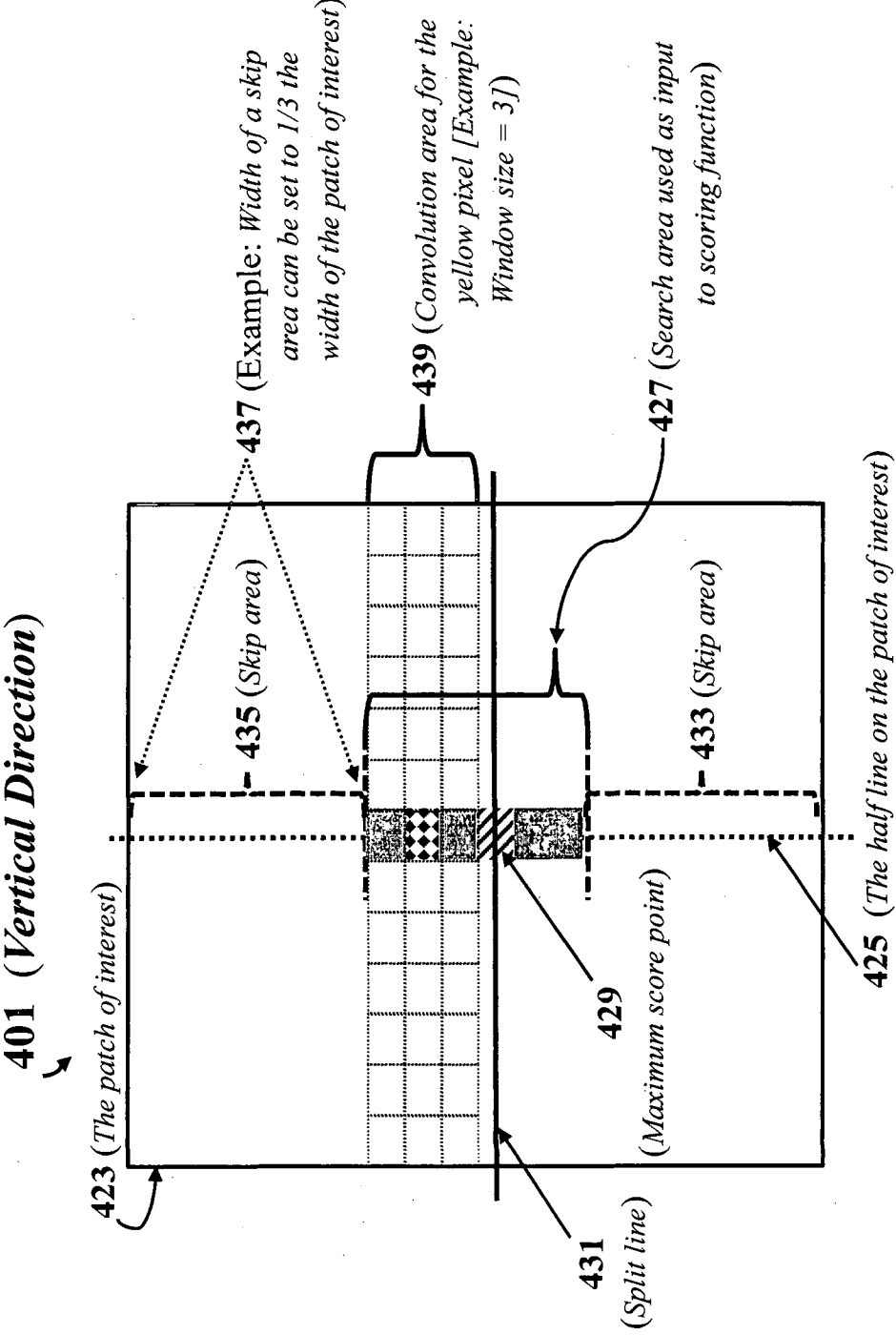
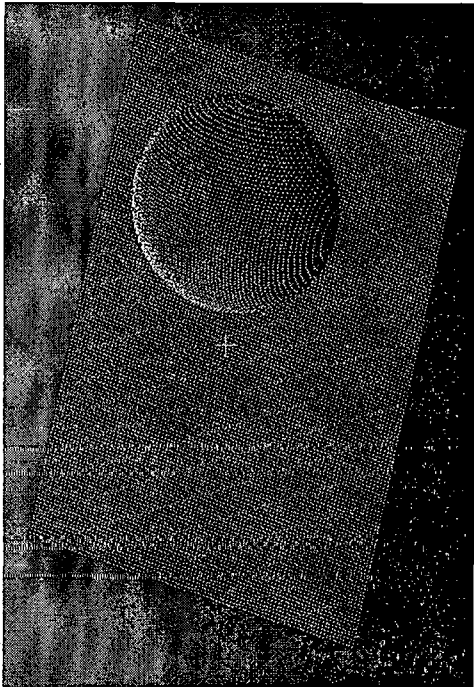


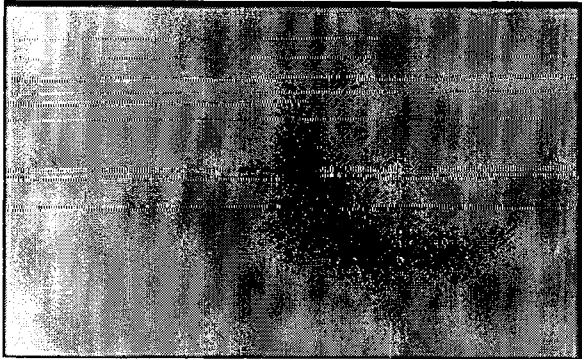
FIG. 4B



510 (3D Point Cloud)

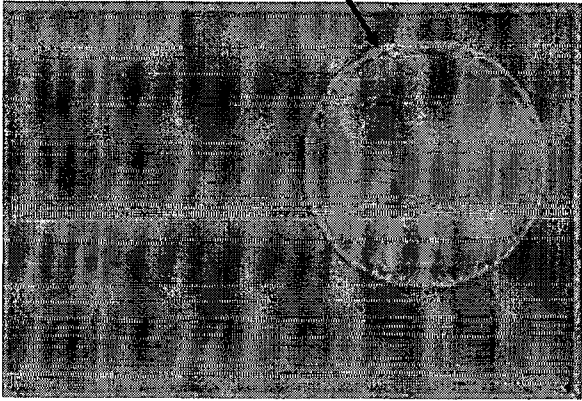
FIG. 5A

530 (GPS Score Map)



520 (Range Map)

FIG. 5B



531 (High Score Part)

FIG. 5C

540 (Prior Art Uniform Splitting)

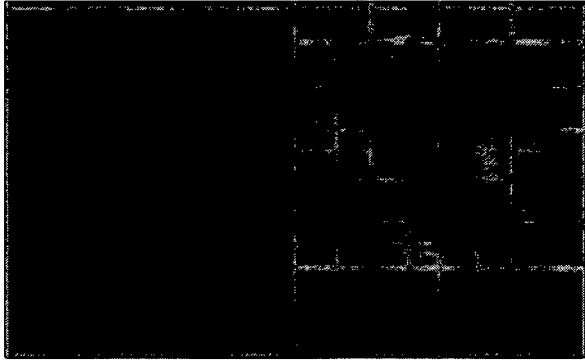


FIG. 5D
(52 patches)

550 (Data-Dependent
Non-Uniform Splitting)

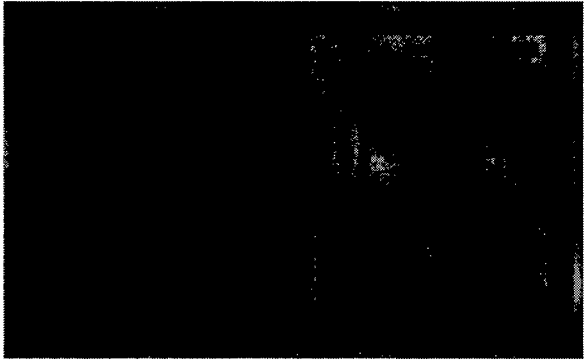


FIG. 5E
(31 patches)

560 (Data-Dependent
Non-Uniform Splitting)



FIG. 5F
(36 patches)

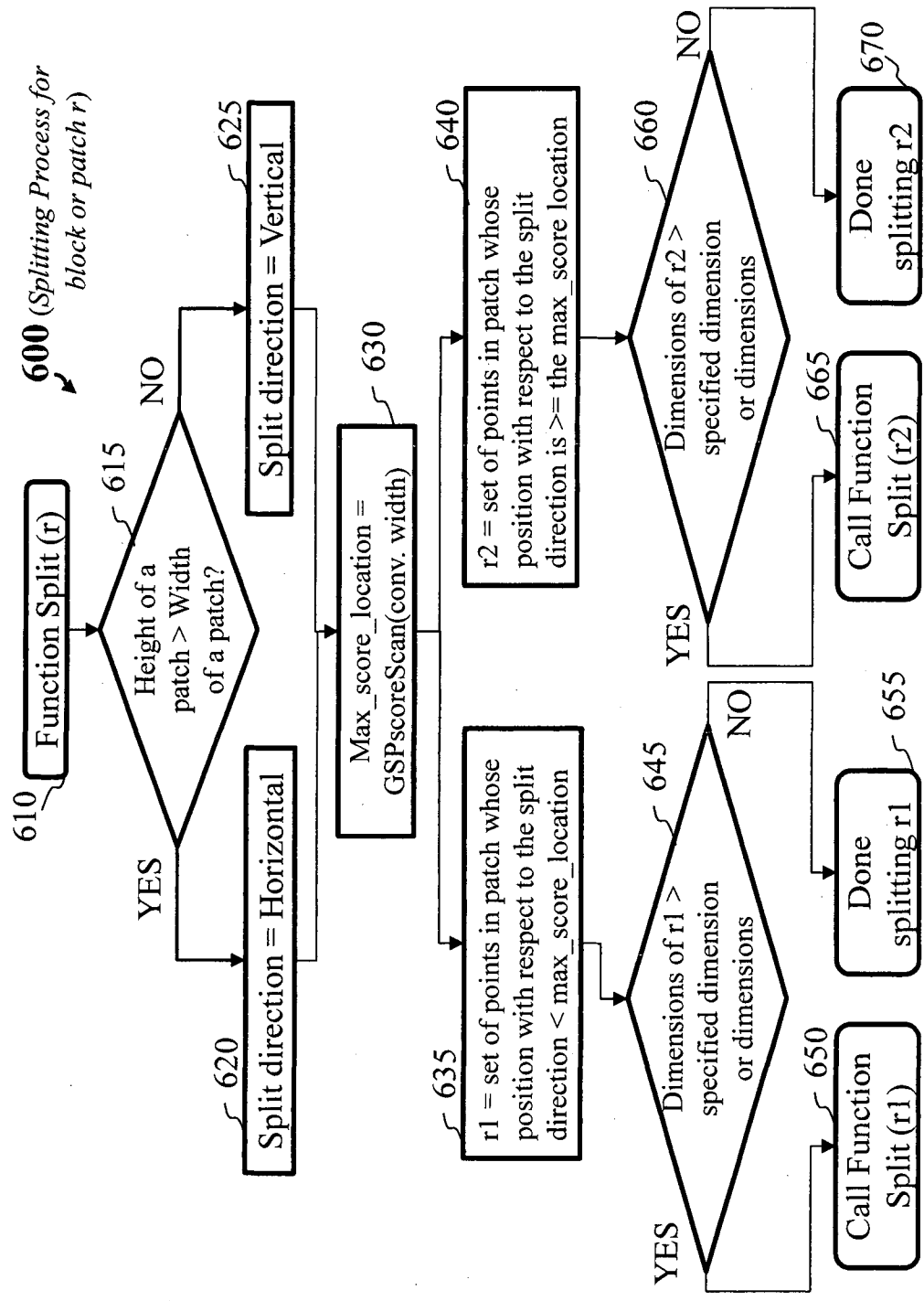


FIG. 6

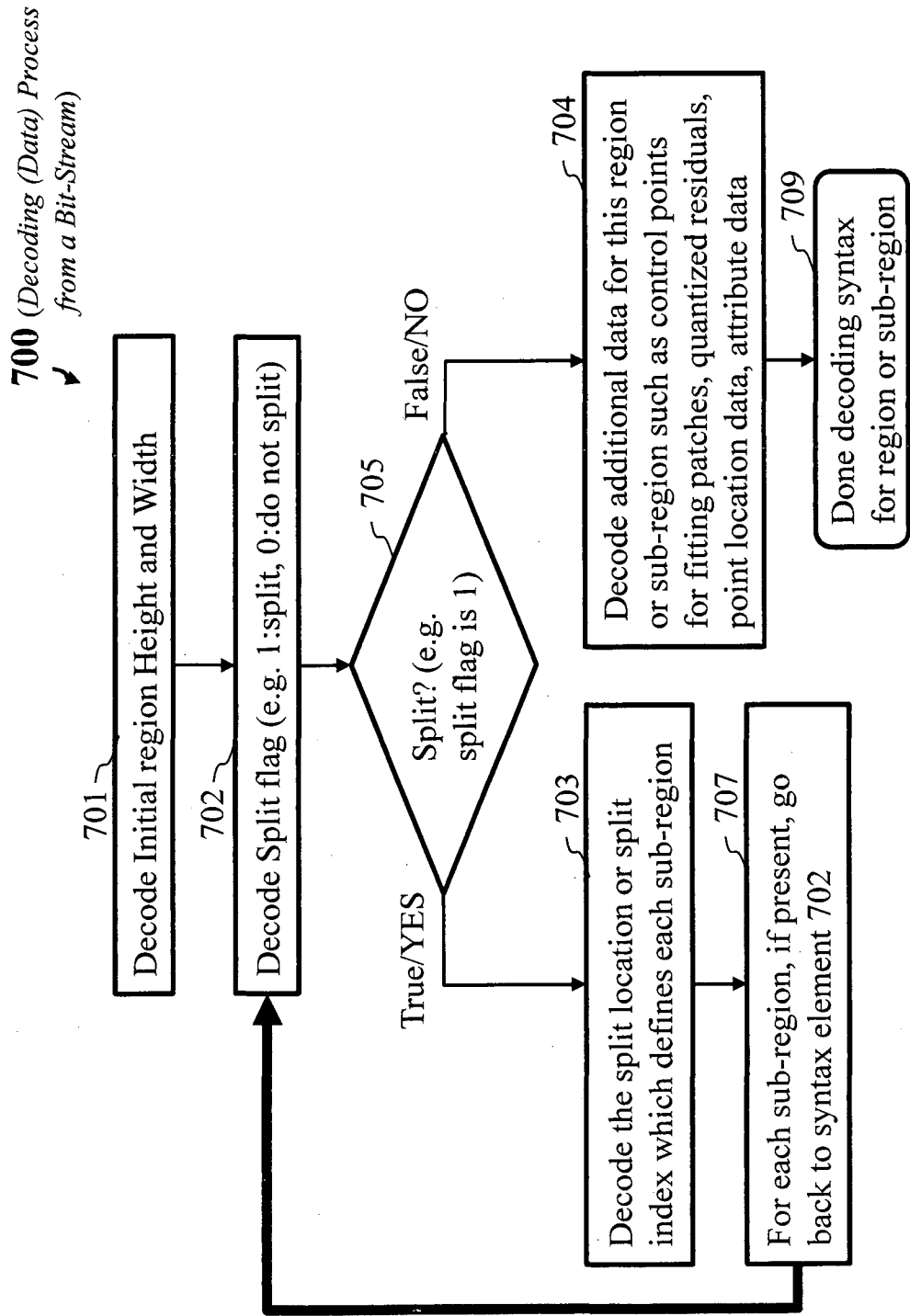
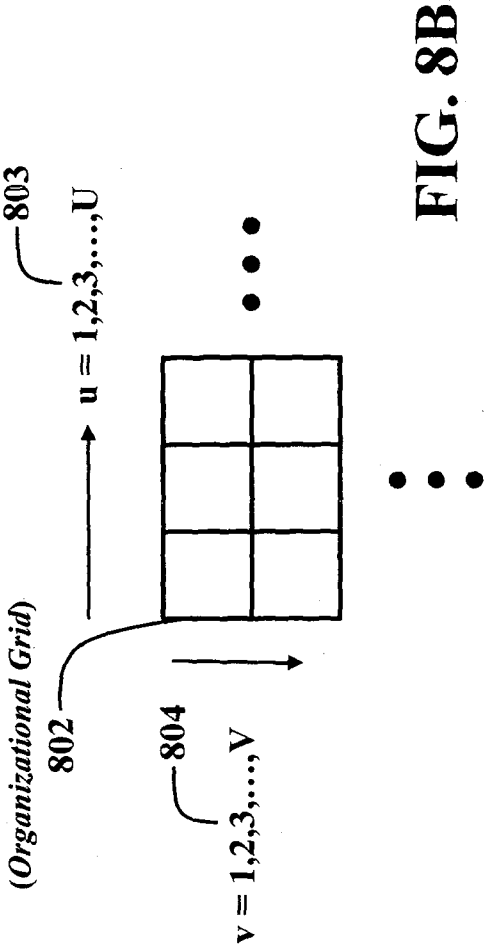
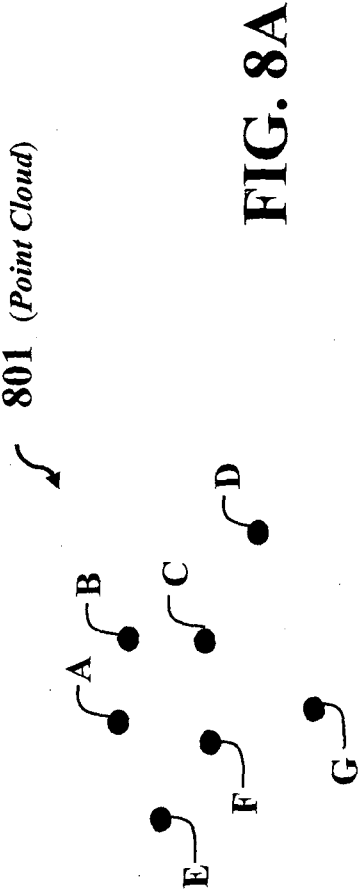


FIG. 7



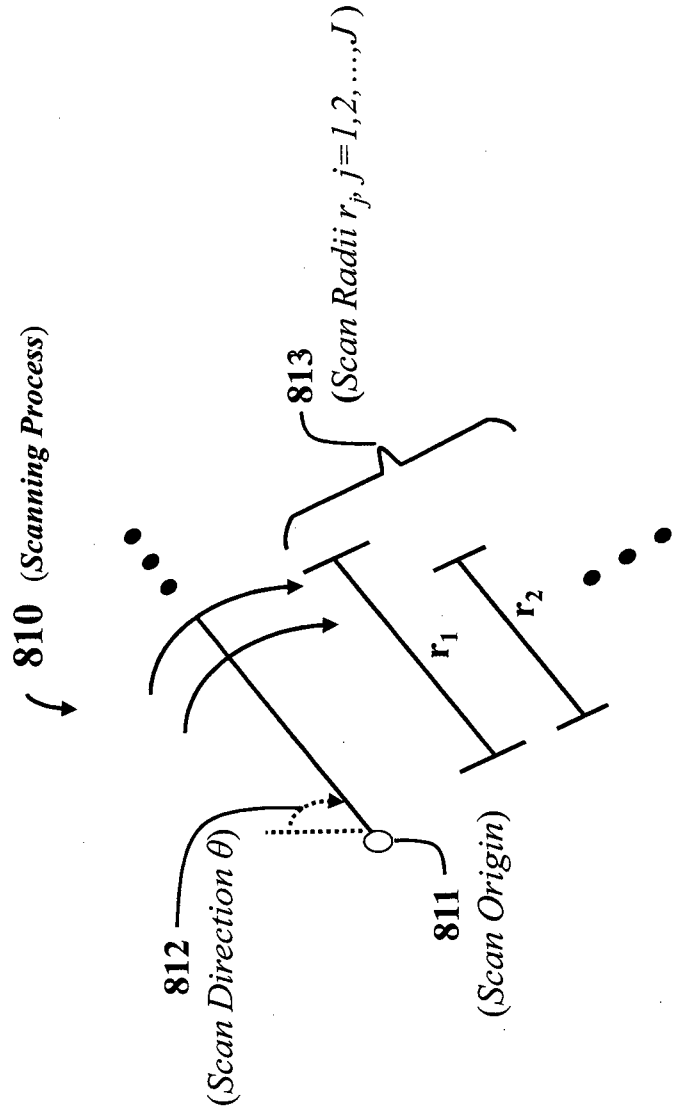


FIG. 8C

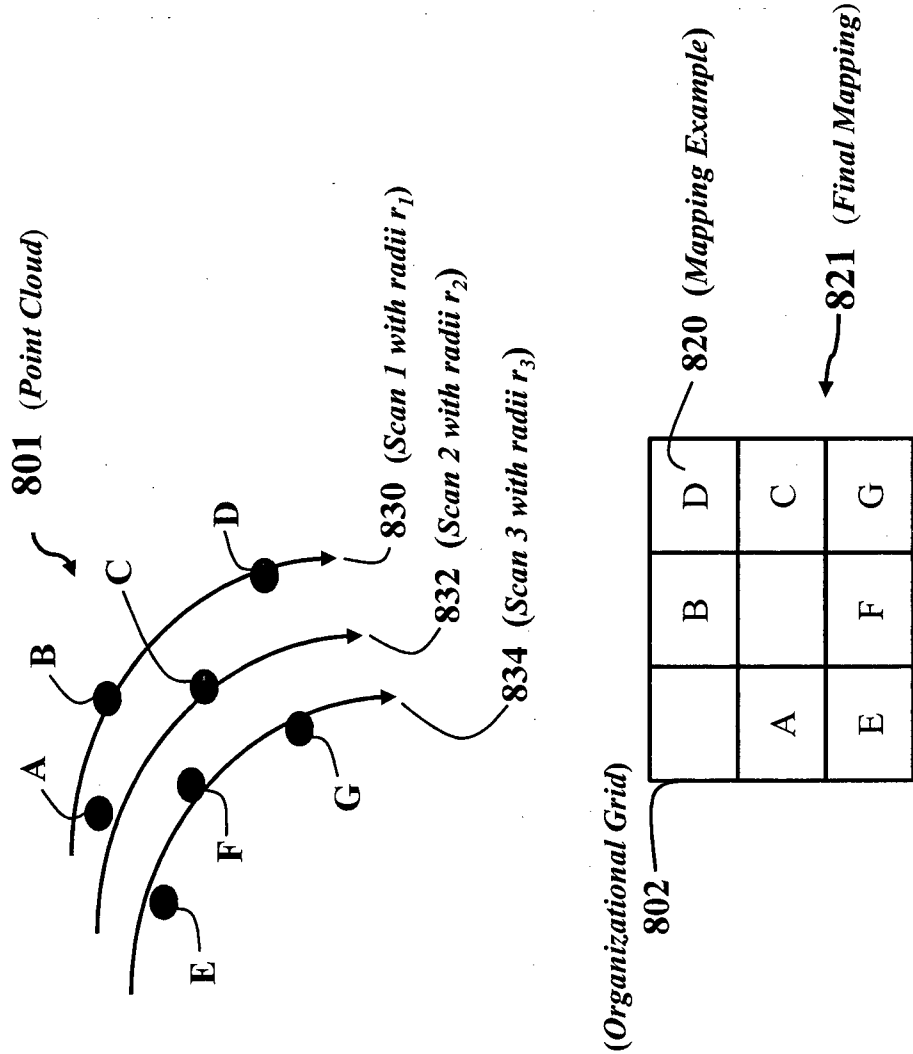


FIG. 8D

INTERNATIONAL SEARCH REPORT

International application No
PCT/JP2018/037496

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/597
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06T H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	ROBERT COHEN ET AL: "Point-based prediction for point cloud compression", 120. MPEG MEETING; 23-10-2017 - 27-10-2017; MACAU; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m41764, 19 October 2017 (2017-10-19), XP030070106,	1-8, 13-22, 24-26
A,P	Introduction Point prediction; page 1, paragraph 1 - paragraph 2.1 Differential prediction from block center Sequential prediction; page 2, paragraph 2.2.1 - page 3, paragraph 3 ----- -/--	23



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

21 February 2019

Date of mailing of the international search report

28/02/2019

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Moschetti, Fulvio

INTERNATIONAL SEARCH REPORT

International application No

PCT/JP2018/037496

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MEKURIA RUFAEL ET AL: "Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, US, vol. 27, no. 4, 1 April 2017 (2017-04-01), pages 828-842, XP011644783, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2016.2543039 [retrieved on 2017-04-03]	1-6, 13-22, 24-26
Y	abstract	7,8,11, 12
A	Introduction; page 828 Related work: basic concepts; page 830, paragraph I.C Coding scheme; page 831, column 2, paragraph II.B - page 832, column 1; figure 4 Bounding Box; page 832, column 2, paragraph III.A; figure 6 Predictive Algorithm; paragraph [IV.A]; figure 8 Objective quality evaluation metric; page 836, column 1, paragraph V.B -----	9,10,23
X	AINALA KHARTIK ET AL: "An improved enhancement layer for octree based point cloud compression with plane projection approximation", PROCEEDINGS OF SPIE; [PROCEEDINGS OF SPIE ISSN 0277-786X VOLUME 10524], SPIE, US, vol. 9971, 27 September 2016 (2016-09-27), pages 99710R-99710R, XP060078033, DOI: 10.1117/12.2237753 ISBN: 978-1-5106-1533-5	1-6, 13-22, 24-26
A	abstract Introduction; page 1 - page 2 MPEG point cloud compression; page 2, paragraph 2 - page 3; figure 1 ----- -/--	23

INTERNATIONAL SEARCH REPORT

International application No

PCT/JP2018/037496

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	COHEN ROBERT A ET AL: "Representation and coding of large-scale 3D dynamic maps", PROCEEDINGS OF SPIE; [PROCEEDINGS OF SPIE ISSN 0277-786X VOLUME 10524], SPIE, US, vol. 9971, 27 September 2016 (2016-09-27), pages 99710T-99710T, XP060078035, DOI: 10.1117/12.2237755 ISBN: 978-1-5106-1533-5	1-6, 13-22, 24-26
A	Representation and compression format.; page 3, paragraph 2.2 - paragraph 2.3 Attribute compression and Intra prediction; block based prediction; page 3, paragraph 3 - page 4, paragraph 3.1; figure 2 Point location compression.; page 6, paragraph 4	23
X	----- DE QUEIROZ RICARDO L ET AL: "Motion-Compensated Compression of Dynamic Voxelized Point Clouds", IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 26, no. 8, 1 August 2017 (2017-08-01), pages 3886-3895, XP011653025, ISSN: 1057-7149, DOI: 10.1109/TIP.2017.2707807 [retrieved on 2017-06-13]	1,13,14
Y	page 3886, paragraph 1	7,8
A	page 3887, column 1, paragraph II page 3889, paragraph V; figures 4,5	2-6, 9-12, 15-26
X	----- BIRENDRA KATHARIYA (UMKC) ET AL: "Dynamic Geometry Compression with Binary Tree Based Prediction and Residual Coding", 118. MPEG MEETING; 3-4-2017 - 7-4-2017; HOBART; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m40607, 31 March 2017 (2017-03-31), XP030068952,	1,13,14
Y	Dynamic geometry coding;	11,12
A	page 1, paragraph 2 Resampling of reference point cloud block; page 3, paragraph 2.2	2-10, 15-26
Y	----- CHEN SIHENG ET AL: "Contour-enhanced resampling of 3D point clouds via graphs", 2017 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), IEEE, 5 March 2017 (2017-03-05), pages 2941-2945, XP033258952, DOI: 10.1109/ICASSP.2017.7952695 [retrieved on 2017-06-16]	11,12
A	page 2941, paragraph 1 - page 2942, paragraph 2	1-10, 13-26

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2018/037496

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of Item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☒ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-6, 13-26

Claims 1-6, 13, 14-26 (wherein claims 2 and 3 are dependent on claim 1; claim 4 is dependent on claim 3; claims 5-6 are also dependent on claim 3. In this group of claims: claims 1-3 and 5-6 relate to generic prediction based residual coding; claim 4 goes into the direction of "order of coding" of points, namely it deals with processing the points in each block according to an order). The technical effect of the feature of claim 4 is mentioned at page 13 of the specification. The scope of the feature of claim 4 is "reducing the energy of the prediction residuals signaled for the case when a chain or sequence of points is signaled". Claim 4 is therefore a first branch starting from claim 3.

Claims 14-26 are not concise with the branch of claims 1-12 and they also scatter into different directions; the subject-matter of claims 14-22, 24-26 is however rendered obvious by D2.

2. claims: 7, 8

Claim 7 (dependent on claim 3): relates to organizing the point cloud by mapping on an organizational grid. The problem solved by the technical features mentioned in this claim is expressed at page 30 of the specification: the selection of regions or group of points in the organizational grid is needed to select related groups of points in the 3D space.

3. claims: 9, 10

Claim 9 and 10 (claim 9 is dependent on claim 3) relate to a method for performing data dependent non uniform partitioning (by employing a scoring function). The problem solved by this branch of claims is that of optimizing the partitioning of the data such that it is more semantically meaningful from the compression point of view (e.g. an object belongs to a single partitioned area and not to multiple partitioned areas).

4. claims: 11, 12

Claims 11 and 12 (claim 11 is dependent on claim 3) relate to resampling of the points in the blocks of points: the resampling is performed to possibly reduce the computational complexity (as mentioned in the specification at page 21, paragraph 52, section resampling).
