



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11)
CỤC SỞ HỮU TRÍ TUỆ
(51)^{2022.01} H04N 19/119; H04N 19/96; H04N 19/70; G06T 9/40; H04N 19/597 (13) B



1-0053517

(21) 1-2022-08363 (22) 31/05/2021
(86) PCT/KR2021/006726 31/05/2021 (87) WO2021/242064 02/12/2021
(30) 10-2020-0065154 29/05/2020 KR
(45) 25/11/2025 452 (43) 25/05/2023 422A
(73) LG ELECTRONICS INC. (KR)
128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Republic of Korea
(72) OH, Hyunmook (KR); OH, Sejin (KR).
(74) Công ty Luật TNHH T&G (TGVN)

(54) THIẾT BỊ TRUYỀN DỮ LIỆU ĐÁM MÂY ĐIỂM, PHƯƠNG PHÁP TRUYỀN DỮ LIỆU ĐÁM MÂY ĐIỂM, THIẾT BỊ NHẬN DỮ LIỆU ĐÁM MÂY ĐIỂM VÀ PHƯƠNG PHÁP NHẬN DỮ LIỆU ĐÁM MÂY ĐIỂM

(21) 1-2022-08363

(57) Sáng chế đề cập đến phương pháp truyền dữ liệu đám mây điểm, phương pháp này có thể bao gồm các bước: thu nhận dữ liệu đám mây điểm; mã hóa, bằng cách áp dụng cây dạng hình học, thông tin dạng hình học bao gồm các vị trí của các điểm của dữ liệu đám mây điểm này; mã hóa thông tin thuộc tính bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm này trên cơ sở cây dạng hình học này; và truyền thông tin dạng hình học được mã hóa này, thông tin thuộc tính được mã hóa này, và thông tin báo hiệu, trong đó bước mã hóa thông tin thuộc tính bao gồm: bước tạo ra các mức độ chi tiết (Level of Detail, LoD) trên cơ sở cây dạng hình học này và bước nén thông tin thuộc tính trên cơ sở các LoD này.

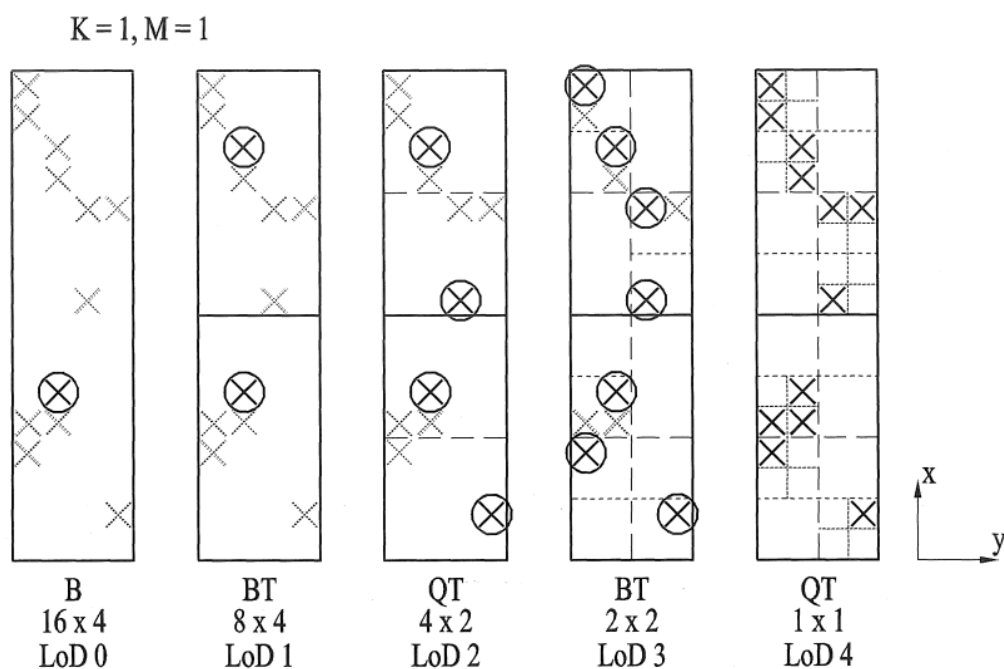


FIG. 23

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các phương án liên quan đến phương pháp và thiết bị để xử lý nội dung đám mây điểm.

Tình trạng kỹ thuật của sáng chế

Nội dung đám mây điểm là nội dung được biểu diễn bởi đám mây điểm, mà nó là tập hợp của các điểm thuộc về hệ tọa độ biểu diễn không gian ba chiều (hoặc dung tích). Nội dung đám mây điểm có thể thể hiện các phương tiện được tạo cấu hình theo ba chiều, và được sử dụng để cung cấp các dịch vụ khác nhau chẳng hạn như thực tế ảo (Virtual Reality, VR), thực tế tăng cường (Augmented Reality, AR), thực tế hỗn hợp (Mixed Reality, MR), thực tế mở rộng (Extended Reality, XR), và các dịch vụ tự lái. Tuy nhiên, cần có đến hàng chục nghìn đến hàng trăm nghìn dữ liệu điểm để biểu diễn nội dung đám mây điểm. Do đó, cần có phương pháp xử lý một cách hiệu quả một lượng lớn dữ liệu điểm.

Bản chất kỹ thuật của sáng chế

Vấn đề kỹ thuật

Mục đích của sáng chế, được đặt ra nhằm giải quyết các vấn đề được mô tả trên đây, là đề xuất thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm để truyền và nhận đám mây điểm một cách hiệu quả.

Một mục đích khác của sáng chế là đề xuất thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm để giải quyết độ trễ và độ phức tạp mã hóa/giải mã.

Một mục đích khác của sáng chế là đề xuất thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm,

và phương pháp nhận dữ liệu đám mây điểm để truyền và nhận một cách hiệu quả luồng bit nén đám mây điểm dạng hình học (geometry-point cloud compression, G-PCC).

Một mục đích khác của sáng chế là đề xuất thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm để so khớp lớp lập mã dạng hình học với việc lập mã có thể định tỷ lệ được của lớp lập mã thuộc tính bằng cách thực hiện việc nén thuộc tính có xem xét đến việc phân bố điểm bất đối xứng.

Một mục đích khác của sáng chế là đề xuất thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm để làm tăng hiệu quả nén thuộc tính của nội dung có phân bố điểm bất đối xứng bằng cách tạo cấu hình lớp có thể định tỷ lệ được dựa trên nút lân cận bất đối xứng khi sử dụng lớp có thể định tỷ lệ được dựa trên LoD trong việc lập mã đám mây điểm có thể định tỷ lệ được.

Các mục đích của sáng chế không bị giới hạn vào các mục đích nêu trên, và các mục đích khác của sáng chế mà chúng không được nhắc tới ở trên sẽ trở nên rõ ràng đối với những người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi xem xét phần mô tả sau đây.

Giải pháp kỹ thuật của sáng chế

Các mục đích trên và các mục đích khác của sáng chế có thể đạt được qua việc đề xuất phương pháp truyền dữ liệu đám mây điểm. Phương pháp này có thể bao gồm các bước thu nhận dữ liệu đám mây điểm, mã hóa thông tin dạng hình học bao gồm các vị trí của các điểm của dữ liệu đám mây điểm này bằng cách áp dụng cây dạng hình học, mã hóa thông tin thuộc tính bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm này dựa trên cây dạng hình học này, và truyền thông tin dạng hình học được mã hóa này, thông tin thuộc tính được mã hóa này, và thông tin báo hiệu.

Trong một phương án, bước mã hóa thông tin thuộc tính có thể bao gồm các bước tạo ra các mức độ chi tiết (Level of Detail, LoD) dựa trên cây dạng hình học này, và nén thông tin thuộc tính dựa trên các LoD này.

Trong một phương án, cây dạng hình học có thể được tạo ra bởi sự kết hợp của hai hay hơn hai thành phần trong số cây bát phân, cây tứ phân, hoặc cây nhị phân.

Trong một phương án, bước tạo ra các LOD có thể bao gồm các bước tạo cấu hình

một hoặc nhiều nút con được phân vùng từ các nút được chiếm giữ của độ sâu hiện tại trong cây dạng hình học làm các lân cận, và chọn một điểm từ trong số các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận này, trong đó bước tạo cấu hình các lân cận và bước chọn này có thể được thực hiện trên mỗi nút trong số các nút được chiếm giữ của độ sâu hiện tại để tạo ra LoD tương ứng với độ sâu hiện tại.

Trong một phương án, số lượng nút con được tạo cấu hình làm các lân cận có thể thay đổi tùy thuộc vào việc liệu loại cây của độ sâu hiện tại là cây bát phân, cây tứ phân, hay cây nhị phân.

Trong một phương án, bước chọn điểm có thể được thực hiện dựa trên thứ tự mã Morton của các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận đã nêu.

Thiết bị truyền dữ liệu đám mây điểm theo các phương án có thể bao gồm bộ thu nhận được tạo cấu hình để thu nhận dữ liệu đám mây điểm, bộ mã hóa dạng hình học được tạo cấu hình để mã hóa thông tin dạng hình học bao gồm các vị trí của các điểm của dữ liệu đám mây điểm này bằng cách áp dụng cây dạng hình học, bộ mã hóa thuộc tính được tạo cấu hình để mã hóa thông tin thuộc tính bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm này dựa trên cây dạng hình học này, và bộ truyền được tạo cấu hình để truyền thông tin dạng hình học được mã hóa này, thông tin thuộc tính được mã hóa này, và thông tin báo hiệu.

Trong một phương án, bộ mã hóa thuộc tính có thể bao gồm bộ tạo ra LoD được tạo cấu hình để tạo ra các mức độ chi tiết (LoD) dựa trên cây dạng hình học, và bộ nén được tạo cấu hình để nén thông tin thuộc tính dựa trên các LoD này.

Trong một phương án, cây dạng hình học có thể được tạo ra bởi sự kết hợp của hai hay hơn hai thành phần trong số cây bát phân, cây tứ phân, hoặc cây nhị phân.

Trong một phương án, bộ tạo ra LOD có thể được tạo cấu hình để tạo cấu hình một hoặc nhiều nút con được phân vùng từ các nút được chiếm giữ của độ sâu hiện tại trong cây dạng hình học làm các lân cận, và chọn một điểm từ trong số các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận này, trong đó việc tạo cấu hình và việc chọn này có thể được thực hiện trên mỗi nút trong số các nút được chiếm giữ của độ sâu hiện tại để tạo ra LoD tương ứng với độ sâu hiện tại.

Trong một phương án, số lượng nút con được tạo cấu hình làm các lân cận có thể

thay đổi tùy thuộc vào việc liệu loại cây của độ sâu hiện tại là cây bát phân, cây tứ phân, hay cây nhị phân.

Trong một phương án, bộ tạo ra LOD có thể chọn điểm dựa trên thứ tự mã Morton của các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận đã nêu.

Phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể bao gồm các bước nhận thông tin dạng hình học, thông tin thuộc tính, và thông tin báo hiệu, giải mã thông tin dạng hình học này dựa trên thông tin báo hiệu này và cây dạng hình học, giải mã thông tin thuộc tính dựa trên thông tin báo hiệu này và cây dạng hình học này, và kết xuất dữ liệu đám mây điểm được xây dựng lại dựa trên thông tin dạng hình học được giải mã và thông tin thuộc tính được giải mã.

Trong một phương án, thông tin dạng hình học được giải mã có thể bao gồm các vị trí của các điểm của dữ liệu đám mây điểm được xây dựng lại, và thông tin thuộc tính được giải mã có thể bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm được xây dựng lại.

Trong một phương án, bước giải mã thông tin thuộc tính có thể bao gồm các bước tạo ra các mức độ chi tiết (Level of Detail, LoD) dựa trên cây dạng hình học này, và xây dựng lại thông tin thuộc tính dựa trên các LoD này.

Trong một phương án, cây dạng hình học có thể được tạo ra bởi sự kết hợp của hai hay hơn hai thành phần trong số cây bát phân, cây tứ phân, hoặc cây nhị phân.

Trong một phương án, bước tạo ra các LOD có thể bao gồm các bước tạo cấu hình một hoặc nhiều nút con được phân vùng từ các nút được chiếm giữ của độ sâu hiện tại trong cây dạng hình học làm các lân cận, và chọn một điểm từ trong số các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận này, trong đó bước tạo cấu hình các lân cận và bước chọn này có thể được thực hiện trên mỗi nút trong số các nút được chiếm giữ của độ sâu hiện tại để tạo ra LoD tương ứng với độ sâu hiện tại.

Trong một phương án, số lượng nút con được tạo cấu hình làm các lân cận có thể thay đổi tùy thuộc vào việc liệu loại cây của độ sâu hiện tại là cây bát phân, cây tứ phân, hay cây nhị phân.

Trong một phương án, bước chọn điểm có thể được thực hiện dựa trên thứ tự mã Morton của các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận đã nêu.

Thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể bao gồm bộ nhận được tạo cấu hình để nhận thông tin dạng hình học, thông tin thuộc tính, và thông tin báo hiệu, bộ giải mã dạng hình học được tạo cấu hình để giải mã thông tin dạng hình học này dựa trên thông tin báo hiệu này và cây dạng hình học, bộ giải mã thuộc tính được tạo cấu hình để giải mã thông tin thuộc tính dựa trên thông tin báo hiệu này và cây dạng hình học này, và bộ kết xuất được tạo cấu hình để kết xuất dữ liệu đám mây điểm được xây dựng lại dựa trên thông tin dạng hình học được giải mã và thông tin thuộc tính được giải mã.

Trong một phương án, thông tin dạng hình học được giải mã có thể bao gồm các vị trí của các điểm của dữ liệu đám mây điểm được xây dựng lại, và thông tin thuộc tính được giải mã có thể bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm được xây dựng lại.

Trong một phương án, bộ giải mã thuộc tính có thể bao gồm bộ tạo ra LOD được tạo cấu hình để tạo ra các mức độ chi tiết (LoD) dựa trên cây dạng hình học, và

bộ xây dựng lại được tạo cấu hình để xây dựng lại thông tin thuộc tính dựa trên các LoD này.

Trong một phương án, cây dạng hình học có thể được tạo ra bởi sự kết hợp của hai hay hơn hai thành phần trong số cây bát phân, cây tứ phân, hoặc cây nhị phân.

Trong một phương án, bộ tạo ra LOD có thể được tạo cấu hình để tạo cấu hình một hoặc nhiều nút con được phân vùng từ các nút được chiếm giữ của độ sâu hiện tại trong cây dạng hình học làm các lân cận, và chọn một điểm từ trong số các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận này, trong đó việc tạo cấu hình và việc chọn này có thể được thực hiện trên mỗi nút trong số các nút được chiếm giữ của độ sâu hiện tại để tạo ra LoD tương ứng với độ sâu hiện tại.

Trong một phương án, số lượng nút con được tạo cấu hình làm các lân cận có thể thay đổi tùy thuộc vào việc liệu loại cây của độ sâu hiện tại là cây bát phân, cây tứ phân, hay cây nhị phân.

Trong một phương án, bộ tạo ra LoD có thể chọn điểm dựa trên thứ tự mã Morton của các điểm thuộc về một hoặc nhiều nút con được tạo cấu hình làm các lân cận đã nêu.

Hiệu quả của sáng chế

Phương pháp truyền dữ liệu đám mây điểm, thiết bị truyền dữ liệu đám mây điểm, phương pháp nhận dữ liệu đám mây điểm, và thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể cung cấp dịch vụ đám mây điểm chất lượng tốt.

Phương pháp truyền dữ liệu đám mây điểm, thiết bị truyền dữ liệu đám mây điểm, phương pháp nhận dữ liệu đám mây điểm, và thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể đạt được các phương pháp lập mã-giải mã (codec) video khác nhau.

Phương pháp truyền dữ liệu đám mây điểm, thiết bị truyền dữ liệu đám mây điểm, phương pháp nhận dữ liệu đám mây điểm, và thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể cung cấp nội dung đám mây điểm phổ quát chẳng hạn như dịch vụ tự lái (hoặc dịch vụ lái tự động).

Phương pháp truyền dữ liệu đám mây điểm, thiết bị truyền dữ liệu đám mây điểm, phương pháp nhận dữ liệu đám mây điểm, và thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể thực hiện việc phân vùng thích ứng theo không gian dữ liệu đám mây điểm cho việc mã hóa và giải mã độc lập dữ liệu đám mây điểm này, qua đó cải thiện việc xử lý song song và cung cấp khả năng định tỷ lệ.

Phương pháp truyền dữ liệu đám mây điểm, thiết bị truyền dữ liệu đám mây điểm, phương pháp nhận dữ liệu đám mây điểm, và thiết bị nhận dữ liệu đám mây điểm theo các phương án có thể thực hiện việc mã hóa và giải mã bằng cách phân vùng dữ liệu đám mây điểm trong các đơn vị là các ô xếp và/hoặc các lát, và báo hiệu dữ liệu cần thiết cho chúng, qua đó cải thiện hiệu quả mã hóa và giải mã của đám mây điểm.

Thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể làm tăng hiệu quả nén thuộc tính của nội dung có phân bố điểm bất đối xứng bằng cách tạo cấu hình lớp có thể định tỷ lệ được sử dụng các nút lân cận bất đối xứng khi sử dụng lớp có thể định tỷ lệ được dựa trên LoD trong việc lập mã đám mây điểm có thể định tỷ lệ được.

Thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể tạo ra các LoD cho việc nén thuộc tính sử dụng cấu trúc cây dạng hình học dựa trên OT/QT/BT được sử dụng cho việc nén dạng hình học, qua đó so khớp

số lượng điểm của dạng hình học với số lượng điểm của thuộc tính cho cấu hình LoD tại độ sâu tương ứng trong phép biểu diễn có thể định tỷ lệ được. Theo các phương án, lớp trung gian có thể được xuất ra dựa trên cấu trúc phân cấp của dữ liệu đám mây điểm. Trường hợp này có thể được định ra là phép biểu diễn có thể định tỷ lệ được.

Thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể được hạn chế để tạo ra LoD (các LoD) chỉ cho độ sâu (các độ sâu) đang khớp với phương pháp tạo ra LoD dựa trên QT (OT trong không gian ba chiều) trong cấu trúc cây dạng hình học được sử dụng cho việc nén dạng hình học. Qua đó, việc không khớp điểm giữa dạng hình học và các thuộc tính có thể được ngăn chặn trong phép biểu diễn có thể định tỷ lệ được.

Thiết bị truyền dữ liệu đám mây điểm, phương pháp truyền dữ liệu đám mây điểm, thiết bị nhận dữ liệu đám mây điểm, và phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể so khớp các điểm/các nút với thông tin thuộc tính theo kiểu phép tương ứng một-một tùy theo độ sâu trong cây dạng hình học khi việc lập mã dạng hình học dựa trên QTBT được thực hiện cho lĩnh vực ứng dụng việc lập mã có thể định tỷ lệ được. Qua đó, phép biểu diễn có thể định tỷ lệ được có thể được cho phép đối với tất cả các độ sâu hoặc đối với một số độ sâu trùng khớp.

Mô tả vắn tắt các hình vẽ

Các hình vẽ kèm theo, mà chúng được đưa vào nhằm cung cấp thêm hiểu biết về sáng chế và được kết hợp trong và cấu thành nên một phần của đơn này, minh họa phương án (các phương án) của sáng chế, và cùng với phần mô tả, có chức năng giải thích nguyên lý của sáng chế.

Fig.1 minh họa hệ thống cung cấp nội dung đám mây điểm làm ví dụ theo các phương án.

Fig.2 là sơ đồ khối minh họa hoạt động cung cấp nội dung đám mây điểm theo các phương án.

Fig.3 minh họa quy trình thu nạp video đám mây điểm làm ví dụ theo các phương án.

Fig.4 minh họa sơ đồ khối làm ví dụ của bộ mã hóa video đám mây điểm theo các phương án.

Fig.5 minh họa ví dụ về các đơn vị ảnh ba chiều (voxel) trong không gian 3D theo các phương án.

Fig.6 minh họa ví dụ về cây bát phân và mã chiếm giữ theo các phương án.

Fig.7 minh họa ví dụ về mẫu dạng nút lân cận theo các phương án.

Fig.8 minh họa ví dụ về cấu hình điểm của nội dung đám mây điểm cho mỗi LOD theo các phương án.

Fig.9 minh họa ví dụ về cấu hình điểm của nội dung đám mây điểm cho mỗi LOD theo các phương án.

Fig.10 minh họa ví dụ về sơ đồ khối của bộ giải mã video đám mây điểm theo các phương án.

Fig.11 minh họa ví dụ về bộ giải mã video đám mây điểm theo các phương án.

Fig.12 minh họa cấu hình cho việc mã hóa video đám mây điểm của thiết bị truyền theo các phương án.

Fig.13 minh họa cấu hình cho việc giải mã video đám mây điểm của thiết bị nhận theo các phương án.

Fig.14 minh họa cấu trúc có thể hoạt động mang tính ví dụ trong mối liên hệ với các phương pháp/thiết bị dữ liệu đám mây điểm theo các phương án.

Fig.15 là sơ đồ minh họa ví dụ về dữ liệu đám mây điểm bao gồm các điểm được phân bố bất đối xứng theo các phương án.

Fig.16 là sơ đồ minh họa ví dụ về việc phân vùng khối lập phương dựa trên cây bát phân trong không gian 3D.

Fig.17-(a) đến Fig.17-(c) là các sơ đồ minh họa các ví dụ về việc phân vùng khối lập phương dựa trên cây tứ phân trong không gian 3D.

Fig.18-(a) đến Fig.18-(c) là các sơ đồ minh họa các ví dụ về việc phân vùng khối lập phương dựa trên cây nhị phân trong không gian 3D.

Fig.19-(a) và Fig.19-(b) thể hiện các ví dụ về việc phân vùng hộp giới hạn trong không gian 2D.

Fig.20 là sơ đồ minh họa ví dụ về việc tạo ra các LOD dựa trên cây bát phân bởi bộ mã hóa thuộc tính theo các phương án.

Fig.21 là sơ đồ minh họa ví dụ về thứ tự mã Morton trong khối lập phương 3D theo các phương án.

Fig.22 là sơ đồ minh họa ví dụ về quy trình chọn thuộc tính theo việc tạo ra LoD cho các nút trong mặt phẳng 2D theo các phương án.

Fig.23 là sơ đồ minh họa một ví dụ khác về quy trình chọn thuộc tính theo việc tạo ra LoD cho các nút trong mặt phẳng 2D bởi bộ mã hóa thuộc tính theo các phương án.

Fig.24-(a) và Fig.24-(b) là các sơ đồ minh họa các ví dụ về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.25-(a) và Fig.25-(b) là các sơ đồ minh họa các ví dụ khác về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trên mặt phẳng 2D.

Fig.26 là sơ đồ minh họa ví dụ về việc tạo ra LoD tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.27 là sơ đồ minh họa một ví dụ khác về việc tạo ra LoD tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.28-(a) và Fig.28(b) là các sơ đồ minh họa các ví dụ về cấu hình của việc bỏ qua LoD tùy theo các điều kiện được mô tả trên đây tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.29 là sơ đồ minh họa một ví dụ khác về cấu hình của việc bỏ qua LoD tùy theo các điều kiện được mô tả trên đây tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.30 là sơ đồ minh họa một ví dụ khác về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.31 minh họa cấu trúc luồng bit mang tính ví dụ dành cho dữ liệu đám mây điểm dành cho việc truyền/nhận theo các phương án.

Fig.32 minh họa cấu trúc luồng bit làm ví dụ dành cho dữ liệu đám mây điểm dành cho việc truyền/nhận theo các phương án.

Fig.33 minh họa mối quan hệ kết nối giữa các phần tử trong luồng bit của dữ liệu

đám mây điểm theo các phương án.

Fig.34 thể hiện phương án về cấu trúc cú pháp của tập tham số trình tự theo sáng chế này.

Fig.35 thể hiện ví dụ về cấu trúc cú pháp của tập tham số dạng hình học theo các phương án.

Fig.36 thể hiện ví dụ về cấu trúc cú pháp của tập tham số thuộc tính theo các phương án.

Fig.37 thể hiện ví dụ về cấu trúc cú pháp của tập tham số thuộc tính bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD theo các phương án.

Fig.38 thể hiện cấu trúc cú pháp mang tính ví dụ của `geometry_slice_bitstream()` theo các phương án.

Fig.39 thể hiện cấu trúc cú pháp mang tính ví dụ của tiêu đề lát dạng hình học theo các phương án.

Fig.40 thể hiện cấu trúc cú pháp mang tính ví dụ của dữ liệu lát dạng hình học theo các phương án.

Fig.41 thể hiện cấu trúc cú pháp mang tính ví dụ của `attribute_slice_bitstream()` theo các phương án.

Fig.42 thể hiện cấu trúc cú pháp mang tính ví dụ của tiêu đề lát thuộc tính theo các phương án.

Fig.42 thể hiện cấu trúc cú pháp mang tính ví dụ của tập tham số thuộc tính bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD theo các phương án.

Fig.44 thể hiện cấu trúc cú pháp mang tính ví dụ của dữ liệu lát thuộc tính theo các phương án.

Fig.45 minh một ví dụ khác về thiết bị truyền đám mây điểm theo các phương án.

Fig.46 là sơ đồ khối chi tiết mang tính ví dụ minh họa bộ mã hóa thuộc tính theo các phương án.

Fig.47 là lưu đồ minh họa phương pháp tạo ra LoD mang tính ví dụ theo các phương án.

Fig.48 minh họa một ví dụ khác về thiết bị nhận đám mây điểm theo các phương án.

Fig.49 là sơ đồ khối chi tiết mang tính ví dụ minh họa bộ giải mã thuộc tính theo

các phương án.

Fig.50 là lưu đồ minh họa phương pháp truyền dữ liệu đám mây điểm theo các phương án.

Fig.51 là lưu đồ minh họa phương pháp nhận dữ liệu đám mây điểm theo các phương án.

Mô tả chi tiết sáng chế

Giờ đây, phần mô tả sẽ được nêu chi tiết theo các phương án làm ví dụ được bộc lộ ở đây, với tham chiếu đến các hình vẽ kèm theo. Để phần mô tả với tham chiếu đến các hình vẽ được ngắn gọn, các thành phần giống nhau hoặc tương đương nhau có thể được cung cấp các số chỉ dẫn giống nhau, và phần mô tả của chúng sẽ không được lặp lại. Cần lưu ý rằng các ví dụ sau đây chỉ là dành cho việc thể hiện sáng chế và không hạn chế phạm vi của sáng chế. Những gì có thể được suy ra một cách dễ dàng bởi chuyên gia trong lĩnh vực kỹ thuật mà sáng chế thuộc về nó từ phần mô tả chi tiết và các ví dụ này của sáng chế sẽ cần được diễn giải là nằm trong phạm vi của sáng chế.

Phần mô tả chi tiết trong bản mô tả này cần phải được phân tích, theo tất cả các khía cạnh, là đều mang tính minh họa và không mang tính hạn chế. Phạm vi bộc lộ cần phải được xác định bởi các yêu cầu bảo hộ đi kèm và các nội dung tương đương hợp pháp của chúng, và tất cả các thay đổi thuộc vào bên trong ý nghĩa và phạm vi tương đương của các yêu cầu bảo hộ đi kèm đều được dự tính là được bao quát trong đó.

Giờ đây tham chiếu chi tiết đến các phương án được ưu tiên của sáng chế, mà các ví dụ về chúng được minh họa trên các hình vẽ kèm theo. Phần mô tả chi tiết được cho dưới đây với tham chiếu đến các hình vẽ kèm theo là nhằm giải thích các phương án làm ví dụ của sáng chế này, chứ không phải chỉ thể hiện các phương án có thể được thi hành theo sáng chế. Phần mô tả chi tiết dưới đây chứa các chi tiết cụ thể để cung cấp cách hiểu kỹ lưỡng đối với sáng chế này. Tuy nhiên, những người có hiểu biết trong lĩnh vực tương ứng sẽ hiểu rõ rằng sáng chế có thể được thực hành mà không có các chi tiết cụ thể này. Mặc dù hầu hết các thuật ngữ được sử dụng trong bản mô tả này đã được lựa chọn từ các thuật ngữ chung được sử dụng rộng rãi trong lĩnh vực này, một số thuật ngữ đã được người nộp đơn lựa chọn tùy ý và ý nghĩa của chúng được giải thích chi tiết trong phần mô tả sau đây khi cần thiết. Vì thế, sáng chế nên được hiểu là dựa trên các ý nghĩa dự kiến của các thuật ngữ thay vì các tên gọi hoặc ý nghĩa đơn giản của chúng. Thêm

vào đó, các hình vẽ và phần mô tả chi tiết sau đây không thể được phân tích như là bị giới hạn trong các phương án được mô tả cụ thể, mà cần được phân tích là bao gồm các nội dung tương đương hoặc thay thế của các phương án được mô tả trên các hình vẽ và phần mô tả chi tiết này.

Fig.1 thể hiện hệ thống cung cấp nội dung đám mây điểm làm ví dụ theo các phương án.

Hệ thống cung cấp nội dung đám mây điểm được minh họa trên Fig.1 có thể bao gồm thiết bị truyền 10000 và thiết bị nhận 10004. Thiết bị truyền 10000 và thiết bị nhận 10004 có khả năng truyền thông có dây hoặc không dây để truyền và nhận dữ liệu đám mây điểm.

Thiết bị truyền dữ liệu đám mây điểm 10000 theo các phương án có thể đạt được và xử lý video đám mây điểm (hoặc nội dung đám mây điểm) và truyền video đám mây điểm (hoặc nội dung đám mây điểm) này. Theo các phương án, thiết bị truyền 10000 có thể bao gồm trạm cố định, hệ thống truyền nhận cơ sở (base transceiver system, BTS), mạng, thiết bị và/hoặc hệ thống trí tuệ nhân tạo (artificial intelligence, AI), rôbot, thiết bị và/hoặc máy chủ AR/VR/XR. Theo các phương án, thiết bị truyền 10000 có thể bao gồm thiết bị, rôbot, xe, thiết bị AR/VR/XR, thiết bị cầm tay, đồ gia dụng, thiết bị Internet vạn vật (Internet of Thing, IoT), và thiết bị/máy chủ AI mà chúng được tạo cấu hình để thực hiện việc truyền thông với trạm cơ sở và/hoặc các thiết bị không dây khác sử dụng công nghệ truy cập radio (ví dụ, RAT mới (New RAT, NR) 5G, phát triển dài hạn (Long Term Evolution, LTE)).

Thiết bị truyền 10000 theo các phương án bao gồm đơn vị thu nhận video đám mây điểm 10001, bộ mã hóa video đám mây điểm 10002, và/hoặc bộ truyền (hoặc môđun truyền thông) 10003.

Đơn vị thu nhận video đám mây điểm 10001 theo các phương án thu nhận video đám mây điểm thông qua quy trình xử lý chẳng hạn như thu nạp, tổng hợp, hoặc tạo ra. Video đám mây điểm là nội dung đám mây điểm được thể hiện bởi đám mây điểm, mà nó là tập các điểm được đặt trong không gian 3D, và có thể được gọi là dữ liệu video đám mây điểm. Video đám mây điểm theo các phương án có thể bao gồm một hoặc nhiều khung. Một khung thể hiện một ảnh/hình ảnh tĩnh. Do đó, video đám mây điểm

có thể bao gồm ảnh/khung/hình ảnh đám mây điểm, và có thể được gọi là ảnh, khung, hoặc hình ảnh đám mây điểm.

Bộ mã hóa video đám mây điểm 10002 theo các phương án mã hóa dữ liệu video đám mây điểm được thu nhận. Bộ mã hóa video đám mây điểm 10002 có thể mã hóa dữ liệu video đám mây điểm dựa trên việc lập mã nén đám mây điểm. Việc lập mã nén đám mây điểm theo các phương án có thể bao gồm việc lập mã nén đám mây điểm dựa trên dạng hình học (geometry-based point cloud compression, G-PCC) và/hoặc việc lập mã nén đám mây điểm dựa trên video (video-based point cloud compression, V-PCC) hoặc việc lập mã thế hệ tiếp theo. Việc lập mã nén đám mây điểm theo các phương án không bị giới hạn ở phương án được mô tả trên đây. Bộ mã hóa video đám mây điểm 10002 có thể xuất ra luồng bit chứa dữ liệu video đám mây điểm được mã hóa. Luồng bit này có thể chứa không chỉ dữ liệu video đám mây điểm được mã hóa, mà cả thông tin báo hiệu liên quan đến việc mã hóa của dữ liệu video đám mây điểm.

Bộ truyền 10003 theo các phương án truyền luồng bit chứa dữ liệu video đám mây điểm được mã hóa. Luồng bit theo các phương án được bao bọc trong tệp tin hoặc phân đoạn (ví dụ, phân đoạn tạo luồng), và được truyền qua các mạng khác nhau chẳng hạn như mạng phát quảng bá và/hoặc mạng băng rộng. Mặc dù không được thể hiện trên hình vẽ, thiết bị truyền 10000 có thể bao gồm bộ bao bọc (hoặc môđun bao bọc) được tạo cấu hình để thực hiện hoạt động bao bọc. Theo các phương án, bộ bao bọc có thể được chứa trong bộ truyền 10003. Theo các phương án, tệp tin hoặc phân đoạn có thể được truyền đến thiết bị nhận 10004 qua mạng, hoặc được lưu trữ trong phương tiện lưu trữ kỹ thuật số (ví dụ, USB, SD, CD, DVD, Blu-ray, HDD, SSD, v.v.). Bộ truyền 10003 theo các phương án có khả năng truyền thông có dây/không dây với thiết bị nhận 10004 (hoặc bộ nhận 10005) qua mạng 4G, 5G, 6G, v.v. Thêm vào đó, bộ truyền có thể thực hiện hoạt động xử lý dữ liệu cần thiết tùy theo hệ thống mạng (ví dụ, hệ thống mạng truyền thông 4G, 5G hoặc 6G). Thiết bị truyền 10000 có thể truyền dữ liệu được bao bọc theo cách tùy theo nhu cầu.

Thiết bị nhận 10004 theo các phương án bao gồm bộ nhận 10005, bộ giải mã video đám mây điểm 10006, và/hoặc bộ kết xuất 10007. Theo các phương án, thiết bị nhận 10004 có thể bao gồm thiết bị, rôbot, xe, thiết bị AR/VR/XR, thiết bị cầm tay, đồ gia dụng, thiết bị Internet vạn vật (Internet of Thing, IoT), và thiết bị/máy chủ AI mà chúng

được tạo cấu hình để thực hiện việc truyền thông với trạm cơ sở và/hoặc các thiết bị không dây khác sử dụng công nghệ truy cập radio (ví dụ, RAT mới (New RAT, NR) 5G, phát triển dài hạn (Long Term Evolution, LTE)).

Bộ nhận 10005 theo các phương án nhận luồng bit chứa dữ liệu video đám mây điểm hoặc tệp tin/phân đoạn, mà luồng bit được bao bọc trong đó, từ mạng hoặc phương tiện lưu trữ. Bộ nhận 10005 có thể thực hiện việc xử lý dữ liệu cần thiết tùy theo hệ thống mạng (ví dụ, hệ thống mạng truyền thông 4G, 5G, 6G, v.v.). Bộ nhận 10005 theo các phương án có thể khử bao bọc tệp tin/phân đoạn nhận được và xuất ra luồng bit. Theo các phương án, bộ nhận 10005 có thể bao gồm bộ khử bao bọc (hoặc môđun khử bao bọc) được tạo cấu hình để thực hiện hoạt động khử bao bọc. Bộ khử bao bọc có thể được thi hành dưới dạng phần tử (hoặc thành phần hoặc môđun) tách biệt khỏi bộ nhận 10005.

Bộ giải mã video đám mây điểm 10006 giải mã luồng bit chứa dữ liệu video đám mây điểm. Bộ giải mã video đám mây điểm 10006 có thể giải mã dữ liệu video đám mây điểm theo phương pháp mà qua đó dữ liệu video đám mây điểm này đã được mã hóa (ví dụ, theo quy trình ngược của hoạt động của bộ mã hóa video đám mây điểm 10002). Theo đó, bộ giải mã video đám mây điểm 10006 có thể giải mã dữ liệu video đám mây điểm bằng cách thực hiện việc lập mã giải nén đám mây điểm, mà nó là quy trình ngược của việc nén đám mây điểm. Việc lập mã giải nén đám mây điểm bao gồm việc lập mã G-PCC.

Bộ kết xuất 10007 kết xuất dữ liệu video đám mây điểm được giải mã. Theo một phương án, bộ kết xuất 10007 có thể kết xuất dữ liệu đám mây điểm được giải mã tùy theo công nhìn. Bộ kết xuất 10007 có thể xuất ra nội dung đám mây điểm bằng cách kết xuất không chỉ dữ liệu video đám mây điểm mà cả dữ liệu âm thanh. Theo các phương án, bộ kết xuất 10007 có thể bao gồm bộ hiển thị được tạo cấu hình để hiển thị nội dung đám mây điểm. Theo các phương án, bộ hiển thị có thể được thi hành dưới dạng thiết bị hoặc thành phần tách biệt thay vì được chứa trong bộ kết xuất 10007.

Các mũi tên được chỉ ra bởi các đường chấm trên hình vẽ biểu diễn đường truyền của thông tin phản hồi được thu nhận bởi thiết bị nhận 10004. Thông tin phản hồi là thông tin để phản ánh tính tương tác với người dùng mà người này tiêu thụ nội dung đám mây điểm, và bao gồm thông tin về người dùng này (ví dụ, thông tin hướng của

đầu, thông tin công nhìn, và dạng tương tự). Cụ thể, khi nội dung đám mây điểm là nội dung dành cho dịch vụ (ví dụ, dịch vụ tự lái, v.v.) mà nó yêu cầu sự tương tác với người dùng, thông tin phản hồi có thể được cung cấp cho phía truyền nội dung (ví dụ, thiết bị truyền 10000) và/hoặc nhà cung cấp dịch vụ. Theo các phương án, thông tin phản hồi có thể được sử dụng trong thiết bị nhận 10004 cũng như thiết bị truyền 10000, hoặc có thể không được cung cấp.

Thông tin hướng của đầu theo các phương án có thể biểu diễn thông tin về vị trí, hướng, góc, và chuyển động của đầu của người dùng. Thiết bị nhận 10004 theo các phương án có thể tính thông tin công nhìn dựa trên thông tin hướng của đầu này. Thông tin công nhìn là thông tin về vùng của video đám mây điểm mà người dùng đang nhìn (tức là, vùng mà người dùng hiện đang nhìn). Tức là, thông tin công nhìn là thông tin về vùng mà người dùng hiện đang nhìn trong video đám mây điểm. Nói cách khác, thông tin công nhìn hoặc vùng công nhìn có thể biểu diễn vùng mà người dùng đang nhìn trong video đám mây điểm. Điểm nhìn là điểm mà người dùng đang nhìn video đám mây điểm, và có thể biểu diễn điểm trung tâm của vùng công nhìn. Tức là, công nhìn là vùng được định tâm trên điểm nhìn, và kích thước và hình dạng của vùng này có thể được xác định bởi trường nhìn (Field Of View, FOV). Theo đó, thiết bị nhận 10004 có thể trích xuất thông tin công nhìn dựa trên FOV dọc hoặc ngang được hỗ trợ bởi thiết bị cũng như thông tin hướng của đầu. Thêm vào đó, thiết bị nhận 10004 có thể thực hiện việc phân tích việc nhìn chăm chú hoặc dạng tương tự dựa trên thông tin hướng của đầu và/hoặc thông tin công nhìn để xác định cách người dùng tiêu thụ video đám mây điểm, vùng mà người dùng nhìn chăm chú vào đó trong video đám mây điểm, và thời gian nhìn chăm chú. Theo các phương án, thiết bị nhận 10004 có thể truyền thông tin phản hồi bao gồm kết quả của việc phân tích việc nhìn chăm chú đến thiết bị truyền 10000. Theo các phương án, thiết bị chẳng hạn như bộ hiển thị VR/XR/AR/MR có thể trích xuất vùng công nhìn dựa trên vị trí/hướng của đầu của người dùng và FOV dọc hoặc ngang được hỗ trợ bởi thiết bị này. Theo các phương án, thông tin hướng của đầu và thông tin công nhìn có thể được gọi là thông tin phản hồi, thông tin báo hiệu, hoặc lý lịch dữ liệu.

Thông tin phản hồi theo các phương án có thể được thu nhận trong quy trình kết xuất và/hoặc hiển thị. Thông tin phản hồi có thể đạt được bởi một hoặc nhiều bộ cảm biến được chứa trong thiết bị nhận 10004. Theo các phương án, thông tin phản hồi có

thể đạt được bởi bộ kết xuất 10007 hoặc phần tử (hoặc thiết bị, thành phần, hoặc dạng tương tự) ngoài tách rời. Các đường chấm trên Fig.1 biểu diễn quy trình truyền thông tin phản hồi đạt được bởi bộ kết xuất 10007. Thông tin phản hồi có thể không chỉ được truyền đến phía truyền, mà còn được tiêu thụ bởi phía nhận. Tức là, hệ thống cung cấp nội dung đám mây điểm có thể xử lý (mã hóa/giải mã/kết xuất) dữ liệu đám mây điểm dựa trên thông tin phản hồi này. Ví dụ, bộ giải mã video đám mây điểm 10006 và bộ kết xuất 10007 ưu tiên có thể giải mã và kết xuất chỉ video đám mây điểm cho vùng hiện đang được nhìn bởi người dùng, dựa trên thông tin phản hồi, tức là, thông tin hướng của đầu và/hoặc thông tin công nhìn.

Thiết bị nhận 10004 có thể truyền thông tin phản hồi đến thiết bị truyền 10000. Thiết bị truyền 10000 (hoặc bộ mã hóa video đám mây điểm 10002) có thể thực hiện hoạt động mã hóa dựa trên thông tin phản hồi. Theo đó, hệ thống cung cấp nội dung đám mây điểm có thể xử lý một cách hiệu quả dữ liệu cần thiết (ví dụ, dữ liệu đám mây điểm tương ứng với vị trí đầu của người dùng) dựa trên thông tin phản hồi thay vì xử lý (mã hóa/giải mã) toàn bộ dữ liệu đám mây điểm, và cung cấp nội dung đám mây điểm đến người dùng.

Theo các phương án, thiết bị truyền 10000 có thể được gọi là bộ mã hóa, thiết bị truyền, bộ truyền, hệ thống truyền, hoặc dạng tương tự, và thiết bị nhận 10004 có thể được gọi là bộ giải mã, thiết bị nhận, bộ nhận, hệ thống nhận, hoặc dạng tương tự.

Dữ liệu đám mây điểm được xử lý trong hệ thống cung cấp nội dung đám mây điểm trên Fig.1 theo các phương án (thông qua một chuỗi các quy trình gồm thu nhận/mã hóa/truyền/giải mã/kết xuất) có thể được gọi là dữ liệu nội dung đám mây điểm hoặc dữ liệu video đám mây điểm. Theo các phương án, dữ liệu nội dung đám mây điểm có thể được sử dụng làm khái niệm bao hàm lý lịch dữ liệu hoặc thông tin báo hiệu liên quan đến dữ liệu đám mây điểm.

Các phần tử của hệ thống cung cấp nội dung đám mây điểm được minh họa trên Fig.1 có thể được thi hành bởi phần cứng, phần mềm, bộ xử lý, và/hoặc sự kết hợp của chúng.

Fig.2 là sơ đồ khối minh họa hoạt động cung cấp nội dung đám mây điểm theo các phương án.

Sơ đồ khối trên Fig.2 thể hiện hoạt động của hệ thống cung cấp nội dung đám mây điểm được mô tả trên Fig.1. Như được mô tả trên đây, hệ thống cung cấp nội dung đám mây điểm có thể xử lý dữ liệu đám mây điểm dựa trên việc lập mã nén đám mây điểm (ví dụ, G-PCC). Hệ thống cung cấp nội dung đám mây điểm theo các phương án (ví dụ, thiết bị truyền đám mây điểm 10000 hoặc đơn vị thu nhận video đám mây điểm 10001) có thể thu nhận video đám mây điểm (20000). Video đám mây điểm được thể hiện bởi đám mây điểm thuộc về hệ tọa độ để biểu diễn không gian 3D. Video đám mây điểm theo các phương án có thể bao gồm tệp tin Ply (định dạng tệp tin đa giác (Polygon File format) hoặc định dạng hình tam giác Stanford (Stanford Triangle format). Khi video đám mây điểm có một hoặc nhiều khung, video đám mây điểm được thu nhận có thể bao gồm một hoặc nhiều tệp tin Ply. Các tệp tin Ply này chứa dữ liệu đám mây điểm, chẳng hạn như dạng hình học và/hoặc các thuộc tính của điểm. Dạng hình học bao gồm các vị trí của các điểm. Vị trí của mỗi điểm có thể được thể hiện bởi các tham số (ví dụ, các giá trị của các trục X, Y, và Z) thể hiện hệ tọa độ ba chiều (ví dụ, hệ tọa độ gồm các trục X, Y và Z). Các thuộc tính bao gồm các thuộc tính của các điểm (ví dụ, thông tin về kết cấu, màu (trong YCbCr hoặc RGB), độ phản xạ r , độ trong suốt, v.v. của mỗi điểm). Điểm có một hoặc nhiều thuộc tính. Ví dụ, điểm có thể có thuộc tính là màu, hoặc hai thuộc tính là màu và độ phản xạ. Theo các phương án, dạng hình học có thể được gọi là các vị trí, thông tin dạng hình học, dữ liệu dạng hình học, hoặc dạng tương tự, và thuộc tính có thể được gọi là các thuộc tính, thông tin thuộc tính, dữ liệu thuộc tính, hoặc dạng tương tự.

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị truyền đám mây điểm 10000 hoặc đơn vị thu nhận video đám mây điểm 10001) có thể đạt được dữ liệu đám mây điểm từ thông tin (ví dụ, thông tin độ sâu, thông tin màu sắc, v.v.) liên quan đến quy trình thu nhận video đám mây điểm.

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị truyền 10000 hoặc bộ mã hóa video đám mây điểm 10002) theo các phương án có thể mã hóa dữ liệu đám mây điểm (20001). Hệ thống cung cấp nội dung đám mây điểm có thể mã hóa dữ liệu đám mây điểm dựa trên việc lập mã nén đám mây điểm. Như được mô tả trên đây, dữ liệu đám mây điểm có thể bao gồm dạng hình học và các thuộc tính của điểm. Theo đó, hệ thống cung cấp nội dung đám mây điểm có thể thực hiện việc mã hóa dạng hình học để

mã hóa dạng hình học và xuất ra luồng bit dạng hình học. Hệ thống cung cấp nội dung đám mây điểm có thể thực hiện việc mã hóa thuộc tính để mã hóa các thuộc tính và xuất ra luồng bit thuộc tính. Theo các phương án, hệ thống cung cấp nội dung đám mây điểm có thể thực hiện việc mã hóa thuộc tính dựa trên việc mã hóa dạng hình học. Luồng bit dạng hình học và luồng bit thuộc tính theo các phương án có thể được dồn kênh và xuất ra dưới dạng một luồng bit. Luồng bit theo các phương án có thể còn chứa thông tin báo hiệu liên quan đến việc mã hóa dạng hình học và việc mã hóa thuộc tính.

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị truyền 10000 hoặc bộ truyền 10003) theo các phương án có thể truyền dữ liệu đám mây điểm được mã hóa (20002). Như được minh họa trên Fig.1, dữ liệu đám mây điểm được mã hóa có thể được biểu diễn bởi luồng bit dạng hình học và luồng bit thuộc tính. Thêm vào đó, dữ liệu đám mây điểm được mã hóa có thể được truyền dưới dạng luồng bit cùng với thông tin báo hiệu liên quan đến việc mã hóa dữ liệu đám mây điểm (ví dụ, thông tin báo hiệu liên quan đến việc mã hóa dạng hình học và việc mã hóa thuộc tính). Hệ thống cung cấp nội dung đám mây điểm có thể bao bọc luồng bit mà nó mang dữ liệu đám mây điểm được mã hóa và truyền nó dưới dạng tệp tin hoặc phân đoạn.

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ nhận 10005) theo các phương án có thể nhận luồng bit chứa dữ liệu đám mây điểm được mã hóa. Thêm vào đó, hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ nhận 10005) có thể phân kênh luồng bit này.

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ giải mã video đám mây điểm 10005) có thể giải mã dữ liệu đám mây điểm được mã hóa (ví dụ, luồng bit dạng hình học, luồng bit thuộc tính) được truyền trong luồng bit này. Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ giải mã video đám mây điểm 10005) có thể giải mã dữ liệu video đám mây điểm dựa trên thông tin báo hiệu liên quan đến việc mã hóa dữ liệu video đám mây điểm được chứa trong luồng bit này. Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ giải mã video đám mây điểm 10005) có thể giải mã luồng bit dạng hình học để xây dựng lại các vị trí (dạng hình học) của các điểm. Hệ thống cung cấp nội dung đám mây điểm có thể xây dựng lại các thuộc tính của các điểm bằng cách giải mã luồng bit thuộc tính dựa trên dạng hình học được xây dựng lại. Hệ thống cung cấp nội dung đám

mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ giải mã video đám mây điểm 10005) có thể xây dựng lại video đám mây điểm dựa trên các vị trí theo dạng hình học được xây dựng lại và các thuộc tính được giải mã.

Hệ thống cung cấp nội dung đám mây điểm theo các phương án (ví dụ, thiết bị nhận 10004 hoặc bộ kết xuất 10007) có thể kết xuất dữ liệu đám mây điểm được giải mã (20004). Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004 hoặc bộ kết xuất 10007) có thể kết xuất dạng hình học và các thuộc tính được giải mã qua quy trình giải mã, sử dụng các phương pháp kết xuất khác nhau. Các điểm trong nội dung đám mây điểm có thể được kết xuất thành đỉnh có độ dày nhất định, khối lập phương có kích thước tối thiểu cụ thể được định tâm trên vị trí đỉnh tương ứng, hoặc hình tròn được định tâm trên vị trí đỉnh tương ứng. Tất cả hoặc một phần của nội dung đám mây điểm được kết xuất được cung cấp cho người dùng qua bộ hiển thị (ví dụ, bộ hiển thị VR/AR, bộ hiển thị thông thường, v.v.).

Hệ thống cung cấp nội dung đám mây điểm (ví dụ, thiết bị nhận 10004) theo các phương án có thể đạt được thông tin phản hồi (20005). Hệ thống cung cấp nội dung đám mây điểm có thể mã hóa và/hoặc giải mã dữ liệu đám mây điểm dựa trên thông tin phản hồi này. Thông tin phản hồi và hoạt động của hệ thống cung cấp nội dung đám mây điểm theo các phương án là giống với thông tin phản hồi và hoạt động được mô tả với tham chiếu đến Fig.1, và do đó phần mô tả chi tiết của chúng được lược bỏ.

Fig.3 minh họa quy trình thu nạp video đám mây điểm làm ví dụ theo các phương án.

Fig.3 minh họa quy trình thu nạp video đám mây điểm làm ví dụ của hệ thống cung cấp nội dung đám mây điểm được mô tả với tham chiếu đến Fig.1 đến Fig.2.

Nội dung đám mây điểm bao gồm video đám mây điểm (các ảnh và/hoặc các video) thể hiện vật thể và/hoặc môi trường được đặt trong các không gian 3D khác nhau (ví dụ, không gian 3D thể hiện môi trường thực, không gian 3D thể hiện môi trường ảo, v.v.). Theo đó, hệ thống cung cấp nội dung đám mây điểm theo các phương án có thể thu nạp video đám mây điểm nhờ sử dụng một hoặc nhiều camera (ví dụ, camera hồng ngoại có khả năng đạt được thông tin độ sâu, camera RGB có khả năng trích xuất thông tin màu sắc tương ứng với thông tin độ sâu này, v.v.), máy chiếu (ví dụ, máy chiếu mẫu dạng hồng ngoại để đạt được thông tin độ sâu), LiDAR, hoặc loại tương tự. Hệ thống cung

cấp nội dung đám mây điểm theo các phương án có thể trích xuất hình dạng của dạng hình học gồm các điểm trong không gian 3D từ thông tin độ sâu và trích xuất các thuộc tính của mỗi điểm từ thông tin màu sắc để đạt được dữ liệu đám mây điểm. Ảnh và/hoặc video theo các phương án có thể được thu nạp dựa trên ít nhất một kỹ thuật trong số kỹ thuật quay mặt vào trong và kỹ thuật quay mặt ra ngoài.

Phần bên trái trên Fig.3 minh họa kỹ thuật quay mặt vào trong. Kỹ thuật quay mặt vào trong chỉ kỹ thuật thu nạp các ảnh của đối tượng trung tâm bằng một hoặc nhiều camera (hoặc các bộ cảm biến camera) được đặt xung quanh đối tượng trung tâm này. Kỹ thuật quay mặt vào trong có thể được sử dụng để tạo ra nội dung đám mây điểm cung cấp ảnh 360 độ của đối tượng chủ đạo đến người dùng (ví dụ, nội dung VR/AR cung cấp ảnh 360 độ của đối tượng (ví dụ, đối tượng chủ đạo chẳng hạn như nhân vật, người chơi, vật thể, hoặc diễn viên) đến người dùng).

Phần bên phải trên Fig.3 minh họa kỹ thuật quay mặt ra ngoài. Kỹ thuật quay mặt ra ngoài chỉ kỹ thuật thu nạp các ảnh của môi trường của đối tượng trung tâm thay vì đối tượng trung tâm này bằng một hoặc nhiều camera (hoặc các bộ cảm biến camera) được đặt xung quanh đối tượng trung tâm này. Kỹ thuật quay mặt ra ngoài có thể được sử dụng để tạo ra nội dung đám mây điểm để cung cấp môi trường xung quanh mà nó xuất hiện từ điểm nhìn của người dùng (ví dụ, nội dung thể hiện môi trường bên ngoài mà nó có thể được cung cấp cho người dùng của xe tự lái).

Như được thể hiện trên Fig.3, nội dung đám mây điểm có thể được tạo ra dựa trên hoạt động thu nạp của một hoặc nhiều camera. Trong trường hợp này, hệ tọa độ có thể là khác nhau giữa các camera, và theo đó hệ thống cung cấp nội dung đám mây điểm có thể hiệu chỉnh một hoặc nhiều camera để thiết lập hệ tọa độ toàn cục trước hoạt động thu nạp. Thêm vào đó, hệ thống cung cấp nội dung đám mây điểm có thể tạo ra nội dung đám mây điểm bằng cách tổng hợp ảnh và/hoặc video tùy ý cùng với ảnh và/hoặc video được thu nạp bởi kỹ thuật thu nạp được mô tả trên đây. Hệ thống cung cấp nội dung đám mây điểm có thể không thực hiện hoạt động thu nạp được mô tả trên Fig.3 khi nó tạo ra nội dung đám mây điểm thể hiện không gian ảo. Hệ thống cung cấp nội dung đám mây điểm theo các phương án có thể thực hiện việc xử lý hậu kỳ trên ảnh và/hoặc video được thu nạp. Nói cách khác, hệ thống cung cấp nội dung đám mây điểm có thể loại bỏ khu vực không mong muốn (ví dụ, hậu cảnh), thực hiện không gian mà các ảnh và/hoặc các

video được thu nạp được kết nối vào đó, và khi có lỗ không gian, thực hiện hoạt động điền đầy lỗ không gian này.

Hệ thống cung cấp nội dung đám mây điểm có thể tạo ra một mảnh nội dung đám mây điểm bằng cách thực hiện việc biến đổi tọa độ trên các điểm của video đám mây điểm đạt được từ mỗi camera. Hệ thống cung cấp nội dung đám mây điểm có thể thực hiện việc biến đổi tọa độ trên các điểm dựa trên các tọa độ của vị trí của mỗi camera. Theo đó, hệ thống cung cấp nội dung đám mây điểm có thể tạo ra nội dung thể hiện một khoảng rộng, hoặc có thể tạo ra nội dung đám mây điểm có mật độ điểm cao.

Fig.4 minh họa bộ mã hóa video đám mây điểm làm ví dụ theo các phương án.

Fig.4 thể hiện ví dụ về bộ mã hóa video đám mây điểm 10002 trên Fig.1. Bộ mã hóa video đám mây điểm xây dựng lại và mã hóa dữ liệu đám mây điểm (ví dụ, các vị trí và/hoặc các thuộc tính của các điểm) để điều chỉnh chất lượng của nội dung đám mây điểm (ví dụ, thành không mất mát, có mất mát, hoặc gần như không mất mát) theo các điều kiện hoặc các ứng dụng mạng. Khi kích thước tổng thể của nội dung đám mây điểm là lớn (ví dụ, nội dung đám mây điểm 60 Gbps được cho với 30 fps), hệ thống cung cấp nội dung đám mây điểm có thể thất bại trong việc tạo luồng nội dung này theo thời gian thực. Theo đó, hệ thống cung cấp nội dung đám mây điểm có thể xây dựng lại nội dung đám mây điểm dựa trên tốc độ bit mục tiêu tối đa để cung cấp nó tương ứng với môi trường mạng hoặc dạng tương tự.

Như được mô tả với tham chiếu đến Fig.1 đến Fig.2, bộ mã hóa video đám mây điểm có thể thực hiện việc mã hóa dạng hình học và việc mã hóa thuộc tính. Việc mã hóa dạng hình học được thực hiện trước việc mã hóa thuộc tính.

Bộ mã hóa video đám mây điểm theo các phương án bao gồm bộ biến đổi tọa độ (biến đổi các tọa độ) 40000, bộ lượng tử hóa (lượng tử hóa và loại bỏ các điểm (tạo đơn vị ảnh ba chiều)) 40001, bộ phân tích cây bát phân (phân tích cây bát phân) 40002, bộ phân tích xấp xỉ bề mặt (phân tích xấp xỉ bề mặt) 40003, bộ mã hóa số học (mã hóa số học) 40004, bộ xây dựng lại dạng hình học (xây dựng lại dạng hình học) 40005, bộ biến đổi màu (biến đổi các màu) 40006, bộ biến đổi thuộc tính (biến đổi các thuộc tính) 40007, bộ biến đổi RAHT (RAHT) 40008, bộ tạo ra LOD (tạo ra LOD) 40009, bộ biến đổi nâng (lifting) (nâng) 40010, bộ lượng tử hóa hệ số (lượng tử hóa các hệ số) 40011, và/hoặc bộ mã hóa số học (mã hóa số học) 40012.

Bộ biến đổi tọa độ 40000, bộ lượng tử hóa 40001, bộ phân tích cây bát phân 40002, bộ phân tích xấp xỉ bề mặt 40003, bộ mã hóa số học 40004, và bộ xây dựng lại dạng hình học 40005 có thể thực hiện việc mã hóa dạng hình học. Việc mã hóa dạng hình học theo các phương án có thể bao gồm việc lập mã dạng hình học cây bát phân, việc lập mã trực tiếp, việc mã hóa dạng hình học trisoup, và việc mã hóa entropy. Việc lập mã trực tiếp và việc mã hóa dạng hình học trisoup được áp dụng có chọn lọc hoặc kết hợp với nhau. Việc mã hóa dạng hình học không bị giới hạn vào ví dụ được mô tả trên đây.

Như được thể hiện trên hình vẽ, bộ biến đổi tọa độ 40000 theo các phương án nhận các vị trí và biến đổi chúng thành các tọa độ. Ví dụ, các vị trí này có thể được biến đổi thành thông tin vị trí trong không gian ba chiều (ví dụ, không gian ba chiều được thể hiện bởi hệ tọa độ XYZ). Thông tin vị trí trong không gian ba chiều theo các phương án có thể được gọi là thông tin dạng hình học.

Bộ lượng tử hóa 40001 theo các phương án lượng tử hoá thông tin dạng hình học này. Ví dụ, bộ lượng tử hóa 40001 có thể lượng tử hóa các điểm dựa trên giá trị vị trí tối thiểu của tất cả các điểm (ví dụ, giá trị tối thiểu trên mỗi trục X, Y, và Z). Bộ lượng tử hóa 40001 thực hiện hoạt động lượng tử hóa gồm việc nhân chênh lệch giữa giá trị vị trí tối thiểu và giá trị vị trí của mỗi điểm với giá trị thang chia tỷ lệ lượng tử hóa được thiết lập trước và sau đó tìm giá trị nguyên gần nhất bằng cách làm tròn xuống giá trị thu được qua phép nhân này. Do đó, một hoặc nhiều điểm có thể có cùng vị trí (hoặc giá trị vị trí) được lượng tử hoá. Bộ lượng tử hóa 40001 theo các phương án thực hiện việc tạo đơn vị ảnh ba chiều dựa trên các vị trí được lượng tử hoá này để xây dựng lại các điểm được lượng tử hoá. Việc tạo đơn vị ảnh ba chiều (voxelization) nghĩa là đơn vị nhỏ nhất thể hiện thông tin vị trí trong không gian 3D. Các điểm của nội dung đám mây điểm (hoặc video đám mây điểm 3D) theo các phương án có thể được chứa trong một hoặc nhiều đơn vị ảnh ba chiều. Thuật ngữ đơn vị ảnh ba chiều (voxel), mà nó là từ ghép của thể tích (volume) và điểm ảnh (pixel), chỉ không gian lập phương 3D được tạo ra khi không gian 3D được phân chia thành các đơn vị (đơn vị=1,0) dựa trên các trục thể hiện không gian 3D này (ví dụ, trục X, trục Y, và trục Z). Bộ lượng tử hóa 40001 có thể so khớp các nhóm điểm trong không gian 3D với các đơn vị ảnh ba chiều. Theo các phương án, một đơn vị ảnh ba chiều có thể bao gồm chỉ một điểm. Theo các phương án, một đơn vị ảnh ba chiều có thể bao gồm một hoặc nhiều điểm. Để thể hiện một đơn vị ảnh

ba chiều dưới dạng một điểm, vị trí của điểm trung tâm của đơn vị ảnh ba chiều có thể được thiết lập dựa trên các vị trí của một hoặc nhiều điểm được chứa trong đơn vị ảnh ba chiều này. Trong trường hợp này, các thuộc tính của tất cả các vị trí được chứa trong một đơn vị ảnh ba chiều có thể được kết hợp và được ấn định cho đơn vị ảnh ba chiều này.

Bộ phân tích cây bát phân 40002 theo các phương án thực hiện việc lập mã dạng hình học cây bát phân (hoặc lập mã cây bát phân) để đưa ra các đơn vị ảnh ba chiều trong cấu trúc cây bát phân. Cấu trúc cây bát phân (octree) thể hiện các điểm được so khớp với các đơn vị ảnh ba chiều, dựa trên cấu trúc cây phân tám (octal tree).

Bộ phân tích xấp xỉ bề mặt 40003 theo các phương án có thể phân tích và tính xấp xỉ cây bát phân. Việc phân tích và tính xấp xỉ cây bát phân theo các phương án là quy trình phân tích vùng chứa nhiều điểm để cung cấp cây bát phân và việc tạo đơn vị ảnh ba chiều một cách hiệu quả.

Bộ mã hóa số học 40004 theo các phương án thực hiện việc mã hóa entropy trên cây bát phân và/hoặc cây bát phân được tính xấp xỉ. Ví dụ, sơ đồ mã hóa này bao gồm việc mã hóa số học. Là kết quả của việc mã hóa này, luồng bit dạng hình học được tạo ra.

Bộ biến đổi màu 40006, bộ biến đổi thuộc tính 40007, bộ biến đổi RAHT 40008, bộ tạo ra LOD 40009, bộ biến đổi nâng 40010, bộ lượng tử hóa hệ số 40011, và/hoặc bộ mã hóa số học 40012 thực hiện việc mã hóa thuộc tính. Như được mô tả trên đây, một điểm có thể có một hoặc nhiều thuộc tính. Việc mã hóa thuộc tính theo các phương án được áp dụng tương đương cho các thuộc tính mà một điểm có. Tuy nhiên, khi thuộc tính (ví dụ, màu) bao gồm một hoặc nhiều phần tử, việc mã hóa thuộc tính được áp dụng độc lập cho mỗi phần tử. Việc mã hóa thuộc tính theo các phương án bao gồm việc lập mã biến đổi màu, việc lập mã biến đổi thuộc tính, việc lập mã biến đổi phân cấp thích ứng vùng (region adaptive hierarchical transform, RAHT), việc lập mã dự đoán lân cận gần nhất phân cấp dựa trên việc nội suy (biến đổi dự đoán), và việc dự đoán lân cận gần nhất phân cấp dựa trên việc nội suy cùng với việc lập mã bước cập nhật/nâng (biến đổi nâng). Tùy thuộc vào nội dung đám mây điểm, việc lập mã RAHT, việc lập mã biến đổi dự đoán và việc lập mã biến đổi nâng được mô tả trên đây có thể được sử dụng có chọn

lọc, hoặc sự kết hợp của một hoặc nhiều sơ đồ lập mã này có thể được sử dụng. Việc mã hóa thuộc tính theo các phương án không bị giới hạn vào ví dụ được mô tả trên đây.

Bộ biến đổi màu 40006 theo các phương án thực hiện việc lập mã biến đổi màu gồm việc biến đổi các giá trị màu (hoặc các kết cấu) được chứa trong các thuộc tính. Ví dụ, bộ biến đổi màu 40006 có thể biến đổi định dạng của thông tin màu sắc (ví dụ, từ RGB thành YCbCr). Hoạt động của bộ biến đổi màu 40006 theo các phương án có thể được áp dụng theo cách tùy chọn theo các giá trị màu được chứa trong các thuộc tính.

Bộ xây dựng lại dạng hình học 40005 theo các phương án xây dựng lại (giải nén) cây bát phân và/hoặc cây bát phân được tính xấp xỉ. Bộ xây dựng lại dạng hình học 40005 xây dựng lại cây bát phân/các đơn vị ảnh ba chiều dựa trên kết quả của việc phân tích sự phân bố của các điểm. Cây bát phân/các đơn vị ảnh ba chiều được xây dựng lại có thể được gọi là dạng hình học được xây dựng lại (dạng hình học được khôi phục).

Bộ biến đổi thuộc tính 40007 theo các phương án thực hiện việc biến đổi thuộc tính để biến đổi các thuộc tính dựa trên dạng hình học được xây dựng lại và/hoặc các vị trí mà việc mã hóa dạng hình học không được thực hiện trên đó. Như được mô tả trên đây, vì các thuộc tính phụ thuộc vào dạng hình học, bộ biến đổi thuộc tính 40007 có thể biến đổi các thuộc tính dựa trên thông tin dạng hình học được xây dựng lại. Ví dụ, dựa trên giá trị vị trí của điểm được chứa trong đơn vị ảnh ba chiều, bộ biến đổi thuộc tính 40007 có thể biến đổi thuộc tính của điểm này tại vị trí này. Như được mô tả trên đây, khi vị trí của trung tâm của đơn vị ảnh ba chiều được thiết lập dựa trên các vị trí của một hoặc nhiều điểm được chứa trong đơn vị ảnh ba chiều này, bộ biến đổi thuộc tính 40007 biến đổi các thuộc tính của một hoặc nhiều điểm này. Khi việc mã hóa dạng hình học trisoup được thực hiện, bộ biến đổi thuộc tính 40007 có thể biến đổi các thuộc tính dựa trên việc mã hóa dạng hình học trisoup.

Bộ biến đổi thuộc tính 40007 có thể thực hiện việc biến đổi thuộc tính bằng cách tính trung bình của các thuộc tính hoặc các giá trị thuộc tính của các điểm lân cận (ví dụ, màu hoặc độ phản xạ của mỗi điểm) trong vị trí/bán kính cụ thể tính từ vị trí (hoặc giá trị vị trí) của trung tâm của mỗi đơn vị ảnh ba chiều. Bộ biến đổi thuộc tính 40007 có thể áp dụng trọng số tùy theo khoảng cách tính từ trung tâm của mỗi điểm khi tính trung bình. Theo đó, mỗi đơn vị ảnh ba chiều có vị trí và thuộc tính (hoặc giá trị thuộc tính) được tính toán.

Bộ biến đổi thuộc tính 40007 có thể tìm kiếm các điểm lân cận tồn tại trong vị trí/bán kính cụ thể tính từ vị trí của trung tâm của mỗi đơn vị ảnh ba chiều dựa trên cây K-D hoặc mã Morton. Cây K-D là cây tìm kiếm nhị phân và nó hỗ trợ cấu trúc dữ liệu có khả năng quản lý các điểm dựa trên các vị trí sao cho việc tìm kiếm lân cận gần nhất (nearest neighbor search, NNS) có thể nhanh chóng được thực hiện. Mã Morton được tạo ra bằng cách đưa ra các tọa độ (ví dụ, (x, y, z)) thể hiện các vị trí 3D của tất cả các điểm dưới dạng các giá trị bit và trộn các bit này. Ví dụ, khi các tọa độ thể hiện vị trí của điểm là $(5, 9, 1)$, các giá trị bit dành cho các tọa độ này là $(0101, 1001, 0001)$. Việc trộn các giá trị bit theo chỉ số bit theo thứ tự z, y , và x cho ra 010001000111 . Giá trị này được biểu diễn dưới dạng số thập phân là 1095. Tức là, giá trị mã Morton của điểm có các tọa độ $(5, 9, 1)$ là 1095. Bộ biến đổi thuộc tính 40007 có thể xếp thứ tự các điểm dựa trên các giá trị mã Morton và thực hiện NNS qua quy trình giao nhau độ sâu đầu tiên (depth-first traversal). Sau hoạt động biến đổi thuộc tính, cây K-D hoặc mã Morton được sử dụng khi NNS là cần thiết cho một quy trình biến đổi khác để lập mã thuộc tính.

Như được thể hiện trên hình vẽ, các thuộc tính được biến đổi được nhập vào bộ biến đổi RAHT 40008 và/hoặc bộ tạo ra LOD 40009.

Bộ biến đổi RAHT 40008 theo các phương án thực hiện việc lập mã RAHT để dự đoán thông tin thuộc tính dựa trên thông tin dạng hình học được xây dựng lại. Ví dụ, bộ biến đổi RAHT 40008 có thể dự đoán thông tin thuộc tính của nút ở mức cao hơn trong cây bát phân dựa trên thông tin thuộc tính liên quan đến nút ở mức thấp hơn trong cây bát phân.

Bộ tạo ra LOD 40009 theo các phương án tạo ra mức độ chi tiết (LOD). LOD theo các phương án là độ chi tiết của nội dung đám mây điểm. Khi giá trị LOD giảm, nó chỉ ra rằng chi tiết của nội dung đám mây điểm bị suy giảm. Khi giá trị LOD tăng, nó chỉ ra rằng chi tiết của nội dung đám mây điểm được tăng cường. Các điểm có thể được phân loại theo LOD.

Bộ biến đổi nâng 40010 theo các phương án thực hiện việc lập mã biến đổi nâng gồm việc biến đổi các thuộc tính của đám mây điểm dựa trên các trọng số. Như được mô tả trên đây, việc lập mã biến đổi nâng có thể được tùy chọn áp dụng.

Bộ lượng tử hóa hệ số 40011 theo các phương án lượng tử hoá các thuộc tính được lập mã thuộc tính dựa trên các hệ số.

Bộ mã hóa số học 40012 theo các phương án mã hóa các thuộc tính được lượng tử hoá dựa trên việc lập mã số học.

Mặc dù không được thể hiện trên hình vẽ, các phần tử của bộ mã hóa video đám mây điểm trên Fig.4 có thể được thi hành bởi phần cứng bao gồm một hoặc nhiều bộ xử lý hoặc mạch tích hợp được tạo cấu hình để truyền thông với một hoặc nhiều bộ nhớ được chứa trong thiết bị cung cấp nội dung đám mây điểm, phần mềm, phần sụn hoặc sự kết hợp của chúng. Một hoặc nhiều bộ xử lý này có thể thực hiện ít nhất một hoạt động và/hoặc chức năng trong số các hoạt động và/hoặc chức năng của các phần tử của bộ mã hóa video đám mây điểm trên Fig.4 được mô tả trên đây. Thêm vào đó, một hoặc nhiều bộ xử lý này có thể hoạt động hoặc thực thi tập các chương trình và/hoặc lệnh phần mềm để thực hiện các hoạt động và/hoặc chức năng của các phần tử của bộ mã hóa video đám mây điểm trên Fig.4. Một hoặc nhiều bộ nhớ theo các phương án có thể bao gồm bộ nhớ truy cập ngẫu nhiên tốc độ cao, hoặc bao gồm bộ nhớ phi khả biến (ví dụ, một hoặc nhiều thiết bị lưu trữ đĩa từ, thiết bị nhớ tác động nhanh, hoặc các thiết bị nhớ trạng thái rắn phi khả biến khác).

Fig.5 thể hiện ví dụ về các đơn vị ảnh ba chiều theo các phương án.

Fig.5 thể hiện các đơn vị ảnh ba chiều được đặt trong không gian 3D được thể hiện bởi hệ tọa độ gồm ba trục, là trục X, trục Y, và trục Z. Như được mô tả với tham chiếu đến Fig.4, bộ mã hóa video đám mây điểm (ví dụ, bộ lượng tử hóa 40001) có thể thực hiện việc tạo đơn vị ảnh ba chiều. Đơn vị ảnh ba chiều (voxel) chỉ không gian lập phương 3D được tạo ra khi không gian 3D được phân chia thành các đơn vị (đơn vị=1,0) dựa trên các trục thể hiện không gian 3D này (ví dụ, trục X, trục Y, và trục Z). Fig.5 thể hiện ví dụ về các đơn vị ảnh ba chiều được tạo ra qua cấu trúc cây bát phân trong đó hộp giới hạn được canh chỉnh trục dạng lập phương được định ra bởi hai cực $(0, 0, 0)$ và $(2^d, 2^d, 2^d)$ được chia nhỏ theo cách đệ quy. Một đơn vị ảnh ba chiều bao gồm ít nhất một điểm. Các tọa độ không gian của đơn vị ảnh ba chiều có thể được ước lượng từ mối quan hệ vị trí với nhóm đơn vị ảnh ba chiều. Như được mô tả trên đây, đơn vị ảnh ba chiều có thuộc tính (chẳng hạn như màu hoặc độ phản xạ) giống như các điểm ảnh của ảnh/video 2D. Các chi tiết về đơn vị ảnh ba chiều là giống với các chi tiết được mô tả với tham chiếu đến Fig.4, và do đó phần mô tả của chúng được lược bỏ.

Fig.6 thể hiện ví dụ về cây bát phân và mã chiếm giữ theo các phương án.

Như được mô tả với tham chiếu đến Fig.1 đến Fig.4, hệ thống cung cấp nội dung đám mây điểm (bộ mã hóa video đám mây điểm 10002) hoặc bộ phân tích cây bát phân 40002 của bộ mã hóa video đám mây điểm thực hiện việc lập mã dạng hình học cây bát phân (hoặc việc lập mã cây bát phân) dựa trên cấu trúc cây bát phân để quản lý một cách hiệu quả vùng và/hoặc vị trí của đơn vị ảnh ba chiều.

Phần trên trên Fig.6 thể hiện cấu trúc cây bát phân. Không gian 3D của nội dung đám mây điểm theo các phương án được thể hiện bởi các trục (ví dụ, trục X, trục Y, và trục Z) của hệ tọa độ này. Cấu trúc cây bát phân được tạo ra bằng cách chia nhỏ theo cách đệ quy hộp giới hạn được canh chỉnh trục dạng lập phương được định ra bởi hai cực $(0, 0, 0)$ và $(2^d, 2^d, 2^d)$. Ở đây, 2^d có thể được thiết lập là giá trị cấu thành nên hộp giới hạn nhỏ nhất xung quanh tất cả các điểm của nội dung đám mây điểm (hoặc video đám mây điểm). Ở đây, d biểu thị độ sâu của cây bát phân. Giá trị của d được xác định trong phương trình 1. Trong phương trình 1, $(x_n^{int}, y_n^{int}, z_n^{int})$ biểu thị các vị trí (hoặc các giá trị vị trí) của các điểm được lượng tử hoá.

Phương trình 1

$$d = \text{Ceil} \left(\text{Log}_2(\text{Max}(x_n^{int}, y_n^{int}, z_n^{int}, n = 1, \dots, N) + 1) \right)$$

Như được thể hiện ở giữa phần trên trên Fig.6, toàn bộ không gian 3D có thể được phân chia thành tám không gian theo việc phân vùng. Mỗi không gian được phân chia được thể hiện bởi khối lập phương với sáu mặt. Như được thể hiện trên phần phía trên bên phải trên Fig.6, mỗi không gian trong số tám không gian này lại được phân chia dựa trên các trục của hệ tọa độ (ví dụ, trục X, trục Y, và trục Z). Theo đó, mỗi không gian được phân chia thành tám không gian nhỏ hơn. Không gian nhỏ hơn được phân chia này cũng được thể hiện bởi khối lập phương với sáu mặt. Sơ đồ phân vùng này được áp dụng cho đến khi nút lá của cây bát phân trở thành đơn vị ảnh ba chiều.

Phần dưới trên Fig.6 thể hiện mã chiếm giữ cây bát phân. Mã chiếm giữ của cây bát phân được tạo ra để chỉ ra liệu mỗi không gian trong số tám không gian được phân chia được tạo ra bằng cách phân chia một không gian có chứa ít nhất một điểm hay không. Theo đó, mã chiếm giữ đơn được thể hiện bởi tám nút con. Mỗi nút con thể hiện sự chiếm giữ của không gian được phân chia, và nút con có giá trị nằm trong 1 bit. Theo đó, mã chiếm giữ được thể hiện dưới dạng mã 8 bit. Tức là, khi ít nhất một điểm được chứa trong không gian tương ứng với nút con, nút này được ấn định giá trị 1. Khi không

có điểm nào được chứa trong không gian tương ứng với nút con (không gian rỗng), nút này được ấn định giá trị 0. Vì mã chiếm giữ được thể hiện trên Fig.6 là 00100001, nó chỉ ra rằng các không gian tương ứng với nút con thứ ba và nút con thứ tám trong số tám nút con này mỗi nút chứa ít nhất một điểm. Như được thể hiện trên hình vẽ, mỗi nút trong số nút con thứ ba và nút con thứ tám đều có tám nút con, và các nút con này được thể hiện bởi mã chiếm giữ 8 bit. Hình vẽ này thể hiện rằng mã chiếm giữ của nút con thứ ba là 10000111, và mã chiếm giữ của nút con thứ tám là 01001111. Bộ mã hóa video đám mây điểm (ví dụ, bộ mã hóa số học 40004) theo các phương án có thể thực hiện việc mã hóa entropy trên các mã chiếm giữ. Để làm tăng hiệu quả nén, bộ mã hóa video đám mây điểm có thể thực hiện việc lập mã nội/liên trên các mã chiếm giữ. Thiết bị nhận (ví dụ, thiết bị nhận 10004 hoặc bộ giải mã video đám mây điểm 10006) theo các phương án xây dựng lại cây bát phân dựa trên các mã chiếm giữ.

Bộ mã hóa video đám mây điểm (ví dụ, bộ phân tích cây bát phân 40002) theo các phương án có thể thực hiện việc tạo đơn vị ảnh ba chiều và việc lập mã cây bát phân để lưu trữ các vị trí của các điểm. Tuy nhiên, các điểm không phải luôn luôn được phân bố đều trong không gian 3D, và theo đó có thể có vùng cụ thể mà ít điểm hơn có mặt trong đó. Theo đó, việc thực hiện tạo đơn vị ảnh ba chiều cho toàn bộ không gian 3D sẽ là không hiệu quả. Ví dụ, khi vùng cụ thể chứa ít điểm, việc tạo đơn vị ảnh ba chiều không cần phải được thực hiện trong vùng cụ thể này.

Theo đó, đối với vùng cụ thể được mô tả trên đây (hoặc nút khác với nút lá của cây bát phân), bộ mã hóa video đám mây điểm theo các phương án có thể bỏ qua việc tạo đơn vị ảnh ba chiều và thực hiện việc lập mã trực tiếp để trực tiếp lập mã các vị trí của các điểm được chứa trong vùng cụ thể này. Các tọa độ của điểm lập mã trực tiếp theo các phương án được gọi là chế độ lập mã trực tiếp (direct coding mode, DCM). Bộ mã hóa video đám mây điểm theo các phương án cũng có thể thực hiện việc mã hóa dạng hình học trisoup, mà nó là để xây dựng lại các vị trí của các điểm trong vùng (hoặc nút) cụ thể dựa trên các đơn vị ảnh ba chiều, dựa trên mô hình bề mặt. Việc mã hóa dạng hình học trisoup là việc mã hóa dạng hình học mà nó biểu diễn đối tượng dưới dạng chuỗi các lưới hình tam giác. Theo đó, bộ giải mã video đám mây điểm có thể tạo ra đám mây điểm từ bề mặt lưới này. Việc lập mã trực tiếp và việc mã hóa dạng hình học trisoup theo các phương án có thể được thực hiện có chọn lọc. Thêm vào đó, việc lập

mã trực tiếp và mã hóa dạng hình học trisoup theo các phương án có thể được thực hiện kết hợp với việc lập mã dạng hình học cây bát phân (hoặc việc lập mã cây bát phân).

Để thực hiện việc lập mã trực tiếp, tùy chọn sử dụng chế độ trực tiếp để áp dụng việc lập mã trực tiếp cần phải được kích hoạt. Nút mà việc lập mã trực tiếp được áp dụng cho nó không phải là nút lá, và các điểm nhỏ hơn ngưỡng cần phải có mặt bên trong nút cụ thể. Thêm vào đó, tổng số lượng các điểm mà việc lập mã trực tiếp cần được áp dụng cho chúng cần phải không vượt quá ngưỡng được thiết lập trước. Khi các điều kiện trên đây được thỏa mãn, bộ mã hóa video đám mây điểm (hoặc bộ mã hóa số học 40004) theo các phương án có thể thực hiện việc lập mã entropy trên các vị trí (hoặc các giá trị vị trí) của các điểm này.

Bộ mã hóa video đám mây điểm (ví dụ, bộ phân tích xấp xỉ bề mặt 40003) theo các phương án có thể xác định mức cụ thể của cây bát phân (mức nhỏ hơn độ sâu d của cây bát phân), và mô hình bề mặt có thể được sử dụng bắt đầu với mức đó để thực hiện việc mã hóa dạng hình học trisoup để xây dựng lại các vị trí của các điểm trong vùng của nút dựa trên các đơn vị ảnh ba chiều (chế độ Trisoup). Bộ mã hóa video đám mây điểm theo các phương án có thể chỉ rõ mức mà việc mã hóa dạng hình học trisoup cần được áp dụng tại đó. Ví dụ, khi mức cụ thể này bằng độ sâu của cây bát phân, bộ mã hóa video đám mây điểm không hoạt động trong chế độ trisoup. Nói cách khác, bộ mã hóa video đám mây điểm theo các phương án có thể hoạt động trong chế độ trisoup chỉ khi mức được chỉ rõ là nhỏ hơn giá trị độ sâu của cây bát phân. Vùng lập phương 3D của các nút tại mức được chỉ rõ này theo các phương án được gọi là khối. Một khối có thể bao gồm một hoặc nhiều đơn vị ảnh ba chiều. Khối hoặc đơn vị ảnh ba chiều này có thể tương ứng với viên gạch. Dạng hình học được biểu diễn dưới dạng bề mặt bên trong mỗi khối. Bề mặt theo các phương án có thể giao với mỗi cạnh của khối nhiều nhất một lần.

Một khối có 12 cạnh, và theo đó có ít nhất 12 chỗ giao trong một khối. Mỗi chỗ giao được gọi là đỉnh (hoặc chóp). Đỉnh có mặt dọc theo cạnh được phát hiện khi có ít nhất một đơn vị ảnh ba chiều được chiếm giữ liền kề với cạnh này trong số tất cả các khối đang chia sẻ cạnh này. Đơn vị ảnh ba chiều được chiếm giữ này theo các phương án chỉ đơn vị ảnh ba chiều chứa điểm. Vị trí của đỉnh được phát hiện dọc theo cạnh này

là vị trí trung bình dọc theo cạnh này của tất cả các đơn vị ảnh ba chiều liền kề với cạnh này trong số tất cả các khối đang chia sẻ cạnh này.

Khi đỉnh được phát hiện, bộ mã hóa video đám mây điểm theo các phương án có thể thực hiện việc mã hóa entropy trên điểm bắt đầu (x, y, z) của cạnh này, vector hướng $(\Delta x, \Delta y, \Delta z)$ của cạnh này, và giá trị vị trí đỉnh (giá trị vị trí tương đối bên trong cạnh này). Khi việc mã hóa dạng hình học trisoup được áp dụng, bộ mã hóa video đám mây điểm theo các phương án (ví dụ, bộ xây dựng lại dạng hình học 40005) có thể tạo ra dạng hình học được khôi phục (dạng hình học được xây dựng lại) bằng cách thực hiện các quy trình xây dựng lại hình tam giác, lấy mẫu lên, và tạo đơn vị ảnh ba chiều.

Các đỉnh được đặt tại cạnh của khối xác định bề mặt mà nó đi qua khối này. Bề mặt này theo các phương án là đa giác không phẳng. Trong quy trình xây dựng lại hình tam giác, bề mặt được thể hiện bởi hình tam giác được xây dựng lại dựa trên điểm bắt đầu của cạnh, vector hướng của cạnh này, và các giá trị vị trí của các đỉnh. Quy trình xây dựng lại hình tam giác được thực hiện theo phương trình 2 thông qua việc: i) tính giá trị trọng tâm (centroid) của mỗi đỉnh, ii) trừ giá trị tâm này khỏi mỗi giá trị đỉnh, và iii) ước lượng tổng các bình phương của các giá trị thu được bởi phép trừ này.

Phương trình 2

$$\textcircled{1} \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad \textcircled{2} \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \\ \bar{z}_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \quad \textcircled{3} \begin{bmatrix} \sigma_x^2 \\ \sigma_y^2 \\ \sigma_z^2 \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} \bar{x}_i^2 \\ \bar{y}_i^2 \\ \bar{z}_i^2 \end{bmatrix}$$

Sau đó, giá trị tối thiểu của tổng này được ước lượng, và quy trình chiếu được thực hiện theo trục với giá trị tối thiểu này. Ví dụ, khi phần tử x là tối thiểu, mỗi đỉnh được chiếu trên trục x đối với tâm của khối, và được chiếu trên mặt phẳng (y, z) . Khi các giá trị thu được qua phép chiếu trên mặt phẳng (y, z) là (a_i, b_i) , giá trị của θ được ước lượng qua $\text{atan2}(b_i, a_i)$, và các đỉnh được xếp thứ tự dựa trên giá trị của θ . Bảng 1 dưới đây thể hiện tổ hợp của các đỉnh để tạo ra hình tam giác tùy theo số lượng các đỉnh. Các đỉnh được xếp thứ tự từ 1 đến n . Bảng 1 dưới đây thể hiện rằng đối với bốn đỉnh, hai hình tam giác có thể được xây dựng theo các tổ hợp của các đỉnh. Hình tam giác thứ nhất có thể bao gồm các đỉnh 1, 2, và 3 trong số các đỉnh được xếp thứ tự, và hình tam giác thứ hai có thể bao gồm các đỉnh 3, 4, và 1 trong số các đỉnh được xếp thứ tự.

Bảng 1 Các hình tam giác được tạo ra từ các đỉnh được xếp thứ tự $1, \dots, n$

n	Các hình tam giác
3	(1,2; 3)
4	(1,2,3), (3,4,1)
5	(1,2,3), (3,4,5), (5,1,3)
6	(1,2,3), (3,4,5), (5,6,1), (1,3,5)
7	(1,2,3), (3,4,5), (5,6,7), (7,1,3), (3,5,7)
8	(1,2,3), (3,4,5), (5,6,7), (7,8,1), (1,3,5), (5,7,1)
9	(1,2,3), (3,4,5), (5,6,7), (7,8,9), (9,1,3), (3,5,7), (7,9,3)
10	(1,2,3), (3,4,5), (5,6,7), (7,8,9), (9,10,1), (1,3,5), (5,7,9), (9,1,5)
11	(1,2,3), (3,4,5), (5,6,7), (7,8,9), (9,10,11), (11,1,3), (3,5,7), (7,9,11), (11,3,7)
12	(1,2,3), (3,4,5), (5,6,7), (7,8,9), (9,10,11), (11,12,1), (1,3,5), (5,7,9), (9,11,1), (1,5,9)

Quy trình lấy mẫu lên được thực hiện để thêm các điểm ở giữa dọc theo cạnh của hình tam giác và thực hiện việc tạo đơn vị ảnh ba chiều. Các điểm được thêm vào được tạo ra dựa trên thừa số lấy mẫu lên và chiều rộng của khối. Các điểm được thêm vào được gọi là các đỉnh được tinh chỉnh. Bộ mã hóa video đám mây điểm theo các phương án có thể tạo đơn vị ảnh ba chiều đối với các đỉnh được tinh chỉnh. Thêm vào đó, bộ mã hóa video đám mây điểm có thể thực hiện việc mã hóa thuộc tính dựa trên các vị trí (hoặc các giá trị vị trí) được tạo đơn vị ảnh ba chiều.

Fig.7 thể hiện ví dụ về mẫu dạng nút lân cận theo các phương án. Để làm tăng hiệu quả nén video đám mây điểm, bộ mã hóa video đám mây điểm theo các phương án có thể thực hiện việc lập mã entropy dựa trên việc lập mã số học thích ứng ngữ cảnh.

Bộ mã hóa video đám mây điểm có thể mã hóa entropy dựa trên việc lập mã số học thích ứng ngữ cảnh để nâng cao hiệu quả nén của video đám mây điểm.

Như được mô tả với tham chiếu đến Fig.1 đến Fig.6, hệ thống cung cấp nội dung đám mây điểm hoặc bộ mã hóa video đám mây điểm 10002 trên Fig.1, hoặc bộ mã hóa video đám mây điểm hoặc bộ mã hóa số học 40004 trên Fig.4 có thể thực hiện việc lập mã entropy trên mã chiếm giữ ngay lập tức. Thêm vào đó, hệ thống cung cấp nội dung đám mây điểm hoặc bộ mã hóa video đám mây điểm này có thể thực hiện việc mã hóa entropy (việc mã hóa nội) dựa trên mã chiếm giữ của nút hiện tại và việc chiếm giữ của các nút lân cận, hoặc thực hiện việc mã hóa entropy (việc mã hóa liên) dựa trên mã chiếm giữ của khung trước đó. Khung theo các phương án thể hiện tập các video đám mây điểm được tạo ra cùng lúc. Hiệu quả nén của việc mã hóa nội/mã hóa liên theo các phương án có thể phụ thuộc vào số lượng nút lân cận được tham chiếu. Khi các bit tăng

lên, hoạt động này trở nên phức tạp, nhưng việc mã hóa có thể được định thiên đến một phía, mà điều này có thể làm tăng hiệu quả nén. Ví dụ, khi ngữ cảnh 3 bit được cho trước, việc lập mã cần được thực hiện nhờ sử dụng $2^3 = 8$ phương pháp. Phần được phân chia cho việc lập mã gây ảnh hưởng đến độ phức tạp của cách thi hành. Theo đó, cần phải đạt được mức hiệu quả nén và độ phức tạp thích hợp.

Fig.7 minh họa quy trình thu mẫu dạng chiếm giữ dựa trên sự chiếm giữ các nút lân cận. Bộ mã hóa video đám mây điểm theo các phương án xác định sự chiếm giữ các nút lân cận của mỗi nút của cây bát phân và thu giá trị của mẫu dạng lân cận. Mẫu dạng nút lân cận được sử dụng để suy ra mẫu dạng chiếm giữ của nút. Phần trên trên Fig.7 thể hiện khối lập phương tương ứng với nút (khối lập phương được đặt ở giữa) và sáu khối lập phương (các nút lân cận) đang chia sẻ ít nhất một mặt với khối lập phương này. Các nút được thể hiện trên hình vẽ này là các nút có cùng độ sâu. Các con số được thể hiện trên hình vẽ này biểu diễn các trọng số (1, 2, 4, 8, 16, và 32) lần lượt được liên kết với sáu nút này. Các trọng số này được ấn định lần lượt theo các vị trí của các nút lân cận này.

Phần dưới trên Fig.7 thể hiện các giá trị mẫu dạng nút lân cận. Giá trị mẫu dạng nút lân cận là tổng các giá trị được nhân với trọng số của nút lân cận được chiếm giữ (nút lân cận chứa điểm). Theo đó, các giá trị mẫu dạng nút lân cận là 0 đến 63. Khi giá trị mẫu dạng nút lân cận là 0, điều này chỉ ra rằng không có nút nào có điểm (không có nút lân cận được chiếm giữ) trong số các nút lân cận của nút này. Khi giá trị mẫu dạng nút lân cận là 63, điều này chỉ ra rằng tất cả các nút lân cận đều là các nút được chiếm giữ. Như được thể hiện trên hình vẽ, vì các nút lân cận mà các trọng số 1, 2, 4 và 8 được ấn định cho chúng là các nút được chiếm giữ, giá trị mẫu dạng nút lân cận là 15, tổng của 1, 2, 4, và 8. Bộ mã hóa video đám mây điểm có thể thực hiện việc lập mã tùy theo giá trị mẫu dạng nút lân cận (ví dụ, khi giá trị mẫu dạng nút lân cận là 63, 64 loại lập mã có thể được thực hiện). Theo các phương án, bộ mã hóa video đám mây điểm có thể làm giảm độ phức tạp lập mã bằng cách thay đổi giá trị mẫu dạng nút lân cận (ví dụ, dựa trên bảng mà qua đó 64 được thay đổi thành 10 hoặc 6).

Fig.8 minh họa ví dụ về cấu hình điểm trong mỗi LOD theo các phương án.

Như được mô tả với tham chiếu đến Fig.1 đến Fig.7, dạng hình học được mã hóa được xây dựng lại (được giải nén) trước khi việc mã hóa thuộc tính được thực hiện. Khi

việc lập mã trực tiếp được áp dụng, hoạt động xây dựng lại dạng hình học có thể bao gồm việc thay đổi cách đặt vị trí của các điểm được lập mã trực tiếp (ví dụ, việc đặt vị trí các điểm được lập mã trực tiếp phía trước của dữ liệu đám mây điểm). Khi việc mã hóa dạng hình học trisoup được áp dụng, quy trình xây dựng lại dạng hình học được thực hiện qua việc xây dựng lại hình tam giác, việc lấy mẫu lên, và việc tạo đơn vị ảnh ba chiều. Vì thuộc tính phụ thuộc vào dạng hình học, việc mã hóa thuộc tính được thực hiện dựa trên dạng hình học được xây dựng lại.

Bộ mã hóa video đám mây điểm (ví dụ, bộ tạo ra LOD 40009) có thể phân loại (hoặc tổ chức lại) các điểm theo LOD. Fig.8 thể hiện nội dung đám mây điểm tương ứng với các LOD. Hình ảnh ngoài cùng bên trái trên Fig.8 thể hiện nội dung đám mây điểm gốc. Hình ảnh thứ hai từ bên trái trên Fig.8 thể hiện việc phân bố các điểm trong LOD thấp nhất, và hình ảnh ngoài cùng bên phải trên Fig.8 thể hiện việc phân bố các điểm trong LOD cao nhất. Tức là, các điểm trong LOD thấp nhất được phân bố thưa thớt, và các điểm trong LOD cao nhất được phân bố dày đặc. Tức là, khi LOD tăng lên theo hướng được chỉ định bởi mũi tên được chỉ ra ở đáy trên Fig.8, không gian (hoặc khoảng cách) giữa các điểm được thu hẹp.

Fig.9 minh họa ví dụ về cấu hình điểm dành cho mỗi LOD theo các phương án.

Như được mô tả với tham chiếu đến Fig.1 đến Fig.8, hệ thống cung cấp nội dung đám mây điểm, hoặc bộ mã hóa video đám mây điểm (ví dụ, bộ mã hóa video đám mây điểm 10002 trên Fig.1, bộ mã hóa video đám mây điểm trên Fig.4, hoặc bộ tạo ra LOD 40009) có thể tạo ra LOD. LOD được tạo ra bằng cách tổ chức lại các điểm thành tập các mức tinh chỉnh tùy theo giá trị khoảng cách LOD (hoặc tập các khoảng cách Euclide) được thiết lập. Quy trình tạo ra LOD được thực hiện không chỉ bởi bộ mã hóa video đám mây điểm, mà cả bởi bộ giải mã video đám mây điểm.

Phần trên trên Fig.9 thể hiện các ví dụ (P0 đến P9) về các điểm của nội dung đám mây điểm được phân bố trong không gian 3D. Trên Fig.9, thứ tự gốc thể hiện thứ tự của các điểm P0 đến P9 trước khi tạo ra LOD. Trên Fig.9, thứ tự dựa trên LOD thể hiện thứ tự của các điểm theo việc tạo ra LOD. Các điểm được tổ chức lại theo LOD. Ngoài ra, LOD cao chứa các điểm thuộc về các LOD thấp hơn. Như được thể hiện trên Fig.9, LOD0 chứa P0, P5, P4 và P2. LOD1 chứa các điểm của LOD0, P1, P6 và P3. LOD2 chứa các điểm của LOD0, các điểm của LOD1, P9, P8 và P7.

Như được mô tả với tham chiếu đến Fig.4, bộ mã hóa video đám mây điểm theo các phương án có thể thực hiện việc lập mã biến đổi dự đoán dựa trên LOD, việc lập mã biến đổi nâng dựa trên LOD, và việc lập mã biến đổi RAHT theo cách có chọn lọc hoặc kết hợp với nhau.

Bộ mã hóa video đám mây điểm theo các phương án có thể tạo ra bộ dự đoán cho các điểm để thực hiện việc lập mã biến đổi dự đoán dựa trên LOD để thiết lập thuộc tính được dự đoán (hoặc giá trị thuộc tính được dự đoán) của mỗi điểm. Tức là, N bộ dự đoán có thể được tạo ra cho N điểm. Bộ dự đoán theo các phương án có thể tính trọng số ($=1/\text{khoảng cách}$) dựa trên giá trị LOD của mỗi điểm, thông tin đánh chỉ số về các điểm lân cận có mặt bên trong khoảng cách được thiết lập đối với mỗi LOD, và khoảng cách đến các điểm lân cận.

Thuộc tính (hoặc giá trị thuộc tính) được dự đoán theo các phương án được thiết lập là trung bình của các giá trị được thu bằng cách nhân các thuộc tính (hoặc các giá trị thuộc tính) (ví dụ, màu, độ phản xạ, v.v.) của các điểm lân cận được thiết lập trong bộ dự đoán của mỗi điểm với trọng số (hoặc giá trị trọng số) được tính dựa trên khoảng cách đến mỗi điểm lân cận. Bộ mã hóa video đám mây điểm theo các phương án (ví dụ, bộ lượng tử hóa hệ số 40011) có thể lượng tử hóa và lượng tử hoá ngược phần dư của mỗi điểm (mà nó có thể được gọi là thuộc tính phần dư, giá trị thuộc tính phần dư, giá trị phần dư dự đoán thuộc tính hoặc giá trị thuộc tính sai số dự đoán v.v.) thu được bằng cách trừ thuộc tính (hoặc giá trị thuộc tính) được dự đoán của mỗi điểm khỏi thuộc tính (cụ thể là, giá trị thuộc tính gốc) của mỗi điểm. Quy trình lượng tử hóa được thực hiện cho giá trị thuộc tính phần dư trong thiết bị truyền được tạo cấu hình như được thể hiện trên bảng 2. Quy trình lượng tử hóa ngược được thực hiện cho giá trị thuộc tính phần dư trong thiết bị nhận được tạo cấu hình như được thể hiện trên bảng 3.

Bảng 2

int PCCQuantization(int value, int quantStep) {
if(value >=0) {
return floor(value / quantStep + 1.0 / 3.0);
} else {
return -floor(-value / quantStep + 1.0 / 3.0);
}
}

Bảng 3

int PCCInverseQuantization(int value, int quantStep) {
if(quantStep ==0) {
return value;
} else {
return value * quantStep;
}
}

Khi bộ dự đoán của mỗi điểm có các điểm lân cận, bộ mã hóa video đám mây điểm (ví dụ, bộ mã hóa số học 40012) theo các phương án có thể thực hiện việc lập mã entropy trên các giá trị thuộc tính phần dư được lượng tử hoá và được lượng tử hoá ngược như được mô tả trên đây.

1) Tạo ra mảng trọng số lượng tử hóa (Quantization Weight, QW) để lưu trữ giá trị trọng số của mỗi điểm. Giá trị ban đầu của tất cả các phần tử của QW là 1,0. Nhân các giá trị QW của các chỉ số bộ dự đoán của các nút lân cận được đăng ký trong bộ dự đoán với trọng số của bộ dự đoán của điểm hiện tại, và cộng các giá trị thu được bởi phép nhân này vào.

2) Quy trình dự đoán nâng: Trừ giá trị thu được bằng cách nhân giá trị thuộc tính của điểm với trọng số khỏi giá trị thuộc tính hiện có để tính giá trị thuộc tính được dự đoán.

3) Tạo ra các mảng tạm thời được gọi là updateweight và cập nhật và khởi tạo các mảng tạm thời này là không.

4) Cộng tích lũy các trọng số được tính bằng cách nhân các trọng số được tính đối với tất cả các bộ dự đoán với trọng số được lưu trữ trong QW tương ứng với chỉ số bộ dự đoán vào mảng updateweight để làm các chỉ số của các nút lân cận. Cộng tích lũy, vào mảng cập nhật này, giá trị thu được bằng cách nhân giá trị thuộc tính của chỉ số của nút lân cận với trọng số được tính.

5) Quy trình cập nhật nâng: Chia các giá trị thuộc tính của mảng cập nhật đối với tất cả các bộ dự đoán cho giá trị trọng số của mảng updateweight của chỉ số bộ dự đoán, và cộng giá trị thuộc tính hiện có với các giá trị thu được bởi phép chia này.

6) Tính các thuộc tính được dự đoán bằng cách nhân các giá trị thuộc tính được cập nhật qua quy trình cập nhật nâng với trọng số được cập nhật qua quy trình dự đoán nâng (được lưu trữ trong QW) đối với tất cả các bộ dự đoán. Bộ mã hóa video đám mây điểm (ví dụ, bộ lượng tử hóa hệ số 40011) theo các phương án lượng tử hoá các giá trị

thuộc tính được dự đoán. Thêm vào đó, bộ mã hóa video đám mây điểm (ví dụ, bộ mã hóa số học 40012) thực hiện việc lập mã entropy trên các giá trị thuộc tính được lượng tử hóa.

Bộ mã hóa video đám mây điểm (ví dụ, bộ biến đổi RAHT 40008) theo các phương án có thể thực hiện việc lập mã biến đổi RAHT trong đó các thuộc tính của các nút có mức cao hơn được dự đoán sử dụng các thuộc tính liên quan đến các nút có mức thấp hơn trong cây bát phân. Việc lập mã biến đổi RAHT là ví dụ về việc lập mã nội thuộc tính qua việc quét ngược cây bát phân. Bộ mã hóa video đám mây điểm theo các phương án quét toàn bộ vùng từ đơn vị ảnh ba chiều và lặp lại quy trình hợp nhất gồm việc hợp nhất các đơn vị ảnh ba chiều thành khối lớn hơn tại mỗi bước cho đến khi nút gốc rễ được chạm đến. Quy trình hợp nhất theo các phương án được thực hiện chỉ trên các nút được chiếm giữ. Quy trình hợp nhất không được thực hiện trên nút rỗng. Quy trình hợp nhất được thực hiện trên nút cao hơn ở ngay trên nút rỗng.

Phương trình 3 dưới đây thể hiện ma trận biến đổi RAHT. Trong phương trình 3, $g_{l\ x,y,z}$ biểu thị giá trị thuộc tính trung bình của các đơn vị ảnh ba chiều tại mức l . $g_{l\ x,y,z}$ có thể được tính dựa trên $g_{l+1\ 2x,y,z}$ và $g_{l+1\ 2x+1,y,z}$. Các trọng số cho $g_{l\ 2x,y,z}$ và $g_{l\ 2x+1,y,z}$ là $w1 = w_{l\ 2x,y,z}$ và $w2 = w_{l\ 2x+1,y,z}$.

Phương trình 3

$$\begin{bmatrix} g_{l-1\ x,y,z} \\ h_{l-1\ x,y,z} \end{bmatrix} = T_{w1\ w2} \begin{bmatrix} g_{l\ 2x,y,z} \\ g_{l\ 2x+1,y,z} \end{bmatrix} \quad T_{w1\ w2} = \frac{1}{\sqrt{w1+w2}} \begin{bmatrix} \sqrt{w1} & \sqrt{w2} \\ -\sqrt{w2} & \sqrt{w1} \end{bmatrix}$$

Ở đây, $g_{l-1\ x,y,z}$ là giá trị thông thấp và được sử dụng trong quy trình hợp nhất tại mức cao hơn tiếp theo. $h_{l-1\ x,y,z}$ biểu thị các hệ số thông cao. Các hệ số thông cao tại mỗi bước được lượng tử hoá và là đối tượng của việc lập mã entropy (ví dụ, việc mã hóa bởi bộ mã hóa số học 40012). Các trọng số được tính là $w_{l-1\ x,y,z} = w_{l\ 2x,y,z} + w_{l\ 2x+1,y,z}$. Nút gốc rễ được tạo ra qua $g_{1\ 0,0,0}$ và $g_{1\ 0,0,1}$ như phương trình 4.

Phương trình 4

$$\begin{bmatrix} g^{DC} \\ h_{0,0,0} \end{bmatrix} = T_{w1000\ w1001} \begin{bmatrix} g_{1\ 0,0,0} \\ g_{1\ 0,0,1} \end{bmatrix}$$

Giá trị của g^{DC} cũng được lượng tử hoá và là đối tượng của việc lập mã entropy giống như các hệ số thông cao.

Fig.10 minh họa bộ giải mã video đám mây điểm theo các phương án.

Bộ giải mã video đám mây điểm được minh họa trên Fig.10 là ví dụ về bộ giải mã video đám mây điểm 10006 được mô tả trên Fig.1, và có thể thực hiện các hoạt động giống hoặc tương tự như các hoạt động của bộ giải mã video đám mây điểm 10006 được minh họa trên Fig.1. Như được thể hiện trên hình vẽ, bộ giải mã video đám mây điểm có thể nhận luồng bit dạng hình học và luồng bit thuộc tính được chứa trong một hoặc nhiều luồng bit. Bộ giải mã video đám mây điểm bao gồm bộ giải mã dạng hình học và bộ giải mã thuộc tính. Bộ giải mã dạng hình học thực hiện việc giải mã dạng hình học trên luồng bit dạng hình học và xuất ra dạng hình học được giải mã. Bộ giải mã thuộc tính thực hiện việc giải mã thuộc tính trên luồng bit thuộc tính dựa trên dạng hình học được giải mã, và xuất ra các thuộc tính được giải mã. Dạng hình học được giải mã và các thuộc tính được giải mã được sử dụng để xây dựng lại nội dung đám mây điểm (đám mây điểm được giải mã).

Fig.11 minh họa bộ giải mã video đám mây điểm theo các phương án.

Bộ giải mã video đám mây điểm được minh họa trên Fig.11 là ví dụ về bộ giải mã video đám mây điểm được minh họa trên Fig.10, và có thể thực hiện hoạt động giải mã, mà nó là quy trình ngược của hoạt động mã hóa của bộ mã hóa video đám mây điểm được minh họa trên Fig.1 đến Fig.9.

Như được mô tả với tham chiếu đến Fig.1 và Fig.10, bộ giải mã video đám mây điểm có thể thực hiện việc giải mã dạng hình học và việc giải mã thuộc tính. Việc giải mã dạng hình học được thực hiện trước việc giải mã thuộc tính.

Bộ giải mã video đám mây điểm theo các phương án bao gồm bộ giải mã số học (giải mã số học) 11000, bộ tổng hợp cây bát phân (tổng hợp cây bát phân) 11001, bộ tổng hợp xấp xỉ bề mặt (tổng hợp xấp xỉ bề mặt) 11002, và bộ xây dựng lại dạng hình học (xây dựng lại dạng hình học) 11003, bộ biến đổi ngược tọa độ (biến đổi ngược các tọa độ) 11004, bộ giải mã số học (giải mã số học) 11005, bộ lượng tử hóa ngược (lượng tử hóa ngược) 11006, bộ biến đổi RAHT 11007, bộ tạo ra LOD (tạo ra LOD) 11008, bộ nâng ngược (nâng ngược) 11009, và/hoặc bộ biến đổi ngược màu (biến đổi ngược màu) 11010.

Bộ giải mã số học 11000, bộ tổng hợp cây bát phân 11001, bộ tổng hợp xấp xỉ bề mặt 11002, và bộ xây dựng lại dạng hình học 11003, và bộ biến đổi ngược tọa độ 11004

có thể thực hiện việc giải mã dạng hình học. Việc giải mã dạng hình học theo các phương án có thể bao gồm việc giải mã trực tiếp và việc giải mã dạng hình học trisoup. Việc giải mã trực tiếp và việc giải mã dạng hình học trisoup được áp dụng có chọn lọc. Việc giải mã dạng hình học không bị giới hạn vào ví dụ được mô tả trên đây, và được thực hiện như là quy trình ngược của việc mã hóa dạng hình học được mô tả với tham chiếu đến Fig.1 đến Fig.9.

Bộ giải mã số học 11000 theo các phương án giải mã luồng bit dạng hình học nhận được dựa trên việc lập mã số học. Hoạt động của bộ giải mã số học 11000 tương ứng với quy trình ngược của bộ mã hóa số học 40004.

Bộ tổng hợp cây bát phân 11001 theo các phương án có thể tạo ra cây bát phân bằng cách thu nhận mã chiếm giữ từ luồng bit dạng hình học được giải mã (hoặc thông tin về dạng hình học đạt được như kết quả của việc giải mã). Mã chiếm giữ được tạo cấu hình như được mô tả chi tiết với tham chiếu đến Fig.1 đến Fig.9.

Khi việc mã hóa dạng hình học trisoup được áp dụng, bộ tổng hợp xấp xỉ bề mặt 11002 theo các phương án có thể tổng hợp bề mặt dựa trên dạng hình học được giải mã và/hoặc cây bát phân được tạo ra.

Bộ xây dựng lại dạng hình học 11003 theo các phương án có thể tái tạo dạng hình học dựa trên bề mặt này và/hoặc dạng hình học được giải mã này. Như được mô tả với tham chiếu đến Fig.1 đến Fig.9, việc lập mã trực tiếp và việc mã hóa dạng hình học trisoup được áp dụng có chọn lọc. Theo đó, bộ xây dựng lại dạng hình học 11003 trực tiếp nhập và thêm thông tin vị trí về các điểm mà việc lập mã trực tiếp được áp dụng với chúng. Khi việc mã hóa dạng hình học trisoup được áp dụng, bộ xây dựng lại dạng hình học 11003 có thể xây dựng lại dạng hình học bằng cách thực hiện các hoạt động xây dựng lại của bộ xây dựng lại dạng hình học 40005, ví dụ, việc xây dựng lại hình tam giác, việc lấy mẫu lên, và việc tạo đơn vị ảnh ba chiều. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.6, và do đó phần mô tả của chúng được lược bỏ. Dạng hình học được xây dựng lại có thể bao gồm hình ảnh hoặc khung đám mây điểm mà chúng không chứa các thuộc tính.

Bộ biến đổi ngược tọa độ 11004 theo các phương án có thể thu nhận các vị trí của các điểm bằng cách biến đổi các tọa độ dựa trên dạng hình học được xây dựng lại.

Bộ giải mã số học 11005, bộ lượng tử hóa ngược 11006, bộ biến đổi RAHT 11007, bộ tạo ra LOD 11008, bộ nâng ngược 11009, và/hoặc bộ biến đổi ngược màu 11010 có thể thực hiện việc giải mã thuộc tính được mô tả với tham chiếu đến Fig.10. Việc giải mã thuộc tính theo các phương án bao gồm việc giải mã biến đổi phân cấp thích ứng vùng (RAHT), việc giải mã dự đoán lân cận gần nhất phân cấp dựa trên việc nội suy (biến đổi dự đoán), và việc dự đoán lân cận gần nhất phân cấp dựa trên việc nội suy cùng với việc giải mã bước cập nhật/nâng (biến đổi nâng). Ba sơ đồ giải mã được mô tả trên đây có thể được sử dụng có chọn lọc, hoặc sự kết hợp của một hoặc nhiều sơ đồ giải mã có thể được sử dụng. Việc giải mã thuộc tính theo các phương án không bị giới hạn vào ví dụ được mô tả trên đây.

Bộ giải mã số học 11005 theo các phương án giải mã luồng bit thuộc tính bằng việc lập mã số học.

Bộ lượng tử hóa ngược 11006 theo các phương án lượng tử hoá ngược thông tin về luồng bit thuộc tính được giải mã hoặc các thuộc tính đạt được như kết quả của việc giải mã, và xuất ra các thuộc tính (hoặc các giá trị thuộc tính) được lượng tử hoá ngược. Việc lượng tử hóa ngược có thể được áp dụng có chọn lọc dựa trên việc mã hóa thuộc tính của bộ mã hóa video đám mây điểm.

Theo các phương án, bộ biến đổi RAHT 11007, bộ tạo ra LOD 11008, và/hoặc bộ nâng ngược 11009 có thể xử lý dạng hình học được xây dựng lại và các thuộc tính được lượng tử hoá ngược. Như được mô tả trên đây, bộ biến đổi RAHT 11007, bộ tạo ra LOD 11008, và/hoặc bộ nâng ngược 11009 có thể thực hiện có chọn lọc hoạt động giải mã tương ứng với việc mã hóa của bộ mã hóa video đám mây điểm.

Bộ biến đổi ngược màu 11010 theo các phương án thực hiện việc lập mã biến đổi ngược để biến đổi ngược giá trị màu (hoặc kết cấu) được chứa trong các thuộc tính được giải mã. Hoạt động của bộ biến đổi ngược màu 11010 có thể được thực hiện có chọn lọc dựa trên hoạt động của bộ biến đổi màu 40006 của bộ mã hóa video đám mây điểm.

Mặc dù không được thể hiện trên hình vẽ, các phần tử của bộ giải mã video đám mây điểm trên Fig.11 có thể được thi hành bởi phần cứng bao gồm một hoặc nhiều bộ xử lý hoặc mạch tích hợp được tạo cấu hình để truyền thông với một hoặc nhiều bộ nhớ được chứa trong thiết bị cung cấp nội dung đám mây điểm, phần mềm, phần sụn hoặc sự kết hợp của chúng. Một hoặc nhiều bộ xử lý này có thể thực hiện ít nhất một hoặc

những hoạt động và/hoặc chức năng trong số các hoạt động và/hoặc chức năng của các phần tử của bộ giải mã video đám mây điểm trên Fig.11 được mô tả trên đây. Thêm vào đó, một hoặc nhiều bộ xử lý này có thể hoạt động hoặc thực thi tập các chương trình và/hoặc lệnh phần mềm để thực hiện các hoạt động và/hoặc chức năng của các phần tử của bộ giải mã video đám mây điểm trên Fig.11.

Fig.12 minh họa thiết bị truyền theo các phương án. Thiết bị truyền được thể hiện trên Fig.12 là ví dụ về thiết bị truyền 10000 trên Fig.1 (hoặc bộ mã hóa video đám mây điểm trên Fig.4).

Thiết bị truyền được minh họa trên Fig.12 có thể thực hiện một hoặc nhiều hoạt động và phương pháp trong số các hoạt động và phương pháp giống với hoặc tương tự như các hoạt động và phương pháp của bộ mã hóa video đám mây điểm được mô tả với tham chiếu đến Fig.1 đến Fig.9. Thiết bị truyền theo các phương án có thể bao gồm đơn vị đầu vào dữ liệu 12000, bộ xử lý lượng tử hóa 12001, bộ xử lý tạo đơn vị ảnh ba chiều 12002, bộ tạo ra mã chiếm giữ cây bát phân 12003, bộ xử lý mô hình bề mặt 12004, bộ xử lý lập mã nội/liên 12005, bộ lập mã số học 12006, bộ xử lý lý lịch dữ liệu 12007, bộ xử lý biến đổi màu 12008, bộ xử lý biến đổi thuộc tính 12009, bộ xử lý biến đổi dự đoán/nâng/RAHT 12010, bộ lập mã số học 12011 và/hoặc bộ xử lý truyền 12012.

Đơn vị đầu vào dữ liệu 12000 theo các phương án nhận hoặc thu nhận dữ liệu đám mây điểm. Đơn vị đầu vào dữ liệu 12000 có thể thực hiện phương pháp hoạt động và/hoặc thu nhận giống với hoặc tương tự với phương pháp hoạt động và/hoặc thu nhận của đơn vị thu nhận video đám mây điểm 10001 (hoặc quy trình thu nhận 20000 được mô tả với tham chiếu đến Fig.2).

Đơn vị đầu vào dữ liệu 12000, bộ xử lý lượng tử hóa 12001, bộ xử lý tạo đơn vị ảnh ba chiều 12002, bộ tạo ra mã chiếm giữ cây bát phân 12003, bộ xử lý mô hình bề mặt 12004, bộ xử lý lập mã nội/liên 12005, và bộ lập mã số học 12006 thực hiện việc mã hóa dạng hình học. Việc mã hóa dạng hình học theo các phương án là giống với hoặc tương tự với việc mã hóa dạng hình học được mô tả với tham chiếu đến Fig.1 đến Fig.9, và do đó phần mô tả chi tiết của nó được lược bỏ.

Bộ xử lý lượng tử hóa 12001 theo các phương án lượng tử hoá dạng hình học (ví dụ, các giá trị vị trí của các điểm). Hoạt động và/hoặc việc lượng tử hóa của bộ xử lý lượng tử hóa 12001 là giống với hoặc tương tự với hoạt động và/hoặc việc lượng tử hóa

của bộ lượng tử hóa 40001 được mô tả với tham chiếu đến Fig.4. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.1 đến Fig.9.

Bộ xử lý tạo đơn vị ảnh ba chiều 12002 theo các phương án tạo đơn vị ảnh ba chiều đối với các giá trị vị trí được lượng tử hóa của các điểm. Bộ xử lý tạo đơn vị ảnh ba chiều 12002 có thể thực hiện hoạt động và/hoặc quy trình giống với hoặc tương tự với hoạt động và/hoặc quy trình tạo đơn vị ảnh ba chiều của bộ lượng tử hóa 40001 được mô tả với tham chiếu đến Fig.4. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.1 đến Fig.9.

Bộ tạo ra mã chiếm giữ cây bát phân 12003 theo các phương án thực hiện việc lập mã cây bát phân trên các vị trí được tạo đơn vị ảnh ba chiều của các điểm dựa trên cấu trúc cây bát phân. Bộ tạo ra mã chiếm giữ cây bát phân 12003 có thể tạo ra mã chiếm giữ. Bộ tạo ra mã chiếm giữ cây bát phân 12003 có thể thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ mã hóa video đám mây điểm (hoặc bộ phân tích cây bát phân 40002) được mô tả với tham chiếu đến Fig.4 và Fig.6. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.1 đến Fig.9.

Bộ xử lý mô hình bề mặt 12004 theo các phương án có thể thực hiện việc mã hóa dạng hình học trigsoup dựa trên mô hình bề mặt để xây dựng lại các vị trí của các điểm trong vùng (hoặc nút) cụ thể trên cơ sở đơn vị ảnh ba chiều. Bộ xử lý mô hình bề mặt 12004 có thể thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ mã hóa video đám mây điểm (ví dụ, bộ phân tích xấp xỉ bề mặt 40003) được mô tả với tham chiếu đến Fig.4. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.1 đến Fig.9.

Bộ xử lý lập mã nội/liên 12005 theo các phương án có thể thực hiện việc lập mã nội/liên trên dữ liệu đám mây điểm. Bộ xử lý lập mã nội/liên 12005 có thể thực hiện việc lập mã giống với hoặc tương tự với việc lập mã nội/liên được mô tả với tham chiếu đến Fig.7. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.7. Theo các phương án, bộ xử lý lập mã nội/liên 12005 có thể được bao gồm trong bộ lập mã số học 12006.

Bộ lập mã số học 12006 theo các phương án thực hiện việc mã hóa entropy trên cây bát phân của dữ liệu đám mây điểm và/hoặc cây bát phân được tính xấp xỉ. Ví dụ,

sơ đồ mã hóa này bao gồm việc mã hóa số học. Bộ lập mã số học 12006 thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ mã hóa số học 40004.

Bộ xử lý lý lịch dữ liệu 12007 theo các phương án xử lý lý lịch dữ liệu về dữ liệu đám mây điểm, ví dụ, giá trị được thiết lập, và cung cấp nó đến quy trình xử lý cần thiết chẳng hạn như việc mã hóa dạng hình học và/hoặc việc mã hóa thuộc tính. Ngoài ra, bộ xử lý lý lịch dữ liệu 12007 theo các phương án có thể tạo ra và/hoặc xử lý thông tin báo hiệu liên quan đến việc mã hóa dạng hình học và/hoặc việc mã hóa thuộc tính. Thông tin báo hiệu theo các phương án có thể được mã hóa riêng biệt khỏi việc mã hóa dạng hình học và/hoặc việc mã hóa thuộc tính. Thông tin báo hiệu theo các phương án có thể được đan xen.

Bộ xử lý biến đổi màu 12008, bộ xử lý biến đổi thuộc tính 12009, bộ xử lý biến đổi dự đoán/nâng/RAHT 12010, và bộ lập mã số học 12011 thực hiện việc mã hóa thuộc tính. Việc mã hóa thuộc tính theo các phương án là giống với hoặc tương tự với việc mã hóa thuộc tính được mô tả với tham chiếu đến Fig.1 đến Fig.9, và do đó phần mô tả chi tiết của nó được lược bỏ.

Bộ xử lý biến đổi màu 12008 theo các phương án thực hiện việc lập mã biến đổi màu để biến đổi các giá trị màu được chứa trong các thuộc tính. Bộ xử lý biến đổi màu 12008 có thể thực hiện việc lập mã biến đổi màu dựa trên dạng hình học được xây dựng lại. Dạng hình học được xây dựng lại là giống như được mô tả với tham chiếu đến Fig.1 đến Fig.9. Ngoài ra, nó thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ biến đổi màu 40006 được mô tả với tham chiếu đến Fig.4. Phần mô tả chi tiết của nó được lược bỏ.

Bộ xử lý biến đổi thuộc tính 12009 theo các phương án thực hiện việc biến đổi thuộc tính để biến đổi các thuộc tính dựa trên dạng hình học được xây dựng lại và/hoặc các vị trí mà việc mã hóa dạng hình học không được thực hiện trên đó. Bộ xử lý biến đổi thuộc tính 12009 thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ biến đổi thuộc tính 40007 được mô tả với tham chiếu đến Fig.4. Phần mô tả chi tiết của nó được lược bỏ. Bộ xử lý biến đổi dự đoán/nâng/RAHT 12010 theo các phương án có thể lập mã các thuộc tính được biến đổi theo một thành phần bất kỳ trong số hoặc theo sự kết hợp của việc lập mã RAHT, việc

lập mã biến đổi dự đoán, và việc lập mã biến đổi nâng. Bộ xử lý biến đổi dự đoán/nâng/RAHT 12010 thực hiện ít nhất một hoạt động trong số các hoạt động giống với hoặc tương tự với các hoạt động của bộ biến đổi RAHT 40008, bộ tạo ra LOD 40009, và bộ biến đổi nâng 40010 được mô tả với tham chiếu đến Fig.4. Thêm vào đó, việc lập mã biến đổi dự đoán, việc lập mã biến đổi nâng, và việc lập mã biến đổi RAHT là giống với việc lập mã biến đổi dự đoán, việc lập mã biến đổi nâng, và việc lập mã biến đổi RAHT được mô tả với tham chiếu đến Fig.1 đến Fig.9, và do đó phần mô tả chi tiết của chúng được lược bỏ.

Bộ lập mã số học 12011 theo các phương án có thể mã hóa các thuộc tính được lập mã dựa trên việc lập mã số học. Bộ lập mã số học 12011 thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp của bộ mã hóa số học 40012.

Bộ xử lý truyền 12012 theo các phương án có thể truyền mỗi luồng bit chứa dạng hình học được mã hóa và/hoặc các thuộc tính được mã hóa và lý lịch dữ liệu, hoặc truyền một luồng bit được tạo cấu hình bằng dạng hình học được mã hóa và/hoặc các thuộc tính được mã hóa và lý lịch dữ liệu này. Khi dạng hình học được mã hóa và/hoặc các thuộc tính được mã hóa và lý lịch dữ liệu theo các phương án được tạo cấu hình thành một luồng bit, luồng bit này có thể bao gồm một hoặc nhiều luồng bit con. Luồng bit theo các phương án có thể chứa thông tin báo hiệu bao gồm tập tham số trình tự (sequence parameter set, SPS) để báo hiệu mức trình tự, tập tham số dạng hình học (geometry parameter set, GPS) để báo hiệu việc lập mã thông tin dạng hình học, tập tham số thuộc tính (attribute parameter set, APS) để báo hiệu việc lập mã thông tin thuộc tính, và tập tham số ô xếp (tile parameter set, TPS hoặc bản kiểm kê ô xếp) để báo hiệu mức ô xếp, và dữ liệu lát. Dữ liệu lát có thể bao gồm thông tin về một hoặc nhiều lát. Một lát theo các phương án có thể bao gồm một luồng bit dạng hình học Geom0⁰ và một hoặc nhiều luồng bit thuộc tính Attr0⁰ và Attr1⁰. TPS (hoặc bản kiểm kê ô xếp) theo các phương án có thể bao gồm thông tin về mỗi ô xếp (ví dụ, thông tin tọa độ và thông tin chiều cao/kích thước về hộp giới hạn) cho một hoặc nhiều ô xếp. Luồng bit dạng hình học có thể chứa tiêu đề và phần tải hữu ích. Tiêu đề của luồng bit dạng hình học theo các phương án có thể chứa bộ nhận dạng tập tham số (geom_parameter_set_id), bộ nhận dạng ô xếp (geom_tile_id) và bộ nhận dạng lát (geom_slice_id) được chứa trong GPS,

và thông tin về dữ liệu được chứa trong phần tải hữu ích. Như được mô tả trên đây, bộ xử lý lý lịch dữ liệu 12007 theo các phương án có thể tạo ra và/hoặc xử lý thông tin báo hiệu và truyền nó đến bộ xử lý truyền 12012. Theo các phương án, các phần tử để thực hiện việc mã hóa dạng hình học và các phần tử để thực hiện việc mã hóa thuộc tính có thể chia sẻ dữ liệu/thông tin với nhau như được chỉ ra bởi các đường chấm. Bộ xử lý truyền 12012 theo các phương án có thể thực hiện hoạt động và/hoặc phương pháp truyền giống với hoặc tương tự với hoạt động và/hoặc phương pháp truyền của bộ truyền 10003. Các chi tiết là giống với các chi tiết được mô tả với tham chiếu đến Fig.1 và Fig.2, và do đó phần mô tả của chúng được lược bỏ.

Fig.13 minh họa thiết bị nhận theo các phương án.

Thiết bị nhận được minh họa trên Fig.13 là ví dụ về thiết bị nhận 10004 trên Fig.1 (hoặc bộ giải mã video đám mây điểm trên Fig.10 và Fig.11). Thiết bị nhận được minh họa trên Fig.13 có thể thực hiện một hoặc nhiều hoạt động và phương pháp trong số các hoạt động và phương pháp giống với hoặc tương tự với các hoạt động và phương pháp của bộ giải mã video đám mây điểm được mô tả với tham chiếu đến Fig.1 đến Fig.11.

Thiết bị nhận theo phương án này có thể bao gồm bộ nhận 13000, bộ xử lý nhận 13001, bộ giải mã số học 13002, bộ xử lý xây dựng lại cây bát phân dựa trên mã chiếm giữ 13003, bộ xử lý mô hình bề mặt (việc xây dựng lại hình tam giác, việc lấy mẫu tăng, việc tạo đơn vị ảnh ba chiều) 13004, và bộ xử lý lượng tử hóa ngược 13005, bộ phân tích cú pháp lý lịch dữ liệu 13006, bộ giải mã số học 13007, bộ xử lý lượng tử hóa ngược 13008, bộ xử lý biến đổi ngược dự đoán/nâng/RAHT 13009, bộ xử lý biến đổi ngược màu 13010, và/hoặc bộ kết xuất 13011. Mỗi phần tử cho việc giải mã theo các phương án có thể thực hiện quy trình ngược của hoạt động của phần tử tương ứng cho việc mã hóa theo các phương án.

Bộ nhận 13000 theo các phương án nhận dữ liệu đám mây điểm. Bộ nhận 13000 có thể thực hiện hoạt động và/hoặc phương pháp nhận giống với hoặc tương tự với hoạt động và/hoặc phương pháp nhận của bộ nhận 10005 trên Fig.1. Phần mô tả chi tiết của nó được lược bỏ.

Bộ xử lý nhận 13001 theo các phương án có thể thu nhận luồng bit dạng hình học và/hoặc luồng bit thuộc tính từ dữ liệu nhận được. Bộ xử lý nhận 13001 có thể được bao gồm trong bộ nhận 13000.

Bộ giải mã số học 13002, bộ xử lý xây dựng lại cây bát phân dựa trên mã chiếm giữ 13003, bộ xử lý mô hình bề mặt 13004, và bộ xử lý lượng tử hóa ngược 1305 có thể thực hiện việc giải mã dạng hình học. Việc giải mã dạng hình học theo các phương án là giống với hoặc tương tự với việc giải mã dạng hình học được mô tả với tham chiếu đến Fig.1 đến Fig.10, và do đó phần mô tả chi tiết của nó được lược bỏ.

Bộ giải mã số học 13002 theo các phương án có thể giải mã luồng bit dạng hình học dựa trên việc lập mã số học. Bộ giải mã số học 13002 thực hiện hoạt động và/hoặc việc lập mã giống với hoặc tương tự với hoạt động và/hoặc việc lập mã của bộ giải mã số học 11000.

Bộ xử lý xây dựng lại cây bát phân dựa trên mã chiếm giữ 13003 theo các phương án có thể xây dựng lại cây bát phân bằng cách thu nhận mã chiếm giữ từ luồng bit dạng hình học được giải mã (hoặc thông tin về dạng hình học đạt được như kết quả của việc giải mã). Bộ xử lý xây dựng lại cây bát phân dựa trên mã chiếm giữ 13003 thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp tạo ra cây bát phân của bộ tổng hợp cây bát phân 11001. Khi việc mã hóa dạng hình học trisoup được áp dụng, bộ xử lý mô hình bề mặt 13004 theo các phương án có thể thực hiện việc giải mã dạng hình học trisoup và việc xây dựng lại dạng hình học liên quan (ví dụ, việc xây dựng lại hình tam giác, việc lấy mẫu lên, việc tạo đơn vị ảnh ba chiều) dựa trên phương pháp mô hình bề mặt. Bộ xử lý mô hình bề mặt 13004 thực hiện hoạt động giống với hoặc tương tự với hoạt động của bộ tổng hợp xấp xỉ bề mặt 11002 và/hoặc bộ xây dựng lại dạng hình học 11003.

Bộ xử lý lượng tử hóa ngược 13005 theo các phương án có thể lượng tử hoá ngược dạng hình học được giải mã.

Bộ phân tích cú pháp lý lịch dữ liệu 13006 theo các phương án có thể phân tích cú pháp lý lịch dữ liệu được chứa trong dữ liệu đám mây điểm nhận được, ví dụ như, giá trị được thiết lập. Bộ phân tích cú pháp lý lịch dữ liệu 13006 có thể chuyển lý lịch dữ liệu đến việc giải mã dạng hình học và/hoặc giải mã thuộc tính. Lý lịch dữ liệu này là giống với lý lịch dữ liệu được mô tả với tham chiếu đến Fig.12, và do đó phần mô tả chi tiết của nó được lược bỏ.

Bộ giải mã số học 13007, bộ xử lý lượng tử hóa ngược 13008, bộ xử lý biến đổi ngược dự đoán/nâng/RAHT 13009 và bộ xử lý biến đổi ngược màu 13010 thực hiện

việc giải mã thuộc tính. Việc giải mã thuộc tính là giống với hoặc tương tự với việc giải mã thuộc tính được mô tả với tham chiếu đến Fig.1 đến Fig.10, và do đó phần mô tả chi tiết của nó được lược bỏ.

Bộ giải mã số học 13007 theo các phương án có thể giải mã luồng bit thuộc tính bằng việc lập mã số học. Bộ giải mã số học 13007 có thể giải mã luồng bit thuộc tính dựa trên dạng hình học được xây dựng lại. Bộ giải mã số học 13007 thực hiện hoạt động và/hoặc việc lập mã giống với hoặc tương tự với hoạt động và/hoặc việc lập mã của bộ giải mã số học 11005.

Bộ xử lý lượng tử hóa ngược 13008 theo các phương án có thể lượng tử hoá ngược luồng bit thuộc tính được giải mã. Bộ xử lý lượng tử hóa ngược 13008 thực hiện hoạt động và/hoặc phương pháp giống với hoặc tương tự với hoạt động và/hoặc phương pháp lượng tử hóa ngược của bộ lượng tử hóa ngược 11006.

Bộ xử lý biến đổi ngược dự đoán/nâng/RAHT 13009 theo các phương án có thể xử lý dạng hình học được xây dựng lại và các thuộc tính được lượng tử hoá ngược. Bộ xử lý biến đổi ngược dự đoán/nâng/RAHT 13009 thực hiện một hoặc nhiều hoạt động và/hoặc việc giải mã mà chúng là giống với hoặc tương tự với các hoạt động và/hoặc việc giải mã của bộ biến đổi RAHT 11007, bộ tạo ra LOD 11008, và/hoặc bộ nâng ngược 11009. Bộ xử lý biến đổi ngược màu 13010 theo các phương án thực hiện việc lập mã biến đổi ngược để biến đổi ngược các giá trị màu (hoặc các kết cấu) được chứa trong các thuộc tính được giải mã. Bộ xử lý biến đổi ngược màu 13010 thực hiện hoạt động và/hoặc việc lập mã biến đổi ngược giống với hoặc tương tự với hoạt động và/hoặc việc lập mã biến đổi ngược của bộ biến đổi ngược màu 11010. Bộ kết xuất 13011 theo các phương án có thể kết xuất dữ liệu đám mây điểm.

Fig.14 thể hiện cấu trúc làm ví dụ kết nối được theo cách hoạt động được với phương pháp/thiết bị truyền và nhận dữ liệu đám mây điểm theo các phương án.

Cấu trúc trên Fig.14 thể hiện cấu hình trong đó ít nhất một thành phần trong số máy chủ 17600, rôbôt 17100, xe tự lái 17200, thiết bị XR 17300, điện thoại thông minh 17400, đồ gia dụng 17500, và/hoặc bộ hiển thị gắn trên đầu (Head-Mount Display, HMD) 17700 được kết nối với mạng đám mây 17000. Rôbôt 17100, xe tự lái 17200, thiết bị XR 17300, điện thoại thông minh 17400, hoặc đồ gia dụng 17500 được gọi là thiết bị. Thêm vào đó, thiết bị XR 17300 có thể tương ứng với thiết bị dữ liệu được nén đám

mây điểm (point cloud compressed, PCC) theo các phương án hoặc có thể được kết nối theo cách hoạt động được với thiết bị PCC này.

Mạng đám mây 17000 có thể biểu diễn mạng mà nó cấu thành nên một phần của cơ sở hạ tầng điện toán đám mây hoặc có mặt trong cơ sở hạ tầng điện toán đám mây. Ở đây, mạng đám mây 17000 có thể được tạo cấu hình nhờ sử dụng mạng 3G, mạng 4G hoặc phát triển dài hạn (LTE), hoặc mạng 5G.

Máy chủ 17600 có thể được kết nối với ít nhất một thành phần trong số rôbot 17100, xe tự lái 17200, thiết bị XR 17300, điện thoại thông minh 17400, đồ gia dụng 17500, và/hoặc HMD 17700 qua mạng đám mây 17000 và có thể hỗ trợ trong ít nhất một phần của việc xử lý của các thiết bị được kết nối 17100 đến 17700.

HMD 17700 biểu diễn một loại cách thi hành trong số các loại cách thi hành của thiết bị XR và/hoặc thiết bị PCC theo các phương án. Thiết bị loại HMD theo các phương án bao gồm đơn vị truyền thông, đơn vị điều khiển, bộ nhớ, đơn vị I/O, đơn vị cảm biến, và đơn vị cấp công suất.

Ở dưới đây, các phương án khác nhau của các thiết bị 17100 đến 17500 mà công nghệ được mô tả trên đây được áp dụng với chúng sẽ được mô tả. Các thiết bị 17100 đến 17500 được minh họa trên Fig.14 có thể được kết nối/được ghép nối theo cách hoạt động được với thiết bị truyền và nhận dữ liệu đám mây điểm theo các phương án được mô tả trên đây.

<PCC+XR>

Thiết bị XR/PCC 17300 có thể sử dụng công nghệ PCC và/hoặc công nghệ XR (AR+VR), và có thể được thi hành dưới dạng HMD, bộ hiển thị ở trên đầu (Head-Up Display, HUD) được cung cấp trên xe, tivi, điện thoại di động, điện thoại thông minh, máy tính, thiết bị đeo được, đồ gia dụng, bảng ký hiệu kỹ thuật số, xe, rôbot đứng yên, hoặc rôbot di động.

Thiết bị XR/PCC 17300 có thể phân tích dữ liệu đám mây điểm 3D hoặc dữ liệu ảnh thu được thông qua các bộ cảm biến khác nhau hoặc từ thiết bị bên ngoài và tạo ra dữ liệu vị trí và dữ liệu thuộc tính về các điểm 3D. Nhờ đó, thiết bị XR/PCC 17300 có thể thu nhận thông tin về không gian xung quanh hoặc đối tượng thực, và kết xuất và xuất ra đối tượng XR. Ví dụ, thiết bị XR/PCC 17300 có thể so khớp đối tượng XR bao

gồm thông tin phụ trợ về đối tượng được nhận ra với đối tượng được nhận ra và xuất ra đối tượng XR được so khớp này.

<PCC+tự lái+XR>

Xe tự lái 17200 có thể được thi hành dưới dạng rôbot di động, xe, xe trên không không người lái, hoặc dạng tương tự bằng cách áp dụng công nghệ PCC và công nghệ XR.

Xe tự lái 17200 mà công nghệ XR/PCC được áp dụng với nó có thể thể hiện xe tự lái được cung cấp phương tiện để cung cấp ảnh XR, hoặc xe tự lái mà nó là mục tiêu của việc điều khiển/tương tác trong ảnh XR. Cụ thể, xe tự lái 17200 mà nó là mục tiêu của việc điều khiển/tương tác trong ảnh XR có thể được phân biệt với thiết bị XR 17300 và có thể được kết nối theo cách hoạt động được với nó.

Xe tự lái 17200, có phương tiện để cung cấp ảnh XR/PCC, có thể thu nhận thông tin cảm biến từ các bộ cảm biến chứa camera, và xuất ra ảnh XR/PCC được tạo ra dựa trên thông tin cảm biến thu được. Ví dụ, xe tự lái 17200 có thể có HUD và xuất ra ảnh XR/PCC đến đó, qua đó cung cấp cho người ngồi trên xe đối tượng XR/PCC tương ứng với đối tượng thực hoặc đối tượng có mặt trên màn hình.

Khi đối tượng XR/PCC được xuất ra đến HUD, ít nhất một phần của đối tượng XR/PCC này có thể được xuất ra để chồng lấn với đối tượng thực mà mắt của người ngồi trên xe được hướng đến nó. Mặt khác, khi đối tượng XR/PCC được xuất ra trên bộ hiển thị được cung cấp bên trong xe tự lái, ít nhất một phần của đối tượng XR/PCC có thể được xuất ra để chồng lấn với đối tượng trên màn hình. Ví dụ, xe tự lái 17200 có thể xuất ra các đối tượng XR/PCC tương ứng với các đối tượng chẳng hạn như đường, một xe khác, đèn giao thông, biển báo giao thông, xe hai bánh, người đi bộ, và tòa nhà.

Công nghệ thực tế ảo (VR), công nghệ thực tế tăng cường (AR), công nghệ thực tế hỗn hợp (MR) và/hoặc công nghệ nén đám mây điểm (PCC) theo các phương án là có thể áp dụng được cho các thiết bị khác nhau.

Nói cách khác, công nghệ VR là công nghệ hiển thị mà nó chỉ cung cấp các ảnh CG của các đối tượng thế giới thực, các hậu cảnh, và dạng tương tự. Mặt khác, công nghệ AR chỉ công nghệ mà nó thể hiện ảnh CG được tạo ra theo cách ảo trên ảnh của đối tượng thực. Công nghệ MR là tương tự với công nghệ AR được mô tả trên đây ở chỗ các đối tượng ảo cần được thể hiện được trộn lẫn và được kết hợp với thế giới thực.

Tuy nhiên, công nghệ MR khác với công nghệ AR ở chỗ công nghệ AR phân biệt rõ ràng giữa đối tượng thực và đối tượng ảo được tạo ra dưới dạng ảnh CG và sử dụng các đối tượng ảo dưới dạng các đối tượng bổ sung cho các đối tượng thực, trong khi công nghệ MR xử lý các đối tượng ảo như là các đối tượng có các đặc điểm tương đương với các đối tượng thực. Cụ thể hơn, ví dụ về các ứng dụng công nghệ MR là dịch vụ ảnh toàn ký.

Gần đây, đôi khi các công nghệ VR, AR, và MR được gọi là công nghệ thực tế mở rộng (extended reality, XR) thay vì được phân biệt một cách rõ ràng với nhau. Theo đó, các phương án của sáng chế là có thể áp dụng được cho công nghệ bất kỳ trong số các công nghệ VR, AR, MR và XR. Việc mã hóa/giải mã dựa trên các kỹ thuật PCC, V-PCC, và G-PCC là có thể áp dụng được cho các công nghệ như vậy.

Phương pháp/thiết bị PCC theo các phương án có thể được áp dụng cho xe mà nó cung cấp dịch vụ tự lái.

Xe cung cấp dịch vụ tự lái được kết nối với thiết bị PCC cho việc truyền thông có dây/không dây.

Khi thiết bị truyền/nhận dữ liệu nén đám mây điểm (PCC) theo các phương án được kết nối với xe cho việc truyền thông có dây/không dây, thiết bị này có thể nhận/xử lý dữ liệu nội dung liên quan đến dịch vụ AR/VR/PCC, mà nó có thể được cung cấp cùng với dịch vụ tự lái, và truyền dữ liệu nội dung này đến xe này. Trong trường hợp trong đó thiết bị truyền/nhận PCC được gắn trên xe, thiết bị truyền/nhận PCC này có thể nhận/xử lý dữ liệu nội dung liên quan đến dịch vụ AR/VR/PCC theo tín hiệu đầu vào người dùng được nhập vào thông qua thiết bị giao diện người dùng và cung cấp dữ liệu nội dung này đến người dùng này. Xe hoặc thiết bị giao diện người dùng theo các phương án có thể nhận tín hiệu đầu vào người dùng. Tín hiệu đầu vào người dùng theo các phương án có thể bao gồm tín hiệu chỉ ra dịch vụ tự lái.

Như được mô tả trên đây, dữ liệu đám mây điểm có thể gồm tập các điểm, và mỗi điểm có thể có thông tin dạng hình học và thông tin thuộc tính. Thông tin dạng hình học là thông tin vị trí ba chiều (xyz) về mỗi điểm, và thông tin thuộc tính là giá trị màu (RGB, YUV, v.v.) và/hoặc độ phản xạ của điểm.

Quy trình mã hóa dữ liệu đám mây điểm bao gồm việc nén thông tin dạng hình học và nén thông tin thuộc tính dựa trên thông tin dạng hình học được xây dựng lại dựa

trên thông tin về dạng hình học được xây dựng lại (dạng hình học được xây dựng lại = dạng hình học được giải mã) thu được dựa trên thông tin vị trí được thay đổi qua việc nén. Thêm vào đó, quy trình giải mã dữ liệu đám mây điểm bao gồm việc nhận luồng bit dạng hình học và luồng bit thuộc tính được mã hóa, giải mã thông tin dạng hình học này, và giải mã thông tin thuộc tính này dựa trên thông tin dạng hình học được xây dựng lại qua việc giải mã này. Theo các phương án, quy trình mã hóa dữ liệu đám mây điểm có thể được thực hiện bởi bộ mã hóa video đám mây điểm 10002 trên Fig.1, việc mã hóa 20001 trên Fig.2, bộ mã hóa video đám mây điểm trên Fig.4, bộ mã hóa video đám mây điểm trên Fig.12, hoặc bộ mã hóa dạng hình học 51003 và bộ mã hóa thuộc tính 51004 trên Fig.45. Quy trình giải mã dữ liệu đám mây điểm có thể được thực hiện bởi bộ giải mã video đám mây điểm 10006 trên Fig.1, việc giải mã 20003 trên Fig.2, bộ giải mã video đám mây điểm trên Fig.11, bộ giải mã video đám mây điểm trên Fig.13, hoặc bộ giải mã dạng hình học 61003 và bộ giải mã thuộc tính 61004 trên Fig.48. Theo một phương án, việc mã hóa thông tin dạng hình học được mô tả dưới đây có thể được thực hiện bởi bộ mã hóa dạng hình học 51003 trên Fig.45, và việc mã hóa thông tin thuộc tính có thể được thực hiện bởi bộ mã hóa thuộc tính 51004 trên Fig.45. Fig.45 và Fig.48 sẽ được mô tả chi tiết sau.

Theo các phương án, bộ mã hóa dạng hình học 51003 có thể sử dụng phương pháp nén dựa trên cây bát phân, dựa trên cây tứ phân (ở dưới đây được gọi là QT) hoặc dựa trên cây nhị phân (ở dưới đây được gọi là BT) cho việc nén thông tin dạng hình học. Trong sáng chế này, để đơn giản, cây bát phân, cây tứ phân, hoặc cây nhị phân được sử dụng cho việc nén dạng hình học sẽ được gọi là cây dạng hình học. Tức là, cây dạng hình học có thể là một hoặc nhiều thành phần trong số cây bát phân, cây tứ phân, hoặc cây nhị phân.

Theo các phương án, để làm tăng hiệu quả của việc nén dạng hình học đối với các điểm được phân bố bất đối xứng, bộ mã hóa dạng hình học 51003 có thể nén thông tin dạng hình học dựa trên cây tứ phân (ở dưới đây được gọi là QT) hoặc cây nhị phân (ở dưới đây được gọi là BT).

Tức là, đối với 8 vị trí nút cấu thành nên cây bát phân, khi các điểm này được phân bố định thiên về phía một mặt phẳng hoặc một đoạn thẳng, số lượng bit được sử dụng cho việc nén dạng hình học có thể được giảm xuống bởi việc sử dụng thêm QT hoặc BT.

Fig.15 là sơ đồ minh họa ví dụ về dữ liệu đám mây điểm bao gồm các điểm được phân bố bất đối xứng theo các phương án.

Theo các phương án, bộ mã hóa dạng hình học 51003 có thể phân vùng hộp giới hạn (mà nó có thể được sử dụng theo cách thay thế được cho nhau với B dưới đây) của dữ liệu đám mây điểm như được thể hiện trên Fig.15 dựa trên một hoặc nhiều OT, QT, và BT để nén thông tin dạng hình học.

Fig.16 là sơ đồ minh họa ví dụ về việc phân vùng khối lập phương dựa trên OT trong không gian 3D.

Trong một phương án, khi các điểm của dữ liệu đám mây điểm được phân bố đều, phương pháp phân vùng dựa trên OT gồm việc phân chia đồng đều khối lập phương (cụ thể là khối lập phương 3D) thành tám khối lập phương có thể được sử dụng cho việc nén thông tin dạng hình học về dữ liệu đám mây điểm. Trong phương pháp này, quy trình phân chia nút được chiếm giữ trong số 8 nút thành 8 khối lập phương bằng cách phân chia mỗi phía của nút thành hai nửa được lặp lại. Việc liệu mỗi nút có được chiếm giữ hay không có thể được chỉ ra dưới dạng 0 hoặc 1 bằng cách sử dụng 1 bit theo thứ tự. Tức là, khi khối lập phương được phân chia thành 8 khối lập phương, 8 bit có thể được sử dụng để chỉ ra liệu 8 nút con này có được chiếm giữ hay không. Ví dụ, khi các nút 0, 3, 4, và 5 được chiếm giữ trong số 8 nút trên Fig.16, sự chiếm giữ đối với các nút tương ứng có thể được chỉ ra dưới dạng 10011100. Tức là, khi phương pháp phân vùng dựa trên OT được thực hiện, một mã chiếm giữ được biểu diễn cho 8 nút con. Nói cách khác, khi khối lập phương 3D được phân vùng dựa trên OT, 8 nút lập phương được tạo ra, và mã chiếm giữ được thể hiện dưới dạng mã 8 bit. Trong một phương án, 8 nút lập phương này có thể được gọi là 8 nút con.

Fig.17-(a) đến Fig.17-(c) là các sơ đồ minh họa các ví dụ về việc phân vùng khối lập phương dựa trên QT trong không gian 3D.

Fig.18-(a) đến Fig.18-(c) là các sơ đồ minh họa các ví dụ về việc phân vùng khối lập phương dựa trên BT trong không gian 3D.

Trong một phương án, khi phân bố điểm của dữ liệu đám mây điểm không đồng đều, số lượng bit được sử dụng để chỉ ra việc chiếm giữ có thể được giảm xuống bằng cách sử dụng QT và/hoặc BT tùy theo các đặc điểm của phân bố điểm.

Theo các phương án, trong trường hợp QT, khối lập phương 3D có thể được phân

vùng dọc theo các trục x-y, các trục y-z, các trục x-z, như trên Fig.17(a) đến Fig.17-(c). Do đó, khi khối lập phương 3D được phân vùng dựa trên QT, bốn nút phỏng lập phương được tạo ra, và mã chiếm giữ được thể hiện dưới dạng mã 4 bit. Trong trường hợp này, mã chiếm giữ có thể được giảm xuống 4 bit so với mã chiếm giữ cho việc phân vùng dựa trên cây bát phân. Trong một phương án, 4 nút phỏng lập phương này có thể được gọi là 4 nút con. Tức là, một nút mẹ có thể có 4 nút con. Ví dụ, khi khối lập phương 3D được phân vùng dọc theo các trục x-y dựa trên QT, và các nút 0 và 6 trong số bốn nút được chiếm giữ, sự chiếm giữ của mỗi nút này có thể được chỉ ra bởi 1001 trong khi chỉ ra rằng việc phân vùng được dựa trên QT theo hướng x-y.

Theo các phương án, trong trường hợp BT, khối lập phương 3D có thể được phân vùng dọc theo trục x, trục y, hoặc trục z, như được thể hiện trên Fig.18-(a) đến Fig.18-(c). Do đó, khi khối lập phương 3D được phân vùng dựa trên BT, hai nút phỏng lập phương được tạo ra, và mã chiếm giữ được thể hiện dưới dạng mã 2 bit. Trong trường hợp này, mã chiếm giữ có thể được giảm xuống 6 bit so với mã chiếm giữ cho việc phân vùng dựa trên OT. Trong một phương án, hai nút phỏng lập phương này có thể được gọi là hai nút con. Tức là, một nút mẹ có thể có hai nút con.

Theo các phương án, thứ tự của OT, QT, và BT cho dữ liệu đám mây điểm có thể được xác định theo các tham số K và M.

Theo các phương án, tham số K có thể biểu thị số lần QTBT được thực hiện khi việc phân vùng hộp giới hạn 3D được bắt đầu, và tham số M có thể biểu thị số lần OT được thực thi ngay lập tức trước khi phân vùng đối với mức nút lá.

Theo các phương án, đối với đám mây điểm có hộp giới hạn mà các kích thước các chiều của nó là 2^{dx} , 2^{dy} , và 2^{dz} , tham số K có thể có giá trị nằm trong khoảng từ 0 đến $\max(dx, dy, dz) - \min(dx, dy, dz)$ như sau. Tham số M có thể có giá trị trong khoảng từ 0 và $\min(dx, dy, dz)$ như sau.

$$K (0 \leq K \leq \max(dx, dy, dz) - \min(dx, dy, dz))$$

Tức là, tham số K biểu thị số lần QT&BT tối đa trước OT (số lần tối đa của QT&BT trước OT).

$$M (0 \leq M \leq \min(dx, dy, dz))$$

Tức là, tham số M biểu thị kích thước tối thiểu của QT&BT. Khi kích thước trục X (dx), kích thước trục Y (dy), và kích thước trục Z (dz) nhỏ hơn hoặc bằng M, QT và

BT có thể không được thực hiện (không có QT&BT nếu $dx, dy, dz \leq M$).

Thêm vào đó, khi kích thước trục X (dx), kích thước trục Y (dy), và kích thước trục Z (dz) là khác nhau (khi $dx \neq dy \neq dz$), BT được thực hiện trước QT. $M=0$ nghĩa là OT được thực hiện ít nhất 0 hoặc nhiều hơn 0 lần ngay trước khi phân vùng đối với mức nút lá. Ví dụ, khi $M=0$, OT có thể không được thực hiện ngay trước khi phân vùng đối với mức nút lá, nhưng OT có thể được thực hiện một số lần được xác định trước, chẳng hạn như một lần hoặc hai lần.

Ở đây, $\min()$, $\text{med}()$, và $\max()$ là các hàm để tìm các giá trị tối thiểu, trung bình, và tối đa, và dx, dy , và dz là các giá trị của $\log_2$ đối với kích thước cho mỗi trục trong số các trục xyz của nút con đối với độ sâu cụ thể.

Theo các phương án, số lần OT/QT/BT có thể được xác định tùy theo kích thước của hộp giới hạn như sau.

$$N(\text{OT}) = \min(dx, dy, dz)$$

$$N(\text{QT}) = \text{med}(dx, dy, dz) - \min(dx, dy, dz)$$

$$N(\text{BT}) = \max(dx, dy, dz) - \text{med}(dx, dy, dz)$$

Ví dụ, khi việc phân vùng dựa trên QT/BT được thực hiện, tổng số lần thực thi việc phân vùng là $N(\text{OT}) + N(\text{QT}) + N(\text{BT}) = \max(dx, dy, dz)$, mà nó là bằng tổng số lần thực thi việc phân vùng dựa trên OT, $\max(dx, dy, dz)$. Thứ tự của OT/QT/BT được xác định bởi các tham số K và M.

Fig.19-(a) và Fig.19-(b) thể hiện các ví dụ về việc phân vùng hộp giới hạn trong không gian 2D. Trên Fig.19-(a) và Fig.19-(b), các dấu x chỉ ra các vị trí nơi các điểm có mặt trong hộp giới hạn.

Fig.19-(a) thể hiện ví dụ về thứ tự phân vùng của dữ liệu đám mây điểm được phân bố trong hộp giới hạn 16x4 trong không gian 2D dựa trên chỉ riêng QT, và Fig.19-(b) thể hiện ví dụ về thứ tự phân vùng của dữ liệu đám mây điểm được phân bố trong hộp giới hạn 16x4 trong không gian 2D dựa trên QT và BT.

Ví dụ trên Fig.19-(a) và Fig.19-(b) có thể được phổ quát hóa thành thứ tự phân vùng trong không gian 3D theo việc sử dụng OT, QT, và BT. Trong trường hợp này, QT trở thành OT trong không gian 3D, và BT trở thành QT hoặc BT trong không gian 3D.

Trong các ví dụ trên Fig.19-(a) và Fig.19-(b), việc phân vùng được thực hiện bốn

lần lên đến nút lá 1x1. Khi việc phân vùng được thực hiện dựa trên chỉ riêng QT (OT trong không gian 3D) như được thể hiện trên Fig.19-(a), việc phân vùng được thực hiện, giả sử rằng hộp giới hạn kích thước 16x4 có kích thước 16x16.

Trên hình vẽ này, việc phân vùng diễn tiến sang bên phải, và nút được chiếm giữ được phân chia bởi các đường liền hoặc chấm theo QT (OT trong không gian 3D) ở mỗi giai đoạn. Kích thước nút đơn vị ở mỗi giai đoạn có thể được biểu diễn là 16x4, 8x8, 4x4, 2x2, hoặc 1x1. Tức là, trên Fig.19-(a), đường chấm hoặc đường liền trong hộp giới hạn ở mỗi giai đoạn chỉ ra các nút tại mỗi độ sâu khi việc phân vùng được thực hiện dựa trên QT.

Khi việc phân vùng được thực hiện dựa trên sự kết hợp của QT (OT trong không gian 3D) và BT (QT trong không gian 3D) như được thể hiện trên Fig.19-(b), hộp giới hạn bất đối xứng 16x4 có thể được sử dụng, và QT có thể được thực hiện hai lần và BT có thể được thực hiện hai lần cho đến khi đạt đến nút lá 1x1. Để làm ví dụ, trong trường hợp Fig.19-(b), các tham số K và M được đặt là K=0 và M=0, và việc phân vùng diễn tiến sang bên phải theo QT, QT, BT, và BT theo thứ tự này. Trong trường hợp này, các kích thước nút đơn vị tại các giai đoạn tương ứng có thể được biểu diễn là 16x4, 8x2, 4x1, 2x1, và 1x1.

Việc nén thông tin thuộc tính có thể được thực hiện bởi bộ mã hóa thuộc tính 51004 dựa trên các mức độ chi tiết (LoD). Trong trường hợp này, bộ mã hóa thuộc tính 51004 có thể tạo ra các LoD sử dụng các phương pháp khác nhau. Trong một phương án, bộ mã hóa thuộc tính 51004 có thể tạo ra các LoD dựa trên cây bát phân.

Theo các phương án, LOD chỉ ra mức độ chi tiết của dữ liệu đám mây điểm (hoặc nội dung đám mây điểm). Việc giảm giá trị LOD nghĩa là việc suy giảm chi tiết của nội dung đám mây điểm, và việc tăng giá trị LOD nghĩa là việc nâng cao chi tiết của nội dung đám mây điểm. Tức là, các LOD được tạo cấu hình sao cho mật độ của các điểm tăng lên khi LOD nâng lên. Các điểm của dạng hình học được xây dựng lại bởi bộ mã hóa dạng hình học 51003, tức là, các vị trí được xây dựng lại, có thể được phân loại theo LOD bởi bộ mã hóa thuộc tính 51004.

Trong một phương án, khi kỹ thuật lập mã biến đổi dự đoán và kỹ thuật lập mã biến đổi nâng được sử dụng, các điểm có thể được nhóm lại bằng cách phân chia các điểm này vào trong các LOD.

Hoạt động này có thể được gọi là quy trình tạo ra LOD, và nhóm có các LOD khác nhau có thể được gọi là tập LOD_l . Ở đây, l biểu thị LOD và là số nguyên bắt đầu từ 0. Chi tiết của nội dung đám mây điểm là thấp nhất trong LOD_0 , và chi tiết của nội dung đám mây điểm tăng lên khi l tăng lên.

Fig.20 là sơ đồ minh họa ví dụ về việc tạo ra các LOD dựa trên cây bát phân bởi bộ mã hóa thuộc tính 51004 theo các phương án. Trên Fig.20, các điểm có thể là các điểm của đám mây điểm được thu nạp, hoặc các điểm của dạng hình học được xây dựng lại bởi bộ mã hóa dạng hình học 51003.

Theo các phương án, khi các LOD được tạo ra dựa trên cây bát phân, mỗi mức độ sâu của cây bát phân có thể được so khớp với mỗi LOD như được thể hiện trên Fig.20. Tức là, trong phương pháp tạo ra LOD dựa trên cây bát phân, các LOD được tạo ra dựa trên nguyên lý đó là khi mức độ sâu tăng lên trong cấu trúc cây bát phân (theo hướng từ gốc đến lá), chi tiết biểu diễn dữ liệu đám mây điểm tăng lên. Theo các phương án, việc tạo cấu hình LOD dựa trên cây bát phân có thể diễn tiến từ nút gốc đến nút lá, hoặc từ nút lá đến nút gốc.

Trong ví dụ này, các điểm tại mỗi mức độ sâu được sắp xếp dựa trên thứ tự mã Morton, và nút thứ nhất trong số các nút con được chọn làm nút tham chiếu hoặc nút được giữ lại của nút mẹ được thể hiện. Số chỉ dẫn 50001 thể hiện ví dụ về các điểm thuộc về LOD N-x. Nói cách khác, các điểm thuộc về LOD N-x tương ứng với số lượng nút được chiếm giữ tại mức độ sâu tương ứng.

Fig.21 là sơ đồ minh họa ví dụ về thứ tự mã Morton trong khối lập phương 3D theo các phương án.

Tức là, mã Morton của mỗi điểm trong đám mây điểm được tạo ra dựa trên các giá trị vị trí x, y, và z của mỗi của điểm. Một khi các mã Morton của các điểm trong đám mây điểm được tạo ra qua quy trình này, các điểm này trong đám mây điểm có thể được sắp xếp theo thứ tự của các mã Morton. Theo các phương án, các điểm này của đám mây điểm có thể được sắp xếp theo thứ tự tăng dần của các mã Morton. Thứ tự của các điểm được sắp xếp theo thứ tự tăng dần của các mã Morton có thể được gọi là thứ tự Morton.

Theo các phương án, khi bộ mã hóa thuộc tính 51004 tạo ra các LOD dựa trên cây bát phân như được mô tả trên đây, khả năng định tỷ lệ theo không gian có thể được hỗ

trợ. Với khả năng định tỷ lệ theo không gian, một đối tượng có thể truy cập đám mây điểm độ phân giải thấp hơn dưới dạng hình nhỏ (thumbnail) với độ phức tạp bộ giải mã nhỏ hơn và/hoặc băng thông nhỏ hơn khi đám mây điểm nguồn là dày đặc. Chức năng khả năng định tỷ lệ theo không gian của dạng hình học có thể được cung cấp qua quy trình mã hóa hoặc giải mã các bit chiếm giữ lên đến mức độ sâu được chọn bằng cách điều chỉnh mức độ sâu của cây bát phân trong quá trình mã hóa/giải mã dạng hình học. Ngoài ra, trong việc mã hóa/giải mã thuộc tính, chức năng khả năng định tỷ lệ theo không gian của thuộc tính cũng có thể được cung cấp qua quy trình gồm việc tạo ra các LOD từ mức độ sâu được chọn của cây bát phân và tạo cấu hình các điểm mà thuộc tính cần được mã hóa/giải mã cho chúng.

Tức là, trong việc mã hóa thuộc tính có thể định tỷ lệ được, độ phân giải có thể tăng lên theo độ sâu của việc mã hóa dạng hình học dựa trên cây bát phân. Nói cách khác, khi cây bát phân được phân vùng, chi tiết tăng lên. Trong trường hợp này, phép biểu diễn có thể định tỷ lệ được có thể được cho phép bằng cách so khớp thuộc tính với nút được chiếm giữ tại mỗi độ sâu (hoặc mức độ sâu). Để đạt được mục đích này, trong quy trình mã hóa thuộc tính có thể định tỷ lệ được, việc tạo ra LoD dựa trên cây bát phân được thực hiện, và việc chọn được thực hiện theo nút đơn vị, không theo khoảng cách. Trong trường hợp này, các nút con có cùng nút là mẹ có thể được định ra là các lân cận (hoặc khoảng tìm kiếm lân cận), và một thuộc tính có thể được chọn trong các nút (hoặc các điểm) lân cận và được khớp với nút mẹ. Ví dụ, thuộc tính của nút con thứ nhất trong số 8 nút con có thể được chọn làm thuộc tính của nút mẹ.

Fig.22 là sơ đồ minh họa ví dụ về quy trình chọn thuộc tính theo việc tạo ra LoD cho các nút (hoặc các điểm) trong mặt phẳng 2D, theo các phương án.

Trên Fig.22, các dấu x chỉ ra các vị trí nơi các điểm có mặt trong hộp giới hạn có kích thước 16x4. Các đường chấm hoặc các đường liền trong hộp giới hạn tại mỗi giai đoạn thể hiện các nút tại mỗi độ sâu theo việc phân vùng dựa trên QT tại mỗi giai đoạn. Tức là, trên Fig.22, việc phân vùng diễn tiến sang bên phải, và kích thước nút đơn vị ở mỗi giai đoạn có thể được biểu diễn là 16x4, 8x8, 4x4, 2x2, hoặc 1x1.

Khi việc tạo ra LoD được thực hiện dựa trên cấu trúc cây được sử dụng cho việc phân vùng như được thể hiện trên Fig.22, hoạt động này có thể diễn tiến từ lá đến gốc, hoặc có thể diễn tiến từ gốc đến lá nếu cần thiết. Các LoD được tạo ra có thể được tạo

cấu hình sao cho LoD tăng lên theo hướng từ gốc đến lá. Thêm vào đó, các hình tròn tại mỗi giai đoạn có thể thể hiện các điểm đang cấu thành nên mỗi LoD, và LoD tối đa (ví dụ, LoD4) có thể chứa tất cả các điểm. Như là một ví dụ khác, trên Fig.22, khi LoD2 được tạo ra dựa trên các điểm đang cấu thành nên LoD3, số lượng điểm cấu thành nên LoD2 là 4 bởi vì số lượng nút được chiếm giữ tại mức độ sâu này là 4.

Như được mô tả trên đây, khi phương pháp nén dạng hình học dựa trên cây bát phân được sử dụng, việc lập mã có thể định tỷ lệ được được hỗ trợ bằng cách tạo cấu hình các thuộc tính tương ứng với cấu hình lớp cây bát phân. Trong trường hợp này, tỷ lệ của độ rộng, độ sâu và độ cao của nút đơn vị được sử dụng tại độ sâu của cây thông tin dạng hình học là giống nhau. Ngoài ra, thông tin dạng hình học và thông tin thuộc tính được chứa tại độ sâu cây bát phân cụ thể được khớp với nhau theo kiểu phép tương ứng một-một. Nói cách khác, một khi bộ mã hóa dạng hình học 51003 thực hiện việc nén dạng hình học dựa trên cây bát phân và bộ mã hóa thuộc tính 51004 tạo ra các LoD dựa trên cây bát phân này và thực hiện việc nén thuộc tính, số lượng điểm cho dạng hình học là giống với số lượng điểm cho thuộc tính. Do đó, khi đám mây điểm được giải mã một phần cây bát phân được cung cấp cho bộ giải mã thuộc tính 61004 trên phía nhận, bộ giải mã thuộc tính 61004 có thể thực hiện một cách chính xác việc giải mã thuộc tính có thể định tỷ lệ được từ mức độ sâu được cung cấp.

Fig.23 là sơ đồ minh họa một ví dụ khác về quy trình chọn thuộc tính theo việc tạo ra LoD cho các nút (hoặc các điểm) trong mặt phẳng 2D bởi bộ mã hóa thuộc tính 51004 theo các phương án.

Trên Fig.23, các dấu x chỉ ra các vị trí nơi các điểm có mặt trong hộp giới hạn có kích thước 16×4 . Các đường chấm hoặc các đường liền trong hộp giới hạn tại mỗi giai đoạn thể hiện các nút tại mỗi độ sâu theo việc phân vùng dựa trên QT hoặc BT tại mỗi giai đoạn. Tức là, Fig.23 thể hiện trường hợp $K=1$ và $M=1$, và việc phân vùng diễn tiến sang bên phải theo BT, QT, BT, và QT theo thứ tự này. Nói cách khác, khi việc phân vùng dựa trên BT được thực hiện dựa trên các tham số $K=1$ và $M=1$, việc phân vùng ban đầu được thực hiện dựa trên BT, và việc phân vùng được thực hiện dựa trên QT trong việc tạo ra nút lá. Theo đó, hộp giới hạn được phân vùng theo BT, QT, BT, và QT theo thứ tự này. Trong trường hợp này, kích thước nút đơn vị ở giai đoạn này và có thể được biểu diễn là 16×4 , 8×4 , 4×2 , 2×2 , hoặc 1×1 .

Kể cả khi việc tạo ra LoD được thực hiện dựa trên cấu trúc cây dạng hình học được sử dụng cho việc phân vùng như được thể hiện trên Fig.23, việc tạo ra LoD có thể diễn tiến từ lá đến gốc hoặc có thể diễn tiến từ gốc đến lá nếu cần thiết. Ngoài ra, các LoD được tạo ra có thể được tạo cấu hình sao cho LoD tăng lên theo hướng từ gốc đến lá. Trong trường hợp này, LoD có thể được tạo cấu hình bằng điểm đại diện của nút tại mỗi giai đoạn. Để làm ví dụ, các hình tròn tại mỗi giai đoạn có thể thể hiện các điểm đang cấu thành nên mỗi LoD, và LoD tối đa (ví dụ, LOD4) có thể chứa tất cả các điểm. Như là một ví dụ khác, trên Fig.23, khi LoD2 được tạo ra dựa trên các điểm đang cấu thành nên LoD3, số lượng điểm cấu thành nên LoD2 là 5 bởi vì số lượng nút được chiếm giữ tại mức độ sâu này là 5.

Khi BT (BT hoặc QT trong không gian 3D) được sử dụng trong việc lập mã dạng hình học như được thể hiện trên Fig.23, LoD có thể được tạo cấu hình tại mức độ sâu cụ thể với số lượng điểm khác với số lượng điểm trong ví dụ trên Fig.22 như được mô tả trên đây. Điều này là bởi vì các điểm được chọn có thể được thay đổi tùy theo việc thiết lập khoảng tìm kiếm lân cận trong việc tạo ra LoD. Ví dụ, khoảng tìm kiếm lân cận (cụ thể là số lượng nút con) có thể là 8 trong cấu trúc cây bát phân. Khoảng tìm kiếm lân cận có thể là 4 trong cấu trúc cây tứ phân. Khoảng tìm kiếm lân cận có thể là 2 trong cấu trúc cây nhị phân.

Theo các phương án, trong trường hợp cấu hình LoD dựa trên QT như được thể hiện trên Fig.22, số lượng điểm thuộc về LoD2 có thể là 4. Mặt khác, trong trường hợp cấu hình LoD dựa trên QTBT như được thể hiện trên Fig.23, số lượng điểm thuộc về LoD2 có thể là 5. Theo đó, khi bộ mã hóa dạng hình học 51003 nén thông tin dạng hình học bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, nhưng bộ mã hóa thuộc tính 51004 tạo cấu hình các LoD dựa trên OT, sự không trùng khớp của các điểm có thể xảy ra giữa dạng hình học và các thuộc tính tại các độ sâu cụ thể, ví dụ, LoD2. Nói cách khác, khi việc trình chiếu có thể định tỷ lệ được thực hiện sử dụng LoD2, số lượng điểm trong dạng hình học có thể khác với số lượng điểm trong thuộc tính cho cấu hình LOD. Kết quả là, các thuộc tính sai có thể được so khớp với dạng hình học, hoặc có thể có các điểm có các thuộc tính mà chúng không được so khớp.

Như vậy, khi OT/QT/BT được sử dụng để phân vùng hợp giới hạn để xem xét phân bố điểm bất đối xứng trong việc nén thông tin dạng hình học, các nút đơn vị được sử

dụng trong mỗi độ sâu của cây dạng hình học có thể có các tỷ lệ về độ rộng, độ sâu và độ cao khác nhau. Ví dụ, trên Fig.23, các nút đơn vị tại các giai đoạn tương ứng theo mặt phẳng 2D là 16×4 , 8×4 , 4×2 , 2×2 , và 1×1 . Trong trường hợp này, khi cấu trúc lớp của việc lập mã dạng hình học không được xem xét trong việc lập mã thuộc tính có thể định tỷ lệ được, phép biểu diễn đám mây điểm có thể định tỷ lệ được có thể là không chính xác do thiếu việc so khớp một-một giữa thông tin dạng hình học và thông tin thuộc tính cho lớp có thể định tỷ lệ được bất kỳ. Nói cách khác, khi bộ mã hóa dạng hình học 51003 nén thông tin dạng hình học bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, và bộ mã hóa thuộc tính 51004 nén thông tin thuộc tính bằng cách tạo ra các LoD dựa trên OT, số lượng điểm cho dạng hình học có thể khác với số lượng điểm cho các thuộc tính. Trong trường hợp này, phép biểu diễn đám mây điểm có thể định tỷ lệ được tại độ sâu tương ứng có thể là không chính xác.

Để giải quyết vấn đề về sự sai lệch này, lớp có thể định tỷ lệ được dựa trên LoD có thể được tạo cấu hình sử dụng nút lân cận bất đối xứng trong việc mã hóa đám mây điểm có thể định tỷ lệ được. Qua đó, hiệu quả nén thuộc tính của nội dung có phân bố điểm bất đối xứng có thể được tăng lên. Nói cách khác, trong trường hợp khi bộ mã hóa dạng hình học 51003 thực hiện việc nén dạng hình học bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, bộ mã hóa thuộc tính 51004 thực hiện việc nén thuộc tính bằng cách tạo ra các LoD dựa trên OT/QT/BT. Theo đó, số lượng điểm dành cho dạng hình học trùng với số lượng điểm dành cho các thuộc tính tại mỗi độ sâu.

Sáng chế đề xuất phương pháp nén dựa trên việc phân vùng nút bất đối xứng trong số các phương pháp dành cho việc lập mã đám mây điểm có thể định tỷ lệ được. Phương pháp nén dựa trên việc phân vùng nút bất đối xứng được mô tả trong sáng chế này có thể được sử dụng làm phương pháp nén thuộc tính có thể định tỷ lệ được khi phương pháp nén dạng hình học dựa trên OT/QT/BT được sử dụng. Phương pháp được đề xuất trong sáng chế này có thể được sử dụng không chỉ cho việc lập mã đám mây điểm có thể định tỷ lệ được, mà còn được sử dụng làm phương pháp để cung cấp theo cách thích ứng các dịch vụ đám mây điểm trong môi trường độ trễ thấp. Thêm vào đó, phương pháp nén thuộc tính dựa trên việc dự đoán được đề xuất trong sáng chế này có thể được sử dụng một cách độc lập với phương pháp nén dạng hình học, và có thể được sử dụng cho việc nén dạng hình học cũng như việc nén thuộc tính dựa trên phương pháp được

đề xuất trong sáng chế này. Phương pháp tạo ra LoD được đề xuất trong sáng chế này có thể được sử dụng không chỉ trong lĩnh vực ứng dụng của việc lập mã có thể định tỷ lệ được mà còn trong việc nén đám mây điểm nói chung.

Tiếp theo, các phương pháp lập mã thuộc tính có thể định tỷ lệ được xem xét đến phân bố điểm bất đối xứng, bao gồm phương pháp tạo ra LoD xem xét đến các nút phồng lập phương và phương pháp lập mã thuộc tính xem xét đến các thay đổi của nút theo các lớp khi việc nén dạng hình học dựa trên OT/QT/BT được sử dụng, hoặc các ràng buộc và việc báo hiệu đối với việc sử dụng việc nén thuộc tính có thể định tỷ lệ được dựa trên OT sẽ được mô tả.

Phương án 1

Khi việc mã hóa dạng hình học được thực hiện bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, nút bất đối xứng (hoặc phồng lập phương) có thể được sử dụng tại mỗi độ sâu của cây dạng hình học này. Trong trường hợp này, khi việc lập mã có thể định tỷ lệ được xem xét, khả năng định tỷ lệ được hỗ trợ trên cơ sở từng độ sâu một trong cây dạng hình học. Phương án 1 là ví dụ về việc lập mã thuộc tính được thực hiện bằng cách tạo ra các LoD theo cấu trúc cây dạng hình học. Trong một phương án của sáng chế, các LoD được tạo ra có xem xét đến nút bất đối xứng của cây dạng hình học cho việc lập mã có thể định tỷ lệ được.

Theo các phương án, trước hết, để xem xét các đặc điểm phân bố của các điểm được phân bố theo hướng cụ thể, việc tìm kiếm lân cận được thực hiện dựa trên nút phi lập phương để làm ví dụ. Theo các phương án, khi các kích thước của nút tại mỗi độ sâu là 2^{dx} , 2^{dy} , và 2^{dz} trong cấu trúc cây dạng hình học được sử dụng trong việc lập mã dạng hình học, dx , dy , và dz có thể có các giá trị khác nhau. Tuy nhiên, trong một phương án, khi (k) được giả sử là độ sâu thứ k , các mối quan hệ $dx(k) \geq dx(k+1)$, $dy(k) \geq dy(k+1)$, và $dz(k) \geq dz(k+1)$ có thể được xác lập. Để làm ví dụ, trong việc tạo ra LoD dựa trên cây bát phân, việc tìm kiếm lân cận có thể được thực hiện dựa trên nút lập phương. Trong trường hợp này, chỉ trường hợp $dx=dy=dz$ là được xem xét.

Thêm vào đó, tiêu chí để chọn các điểm cho LoD cho khoảng tìm kiếm lân cận được cho tại độ sâu tương ứng có thể được tạo cấu hình. Khoảng tìm kiếm lân cận được cho tại độ sâu này tùy thuộc vào việc liệu hoạt động được dựa trên OT, QT, hay BT. Ví dụ, khoảng tìm kiếm lân cận (cụ thể là số lượng nút con) cho hoạt động dựa trên OT có

thể là 8. Khoảng tìm kiếm lân cận cho hoạt động dựa trên QT có thể là 4. Khoảng tìm kiếm lân cận cho hoạt động dựa trên BT có thể là 2. Trong một phương án, điểm thứ nhất được sắp xếp theo thứ tự của thứ tự mã Morton (cụ thể là, nút được chiếm giữ thứ nhất trong số các nút được chiếm giữ được sắp xếp theo thứ tự của thứ tự mã Morton) có thể được chọn trong khoảng tìm kiếm lân cận. Trong một phương án khác, các điểm tại các vị trí khác nhau có thể được chọn theo độ sâu. Ví dụ, khi điểm thứ nhất trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận được chọn cho LoD thứ k , điểm cuối cùng trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận có thể được chọn cho LoD thứ $k-1$. Trong một phương án khác, điểm mà nó nằm ở khoảng cách gần so với tâm của nút này có thể được chọn. Trong một phương án, thông tin (ví dụ, `LoD_sampling_method`) để nhận dạng phương pháp chọn điểm được sử dụng tại một độ sâu có thể được báo hiệu qua thông tin báo hiệu. Thông tin báo hiệu này có thể ít nhất là tập tham số thuộc tính hoặc tiêu đề lát thuộc tính. Trong sáng chế này, để đơn giản, thông tin (ví dụ, `LoD_sampling_method`) để nhận dạng phương pháp chọn điểm được sử dụng tại một độ sâu có thể được gọi là thông tin để nhận dạng phương pháp lấy mẫu LoD.

Fig.24-(a) và Fig.24-(b) là các sơ đồ minh họa các ví dụ về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D.

Fig.24-(a) và Fig.24-(b) minh họa các ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hộp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo QT, QT, BT, và BT theo thứ tự này dựa trên các tham số được đặt là $K=0$ và $M=0$. Theo các phương án, bộ mã hóa thuộc tính 51004 có thể thực hiện việc tạo ra LoD dựa trên cấu trúc cây dạng hình học có nút không vuông trên Fig.24. Trong trường hợp này, các LoD có thể được tạo ra có xem xét đến nút bất đối xứng tại mỗi độ sâu của cây dạng hình học. Ví dụ, khi LoD4 được giả sử là LoD tối đa, LoD4 chứa tất cả các điểm của hộp giới hạn. Thêm vào đó, đối với cấu hình LoD thấp hơn (LoD được đánh số nhỏ), các lân cận có thể được tạo cấu hình trong đơn vị nút có kích thước nút đơn vị của độ sâu tương ứng với LoD thấp hơn của kích thước nút đơn vị của LoD thấp hơn này tương ứng với LoD thấp hơn trong số các điểm đang cấu thành nên LoD cao hơn (LoD được đánh số lớn), và điểm

cụ thể có thể được chọn từ trong số các điểm được tạo cấu hình làm các lân cận này. Qua đó, các LoD thấp hơn có thể được tạo cấu hình. Trên Fig.24, các kích thước nút đơn vị tại các độ sâu tương ứng có thể được biểu diễn là 16×4 , 8×2 , 4×1 , 2×1 , và 1×1 . Ví dụ, LoD3 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận (khoảng tìm kiếm lân cận) trong đơn vị nút (BT) có kích thước 2×1 trong số các điểm đang cấu thành nên LoD4 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD2 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận trong đơn vị nút (BT) có kích thước 4×1 trong số các điểm đang cấu thành nên LoD3 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD1 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận trong đơn vị nút (QT) có kích thước 8×2 trong số các điểm đang cấu thành nên LoD2 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD0 có thể được tạo cấu hình bằng cách tạo cấu hình bốn điểm được chứa trong LoD1 làm các lân cận và chọn điểm cụ thể trong số bốn điểm được tạo cấu hình làm các lân cận này.

Fig.24-(a) minh họa ví dụ về việc chọn điểm thứ nhất xét về mã Morton trong quy trình chọn điểm đối với mỗi độ sâu. Tức là, đối với LoD3 đến LoD0, khi chọn, điểm cụ thể được chọn từ trong số các điểm được tạo cấu hình làm các lân cận, điểm thứ nhất được chọn từ các điểm lân cận được sắp xếp theo thứ tự mã Morton.

Fig.24-(b) minh họa ví dụ về việc chọn điểm thứ nhất hoặc điểm cuối cùng xét về mã Morton theo độ sâu trong quy trình chọn điểm. Cụ thể hơn, hình vẽ này minh họa ví dụ về việc chọn luân phiên các điểm tại các vị trí thứ nhất và cuối cùng xét về mã Morton tại các độ sâu liên tiếp trong quy trình chọn điểm. Nói cách khác, đối với LoD3, điểm thứ nhất có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD2, điểm cuối cùng có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD1, điểm thứ nhất có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD0, điểm cuối cùng có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận.

Ngược với Fig.24-(a), phân bố điểm trên Fig.24-(b) có thể được tạo cấu hình sao cho các điểm của LOD thấp hơn là gần hơn với tâm của nút. Các điểm 50003 và 50005 trên Fig.24-(a) là các điểm thứ nhất được chọn dựa trên mã Morton tại các độ sâu tương

ứng, và các điểm 50007 và 50009 trên Fig.24-(b) là các điểm cuối cùng được chọn dựa trên mã Morton tại các độ sâu tương ứng.

Fig.25-(a) và Fig.25-(b) là các sơ đồ minh họa các ví dụ khác về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trên mặt phẳng 2D.

Fig.25-(a) và Fig.25-(b) minh họa các ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hộp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, QT, QT, và BT theo thứ tự này dựa trên các tham số được đặt là $K=1$ và $M=0$. Theo các phương án, bộ mã hóa thuộc tính 51004 có thể thực hiện việc tạo ra LoD dựa trên cấu trúc cây dạng hình học có nút không vuông trên Fig.25. Trong trường hợp này, các LoD có thể được tạo ra có xem xét đến nút bất đối xứng tại mỗi độ sâu của cây dạng hình học. Ví dụ, khi LoD4 được giả sử là LoD tối đa, LoD4 chứa tất cả các điểm của hộp giới hạn. Thêm vào đó, đối với cấu hình LoD thấp hơn (LoD được đánh số nhỏ), các lân cận có thể được tạo cấu hình trong đơn vị nút có kích thước nút đơn vị của độ sâu tương ứng với LoD thấp hơn của kích thước nút đơn vị của LoD thấp hơn này tương ứng với LoD thấp hơn trong số các điểm đang cấu thành nên LoD cao hơn (LoD được đánh số lớn), và điểm cụ thể có thể được chọn từ trong số các điểm được tạo cấu hình làm các lân cận này. Qua đó, các LoD thấp hơn có thể được tạo cấu hình. Trên Fig.25, các kích thước nút đơn vị tại các độ sâu tương ứng có thể được biểu diễn là 16×4 , 18×4 , 4×2 , 2×1 , và 1×1 . Ví dụ, LoD3 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận (khoảng tìm kiếm lân cận) trong đơn vị nút (BT) có kích thước 2×1 trong số các điểm đang cấu thành nên LoD4 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD2 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận trong đơn vị nút (QT) có kích thước 4×2 trong số các điểm đang cấu thành nên LoD3 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD1 có thể được tạo cấu hình bằng cách tạo cấu hình các lân cận trong đơn vị nút (QT) có kích thước 8×4 trong số các điểm đang cấu thành nên LoD2 và chọn điểm cụ thể trong số các điểm được tạo cấu hình làm các lân cận này. LoD0 có thể được tạo cấu hình bằng cách tạo cấu hình hai điểm được chứa trong LoD1 làm các lân cận và chọn điểm cụ thể trong số hai điểm được tạo cấu hình làm các lân cận này.

Fig.25-(a) minh họa ví dụ về việc chọn điểm thứ nhất xét về mã Morton trong quy trình chọn điểm đối với mỗi độ sâu. Tức là, đối với LoD3 đến LoD0, khi chọn, điểm cụ thể được chọn từ trong số các điểm được tạo cấu hình làm các lân cận, điểm thứ nhất được chọn từ các điểm lân cận được sắp xếp theo thứ tự mã Morton.

Fig.25-(b) minh họa ví dụ về việc chọn điểm thứ nhất hoặc điểm cuối cùng xét về mã Morton theo độ sâu trong quy trình chọn điểm. Cụ thể hơn, hình vẽ này minh họa ví dụ về việc chọn luân phiên các điểm tại các vị trí thứ nhất và cuối cùng xét về mã Morton tại các độ sâu liên tiếp trong quy trình chọn điểm. Nói cách khác, đối với LoD3, điểm thứ nhất có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD2, điểm cuối cùng có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD1, điểm thứ nhất có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận. Đối với LoD0, điểm cuối cùng có thể được chọn trong số các điểm được sắp xếp theo thứ tự mã Morton trong khoảng tìm kiếm lân cận.

Như trong trường hợp trên Fig.24-(a) và Fig.24-(b), có thể thấy rằng, ngược với Fig.25-(a), phân bố điểm trên Fig.25-(b) có thể được tạo cấu hình sao cho các điểm của LOD thấp hơn là gần hơn với tâm của nút.

Phương án 2

Khi việc lập mã dạng hình học được thực hiện bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, nút bất đối xứng (hoặc phỏng lập phương) có thể được sử dụng tại mỗi độ sâu của cây dạng hình học này. Trong trường hợp này, khi việc lập mã có thể định tỷ lệ được xem xét, khả năng định tỷ lệ được hỗ trợ trên cơ sở từng độ sâu một trong cây dạng hình học. Phương án 2 là ví dụ về việc thêm các ràng buộc để xem xét các đặc điểm của việc tạo ra LoD dựa trên cây bát phân trong việc lập mã dạng hình học. Tức là, khi nút lập phương được đảm bảo đối với khoảng độ sâu nhất định từ mức lá (cụ thể là, khi kích thước của nút đơn vị là giống nhau theo các hướng xyz), khả năng định tỷ lệ có thể được đảm bảo đối với khoảng độ sâu tương ứng bằng cách cho phép bộ mã hóa thuộc tính 51004 thực hiện việc tạo ra LoD dựa trên cây bát phân. Để đạt được mục đích này, trong sáng chế này, việc nén thuộc tính có thể được hạn chế để được thực hiện sử dụng chỉ LoD (các LoD) khi các độ sâu gần với nút lá là các nút đối xứng.

Fig.26 là sơ đồ minh họa ví dụ về việc tạo ra LoD tại một số độ sâu trong cấu trúc

cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D. Fig.26 minh họa ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hộp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, QT, BT, và QT theo thứ tự này dựa trên các tham số được đặt là $K=1$ và $M=1$. Trên Fig.26, giả sử rằng hai mức độ sâu đang chứa mức lá có nút lập phương. Trong trường hợp này, như là một phương án, bộ mã hóa thuộc tính 51004 có thể tạo ra các LoD chỉ tại hai mức độ sâu liên tiếp đang chứa mức lá.

Fig.27 là sơ đồ minh họa một ví dụ khác về việc tạo ra LoD tại một số độ sâu của cấu trúc dạng hình học trong đó dữ liệu đám mây điểm được phân bố trên mặt phẳng 2D được phân vùng. Fig.27 minh họa ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hộp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, BT, QT, và QT theo thứ tự này dựa trên các tham số được đặt là $K=2$ và $M=0$. Trên Fig.27, giả sử rằng ba mức độ sâu đang chứa mức lá có nút lập phương. Trong trường hợp này, như là một phương án, bộ mã hóa thuộc tính 51004 có thể tạo ra các LoD chỉ tại ba mức độ sâu liên tiếp đang chứa mức lá.

Trong sáng chế này, sự không khớp điểm giữa dạng hình học và các thuộc tính có thể được ngăn chặn trong việc trình chiếu có thể định tỷ lệ được bằng cách áp đặt ràng buộc giới hạn rằng LoD (các LoD) cần phải được tạo ra chỉ cho các độ sâu mà chúng khớp với phương pháp tạo ra LoD dựa trên QT (OT trong không gian 3D) trong cấu trúc dạng hình học được sử dụng cho việc nén dạng hình học.

Trong một phương án, thông tin (ví dụ, `max_num_LoD`) chỉ ra số lượng LOD tối đa và/hoặc thông tin độ sâu bắt đầu LoD (ví dụ, `LoD_starting_depth`) có thể được báo hiệu qua thông tin báo hiệu. Thông tin báo hiệu này có thể ít nhất là tập tham số thuộc tính hoặc tiêu đề lát thuộc tính.

Nói cách khác, số lượng LoD được sử dụng trong việc lập mã có thể định tỷ lệ được có thể được thông báo cho bộ giải mã thuộc tính 61004 của thiết bị nhận bằng thông tin báo hiệu (ví dụ, `max_num_LoD`) chỉ ra số lượng LOD tối đa, hoặc vị trí mà LoD0 bắt đầu tại đó có thể được báo hiệu cho bộ giải mã thuộc tính 61004 của thiết bị nhận bằng cách báo hiệu thông tin độ sâu bắt đầu LoD (ví dụ, `LoD_starting_depth`).

Theo các phương án, thông tin độ sâu bắt đầu LoD (ví dụ, `LoD_starting_depth`) có thể chỉ ra độ sâu LoD tối đa mà nó có thể được so khớp với việc tạo ra LoD dựa trên cây bát phân. Tức là, nó có thể chỉ ra số lượng độ sâu từ mức lá. Để làm ví dụ, trên Fig.26, số lượng độ sâu được chỉ ra bởi thông tin độ sâu bắt đầu LoD (ví dụ, `LoD_starting_depth`) là 2, mà nó là độ sâu tương ứng với LoD0. Để làm ví dụ, trên Fig.27, số lượng độ sâu khớp với việc tạo ra LoD dựa trên cây bát phân bắt đầu từ mức lá có thể là 3, mà nó là độ sâu tương ứng với LoD0.

Theo các phương án, thông tin (ví dụ, `max_num_LoD`) chỉ ra số lượng LOD tối đa và thông tin độ sâu bắt đầu LoD (ví dụ, `LoD_starting_depth`) có thể có các giá trị sau đây.

$$\text{max_num_LoD} = \text{max_dimension} - \text{min_dimension} - M$$

$$\text{LoD_starting_depth} = \text{max_dimension} - M$$

Thêm vào đó, ràng buộc rằng $K+M = \text{max_dimension} - \text{min_dimension}$ cần phải được xác lập khi việc lập mã dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003 có thể được thêm vào.

Theo các phương án, `max_dimension` biểu thị $\max(dx, dy, dz)$, và `min_dimension` biểu thị $\min(dx, dy, dz)$.

Phương án 3

Khi việc lập mã dạng hình học được thực hiện bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, nút bất đối xứng (hoặc phỏng lập phương) có thể được sử dụng tại mỗi độ sâu của cây dạng hình học này. Khi việc lập mã có thể định tỷ lệ được xem xét, khả năng định tỷ lệ được hỗ trợ trên cơ sở từng độ sâu một trong cây dạng hình học. Phương án 3 là ví dụ trong đó việc tạo ra LoD được bỏ qua tại độ sâu (các độ sâu) của cây dạng hình học nơi sự không khớp điểm giữa dạng hình học và các thuộc tính xảy ra hoặc LoD (các LoD) được tạo ra tại độ sâu (các độ sâu) này không được sử dụng trong việc nén thuộc tính, và thông tin (ví dụ, `LoD_skip_flag`) chỉ ra việc bỏ qua này được báo hiệu qua thông tin báo hiệu. Thông tin báo hiệu này có thể ít nhất là tập tham số thuộc tính hoặc tiêu đề lát thuộc tính.

Trong một phương án, việc tạo ra các LoD tương ứng với độ sâu (các độ sâu) của cây dạng hình học nơi sự không khớp điểm giữa dạng hình học và các thuộc tính xảy ra có thể được bỏ qua, và thông tin bỏ qua LoD (ví dụ, `LoD_skip_flag`) chỉ ra việc bỏ qua

này có thể được báo hiệu và được thông báo cho bộ giải mã thuộc tính 61004 của thiết bị nhận. Trong trường hợp này, trong một phương án, LoD (các LoD) có thể được tạo ra tại độ sâu (các độ sâu) còn lại ngoại trừ độ sâu (các độ sâu) của dạng hình học nơi sự không khớp xảy ra, khi xem xét đến nút bất đối xứng cho mỗi độ sâu. Ví dụ, thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) bằng 1 có thể chỉ ra rằng điểm là không khớp giữa dạng hình học và các thuộc tính tại độ sâu này của cây dạng hình học, và/hoặc rằng việc tạo ra LoD đã được bỏ qua tại độ sâu này. Thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) bằng 0 có thể chỉ ra rằng các điểm là khớp giữa dạng hình học và các thuộc tính tại độ sâu này, và/hoặc rằng LoD đã được tạo ra tại độ sâu này.

Trong một phương án khác, LoD (các LoD) được tạo ra tại độ sâu (các độ sâu) của cây dạng hình học nơi sự không khớp điểm giữa dạng hình học và các thuộc tính xảy ra có thể không được sử dụng cho việc nén thuộc tính, và thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) chỉ ra việc bỏ qua này có thể được báo hiệu và được thông báo cho bộ giải mã thuộc tính 61004 của thiết bị nhận. Trong trường hợp này, trong một phương án, việc nén thuộc tính có thể được thực hiện sử dụng LoD (các LoD) được tạo ra tại độ sâu (các độ sâu) còn lại ngoại trừ độ sâu (các độ sâu) của dạng hình học đang trải qua sự không khớp khi xem xét đến nút bất đối xứng cho mỗi độ sâu. Ví dụ, thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) bằng 1 có thể chỉ ra rằng điểm là không khớp giữa dạng hình học và các thuộc tính tại độ sâu này của cây dạng hình học, và/hoặc rằng các LoD được tạo ra tại độ sâu này không được sử dụng cho việc nén thuộc tính. Thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) bằng 0 có thể chỉ ra rằng các điểm là khớp giữa dạng hình học và các thuộc tính tại độ sâu này, và/hoặc rằng các LoD được tạo ra tại độ sâu này được sử dụng cho việc nén thuộc tính.

Tức là, khi giá trị của thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) là 1, LoD tương ứng với độ sâu này có thể không được sử dụng cho việc nén thuộc tính. Theo cách khác, LoD có thể được định ra cho độ sâu này, nhưng có thể không được sử dụng trong việc trình chiếu có thể định tỷ lệ được.

Theo các phương án, việc thiết lập thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) có thể tuân thủ theo các điều kiện sau đây.

if dx_ot=dx & dy_ot=dy & dz_ot=dz	LoD_skip_flag = 0
else	LoD_skip_flag = 1

Ở đây, dx_{ot} , dy_{ot} , và dz_{ot} có thể biểu thị kích thước nút trục xyz trong hộp giới hạn nơi các điểm thực sự được phân bố khi việc phân vùng dựa trên OT được sử dụng, và dx , dy , và dz có thể biểu thị kích thước nút trục xyz nơi việc phân vùng dựa trên QTBT được sử dụng.

Nói cách khác, thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) có thể được thiết lập là 0 chỉ nếu kích thước nút trục x (dx_{ot}) khi việc phân vùng dựa trên OT được sử dụng là giống với kích thước nút trục x (dx) trong hộp giới hạn khi việc phân vùng dựa trên QTBT được sử dụng, kích thước nút trục y (dy_{ot}) trong hộp giới hạn khi việc phân vùng dựa trên OT được sử dụng là giống với kích thước nút trục y (dy) khi việc phân vùng dựa trên QTBT được sử dụng, và kích thước nút trục z (dz_{ot}) trong hộp giới hạn khi việc phân vùng dựa trên OT được sử dụng là giống với kích thước nút trục z (dz) khi việc phân vùng dựa trên QTBT được sử dụng. Ngoài ra, nếu các kích thước này là không giống nhau trên trục bất kỳ trong số các trục xyz, thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) có thể được thiết lập là 1.

Nói cách khác, trong trường hợp bỏ qua LoD, cấu trúc phân cấp được cung cấp bởi việc lập mã có thể định tỷ lệ được, nhưng LoD này không được so khớp về mặt cấu trúc với dạng hình học. Theo đó, thông tin bỏ qua LoD (ví dụ, LoD_skip_flag) có thể được sử dụng để thông báo rằng việc xuất ra (hoặc sử dụng) LoD này cho việc nén thuộc tính là không thích hợp.

Fig.28-(a) và Fig.28(b) là các sơ đồ minh họa các ví dụ về thiết lập của việc bỏ qua LoD tùy theo các điều kiện được mô tả trên đây tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D. Fig.28-(a) và Fig.28-(b) minh họa các ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hộp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, QT, BT, và QT theo thứ tự này, dựa trên các tham số được đặt là $K=1$ và $M=1$. Fig.28(a) minh họa ví dụ trong đó việc bỏ qua LoD được thiết lập cho độ sâu thứ ba dựa trên nút gốc, và Fig.28(b) minh họa ví dụ trong đó việc bỏ qua LoD được thiết lập cho ba độ sâu dựa trên nút gốc.

Theo các phương án, trên Fig.28-(a), $LoD0$ và $LoD1$ thể hiện trường hợp nơi một nút phỏng lập phương được chứa trong nút lập phương. Trong trường hợp này, các LoD

có thể được thiết lập để không được bỏ qua bởi vì chúng khớp với dạng hình học.

Fig.29 là sơ đồ minh họa một ví dụ khác về thiết lập của việc bỏ qua LoD tùy theo các điều kiện được mô tả trên đây tại một số độ sâu trong cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D. Fig.29 minh họa ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hợp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, BT, QT, và QT theo thứ tự này dựa trên các tham số được đặt là $K=2$ và $M=0$. Trong ví dụ trên Fig.29, việc bỏ qua LoD được thiết lập đối với hai độ sâu.

Fig.30 là sơ đồ minh họa một ví dụ khác về việc tạo ra LoD cho mỗi độ sâu dựa trên cấu trúc cây dạng hình học được sử dụng để phân vùng dữ liệu đám mây điểm được phân bố trong mặt phẳng 2D. Fig.30 minh họa ví dụ trong đó QT/BT được sử dụng cho việc phân vùng hợp giới hạn khi việc mã hóa dạng hình học được thực hiện bởi bộ mã hóa dạng hình học 51003, và cây dạng hình học được tạo cấu hình theo BT, BT, QT, và QT theo thứ tự này dựa trên các tham số được đặt là $K=2$ và $M=0$. Theo các phương án, bộ mã hóa thuộc tính 51004 có thể thực hiện việc tạo ra LoD dựa trên cấu trúc cây dạng hình học có nút phi lập phương trên Fig.30.

Theo các phương án, thông tin bỏ qua LoD (ví dụ, `LoD_skip_flag`) cũng có thể được áp dụng cho cấu trúc cây dạng hình học như được thể hiện trên Fig.30. Tức là, có thể thấy rằng mỗi độ sâu của dạng hình học trên Fig.30 có cùng kích thước nút như kích thước nút trong phương pháp tạo ra LoD dựa trên QT. Do đó, khi đặc điểm này được nhận dạng từ trước bởi bộ mã hóa thuộc tính 51004, thông tin bỏ qua LoD (ví dụ, `LoD_skip_flag`) có thể được báo hiệu sao cho thông tin bỏ qua LoD (ví dụ, `LoD_skip_flag`)=0 được thiết lập cho tất cả các LoD.

Theo các phương án, bộ mã hóa dạng hình học 51003 và/hoặc bộ mã hóa thuộc tính 51004 của thiết bị truyền có thể thực hiện việc nén thông tin dạng hình học và/hoặc việc nén thông tin thuộc tính hoặc thực hiện việc mã hóa thuộc tính có thể định tỷ lệ được bằng cách kết hợp một hoặc nhiều phương án trong số các phương án 1 đến 3 được mô tả trên đây. Thêm vào đó, bộ giải mã dạng hình học 61003 và/hoặc bộ giải mã thuộc tính 61004 của thiết bị nhận có thể thực hiện việc xây dựng lại thông tin dạng hình học và/hoặc xây dựng lại thông tin thuộc tính hoặc thực hiện việc giải mã thuộc tính có thể

định tỷ lệ được bằng cách kết hợp một hoặc nhiều phương án trong số các phương án 1 đến 3 được mô tả trên đây.

Tiếp theo, thông tin báo hiệu để báo hiệu thông tin liên quan đến việc nén thuộc tính dựa trên LoD được mô tả trên đây cấu trúc luồng bit chứa thông tin báo hiệu này sẽ được mô tả. Theo các phương án, thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể bao gồm ít nhất một thông tin trong số thông tin để nhận dạng phương pháp lấy mẫu LoD (ví dụ, `LoD_sampling_method`), thông tin chỉ ra số lượng LoD tối đa (ví dụ, `max_num_LoD`), thông tin độ sâu bắt đầu LoD (ví dụ, `LoL_starting_depth`), hoặc thông tin bỏ qua LoD (ví dụ, `LoD_skip_flag`). Thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể được gọi là thông tin liên quan đến việc mã hóa thuộc tính dựa trên LoD.

Theo các cấu hình, khi thông tin dạng hình học được mã hóa bởi bộ mã hóa video đám mây điểm, kết quả là luồng bit dạng hình học được tạo ra. Khi thông tin thuộc tính được mã hóa, kết quả là luồng bit thuộc tính được tạo ra. Luồng bit dạng hình học và luồng bit thuộc tính theo các phương án có thể được dồn kênh và xuất ra dưới dạng một luồng bit. Luồng bit theo các phương án có thể còn bao gồm thông tin báo hiệu liên quan đến việc mã hóa dạng hình học và việc mã hóa thuộc tính. Trong một phương án, thông tin báo hiệu có thể bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD.

Fig.31 minh họa ví dụ về cấu trúc luồng bit của dữ liệu đám mây điểm dành cho việc truyền/nhận theo các phương án.

Khi luồng bit dạng hình học, luồng bit thuộc tính, và/hoặc luồng bit báo hiệu (hoặc thông tin báo hiệu) theo các phương án được tạo cấu hình thành một luồng bit (hoặc luồng bit G-PCC), luồng bit này có thể bao gồm một hoặc nhiều luồng bit con. Luồng bit theo các phương án có thể bao gồm tập tham số trình tự (SPS) cho việc báo hiệu mức trình tự, tập tham số dạng hình học (GPS) cho việc báo hiệu việc lập mã thông tin dạng hình học, một hoặc nhiều tập tham số thuộc tính (APS) (APS_0 , APS_1) cho việc báo hiệu việc lập mã thông tin thuộc tính, tập tham số ô xếp (TPS hoặc bản kiểm kê ô xếp) cho việc báo hiệu mức ô xếp, và một hoặc nhiều lát (lát 0 đến lát n). Tức là, luồng bit của dữ liệu đám mây điểm theo các phương án có thể bao gồm một hoặc nhiều ô xếp, và mỗi ô xếp có thể là nhóm các lát chứa một hoặc nhiều lát (lát 0 đến lát n). Bản kiểm kê ô xếp (hoặc TPS) theo các phương án có thể chứa thông tin về mỗi ô xếp trong số một

hoặc nhiều ô xếp này (ví dụ, thông tin giá trị tọa độ và thông tin chiều cao/kích thước về hộp giới hạn ô xếp). Mỗi lát có thể bao gồm một luồng bit dạng hình học (Geom0) và một hoặc nhiều luồng bit thuộc tính (Attr0 và Attr1). Ví dụ, lát thứ nhất (lát 0) có thể bao gồm luồng bit dạng hình học (Geom0⁰) và một hoặc nhiều luồng bit thuộc tính (Attr0⁰, Attr1⁰).

Luồng bit dạng hình học trong mỗi lát có thể gồm tiêu đề lát dạng hình học (geom_slice_header) và dữ liệu lát dạng hình học (geom_slice_data). Theo các phương án, geom_slice_header có thể bao gồm thông tin nhận dạng (geom_parameter_set_id), bộ nhận dạng ô xếp (geom_tile_id), và bộ nhận dạng lát (geom_slice_id) cho tập tham số được chứa trong GPS, và thông tin (geomBoxOrigin, geom_box_log2_scale, geom_max_node_size_log2, geom_num_points) về dữ liệu được chứa trong dữ liệu lát dạng hình học (geom_slice_data). geomBoxOrigin là thông tin gốc hộp dạng hình học chỉ ra gốc của hộp của dữ liệu lát dạng hình học, geom_box_log2_scale là thông tin chỉ ra thang loga của dữ liệu lát dạng hình học, geom_max_node_size_log2 là thông tin chỉ ra kích thước của nút cây bát phân dạng hình học ở gốc rễ, và geom_num_points là thông tin liên quan đến số lượng điểm của dữ liệu lát dạng hình học. Theo các phương án, geom_slice_data có thể bao gồm thông tin dạng hình học (hoặc dữ liệu dạng hình học) về dữ liệu đám mây điểm trong lát tương ứng.

Mỗi luồng bit thuộc tính trong mỗi lát có thể gồm tiêu đề lát thuộc tính (attr_slice_header) và dữ liệu lát thuộc tính (attr_slice_data). Theo các phương án, attr_slice_header có thể bao gồm thông tin về dữ liệu lát thuộc tính tương ứng. Dữ liệu lát thuộc tính có thể chứa thông tin thuộc tính (hoặc dữ liệu thuộc tính) về dữ liệu đám mây điểm trong lát tương ứng. Khi có nhiều luồng bit thuộc tính trong một lát, mỗi luồng bit trong số các luồng bit này có thể chứa thông tin thuộc tính khác nhau. Ví dụ, một luồng bit thuộc tính có thể chứa thông tin thuộc tính tương ứng với màu, và một luồng bit thuộc tính khác có thể chứa thông tin thuộc tính tương ứng với độ phản xạ.

Fig.32 minh họa cấu trúc luồng bit làm ví dụ dành cho dữ liệu đám mây điểm dành cho việc truyền/nhận theo các phương án.

Fig.33 minh họa mối quan hệ kết nối giữa các phần tử trong luồng bit của dữ liệu đám mây điểm theo các phương án.

Cấu trúc luồng bit cho dữ liệu đám mây điểm được minh họa trên Fig.32 và Fig.33 có thể thể hiện cấu trúc luồng bit cho dữ liệu đám mây điểm được thể hiện trên Fig.31.

Theo các phương án, SPS có thể bao gồm bộ nhận dạng (`seq_parameter_set_id`) để nhận dạng SPS này, và GPS có thể bao gồm bộ nhận dạng (`geom_parameter_set_id`) để nhận dạng GPS này và bộ nhận dạng (`seq_parameter_set_id`) chỉ ra SPS hoạt động mà GPS này thuộc về nó. APS có thể bao gồm bộ nhận dạng (`attr_parameter_set_id`) để nhận dạng APS này và bộ nhận dạng (`seq_parameter_set_id`) chỉ ra SPS hoạt động mà APS này thuộc về nó. Theo các phương án, dữ liệu dạng hình học có thể bao gồm tiêu đề lát dạng hình học và dữ liệu lát dạng hình học. Tiêu đề lát dạng hình học có thể bao gồm bộ nhận dạng (`geom_parameter_set_id`) của GPS hoạt động sẽ được tham chiếu đến bởi lát dạng hình học tương ứng. Hơn nữa, tiêu đề lát dạng hình học có thể còn bao gồm bộ nhận dạng (`geom_slice_id`) để nhận dạng lát dạng hình học tương ứng và/hoặc bộ nhận dạng (`geom_tile_id`) để nhận dạng ô xếp tương ứng. Dữ liệu lát dạng hình học có thể bao gồm luồng bit dạng hình học thuộc về lát tương ứng. Theo các phương án, dữ liệu thuộc tính có thể bao gồm tiêu đề lát thuộc tính và dữ liệu lát thuộc tính. Tiêu đề lát thuộc tính có thể bao gồm bộ nhận dạng (`attr_parameter_set_id`) của APS hoạt động sẽ được tham chiếu đến bởi lát thuộc tính tương ứng và bộ nhận dạng (`geom_slice_id`) để nhận dạng lát dạng hình học liên quan đến lát thuộc tính này. Dữ liệu lát thuộc tính có thể bao gồm luồng bit thuộc tính thuộc về lát tương ứng.

Tức là, lát dạng hình học tham chiếu đến GPS, và GPS tham chiếu đến SPS. Thêm vào đó, SPS liệt kê các thuộc tính khả dụng, ấn định bộ nhận dạng cho mỗi thuộc tính, và nhận dạng phương pháp giải mã. Lát thuộc tính được ánh xạ để xuất ra các thuộc tính theo bộ nhận dạng. Lát thuộc tính có sự phụ thuộc vào lát dạng hình học (được giải mã) đứng trước và APS. APS tham chiếu đến SPS.

Theo các phương án, các tham số cần thiết cho việc mã hóa dữ liệu đám mây điểm có thể được định mới ra trong tập tham số của dữ liệu đám mây điểm và/hoặc tiêu đề lát tương ứng. Ví dụ, khi việc mã hóa thông tin thuộc tính được thực hiện, các tham số có thể được thêm vào APS. Khi việc mã hóa dựa trên ô xếp được thực hiện, các tham số có thể được thêm vào tiêu đề ô xếp và/hoặc lát

Như được thể hiện trên Fig.31 đến Fig.33, luồng bit của dữ liệu đám mây điểm cung cấp các ô xếp hoặc các lát sao cho dữ liệu đám mây điểm này có thể được phân

vùng và được xử lý theo các vùng. Theo các phương án, các vùng tương ứng của luồng bit có thể có các tầm quan trọng khác nhau. Theo đó, khi dữ liệu đám mây điểm được phân vùng thành các ô xếp, bộ lọc (phương pháp mã hóa) khác nhau và đơn vị lọc khác nhau có thể được áp dụng cho mỗi ô xếp. Khi dữ liệu đám mây điểm được phân vùng thành các lát, bộ lọc khác nhau và đơn vị lọc khác nhau có thể được áp dụng cho mỗi chỗ lát.

Khi dữ liệu đám mây điểm được phân vùng và được nén, thiết bị truyền và thiết bị nhận theo các phương án có thể truyền và nhận luồng bit trong cấu trúc cú pháp mức cao để truyền có chọn lọc thông tin thuộc tính trong các vùng được phân vùng.

Thiết bị truyền theo các phương án có thể truyền dữ liệu đám mây điểm theo cấu trúc luồng bit như được thể hiện trên Fig.31 đến Fig.33. Theo đó, phương pháp áp dụng các hoạt động mã hóa khác nhau và sử dụng phương pháp mã hóa chất lượng tốt cho vùng quan trọng có thể được đề xuất. Thêm vào đó, việc mã hóa và truyền có hiệu quả có thể được hỗ trợ tùy theo các đặc điểm của dữ liệu đám mây điểm, và các giá trị thuộc tính có thể được cung cấp tùy theo các yêu cầu của người dùng.

Thiết bị nhận theo các phương án có thể nhận dữ liệu đám mây điểm theo cấu trúc luồng bit như được thể hiện trên Fig.31 đến Fig.33. Theo đó, các phương pháp lọc (giải mã) khác nhau có thể được áp dụng cho các vùng tương ứng (các vùng được phân vùng thành các ô xếp hoặc thành các lát), thay vì phương pháp giải mã (lọc) phức tạp đang được áp dụng cho toàn bộ dữ liệu đám mây điểm. Do đó, chất lượng ảnh tốt hơn trong vùng quan trọng được cung cấp cho người dùng và độ trễ thích hợp đối với hệ thống được đảm bảo.

Trường, mà nó là thuật ngữ được sử dụng trong các cú pháp của sáng chế được mô tả sau, có thể có cùng ý nghĩa như tham số hoặc phần tử.

Fig.34 thể hiện phương án về cấu trúc cú pháp của tập tham số trình tự (SPS) (`seq_parameter_set()`) theo sáng chế này. SPS có thể chứa thông tin trình tự về luồng bit dữ liệu đám mây điểm.

SPS theo các phương án có thể bao gồm trường `main_profile_compatibility_flag`, trường `unique_point_positions_constraint_flag`, trường `level_idc`, trường `sps_seq_parameter_set_id`, trường `sps_bounding_box_present_flag`, trường

sps_source_scale_factor_numerator_minus1, trường
sps_source_scale_factor_denominator_minus1, trường sps_num_attribute_sets, trường
log2_max_frame_idx, trường axis_coding_order, trường
sps_bypass_stream_enabled_flag, và trường sps_extension_flag.

Trường main_profile_compatibility_flag có thể chỉ ra việc liệu luồng bit có tuân thủ theo hồ sơ (profile) chính hay không. Ví dụ, main_profile_compatibility_flag bằng 1 có thể chỉ ra rằng luồng bit có tuân thủ theo hồ sơ chính. Ví dụ, main_profile_compatibility_flag bằng 0 có thể chỉ ra rằng luồng bit tuân thủ theo hồ sơ khác hồ sơ chính.

Khi unique_point_positions_constraint_flag bằng 1, trong mỗi khung đám mây điểm mà nó được tham chiếu đến bởi SPS hiện tại, tất cả các điểm đầu ra đều có thể có những vị trí độc nhất. Khi unique_point_positions_constraint_flag bằng 0, trong khung đám mây điểm bất kỳ mà nó được tham chiếu đến bởi SPS hiện tại, hai hay hơn hai điểm đầu ra có thể có cùng vị trí. Ví dụ, kể cả khi tất cả các điểm đều là độc nhất trong các lát tương ứng, các lát trong khung và các điểm khác có thể chồng lặp. Trong trường hợp này, unique_point_positions_constraint_flag được đặt là 0.

level_idc chỉ ra mức mà luồng bit tuân thủ theo nó.

sps_seq_parameter_set_id cung cấp bộ nhận dạng cho SPS cho việc tham chiếu bởi các phân tử cú pháp khác.

Trường sps_bounding_box_present_flag chỉ ra liệu hộp giới hạn có có mặt trong SPS hay không. Ví dụ, sps_bounding_box_present_flag bằng 1 chỉ ra rằng hộp giới hạn có có mặt trong SPS, và sps_bounding_box_present_flag bằng 0 chỉ ra rằng kích thước của hộp giới hạn là không được định rõ.

Theo các phương án, khi sps_bounding_box_present_flag bằng 1, SPS có thể còn bao gồm trường sps_bounding_box_offset_x, trường sps_bounding_box_offset_y, trường sps_bounding_box_offset_z, trường sps_bounding_box_offset_log2_scale, trường sps_bounding_box_size_width, trường sps_bounding_box_size_height, và trường sps_bounding_box_size_depth.

sps_bounding_box_offset_x chỉ ra phần bù (offset) x của hộp giới hạn nguồn theo các tọa độ Đề các (Cartesian). Khi phần bù x của hộp giới hạn nguồn không có mặt, giá trị của sps_bounding_box_offset_x là 0.

`sps_bounding_box_offset_y` chỉ ra phần bù y của hộp giới hạn nguồn theo các tọa độ Đề các. Khi phần bù y của hộp giới hạn nguồn không có mặt, giá trị của `sps_bounding_box_offset_y` là 0.

`sps_bounding_box_offset_z` chỉ ra phần bù z của hộp giới hạn nguồn theo các tọa độ Đề các. Khi phần bù z của hộp giới hạn nguồn không có mặt, giá trị của `sps_bounding_box_offset_z` là 0.

`sps_bounding_box_offset_log2_scale` chỉ ra hệ số thang chia tỷ lệ để định tỷ lệ cho các phần bù hộp giới hạn nguồn x, y, và z được lượng tử hoá.

`sps_bounding_box_size_width` chỉ ra chiều rộng của hộp giới hạn nguồn theo các tọa độ Đề các. Khi chiều rộng của hộp giới hạn nguồn không có mặt, giá trị của `sps_bounding_box_size_width` có thể là 1.

`sps_bounding_box_size_height` chỉ ra chiều cao của hộp giới hạn nguồn theo các tọa độ Đề các. Khi chiều cao của hộp giới hạn nguồn không có mặt, giá trị của `sps_bounding_box_size_height` có thể là 1.

`sps_bounding_box_size_depth` chỉ ra chiều sâu của hộp giới hạn nguồn theo các tọa độ Đề các. Khi chiều sâu của hộp giới hạn nguồn không có mặt, giá trị của `sps_bounding_box_size_depth` có thể là 1.

`sps_source_scale_factor_numerator_minus1` cộng 1 chỉ ra tử số hệ số thang chia tỷ lệ của đám mây điểm nguồn.

`sps_source_scale_factor_denominator_minus1` cộng 1 chỉ ra mẫu số hệ số thang chia tỷ lệ của đám mây điểm nguồn.

`sps_num_attribute_sets` chỉ ra số lượng thuộc tính được lập mã trong luồng bit.

SPS theo các phương án bao gồm câu lệnh lặp lại được lặp lại với số lần nhiều bằng giá trị của trường `sps_num_attribute_sets`. Trong một phương án, *i* được khởi tạo là 0, và được tăng lên 1 mỗi lần câu lệnh lặp lại được thực thi. Câu lệnh lặp lại này được lặp lại cho đến khi giá trị của *i* trở thành bằng với giá trị của trường `sps_num_attribute_sets`. Câu lệnh lặp lại có thể bao gồm trường `attribute_dimension_minus1[i]` và trường `attribute_instance_id[i]`. `attribute_dimension_minus1[i]` cộng 1 chỉ ra số lượng thành phần của thuộc tính thứ *i*.

Trường `attribute_instance_id[i]` chỉ rõ ID của thực thể (instance) của thuộc tính thứ *i*.

Theo các phương án, khi giá trị của trường `attribute_dimension_minus1[i]` lớn hơn 1, câu lệnh lặp lại có thể còn bao gồm trường `attribute_secondary_bitdepth_minus1[i]`, trường `attribute_cicp_colour_primaries[i]`, trường `attribute_cicp_transfer_characteristics[i]`, trường `attribute_cicp_matrix_coeffs[i]`, và trường `attribute_cicp_video_full_range_flag[i]`.

`attribute_secondary_bitdepth_minus1[i]` cộng 1 chỉ rõ độ sâu bit cho thành phần thứ cấp của tín hiệu (các tín hiệu) thuộc tính thứ *i*.

`attribute_cicp_colour_primaries[i]` chỉ ra các tọa độ sắc độ của các nguyên tố (primary) nguồn thuộc tính màu của thuộc tính thứ *i*.

`attribute_cicp_transfer_characteristics[i]` hoặc là chỉ ra hàm chuyển đặc điểm quang-điện tử tham chiếu của thuộc tính màu dưới dạng hàm của cường độ quang học tuyến tính đầu vào nguồn với khoảng giá trị thực danh nghĩa là 0 đến 1 hoặc là chỉ ra ngược của hàm chuyển đặc điểm quang-điện tử tham chiếu dưới dạng hàm của cường độ quang học tuyến tính đầu ra.

`attribute_cicp_matrix_coeffs[i]` mô tả các hệ số ma trận được sử dụng trong việc dẫn ra các tín hiệu độ chói và sắc độ từ các nguyên tố xanh lục, xanh lam, và đỏ, hoặc Y, Z, và X của thuộc tính thứ *i*.

`attribute_cicp_video_full_range_flag[i]` chỉ rõ mức đen và khoảng của các tín hiệu độ chói và sắc độ như được dẫn ra từ các tín hiệu thành phần được định giá trị thực E'Y, E'PB, và E'PR hoặc E'R, E'G, và E'B của thuộc tính thứ *i*.

`known_attribute_label_flag[i]` chỉ ra việc liệu trường `known_attribute_label[i]` hay trường `attribute_label_four_bytes[i]` được báo hiệu đối với thuộc tính thứ *i*. Ví dụ, `known_attribute_label_flag[i]` bằng 0 chỉ ra trường `known_attribute_label[i]` được báo hiệu đối với thuộc tính thứ *i*. `known_attribute_label_flag[i]` bằng 1 chỉ ra rằng trường `attribute_label_four_bytes[i]` được báo hiệu đối với thuộc tính thứ *i*.

`known_attribute_label[i]` chỉ rõ loại của thuộc tính thứ *i*. Ví dụ, `known_attribute_label[i]` bằng 0 có thể chỉ rõ rằng thuộc tính thứ *i* là màu. `known_attribute_label[i]` bằng 1 có thể chỉ rõ rằng thuộc tính thứ *i* là độ phản xạ. `known_attribute_label[i]` bằng 2 có thể chỉ rõ rằng thuộc tính thứ *i* là chỉ số khung. Ngoài ra, `known_attribute_label[i]` bằng 4 chỉ rõ rằng thuộc tính thứ *i* là độ trong suốt. `known_attribute_label[i]` bằng 5 chỉ rõ rằng thuộc tính thứ *i* là các pháp tuyến.

`attribute_label_four_bytes[i]` chỉ ra loại thuộc tính đã biết bằng mã 4 byte.

Theo các phương án, `attribute_label_four_bytes[i]` bằng 0 có thể chỉ ra rằng thuộc tính thứ *i* là màu. `attribute_label_four_bytes[i]` bằng 1 có thể chỉ ra rằng thuộc tính thứ *i* là độ phản xạ. `attribute_label_four_bytes[i]` bằng 2 có thể chỉ ra rằng thuộc tính thứ *i* là chỉ số khung. `attribute_label_four_bytes[i]` bằng 4 có thể chỉ ra rằng thuộc tính thứ *i* là độ trong suốt. `attribute_label_four_bytes[i]` bằng 5 có thể chỉ ra rằng thuộc tính thứ *i* là các pháp tuyến.

`log2_max_frame_idx` chỉ ra số lượng bit được sử dụng để báo hiệu biến số cú pháp `frame_idx`.

`axis_coding_order` chỉ rõ sự tương ứng giữa các nhãn trục đầu ra X, Y, và Z và ba thành phần vị trí trong đám mây điểm được xây dựng lại `RecPic [pointidx] [axis]` với `axis=0..2`.

`sps_bypass_stream_enabled_flag` bằng 1 chỉ rõ rằng chế độ lập mã đi vòng có thể được sử dụng khi đọc luồng bit. Như là một ví dụ khác, `sps_bypass_stream_enabled_flag` bằng 0 chỉ rõ rằng chế độ lập mã đi vòng không được sử dụng khi đọc luồng bit.

`sps_extension_flag` chỉ rõ liệu cấu trúc cú pháp `sps_extension_data` có có mặt trong cấu trúc cú pháp SPS hay không. Ví dụ, `sps_extension_present_flag` bằng 1 chỉ ra rằng cấu trúc cú pháp `sps_extension_data` có mặt trong cấu trúc cú pháp SPS. `sps_extension_present_flag` bằng 0 chỉ ra rằng cấu trúc cú pháp này không có mặt.

Khi giá trị của trường `sps_extension_flag` bằng 1, SPS theo các phương án có thể còn bao gồm trường `sps_extension_data_flag`.

`sps_extension_data_flag` có thể có giá trị bất kỳ.

Fig.35 thể hiện phương án về cấu trúc cú pháp của GPS (`geometry_parameter_set()`) theo sáng chế này. GPS có thể bao gồm thông tin về phương pháp mã hóa thông tin dạng hình học của dữ liệu đám mây điểm được chứa trong một hoặc nhiều lát.

Theo các phương án, GPS có thể bao gồm trường `gps_geom_parameter_set_id`, trường `gps_seq_parameter_set_id`, trường `gps_box_present_flag`, trường `unique_geometry_points_flag`, trường `geometry_planar_mode_flag`, trường `geometry_angular_mode_flag`, trường `neighbour_context_restriction_flag`, trường `inferred_direct_coding_mode_enabled_flag`, trường `bitwise_occupancy_coding_flag`,

trường `adjacent_child_contextualization_enabled_flag`, trường `log2_neighbour_avail_boundary`, trường `log2_intra_pred_max_node_size`, trường `log2_trisoup_node_size`, trường `geom_scaling_enabled_flag`, trường `gps_implicit_geom_partition_flag`, và trường `gps_extension_flag`.

Trường `gps_geom_parameter_set_id` cung cấp bộ nhận dạng cho GPS cho việc tham chiếu bởi các phần tử cú pháp khác.

Trường `gps_seq_parameter_set_id` chỉ rõ giá trị của `sps_seq_parameter_set_id` đối với SPS hoạt động.

Trường `gps_box_present_flag` chỉ rõ liệu thông tin hộp giới hạn bổ sung có được cung cấp trong tiêu đề lát dạng hình học mà nó tham chiếu đến GPS hiện tại hay không. Ví dụ, trường `gps_box_present_flag` bằng 1 có thể chỉ rõ rằng thông tin hộp giới hạn bổ sung được cung cấp trong tiêu đề lát dạng hình học mà nó tham chiếu GPS hiện tại. Theo đó, khi trường `gps_box_present_flag` bằng 1, GPS có thể còn bao gồm trường `gps_gsh_box_log2_scale_present_flag`.

Trường `gps_gsh_box_log2_scale_present_flag` chỉ rõ liệu trường `gps_gsh_box_log2_scale` có được báo hiệu trong mỗi tiêu đề lát dạng hình học mà nó tham chiếu đến GPS hiện tại hay không. Ví dụ, trường `gps_gsh_box_log2_scale_present_flag` bằng 1 có thể chỉ rõ rằng trường `gps_gsh_box_log2_scale` được báo hiệu trong mỗi tiêu đề lát dạng hình học mà nó tham chiếu đến GPS hiện tại. Như là một ví dụ khác, trường `gps_gsh_box_log2_scale_present_flag` bằng 0 có thể chỉ rõ rằng trường `gps_gsh_box_log2_scale` không được báo hiệu trong mỗi tiêu đề lát dạng hình học và thang chia tỷ lệ chung đối với tất cả các lát được báo hiệu trong trường `gps_gsh_box_log2_scale` của GPS hiện tại.

Khi trường `gps_gsh_box_log2_scale_present_flag` bằng 0, GPS có thể còn bao gồm trường `gps_gsh_box_log2_scale`.

Trường `gps_gsh_box_log2_scale` chỉ ra hệ số thang chia tỷ lệ chung của góc hộp giới hạn cho tất cả các lát mà chúng tham chiếu đến GPS hiện tại.

`unique_geometry_points_flag` chỉ ra liệu tất cả các điểm đầu ra có có các vị trí độc nhất trong một lát trong tất cả các lát hiện đang tham chiếu đến GPS hay không. Ví dụ, `unique_geometry_points_flag` bằng 1 chỉ ra rằng trong tất cả các lát mà chúng tham

chiều đến GPS hiện tại, tất cả các điểm đầu ra đều có các vị trí độc nhất bên trong một lát. Trường `unique_geometry_points_flag` bằng 0 chỉ ra rằng trong tất cả các lát mà chúng tham chiếu đến GPS hiện tại, hai hay hơn hai điểm đầu ra có thể cùng vị trí bên trong một lát.

Trường `geometry_planar_mode_flag` chỉ ra liệu chế độ lập mã phẳng có được kích hoạt hay không. Ví dụ, `geometry_planar_mode_flag` bằng 1 chỉ ra rằng chế độ lập mã phẳng đang hoạt động. `geometry_planar_mode_flag` bằng 0 chỉ ra rằng chế độ lập mã phẳng đang không hoạt động.

Khi giá trị của trường `geometry_planar_mode_flag` là 1, tức là, đúng (TRUE), GPS có thể còn bao gồm trường `geom_planar_mode_th_idcm`, trường `geom_planar_mode_th[1]`, và trường `geom_planar_mode_th[2]`.

Trường `geom_planar_mode_th_idcm` có thể chỉ rõ giá trị của ngưỡng của việc kích hoạt đối với chế độ lập mã trực tiếp.

`geom_planar_mode_th[i]` chỉ rõ, với i nằm trong khoảng $0 \dots 2$, giá trị của ngưỡng của việc kích hoạt đối với chế độ lập mã phẳng theo hướng khả dĩ nhất thứ i để chế độ lập mã phẳng được hữu hiệu.

`geometry_angular_mode_flag` chỉ ra liệu chế độ lập mã góc có đang hoạt động hay không. Ví dụ, trường `geometry_angular_mode_flag` bằng 1 có thể chỉ ra rằng chế độ lập mã góc đang hoạt động. Trường `geometry_angular_mode_flag` bằng 0 có thể chỉ ra rằng chế độ lập mã góc đang không hoạt động.

Khi giá trị của trường `geometry_angular_mode_flag` là 1, tức là, đúng (TRUE), GPS có thể còn bao gồm trường `lidar_head_position[0]`, trường `lidar_head_position[1]`, trường `lidar_head_position[2]`, trường `number_lasers`, trường `planar_buffer_disabled`, trường `implicit_qtbt_angular_max_node_min_dim_log2_to_split_z`, và trường `implicit_qtbt_angular_max_diff_to_split_z`.

Trường `lidar_head_position[0]`, trường `lidar_head_position[1]`, và trường `lidar_head_position[2]` có thể chỉ rõ các tọa độ (X, Y, Z) của đầu lidar trong hệ tọa độ với các trục bên trong.

`number_lasers` chỉ rõ số lượng laze được sử dụng cho chế độ lập mã góc.

GPS theo các phương án bao gồm câu lệnh lặp lại được lặp lại với số lần nhiều bằng giá trị của trường `number_lasers`. Trong một phương án, i được khởi tạo là 0, và

được tăng lên 1 mỗi lần câu lệnh lặp lại được thực thi. Câu lệnh lặp lại này được lặp lại cho đến khi giá trị của *i* trở thành bằng với giá trị của trường `number_lasers`. Câu lệnh lặp lại này có thể bao gồm trường `laser_angle[i]` và trường `laser_correction[i]`.

`laser_angle[i]` chỉ rõ tang của góc nâng của laze thứ *i* so với mặt phẳng ngang được định ra bởi các trục bên trong thứ 0 và thứ 1.

`laser_correction[i]` chỉ rõ việc hiệu chỉnh, dọc theo trục bên trong thứ hai, về vị trí laze thứ *i* so với `lidar_head_position[2]`.

`planar_buffer_disabled` bằng 1 chỉ ra rằng việc theo dõi các nút gần nhất sử dụng bộ đệm không được sử dụng trong quy trình lập mã cờ chế độ phẳng và vị trí mặt phẳng trong chế độ phẳng. `planar_buffer_disabled` bằng 0 chỉ ra rằng việc theo dõi các nút gần nhất sử dụng bộ đệm có được sử dụng.

`implicit_qtbt_angular_max_node_min_dim_log2_to_split_z` chỉ rõ giá trị log2 của kích thước nút mà dưới đó thì việc phân chia theo phương ngang các nút được ưu tiên so với việc phân chia theo phương dọc.

`implicit_qtbt_angular_max_diff_to_split_z` chỉ rõ giá trị log2 của tỷ lệ kích thước nút theo phương dọc trên kích thước nút theo phương ngang tối đa được phép đối với nút.

`neighbour_context_restriction_flag` bằng 0 chỉ ra rằng việc chiếm giữ nút dạng hình học của nút hiện tại được lập mã bằng các ngữ cảnh được xác định từ các nút lân cận mà chúng được đặt bên trong nút mẹ của nút hiện tại. `neighbour_context_restriction_flag` bằng 1 chỉ ra rằng việc chiếm giữ nút dạng hình học của nút hiện tại được lập mã bằng các ngữ cảnh được xác định từ các nút lân cận mà chúng được đặt bên trong hoặc bên ngoài nút mẹ của nút hiện tại.

Trường `inferred_direct_coding_mode_enabled_flag` chỉ ra liệu trường `direct_mode_flag` có có mặt trong cú pháp nút dạng hình học hay không. Ví dụ, trường `inferred_direct_coding_mode_enabled_flag` bằng 1 chỉ ra rằng trường `direct_mode_flag` có thể có mặt trong cú pháp nút dạng hình học. Ví dụ, trường `inferred_direct_coding_mode_enabled_flag` bằng 0 chỉ ra rằng trường `direct_mode_flag` không có mặt trong cú pháp nút dạng hình học.

Trường `bitwise_occupancy_coding_flag` chỉ ra liệu việc chiếm giữ nút dạng hình học có được mã hóa nhờ sử dụng việc ngữ cảnh hóa từng bit (bitwise) của bản đồ chiếm

giữ phần tử cú pháp hay không. Ví dụ, trường `bitwise_occupancy_coding_flag` bằng 1 chỉ ra rằng việc chiếm giữ nút dạng hình học được mã hóa nhờ sử dụng việc ngữ cảnh hóa từng bit của phần tử cú pháp `occupancy_map`. Ví dụ, trường `bitwise_occupancy_coding_flag` bằng 0 chỉ ra rằng việc chiếm giữ nút dạng hình học được mã hóa nhờ sử dụng phần tử cú pháp được mã hóa từ điển `occupancy_byte`.

Trường `adjacent_child_contextualization_enabled_flag` chỉ ra liệu con liền kề của các nút cây bát phân lân cận có được sử dụng cho việc ngữ cảnh hóa chiếm giữ từng bit hay không. Ví dụ, trường `adjacent_child_contextualization_enabled_flag` bằng 1 chỉ ra rằng con liền kề của các nút cây bát phân lân cận được sử dụng cho việc ngữ cảnh hóa chiếm giữ từng bit. Ví dụ, trường `adjacent_child_contextualization_enabled_flag` bằng 0 chỉ ra rằng con của các nút cây bát phân lân cận không được sử dụng cho việc ngữ cảnh hóa chiếm giữ. Trường `log2_neighbour_avail_boundary` chỉ rõ giá trị của biến số `NeighbAvailBoundary` mà nó được sử dụng trong quy trình giải mã.

Ví dụ, khi trường `neighbour_context_restriction_flag` bằng 1, `NeighbAvailabilityMask` có thể được thiết lập bằng 1. Ví dụ, khi trường `neighbour_context_restriction_flag` bằng 0, `NeighbAvailabilityMask` có thể được thiết lập bằng $1 \ll \log2_neighbour_avail_boundary$.

Trường `log2_intra_pred_max_node_size` chỉ rõ kích thước nút cây bát phân thích hợp cho việc dự đoán nội việc chiếm giữ.

Trường `log2_trisoup_node_size` chỉ rõ biến số `TrisoupNodeSize` làm kích thước của các nút hình tam giác.

`geom_scaling_enabled_flag` chỉ ra liệu quy trình định tỷ lệ cho các vị trí dạng hình học có được áp dụng trong quy trình giải mã lát dạng hình học hay không. Ví dụ, trường `geom_scaling_enabled_flag` bằng 1 chỉ rõ rằng quy trình định tỷ lệ cho các vị trí dạng hình học có được áp dụng trong quy trình giải mã lát dạng hình học. Trường `geom_scaling_enabled_flag` bằng 0 chỉ rõ rằng các vị trí dạng hình học không yêu cầu việc định tỷ lệ.

`geom_base_qp` chỉ ra giá trị cơ sở của tham số lượng tử hóa vị trí dạng hình học.

`gps_implicit_geom_partition_flag` chỉ ra liệu phân vùng dạng hình học ngầm định có được cho phép đối với trình tự hoặc lát hay không. Ví dụ, trường `gps_implicit_geom_partition_flag` bằng 1 chỉ rõ rằng phân vùng dạng hình học ngầm

định có được cho phép đối với trình tự hoặc lát. `gps_implicit_geom_partition_flag` bằng 0 chỉ rõ rằng phân vùng dạng hình học ngầm định bị vô hiệu hóa đối với trình tự hoặc lát. Khi `gps_implicit_geom_partition_flag` bằng 1, hai trường sau đây, tức là, trường `gps_max_num_implicit_qtbt_before_ot` và trường `gps_min_size_implicit_qtbt`, được báo hiệu.

`gps_max_num_implicit_qtbt_before_ot` chỉ rõ số lượng phân vùng QT và BT ngầm định tối đa trước các phân vùng OT. Khi đó, biến số K được khởi tạo bởi `gps_max_num_implicit_qtbt_before_ot` như sau.

$$K = \text{gps_max_num_implicit_qtbt_before_ot}.$$

`gps_min_size_implicit_qtbt` chỉ rõ kích thước tối thiểu của các phân vùng QT và BT ngầm định. Khi đó, biến số M được khởi tạo bởi `gps_min_size_implicit_qtbt` như sau.

$$M = \text{gps_min_size_implicit_qtbt}$$

`gps_extension_flag` chỉ ra liệu cấu trúc cú pháp `gps_extension_data` có có mặt trong cấu trúc cú pháp GPS hay không. Ví dụ, `gps_extension_flag` bằng 1 chỉ ra rằng cấu trúc cú pháp `gps_extension_data` có mặt trong cú pháp GPS. Ví dụ, `gps_extension_flag` bằng 0 chỉ ra rằng cấu trúc cú pháp `gps_extension_data` không có mặt trong cú pháp GPS.

Khi `gps_extension_flag` bằng 1, GPS theo các phương án có thể còn bao gồm trường `gps_extension_data_flag`.

`gps_extension_data_flag` có thể có giá trị bất kỳ. Sự có mặt và giá trị của nó không làm ảnh hưởng đến sự hợp cách của bộ giải mã đối với các hồ sơ.

Fig.36 thể hiện phương án về cấu trúc cú pháp của tập tham số thuộc tính (APS) (`attribute_parameter_set()`) theo sáng chế này. APS theo các phương án có thể chứa thông tin về phương pháp mã hóa thông tin thuộc tính về dữ liệu đám mây điểm được chứa trong một hoặc nhiều lát.

APS theo các phương án có thể bao gồm trường `aps_attr_parameter_set_id`, trường `aps_seq_parameter_set_id`, trường `attr_coding_type`, trường `aps_attr_initial_qp`, trường `aps_attr_chroma_qp_offset`, trường `aps_slice_qp_delta_present_flag`, và trường `aps_extension_flag`.

Trường `aps_attr_parameter_set_id` cung cấp bộ nhận dạng cho APS cho việc tham chiếu bởi các phần tử cú pháp khác.

Trường `aps_seq_parameter_set_id` chỉ rõ giá trị của `sps_seq_parameter_set_id` đối với SPS hoạt động.

Trường `attr_coding_type` chỉ ra loại lập mã cho thuộc tính.

Theo các phương án, trường `attr_coding_type` bằng 0 có thể chỉ ra việc nâng trọng số dự đoán làm loại lập mã. Trường `attr_coding_type` bằng 1 có thể chỉ ra RAHT làm loại lập mã. Trường `attr_coding_type` bằng 2 có thể chỉ ra việc nâng trọng số cố định.

Trường `aps_attr_initial_qp` chỉ rõ giá trị ban đầu của biến số `SliceQp` cho mỗi lát đang tham chiếu đến APS.

Trường `aps_attr_chroma_qp_offset` chỉ rõ các phần bù đối với tham số lượng tử hóa ban đầu được báo hiệu bởi cú pháp `aps_attr_initial_qp`.

Trường `aps_slice_qp_delta_present_flag` chỉ rõ liệu các phần tử cú pháp `ash_attr_qp_delta_luma` và `ash_attr_qp_delta_chroma` có có mặt trong tiêu đề lát thuộc tính (attribute slice header, ASH) hay không. Ví dụ, trường `aps_slice_qp_delta_present_flag` bằng 1 chỉ rõ rằng các phần tử cú pháp `ash_attr_qp_delta_luma` và `ash_attr_qp_delta_chroma` có mặt trong ASH. Ví dụ, trường `aps_slice_qp_delta_present_flag` chỉ rõ rằng các phần tử cú pháp `ash_attr_qp_delta_luma` và `ash_attr_qp_delta_chroma` không có mặt trong ASH.

Khi giá trị của trường `attr_coding_type` là 0 hoặc 2, tức là, loại lập mã là việc nâng trọng số dự đoán hoặc việc nâng trọng số cố định, APS theo các phương án có thể còn bao gồm trường `lifting_num_pred_nearest_neighbours_minus1`, trường `lifting_search_range_minus1`, và trường `lifting_neighbour_bias[k]`.

`lifting_num_pred_nearest_neighbours` cộng 1 chỉ rõ số lượng lân cận gần nhất tối đa sẽ được sử dụng cho việc dự đoán. Theo các phương án, giá trị của `NumPredNearestNeighbours` được đặt bằng `lifting_num_pred_nearest_neighbours`.

`lifting_search_range_minus1` cộng 1 chỉ rõ khoảng tìm kiếm được sử dụng để xác định các lân cận gần nhất sẽ được sử dụng cho việc dự đoán và để xây dựng các mức độ chi tiết (các LOD) dựa trên khoảng cách. Biến số `LiftingSearchRange` để chỉ rõ khoảng tìm kiếm có thể được thu bằng cách cộng 1 vào giá trị của trường `lifting_search_range_minus1` ($\text{LiftingSearchRange} = \text{lifting_search_range_minus1} + 1$).

Trường `lifting_neighbour_bias[k]` chỉ rõ thiên hướng được sử dụng để định trọng số cho các thành phần thứ `k` trong phép tính khoảng cách Euclide giữa hai điểm như một phần của quy trình dẫn ra lân cận gần nhất.

Khi giá trị của trường `attr_coding_type` là 2, tức là, khi loại lập mã chỉ ra việc nâng trọng số cố định, APS theo các phương án có thể còn bao gồm trường `lifting_scalability_enabled_flag`.

Trường `lifting_scalability_enabled_flag` chỉ rõ liệu quy trình giải mã thuộc tính có cho phép kết quả giải mã cây bất phân được xén bớt đối với các điểm dạng hình học đầu vào hay không. Ví dụ, trường `lifting_scalability_enabled_flag` bằng 1 chỉ rõ rằng quy trình giải mã thuộc tính có cho phép kết quả giải mã cây bất phân được xén bớt đối với các điểm dạng hình học đầu vào. Trường `lifting_scalability_enabled_flag` bằng 0 chỉ rõ rằng quy trình giải mã thuộc tính yêu cầu kết quả giải mã cây bất phân hoàn chỉnh đối với các điểm dạng hình học đầu vào.

Theo các phương án, khi giá trị của trường `lifting_scalability_enabled_flag` là sai (FALSE), APS có thể còn bao gồm trường `lifting_num_detail_levels_minus1`.

Trường `lifting_num_detail_levels_minus1` chỉ rõ số lượng mức độ chi tiết cho việc lập mã thuộc tính. Biến số `LevelDetailCount` để chỉ rõ số lượng LOD có thể được thu bằng cách cộng 1 vào giá trị của trường `lifting_num_detail_levels_minus1`. ($\text{LevelDetailCount} = \text{lifting_num_detail_levels_minus1} + 1$).

Theo các phương án, khi giá trị của trường `lifting_num_detail_levels_minus1` lớn hơn 1, APS có thể còn bao gồm trường `lifting_lod_regular_sampling_enabled_flag`.

Trường `lifting_lod_regular_sampling_enabled_flag` chỉ rõ liệu các mức độ chi tiết (các LOD) có được xây dựng bởi chiến lược lấy mẫu thông thường hay không. Ví dụ, trường `lifting_lod_regular_sampling_enabled_flag` bằng 1 chỉ rõ rằng các mức độ chi tiết (LOD) được xây dựng bằng cách sử dụng chiến lược lấy mẫu thông thường. Trường `lifting_lod_regular_sampling_enabled_flag` bằng 0 chỉ rõ rằng chiến lược lấy mẫu dựa trên khoảng cách được sử dụng thay vào đó.

Theo các phương án, khi giá trị của trường `lifting_scalability_enabled_flag` là sai (FALSE), APS có thể còn bao gồm câu lệnh lặp lại được lặp lại với số lần nhiều bằng giá trị của trường `lifting_num_detail_levels_minus1`. Trong một phương án, chỉ số (`idx`) được khởi tạo là 0 và được tăng lên 1 mỗi lần câu lệnh lặp lại được thực thi, và câu lệnh

lặp lại được lặp lại cho đến khi chỉ số (idx) là lớn hơn giá trị của trường `lifting_num_detail_levels_minus1`. Câu lệnh lặp lại này có thể bao gồm trường `lifting_sampling_period_minus2 [idx]` khi giá trị của trường `lifting_lod_decimation_enabled_flag` là đúng (ví dụ, 1), và có thể bao gồm trường `lifting_sampling_distance_squared_scale_minus1 [idx]` khi giá trị của trường `lifting_lod_regular_sampling_enabled_flag` là sai (ví dụ, 0). Ngoài ra, khi giá trị của idx là khác 0 (`idx != 0`), trường `lifting_sampling_distance_squared_offset [idx]` có thể được chứa thêm.

`lifting_sampling_period_minus2 [idx]` cộng 2 chỉ rõ chu kỳ lấy mẫu đối với mức độ chi tiết idx.

`lifting_sampling_distance_squared_scale_minus1 [idx]` cộng 1 chỉ rõ hệ số thang chia tỷ lệ dành cho việc dẫn ra bình phương của khoảng cách lấy mẫu đối với mức độ chi tiết idx.

Trường `lifting_sampling_distance_squared_offset [idx]` chỉ rõ phần bù của việc dẫn ra bình phương của khoảng cách lấy mẫu đối với mức độ chi tiết idx.

Khi giá trị của trường `attr_coding_type` là 0, tức là, khi loại lập mã là việc nâng trọng số dự đoán, APS theo các phương án có thể còn bao gồm trường `lifting_adaptive_prediction_threshold`, trường `lifting_intra_lod_prediction_num_layers`, trường `lifting_max_num_direct_predictors`, và trường `inter_component_prediction_enabled_flag`.

Trường `lifting_adaptive_prediction_threshold` chỉ rõ ngưỡng để cho phép việc dự đoán thích ứng. Theo các phương án, biến số `AdaptivePredictionThreshold` để chỉ rõ ngưỡng để chuyển đổi chế độ chọn bộ dự đoán thích ứng được đặt thành giá trị của trường `lifting_adaptive_prediction_threshold` (`AdaptivePredictionThreshold = lifting_adaptive_prediction_threshold`).

Trường `lifting_intra_lod_prediction_num_layers` chỉ rõ số lượng lớp LOD nơi các điểm được giải mã trong cùng lớp LOD có thể được tham chiếu để tạo ra giá trị dự đoán của điểm mục tiêu. Ví dụ, trường `lifting_intra_lod_prediction_num_layers` bằng `LevelDetailCount` chỉ ra rằng điểm mục tiêu có thể tham chiếu đến các điểm được giải mã trong cùng lớp LOD đối với tất cả các lớp LOD. Ví dụ, trường `lifting_intra_lod_prediction_num_layers` bằng 0 chỉ ra rằng điểm mục tiêu không thể

tham chiếu đến các điểm được giải mã trong cùng lớp LOD đối với các lớp LOD bất kỳ. Trường `lifting_max_num_direct_predictors` chỉ rõ số lượng bộ dự đoán tối đa sẽ được sử dụng cho việc dự đoán trực tiếp. Giá trị của trường `lifting_max_num_direct_predictors` sẽ phải nằm trong khoảng từ 0 đến `LevelDetailCount`.

Trường `inter_component_prediction_enabled_flag` chỉ rõ liệu thành phần sơ cấp của thuộc tính đa thành phần có được sử dụng để dự đoán giá trị được xây dựng lại của các thành phần phi sơ cấp hay không. Ví dụ, trường `inter_component_prediction_enabled_flag` bằng 1 chỉ rõ rằng thành phần sơ cấp của thuộc tính đa thành phần có được sử dụng để dự đoán giá trị được xây dựng lại của các thành phần phi sơ cấp. Trường `inter_component_prediction_enabled_flag` bằng 0 chỉ rõ rằng tất cả các thành phần thuộc tính đều được xây dựng lại một cách độc lập.

Theo các phương án, khi giá trị của trường `attr_coding_type` là 1, tức là, khi loại lập mã thuộc tính là RAHT, APS có thể còn bao gồm trường `raht_prediction_enabled_flag`.

Trường `raht_prediction_enabled_flag` chỉ rõ liệu việc dự đoán trọng số biến đổi từ các điểm lân cận có được cho phép trong quy trình giải mã RAHT hay không. Ví dụ, trường `raht_prediction_enabled_flag` bằng 1 chỉ rõ việc dự đoán trọng số biến đổi từ các điểm lân cận có được cho phép trong quy trình giải mã RAHT. Trường `raht_prediction_enabled_flag` bằng 0 chỉ rõ rằng việc dự đoán trọng số biến đổi được vô hiệu hóa trong quy trình giải mã RAHT.

Theo các phương án, khi giá trị của trường `raht_prediction_enabled_flag` là đúng (TRUE), APS có thể còn bao gồm trường `raht_prediction_threshold0` và trường `raht_prediction_threshold1`.

Trường `raht_prediction_threshold0` chỉ rõ ngưỡng để chấm dứt việc dự đoán trọng số biến đổi từ các điểm lân cận.

Trường `raht_prediction_threshold1` chỉ rõ ngưỡng để bỏ qua việc dự đoán trọng số biến đổi từ các điểm lân cận.

Trường `aps_extension_flag` chỉ rõ liệu cấu trúc cú pháp `aps_extension_data_flag` có có mặt trong cấu trúc cú pháp APS hay không. Ví dụ, `aps_extension_flag` bằng 1 chỉ ra rằng cấu trúc cú pháp `aps_extension_data` có có mặt trong cấu trúc cú pháp APS. Ví

dụ, `aps_extension_flag` bằng 0 chỉ ra rằng cấu trúc cú pháp `aps_extension_data` không có mặt trong cấu trúc cú pháp APS.

Khi giá trị của trường `aps_extension_flag` bằng 1, APS theo các phương án có thể còn bao gồm trường `aps_extension_data_flag`.

Trường `aps_extension_data_flag` có thể có giá trị bất kỳ. Sự có mặt và giá trị của nó không làm ảnh hưởng đến sự hợp cách của bộ giải mã đối với các hồ sơ.

APS theo các phương án có thể còn bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD.

Fig.37 thể hiện phương án về cấu trúc cú pháp của APS (`attribute_parameter_set()`) bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD theo các phương án.

Trên Fig.37, thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể bao gồm trường `constrained_QTBT_flag`, trường `num_LoD_minus1`, và trường `LoD_sampling_method`.

Trường `constrained_QTBT_flag` có thể chỉ ra liệu có ràng buộc trong việc tạo ra cấu trúc cây dành cho việc nén thông tin dạng hình học và/hoặc việc nén thông tin thuộc tính bởi bộ mã hóa video đám mây điểm hay không. Ví dụ, `constrained_QTBT_flag` bằng 1 có thể chỉ ra rằng có ràng buộc trong việc tạo ra cấu trúc cây. Theo các phương án, ràng buộc này có thể là điều kiện đối với việc lập mã thuộc tính có thể định tỷ lệ được và/hoặc có thể chỉ ra rằng việc tạo ra LoD dựa trên OT có thể được sử dụng đối với độ sâu cụ thể hoặc độ sâu thấp hơn (xem phần mô tả của phương án 2 được cho trên đây). Ví dụ, giá trị của trường `constrained_QTBT_flag` bằng 1 có thể được sử dụng để chỉ ra rằng bộ giải mã thuộc tính 61004 của thiết bị nhận có thể tạo ra các LoD sử dụng cây dạng hình học dựa trên OT/QT/BT. Như là một ví dụ khác, giá trị của trường `constrained_QTBT_flag` bằng 1 có thể được sử dụng để chỉ ra rằng bộ giải mã thuộc tính 61004 của thiết bị nhận có thể sử dụng một số độ sâu (ví dụ, bắt đầu từ nút lá) của cây dạng hình học cho việc tạo ra LoD dựa trên cây bát phân.

Theo các phương án, khi giá trị của trường `constrained_QTBT_flag` là 1, thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm trường `max_num_LoD` chỉ ra số lượng LOD tối đa và trường `LoD_starting_depth` chỉ ra điểm nơi LoD bắt đầu. Theo các phương án, trường `max_num_LoD` có thể chỉ ra kích thước độ sâu từ một mức lá trong cây dạng hình học. Ngoài ra, bộ giải mã thuộc tính 61004

của thiết bị nhận có thể giải mã thông tin thuộc tính bằng cách tạo ra LoD (các LoD) dựa trên cây bát phân chỉ cho độ sâu (các độ sâu) tương ứng trong cây dạng hình học dựa trên trường `constrained_QTBT_flag`, trường `max_num_LoD`, và trường `LoD_starting_depth`. Giá trị của trường `constrained_QTBT_flag` bằng 0 có thể được sử dụng để chỉ ra rằng bộ giải mã thuộc tính 61004 của thiết bị nhận cần phải sử dụng việc tạo ra LoD dựa trên cây bát phân.

`num_LoD_minus1` cộng 1 có thể chỉ ra số lượng LoD. Theo cách khác, nó có thể được báo hiệu cùng với trường `num_tree_depth_minus1` để chỉ ra độ sâu trong cây dạng hình học. Để chỉ ra số lượng độ sâu của cây, trường `num_tree_depth_minus1` có thể được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD hoặc tiêu đề lát dạng hình học.

Trường `LoD_sampling_method` có thể chỉ ra phương pháp lấy mẫu để chọn điểm trong khoảng tìm kiếm lân cận (hoặc lân cận được tạo cấu hình) khi tạo ra các LoD. `LoD_sampling_method` bằng 0 có thể chỉ ra rằng điểm thứ nhất có thể được chọn từ trong số các điểm được tạo cấu hình (hoặc được chứa) trong lân cận. Trong một phương án, điểm thứ nhất nghĩa là điểm thứ nhất trong số các điểm được sắp xếp theo thứ tự mã Morton bên trong khoảng tìm kiếm lân cận. `LoD_sampling_method` bằng 1 có thể chỉ ra rằng điểm thứ nhất và điểm cuối cùng có thể được chọn luân phiên cho mỗi độ sâu. Ở đây, điểm thứ nhất thể hiện điểm thứ nhất trong số các điểm được sắp xếp theo thứ tự mã Morton bên trong khoảng tìm kiếm lân cận của độ sâu này, và điểm cuối cùng thể hiện điểm cuối cùng trong số các điểm được sắp xếp theo thứ tự mã Morton bên trong khoảng tìm kiếm lân cận của độ sâu này. `LoD_sampling_method` bằng 2 có thể chỉ ra rằng điểm có khoảng cách Euclide nhỏ so với tâm lân cận có thể được chọn từ trong số các điểm bên trong khoảng tìm kiếm lân cận. Nói cách khác, `LoD_sampling_method` bằng 2 có thể chỉ ra rằng điểm gần với tâm nút có thể được chọn.

Theo các phương án, khi giá trị của trường `constrained_QTBT_flag` là đúng (ví dụ, 1), thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm trường `max_LoD_depth_for_octree_based_LoD_generation`.

Khi giá trị của trường `constrained_QTBT_flag` là 1, trường `max_LoD_depth_for_octree_based_LoD_generation` có thể chỉ ra độ sâu LoD tối đa mà nó có thể được so khớp với việc tạo ra LoD dựa trên cây bát phân. Theo các phương án,

độ sâu LoD tối đa có thể chỉ ra kích thước độ sâu từ mức lá của cây dạng hình học. Theo các phương án, trường `max_LoD_depth_for_octree_based_LoD_generation` có thể có cùng ý nghĩa như trường `LoD_starting_depth`.

Theo các phương án, thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm vòng lặp mà nó được lặp lại với số lần nhiều bằng giá trị của trường `num_LoD_minus1`. Trong trường hợp này, theo một phương án, `i` có thể được khởi tạo là 0, và được tăng lên 1 mỗi lần vòng lặp được thực thi. Vòng lặp có thể được lặp lại cho đến khi `i` đạt đến giá trị của trường `num_LoD_minus1`. Vòng lặp này có thể bao gồm trường `cubic_node_flag`, trường `node_size_x`, trường `node_size_y`, và trường `node_size_z`.

`cubic_node_flag` bằng 1 có thể chỉ ra rằng nút đơn vị được sử dụng cho việc tìm kiếm lân cận trong việc tạo ra LoD là nút có hình dạng khối lập phương (cụ thể là, khối sáu mặt đều) có cùng chiều dài trên các cạnh `xyz`. `cubic_node_flag` bằng 0 có thể chỉ ra rằng nút đơn vị được sử dụng cho việc tìm kiếm lân cận trong việc tạo ra LoD là phỏng lập phương (hoặc hình hộp chữ nhật hoặc hình dạng phi lập phương) có các chiều dài khác nhau trên các cạnh `xyz`.

Khi giá trị của trường `cubic_node_flag` là 0, thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm trường `node_direction`.

Trường `node_direction` có thể chỉ ra sự định hướng (cụ thể là, hướng của việc phân vùng) của nút có hình dạng phỏng lập phương. Theo các phương án, các giá trị 0 đến 5 có thể lần lượt chỉ ra những sự định hướng `z`, `y`, `x`, `yz`, `xz`, và `xy` của các nút. Ví dụ, khi kích thước của nút trên các trục được biểu diễn dưới dạng 2^{dx} , 2^{dy} , và 2^{dz} , $(k, k, 2k)$, $(k, 2k, k)$, $(2k, k, k)$, $(k, 2k, 2k)$, $(2k, k, 2k)$, và $(2k, 2k, k)$ có thể được thu, ở đây, k biểu thị kích thước đơn vị. Theo cách khác, kích thước của nút trên các trục `xyz` có thể được báo hiệu sử dụng các trường `node_size_x`, `node_size_y`, và `node_size_z`.

Các trường `node_size_x`, `node_size_y`, và `node_size_z` có thể chỉ ra kích thước của nút trên các trục `xyz`.

Theo các phương án, khi giá trị của trường `constrained_QTBT_flag` là sai 1, vòng lặp có thể còn bao gồm trường `no_representation_flag`.

Khi việc tạo ra LoD dựa trên cây bất phân được thực hiện trong việc mã hóa thuộc tính, trường `no_representation_flag` có thể chỉ ra liệu độ sâu tương ứng có là thích hợp

cho phép biểu diễn có thể định tỷ lệ được hay không. Theo cách khác, trường `no_representation_flag` có thể được sử dụng để trực tiếp thông báo rằng số lượng điểm đối với dạng hình học là khác với số lượng điểm đối với thuộc tính tại độ sâu này. Ví dụ, trường `no_representation_flag` bằng 1 có thể chỉ ra rằng độ sâu này là không thích hợp cho phép biểu diễn có thể định tỷ lệ được. Trường `no_representation_flag` bằng 0 có thể chỉ ra rằng độ sâu này là thích hợp cho phép biểu diễn có thể định tỷ lệ được hoặc rằng số lượng điểm đối với dạng hình học là giống với số lượng điểm đối với thuộc tính tại độ sâu này. Như là một ví dụ khác, trường `no_representation_flag` có thể chỉ ra rằng LoD là LoD được tạo ra tại độ sâu của cây dạng hình học nơi việc không khớp điểm giữa dạng hình học và thuộc tính xảy ra. Trong trường hợp này, LoD có thể không được sử dụng cho việc nén thuộc tính, và bộ giải mã thuộc tính 61004 của thiết bị nhận có thể không sử dụng LoD này cho việc xây dựng lại thuộc tính dựa trên thông tin này. Trường `no_representation_flag` có thể có ý nghĩa giống với trường `LoD_skip_flag`. Theo cách khác, trường `LoD_skip_flag` có thể được chứa thêm trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD. Theo cách khác, bộ giải mã thuộc tính 61004 của thiết bị nhận có thể xác định liệu LoD có được bỏ qua hay không dựa trên thông tin được chứa trong vòng lặp.

Theo các phương án, thông tin liên quan đến việc nén thuộc tính dựa trên LoD trên Fig.37 có thể được chứa trong câu lệnh `if(LodParametersPresent)` hoặc câu lệnh `if(attr_coding_type == 0)` trong APS trên Fig.36. Điều này đơn thuần chỉ là một phương án, và thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể được chứa tại vị trí bất kỳ trong APS trên Fig.36. Fig.38 thể hiện phương án về cấu trúc cú pháp của luồng bit lát dạng hình học theo sáng chế này.

Luồng bit lát dạng hình học (`geometry_slice_bitstream()`) theo các phương án có thể bao gồm tiêu đề lát dạng hình học (`geometry_slice_header()`) và dữ liệu lát dạng hình học (`geometry_slice_data()`). Nó có thể được gọi là lát dạng hình học. Ngoài ra, luồng bit lát thuộc tính có thể được gọi là lát thuộc tính.

Fig.38 thể hiện cấu trúc cú pháp mang tính ví dụ của `geometry_slice_bitstream()` theo các phương án.

Luồng bit lát dạng hình học (`geometry_slice_bitstream()`) theo các phương án có thể bao gồm tiêu đề lát dạng hình học (`geometry_slice_header()`) và dữ liệu lát dạng hình học (`geometry_slice_data()`).

Fig.39 thể hiện phương án về cấu trúc cú pháp của tiêu đề lát dạng hình học (`geometry_slice_header()`) theo sáng chế này.

Luồng bit được truyền bởi thiết bị truyền (hoặc luồng bit được nhận bởi thiết bị nhận) theo các phương án có thể chứa một hoặc nhiều lát. Mỗi lát có thể bao gồm lát dạng hình học và lát thuộc tính. Lát dạng hình học bao gồm tiêu đề lát dạng hình học (`geometry slice header`, GSH). Lát thuộc tính bao gồm tiêu đề lát thuộc tính (ASH).

Tiêu đề lát dạng hình học (`geometry_slice_header()`) theo các phương án có thể bao gồm trường `gsh_geometry_parameter_set_id`, trường `gsh_tile_id`, trường `gsh_slice_id`, trường `frame_idx`, trường `gsh_num_points`, và trường `byte_alignment()`.

Khi giá trị của trường `gps_box_present_flag` được chứa trong GPS là đúng (ví dụ, 1), và giá trị của trường `gps_gsh_box_log2_scale_present_flag` là đúng (ví dụ, 1), tiêu đề lát dạng hình học (`geometry_slice_header()`) theo các phương án có thể còn bao gồm trường `gsh_box_log2_scale`, trường `gsh_box_origin_x`, trường `gsh_box_origin_y`, và trường `gsh_box_origin_z`.

`gsh_geometry_parameter_set_id` chỉ rõ giá trị của `gps_geom_parameter_set_id` của GPS hoạt động.

Trường `gsh_tile_id` chỉ rõ giá trị của id của ô xếp mà nó được tham chiếu bởi GSH. `gsh_slice_id` chỉ rõ ID của lát cho việc tham chiếu bởi các phần tử cú pháp khác.

Trường `frame_idx` chỉ ra $\log_2_max_frame_idx + 1$ bit ít quan trọng nhất của bộ đếm số lượng khung mang tính khái niệm. Các lát liên tiếp với các giá trị khác nhau của `frame_idx` tạo ra các phần của các khung đám mây điểm đầu ra khác nhau. Các lát liên tiếp với các giá trị giống nhau của `frame_idx` mà không có đơn vị dữ liệu đánh dấu biên của khung xen giữa tạo ra các phần của khung đám mây điểm đầu ra giống nhau.

Trường `gsh_num_points` chỉ ra số lượng điểm được lập mã tối đa trong lát. Theo các phương án, việc `gsh_num_points` lớn hơn hoặc bằng số lượng điểm được giải mã trong lát là yêu cầu đối với sự hợp cách của luồng bit.

Trường `gsh_box_log2_scale` chỉ rõ hệ số định tỷ lệ của góc hộp giới hạn cho lát.

Trường `gsh_box_origin_x` chỉ rõ giá trị `x` của gốc hộp giới hạn được định tỷ lệ bởi giá trị của trường `gsh_box_log2_scale`.

Trường `gsh_box_origin_y` chỉ rõ giá trị `y` của gốc hộp giới hạn được định tỷ lệ bởi giá trị của trường `gsh_box_log2_scale`.

Trường `gsh_box_origin_z` chỉ rõ giá trị `z` của gốc hộp giới hạn được định tỷ lệ bởi giá trị của trường `gsh_box_log2_scale`.

Ở đây, các biến số `slice_origin_x`, `slice_origin_y`, và `slice_origin_z` có thể được dẫn ra như sau.

Khi `gps_gsh_box_log2_scale_present_flag` bằng 0, `originScale` được đặt là `gsh_box_log2_scale`.

Khi `gps_gsh_box_log2_scale_present_flag` bằng 1, `originScale` được đặt là `gps_gsh_box_log2_scale`.

Khi `gps_box_present_flag` bằng 0, các giá trị của các biến số `slice_origin_x`, `slice_origin_y`, và `slice_origin_z` được suy ra là 0.

Khi `gps_box_present_flag` bằng 1, các phương trình sau đây sẽ được áp dụng với các biến số `slice_origin_x`, `slice_origin_y`, và `slice_origin_z`.

$$\text{slice_origin_x} = \text{gsh_box_origin_x} \ll \text{originScale}$$

$$\text{slice_origin_y} = \text{gsh_box_origin_y} \ll \text{originScale}$$

$$\text{slice_origin_z} = \text{gsh_box_origin_z} \ll \text{originScale}$$

Khi giá trị của trường `gps_implicit_geom_partition_flag` là đúng (cụ thể là, 0), tiêu đề lát dạng hình học ((`geometry_slice_header()`)) có thể còn bao gồm trường `gsh_log2_max_nodesize_x`, trường `gsh_log2_max_nodesize_y_minus_x`, và trường `gsh_log2_max_nodesize_z_minus_y`. Khi giá trị của trường `gps_implicit_geom_partition_flag` là sai (cụ thể là, 1), tiêu đề lát dạng hình học có thể còn bao gồm trường `gsh_log2_max_nodesize`.

Trường `gsh_log2_max_nodesize_x` chỉ rõ kích thước hộp giới hạn theo chiều `x`, cụ thể là, `MaxNodeSizeXLog2` mà nó được sử dụng trong quy trình giải mã như sau.

$$\text{MaxNodeSizeXLog2} = \text{gsh_log2_max_nodesize_x}$$

$$\text{MaxNodeSizeX} = 1 \ll \text{MaxNodeSizeXLog2}$$

Trường `gsh_log2_max_nodsize_y_minus_x` chỉ rõ kích thước hộp giới hạn theo chiều y, cụ thể là, `MaxNodeSizeYLog2` mà nó được sử dụng trong quy trình giải mã như sau.

$$\text{MaxNodeSizeYLog2} = \text{gsh_log2_max_nodsize_y_minus_x} + \text{MaxNodeSizeXLog2}.$$

$$\text{MaxNodeSizeY} = 1 \ll \text{MaxNodeSizeYLog2}.$$

Trường `gsh_log2_max_nodsize_z_minus_y` chỉ rõ kích thước hộp giới hạn theo chiều z, cụ thể là, `MaxNodeSizeZLog2` mà nó được sử dụng trong quy trình giải mã như sau.

$$\text{MaxNodeSizeZLog2} = \text{gsh_log2_max_nodsize_z_minus_y} + \text{MaxNodeSizeYLog2}$$

$$\text{MaxNodeSizeZ} = 1 \ll \text{MaxNodeSizeZLog2}$$

Khi giá trị của trường `gps_implicit_geom_partition_flag` là 1, `gsh_log2_max_nodsize` được thu như sau.

$$\text{gsh_log2_max_nodsize} = \max\{\text{MaxNodeSizeXLog2}, \text{MaxNodeSizeYLog2}, \text{MaxNodeSizeZLog2}\}$$

Trường `gsh_log2_max_nodsize` chỉ rõ kích thước của nút cây bát phân dạng hình học ở gốc khi `gps_implicit_geom_partition_flag` bằng 0.

Ở đây, các biến số `MaxNodeSize` và `MaxGeometryOctreeDepth` được dẫn ra như sau.

$$\text{MaxNodeSize} = 1 \ll \text{gsh_log2_max_nodsize}$$

$$\text{MaxGeometryOctreeDepth} = \text{gsh_log2_max_nodsize} - \text{log2_trisoup_node_size}$$

Khi giá trị của trường `geom_scaling_enabled_flag` là đúng (TRUE), tiêu đề lát dạng hình học (`geometry_slice_header()`) theo các phương án có thể còn bao gồm trường `geom_slice_qp_offset` và trường `geom_octree_qp_offsets_enabled_flag`.

Trường `geom_slice_qp_offset` chỉ rõ phần bù đối với tham số lượng tử hóa dạng hình học cơ sở `geom_base_qp`.

Trường `geom_octree_qp_offsets_enabled_flag` chỉ rõ liệu trường `geom_octree_qp_ofsets_depth` có có mặt trong tiêu đề lát dạng hình học hay không. Ví dụ, `geom_octree_qp_offsets_enabled_flag` bằng 1 chỉ rõ rằng trường `geom_octree_qp_ofsets_depth` có có mặt trong tiêu đề lát dạng hình học.

`geom_octree_qp_offsets_enabled_flag` bằng 0 chỉ rõ rằng trường `geom_octree_qp_ofsets_depth` không có mặt.

Trường `geom_octree_qp_offsets_depth` chỉ rõ độ sâu của cây bát phân dạng hình học.

Fig.40 thể hiện phương án về cấu trúc cú pháp của dữ liệu lát dạng hình học (`geometry_slice_data()`) theo sáng chế này. Dữ liệu lát dạng hình học (`geometry_slice_data()`) theo các phương án có thể mang luồng bit dạng hình học thuộc về lát tương ứng.

`geometry_slice_data()` theo các phương án có thể bao gồm câu lệnh lặp lại thứ nhất được lặp lại với số lần nhiều bằng giá trị của `MaxGeometryOctreeDepth`. Trong một phương án, độ sâu (`depth`) được khởi tạo là 0 và được tăng lên 1 mỗi lần câu lệnh lặp lại được thực thi, và câu lệnh lặp lại thứ nhất được lặp lại cho đến khi độ sâu này trở thành bằng với `MaxGeometryOctreeDepth`. Câu lệnh lặp lại thứ nhất có thể bao gồm câu lệnh vòng lặp thứ hai được lặp lại với số lần nhiều bằng giá trị của `NumNodesAtDepth`. Trong một phương án, `nodeidx` được khởi tạo là 0 và được tăng lên 1 mỗi lần câu lệnh lặp lại này được thực thi. Câu lệnh lặp lại thứ hai này được lặp lại cho đến khi `nodeidx` trở thành bằng với `NumNodesAtDepth`. Câu lệnh lặp lại thứ hai có thể bao gồm $xN = \text{NodeX}[\text{depth}][\text{nodeIdx}]$, $yN = \text{NodeY}[\text{depth}][\text{nodeIdx}]$, $zN = \text{NodeZ}[\text{depth}][\text{nodeIdx}]$, và `geometry_node(depth, nodeIdx, xN, yN, zN)`. `MaxGeometryOctreeDepth` chỉ ra giá trị tối đa của độ sâu cây bát phân dạng hình học, và `NumNodesAtDepth[depth]` chỉ ra số lượng nút cần được giải mã tại độ sâu tương ứng. Các biến số `NodeX[depth][nodeIdx]`, `NodeY[depth][nodeIdx]`, và `NodeZ[depth][nodeIdx]` chỉ ra các tọa độ x, y, z của nút thứ `idx` theo thứ tự giải mã tại độ sâu đã cho. Luồng bit dạng hình học của nút có độ sâu này được truyền qua `geometry_node(depth, nodeIdx, xN, yN, zN)`.

Dữ liệu lát dạng hình học (`geometry_slice_data()`) theo các phương án có thể còn bao gồm `geometry_trisoup_data()` khi giá trị của trường `log2_trisoup_node_size` lớn hơn 0. Tức là, khi kích thước của các nút hình tam giác là lớn hơn 0, luồng bit dạng hình học là đối tượng của việc mã hóa dạng hình học trisoup được truyền qua `geometry_trisoup_data()`.

Fig.41 thể hiện phương án về cấu trúc cú pháp của `attribute_slice_bitstream()` theo sáng chế này.

Luồng bit lát thuộc tính (`attribute_slice_bitstream()`) theo các phương án có thể bao gồm tiêu đề lát thuộc tính (`attribute_slice_header()`) và dữ liệu lát thuộc tính (`attribute_slice_data()`).

Fig.42 thể hiện phương án về cấu trúc cú pháp của tiêu đề lát thuộc tính (`attribute_slice_header()`) theo sáng chế này.

Tiêu đề lát thuộc tính (`attribute_slice_header()`) theo các phương án có thể bao gồm trường `ash_attr_parameter_set_id`, trường `ash_attr_sps_attr_idx`, trường `ash_attr_geom_slice_id`, trường `ash_attr_layer_qp_delta_present_flag`, và trường `ash_attr_region_qp_delta_present_flag`.

Khi giá trị của trường `aps_slice_qp_delta_present_flag` của APS là đúng (ví dụ, 1), tiêu đề lát thuộc tính (`attribute_slice_header()`) theo các phương án có thể còn bao gồm trường `ash_attr_qp_delta_luma`. Khi giá trị của trường `attribute_dimension_minus1[ash_attr_sps_attr_idx]` lớn hơn 0, tiêu đề lát thuộc tính có thể còn bao gồm trường `ash_attr_qp_delta_chroma`.

Trường `ash_attr_parameter_set_id` chỉ rõ giá trị của trường `aps_attr_parameter_set_id` của APS hoạt động hiện tại.

Trường `ash_attr_sps_attr_idx` chỉ rõ tập thuộc tính trong SPS hoạt động hiện tại.

Trường `ash_attr_geom_slice_id` chỉ rõ giá trị của trường `gsh_slice_id` của tiêu đề lát dạng hình học hiện tại.

Trường `ash_attr_qp_delta_luma` chỉ rõ tham số lượng tử hóa qp delta độ chói được dẫn ra từ qp lát ban đầu trong tập tham số thuộc tính hoạt động.

Trường `ash_attr_qp_delta_chroma` chỉ rõ qp delta sắc độ được dẫn ra từ qp lát ban đầu trong tập tham số thuộc tính hoạt động.

Các biến số `InitialSliceQpY` và `InitialSliceQpC` được dẫn ra như sau.

$$\text{InitialSliceQpY} = \text{aps_attrattr_initial_qp} + \text{ash_attr_qp_delta_luma}$$

$$\text{InitialSliceQpC} = \text{aps_attrattr_initial_qp} + \text{aps_attr_chroma_qp_offset} + \text{ash_attr_qp_delta_chroma}$$

Trường `ash_attr_layer_qp_delta_present_flag` chỉ rõ liệu trường `ash_attr_layer_qp_delta_luma` và trường `ash_attr_layer_qp_delta_chroma` có có mặt trong ASH cho mỗi lớp hay không. Ví dụ, khi giá trị của trường `ash_attr_layer_qp_delta_present_flag` là 1, nó chỉ ra rằng trường

ash_attr_layer_qp_delta_luma và trường ash_attr_layer_qp_delta_chroma có mặt trong ASH. Khi giá trị này là 0, nó chỉ ra rằng các trường này không có mặt.

Khi giá trị của trường ash_attr_layer_qp_delta_present_flag là đúng (TRUE), ASH có thể còn bao gồm trường ash_attr_num_layer_qp_minus1.

ash_attr_num_layer_qp_minus1 cộng 1 chỉ ra số lượng lớp mà trường ash_attr_qp_delta_luma và trường ash_attr_qp_delta_chroma được báo hiệu qua đó. Khi trường ash_attr_num_layer_qp không được báo hiệu, giá trị của trường ash_attr_num_layer_qp sẽ là 0. Theo các phương án, NumLayerQp chỉ rõ số lượng lớp có thể được thu bằng cách cộng 1 vào giá trị của trường ash_attr_num_layer_qp_minus1 ($\text{NumLayerQp} = \text{ash_attr_num_layer_qp_minus1} + 1$).

Theo các phương án, khi giá trị của trường ash_attr_layer_qp_delta_present_flag là đúng (TRUE), tiêu đề lát dạng hình học có thể bao gồm vòng lặp được lặp lại với số lần nhiều bằng giá trị của NumLayerQp. Trong trường hợp này, trong một phương án, i có thể được khởi tạo là 0 và được tăng lên 1 mỗi lần vòng lặp được thực thi, và vòng lặp được lặp lại cho đến khi giá trị của i đạt đến giá trị của NumLayerQp. Vòng lặp này chứa trường ash_attr_layer_qp_delta_luma[i]. Ngoài ra, khi giá trị của trường attribute_dimension_minus1[ash_attr_sps_attr_idx] lớn hơn 0, vòng lặp có thể còn bao gồm trường ash_attr_layer_qp_delta_chroma[i].

Trường ash_attr_layer_qp_delta_luma chỉ ra tham số lượng tử hóa qp delta độ chói từ InitialSliceQpY trong mỗi lớp.

Trường ash_attr_layer_qp_delta_chroma chỉ ra tham số lượng tử hóa qp delta sắc độ từ InitialSliceQpC trong mỗi lớp.

Các biến số SliceQpY[i] và SliceQpC[i] với $i = 0, \dots, \text{NumLayerQpNumQP} - 1$ được dẫn ra như sau.

```
for ( i = 0; i < NumLayerQpNumQP; i++) {
    SliceQpY[i] = InitialSliceQpY + ash_attr_layer_qp_delta_luma[i]
    SliceQpC[i] = InitialSliceQpC + ash_attr_layer_qp_delta_chroma[i]
}
```

ash_attr_region_qp_delta_present_flag bằng 1 chỉ ra rằng ash_attr_region_qp_delta, gốc hộp giới hạn vùng, và kích thước là có mặt trong tiêu đề lát thuộc tính (attribute_slice_header()) hiện tại theo các phương án.

ash_attr_region_qp_delta_present_flag bằng 0 chỉ ra rằng ash_attr_region_qp_delta, gốc hộp giới hạn vùng, và kích thước không có mặt trong tiêu đề lát thuộc tính hiện tại.

Tức là, khi giá trị của trường ash_attr_layer_qp_delta_present_flag là 1, tiêu đề lát thuộc tính có thể còn bao gồm trường ash_attr_qp_region_box_origin_x, trường ash_attr_qp_region_box_origin_y, trường ash_attr_qp_region_box_origin_z, trường ash_attr_qp_region_box_width, trường ash_attr_qp_region_box_height, trường ash_attr_qp_region_box_depth, và trường ash_attr_region_qp_delta.

Trường ash_attr_qp_region_box_origin_x chỉ ra phần bù x của hộp giới hạn vùng so với slice_origin_x.

Trường ash_attr_qp_region_box_origin_y chỉ ra phần bù y của hộp giới hạn vùng so với slice_origin_y.

Trường ash_attr_qp_region_box_origin_z chỉ ra phần bù z của hộp giới hạn vùng so với slice_origin_z.

Trường ash_attr_qp_region_box_size_width chỉ ra chiều rộng của hộp giới hạn vùng.

Trường ash_attr_qp_region_box_size_height chỉ ra chiều cao của hộp giới hạn vùng.

Trường ash_attr_qp_region_box_size_depth chỉ ra chiều sâu của hộp giới hạn vùng.

Trường ash_attr_region_qp_delta chỉ ra qp_delta từ SliceQpY[i] và SliceQpC[i] của vùng được chỉ rõ bởi trường ash_attr_qp_region_box.

Theo các phương án, biến số RegionboxDeltaQp chỉ rõ tham số lượng tử hóa delta hộp vùng được đặt là bằng với giá trị của trường ash_attr_region_qp_delta (RegionboxDeltaQp = ash_attr_region_qp_delta).

Tiêu đề lát thuộc tính theo các phương án có thể còn bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD.

Fig.43 là sơ đồ minh họa phương án về cấu trúc cú pháp của tiêu đề lát thuộc tính (attribute_slice_header()) chứa thông tin liên quan đến việc nén thuộc tính dựa trên LoD theo các phương án.

Thông tin liên quan đến việc nén thuộc tính dựa trên LoD trên Fig.43 có thể bao gồm trường constrained_QTBT_flag, trường num_LoD_minus1, và trường

LoD_sampling_method. Thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm trường max_LoD_depth_for_octree_based_LoD_generation tùy theo giá trị của trường constrained_QTBT_flag. Thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể còn bao gồm vòng lặp mà nó được lặp lại với số lần nhiều bằng giá trị của trường num_LoD_minus1. Vòng lặp này có thể bao gồm trường cubic_node_flag, trường node_size_x, trường node_size_y, và trường node_size_z. Vòng lặp này có thể còn bao gồm trường node_direction tùy theo giá trị của trường cubic_node_flag, và có thể còn bao gồm trường no_representation_flag tùy theo giá trị của trường constrained_QTBT_flag.

Các trường được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD trên Fig.43 có cùng ý nghĩa như các trường trên Fig.37. Theo đó, để biết các chi tiết về các trường này, hãy xem phần mô tả chi tiết trên Fig.37.

Theo các phương án, thông tin liên quan đến việc nén thuộc tính dựa trên LoD trên Fig.43 có thể được chứa trước câu lệnh if (aps_slice_qp_delta_present_flag) trong tiêu đề lát thuộc tính trên Fig.43. Điều này đơn thuần chỉ là một phương án, và thông tin liên quan đến việc nén thuộc tính dựa trên LoD có thể được chứa tại vị trí bất kỳ trong APS trên Fig.43.

Fig.44 là sơ đồ thể hiện phương án về cấu trúc cú pháp của dữ liệu lát thuộc tính (attribute_slice_data()) theo sáng chế này. Dữ liệu lát thuộc tính (attribute_slice_data()) theo các phương án có thể mang luồng bit thuộc tính thuộc về lát tương ứng. Dữ liệu lát thuộc tính theo các phương án có thể bao gồm thuộc tính hoặc dữ liệu liên quan đến thuộc tính trong mối quan hệ với một số hoặc tất cả các đám mây điểm.

Trên Fig.44, trường loạt không (zerorun) chỉ rõ số lượng 0 đứng trước predIndex hoặc phần dư.

Trường predIndex[i] chỉ rõ chỉ số bộ dự đoán cho việc giải mã giá trị của điểm thứ i của thuộc tính. Giá trị của trường predIndex[i] nằm trong khoảng từ 0 đến giá trị của trường max_num_predictors.

Các hoạt động của các phương án được mô tả trên đây có thể được thực hiện qua các phần tử của thiết bị/phương pháp truyền/nhận đám mây điểm theo các phương án được mô tả dưới đây. Mỗi phần tử theo các phương án có thể tương ứng với phần cứng, phần mềm, bộ xử lý và/hoặc sự kết hợp của chúng. Mặc dù phương án này mô tả phương

pháp nén thông tin thuộc tính của dữ liệu đám mây điểm, phương pháp được mô tả ở đây có thể được áp dụng cho việc nén thông tin dạng hình học và các phương pháp nén khác.

Fig.45 minh một ví dụ khác về thiết bị truyền đám mây điểm theo các phương án. Các phần tử của thiết bị truyền đám mây điểm được thể hiện trên Fig.45 có thể được thi hành dưới dạng phần cứng, phần mềm, bộ xử lý và/hoặc sự kết hợp của chúng.

Theo các phương án, thiết bị truyền đám mây điểm có thể bao gồm bộ xử lý báo hiệu 51002, bộ mã hóa dạng hình học 51003, bộ mã hóa thuộc tính 51004, và bộ xử lý truyền 51005.

Bộ mã hóa dạng hình học 51003 và bộ mã hóa thuộc tính 51004 có thể thực hiện một phần hoặc toàn bộ các hoạt động được mô tả trong mối quan hệ với bộ mã hóa video đám mây điểm 10002 trên Fig.1, việc mã hóa 20001 trên Fig.2, bộ mã hóa video đám mây điểm trên Fig.4, và bộ mã hóa video đám mây điểm trên Fig.12.

Trong một phương án, bộ mã hóa dạng hình học 51003 tạo ra cây dạng hình học bằng cách phân vùng dữ liệu đám mây điểm gồm các điểm được phân bố một cách đối xứng hoặc bất đối xứng như được mô tả trên đây dựa trên OT hoặc OT/QT/BT, và nén thông tin dạng hình học của dữ liệu đám mây điểm dựa trên cây dạng hình học được tạo ra.

Để biết chi tiết về việc tạo ra cây dạng hình học bằng cách phân vùng dữ liệu đám mây điểm dựa trên OT hoặc OT/QT/BT bởi bộ mã hóa dạng hình học 51003, hãy tham khảo phần mô tả về Fig.16 đến Fig.30 được cho trên đây.

Bộ mã hóa dạng hình học 51003 xây dựng lại thông tin dạng hình học dựa trên các vị trí được thay đổi qua việc nén, và xuất ra thông tin dạng hình học được xây dựng lại (hoặc được giải mã) đến bộ mã hóa thuộc tính 51004.

Bộ mã hóa thuộc tính 51004 nén thông tin thuộc tính dựa trên các vị trí mà việc mã hóa dạng hình học chưa được thực hiện trên đó và/hoặc thông tin dạng hình học được xây dựng lại. Bộ mã hóa thuộc tính 51004 có thể thực hiện việc mã hóa thông tin thuộc tính hoặc thực hiện việc mã hóa thuộc tính có thể định tỷ lệ được bằng cách kết hợp một hoặc hai hay hơn hai phương án trong số các phương án 1 đến 3.

Bộ xử lý báo hiệu 51002 có thể tạo ra và/hoặc xử lý thông tin báo hiệu cần thiết cho việc mã hóa/giải mã/kết xuất thông tin dạng hình học và thông tin thuộc tính, và

cung cấp thông tin này cho bộ mã hóa dạng hình học 51003, bộ mã hóa thuộc tính 51004, và/hoặc bộ xử lý truyền 51005. Theo cách khác, bộ xử lý báo hiệu 51002 có thể nhận thông tin báo hiệu được tạo ra bởi bộ mã hóa dạng hình học 51003, bộ mã hóa thuộc tính 51004 và/hoặc bộ xử lý truyền 51005. Bộ xử lý báo hiệu 51002 có thể cung cấp thông tin (ví dụ, thông tin hướng của đầu và/hoặc thông tin cổng nhìn) được phản hồi từ thiết bị nhận cho bộ mã hóa dạng hình học 51003, bộ mã hóa thuộc tính 51004, và/hoặc bộ xử lý truyền 51005.

Trong sáng chế này, thông tin báo hiệu có thể được báo hiệu và được truyền trong các đơn vị là các tập tham số (tập tham số trình tự (SPS), tập tham số dạng hình học (GPS), tập tham số thuộc tính (APS), tập tham số ô xếp (TPS) (hoặc bản kiểm kê ô xếp), v.v.). Ngoài ra, nó có thể được báo hiệu và được truyền trong các đơn vị lập mã của mỗi ảnh, chẳng hạn như các lát hoặc các ô xếp.

Theo các phương án, bộ xử lý báo hiệu 51002 có thể báo hiệu thông tin liên quan đến việc nén thuộc tính dựa trên LoD được mô tả trên đây cho ít nhất là APS hoặc tiêu đề lát thuộc tính để cung cấp thông tin này cho bộ mã hóa dạng hình học 51003, bộ mã hóa thuộc tính 51004 và/hoặc bộ xử lý truyền 51005.

Phương pháp/thiết bị theo các phương án có thể báo hiệu thông tin liên quan để bổ sung thêm/thực hiện các hoạt động của các phương án. Thông tin báo hiệu theo các phương án có thể được sử dụng bởi thiết bị truyền và/hoặc thiết bị nhận.

Bộ xử lý truyền 51005 có thể thực hiện hoạt động và/hoặc phương pháp truyền giống hoặc tương tự với hoạt động và/hoặc phương pháp truyền của bộ xử lý truyền 12012 trên Fig.12, và thực hiện hoạt động và/hoặc phương pháp truyền giống hoặc tương tự với hoạt động và/hoặc phương pháp truyền của bộ truyền 1003 trên Fig.1. Để biết chi tiết, hãy tham khảo phần mô tả trên Fig.1 hoặc Fig.12. Phần mô tả của chúng sẽ được lược bỏ ở đây.

Bộ xử lý truyền 51005 có thể truyền luồng bit dạng hình học được xuất ra từ bộ mã hóa dạng hình học 51003, luồng bit thuộc tính được xuất ra từ bộ mã hóa thuộc tính 51004, và luồng bit báo hiệu được xuất ra từ bộ xử lý báo hiệu 51002 bằng cách dồn kênh các luồng bit này thành một luồng bit hoặc bằng cách bao bọc các luồng bit này trong tệp tin hoặc phân đoạn. Trong một phương án của sáng chế, tệp tin này có thể nằm ở định dạng tệp tin ISOBMFF.

Theo các phương án, tệp tin hoặc phân đoạn có thể được truyền đến thiết bị nhận hoặc được lưu trữ trong phương tiện lưu trữ kỹ thuật số (ví dụ, USB, SD, CD, DVD, Blu-ray, HDD, SSD, v.v.). Bộ xử lý truyền 51005 theo các phương án có khả năng thực hiện việc truyền thông có dây/không dây với thiết bị nhận qua mạng 4G, 5G, 6G, v.v. Thêm vào đó, bộ xử lý truyền 51005 có thể thực hiện hoạt động xử lý dữ liệu cần thiết tùy theo hệ thống mạng (ví dụ, hệ thống mạng truyền thông 4G, 5G hoặc 6G). Ngoài ra, bộ xử lý truyền 51005 có thể truyền dữ liệu được bao bọc theo cách tùy theo nhu cầu.

Fig.46 là sơ đồ khối chi tiết mang tính ví dụ về bộ mã hóa thuộc tính 51004 theo các phương án. Các phần tử của bộ mã hóa thuộc tính được thể hiện trên Fig.46 có thể được thi hành dưới dạng phần cứng, phần mềm, bộ xử lý và/hoặc sự kết hợp của chúng v.v

Theo các phương án, bộ mã hóa thuộc tính 51004 có thể bao gồm bộ tạo ra LoD 53001, bộ nén thuộc tính 53002, bộ biến đổi thuộc tính 53003, và bộ lượng tử hóa 53004.

Bộ tạo ra LoD 53001 tạo ra các LoD dựa trên các vị trí mà việc mã hóa dạng hình học chưa được thực hiện trên đó và/hoặc thông tin dạng hình học được xây dựng lại. Bộ tạo ra LoD 53001 tạo ra các LoD qua một hoặc sự kết hợp của hai hay hơn hai phương án trong số các phương án 1 đến 3 đã được mô tả trên đây dựa trên Fig.16 đến Fig.30.

Tức là, bộ tạo ra LoD 53001 tạo ra các LoD dựa trên cấu trúc cây dạng hình học như được mô tả trên đây, và thực hiện việc lấy mẫu con bằng cách định ra nút của độ sâu dạng hình học tương ứng làm lân cận (hoặc khoảng tìm kiếm lân cận) khi tạo ra các LoD. Sau đó, cây dạng hình học được xây dựng và thông tin kích thước nút theo lớp LoD được báo hiệu qua thông tin liên quan đến việc nén thuộc tính dựa trên LoD, ví dụ, trường `num_LoD_minus1`, trường `cubic_node_flag`, trường `node_direction`, trường `node_size_x`, trường `node_size_y`, và trường `node_size_z`. Thêm vào đó, khi bộ tạo ra LoD 53001 được tạo cấu hình để thực hiện việc tạo ra LoD dựa trên cây bát phân sử dụng cây dạng hình học, thông tin liên quan có thể được báo hiệu qua thông tin liên quan đến việc nén thuộc tính dựa trên LoD, ví dụ, trường `constrained_QTBT_flag` và trường `max_LoD_depth_for_octree_based_LoD_generation`. Ngoài ra, phương pháp lấy mẫu LoD có thể được phân phát qua thông tin liên quan đến việc nén thuộc tính dựa trên LoD, ví dụ, trường `LoD_sampling_method`. Để biết chi tiết về trường `num_LoD_minus1`, trường `cubic_node_flag`, trường `node_direction`, trường

node_size_x, trường node_size_y, trường node_size_z, trường constrained_QTBT_flag, trường max_LoD_depth_for_octree_based_LoD_generation, và trường LoD_sampling_method được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD, hãy tham khảo phần mô tả về Fig.37.

Bộ nén thuộc tính 53002 thực hiện việc dự đoán thuộc tính dựa trên các LoD được tạo ra bởi bộ tạo ra LoD 53001, và tạo ra thông tin thuộc tính phần dư dựa trên thông tin thuộc tính được dự đoán. Thông tin thuộc tính phần dư được biến đổi vào trong miền nén bởi bộ biến đổi thuộc tính 53003 và sau đó được lượng tử hoá bởi bộ lượng tử hóa 53004. Sau đó, thông tin thuộc tính phần dư được lượng tử hoá được lập mã entropy và được xuất ra dưới dạng luồng bit thuộc tính. Thông tin thuộc tính phần dư có ý nghĩa giống với thông tin thuộc tính sai số dự đoán.

Fig.47 là lưu đồ minh họa phương pháp tạo ra LoD dựa trên cây dạng hình học mang tính ví dụ theo các phương án.

Trước hết, giả sử rằng thông tin liên quan đến việc nén thuộc tính dựa trên LoD chứa thông tin về cấu trúc cây dạng hình học dựa trên OT hoặc dựa trên OT/QT/BT, được sử dụng bởi bộ mã hóa dạng hình học 51003 để nén thông tin dạng hình học, được cung cấp cho bộ mã hóa thuộc tính 51004 qua bộ xử lý báo hiệu 51002. Ngoài ra, giả sử rằng độ sâu của cây dạng hình học là N.

Trong hoạt động 55001, khi dữ liệu đám mây điểm, thông tin liên quan đến việc nén thuộc tính dựa trên LoD chứa thông tin về cây dạng hình học, và thông tin độ sâu (N) được nhập vào, xác định được rằng liệu giá trị của trường constrained_QTBT_flag được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD có là 1 hay không.

Khi giá trị của trường constrained_QTBT_flag là 1, quy trình diễn tiến đến hoạt động 55002. Khi giá trị của trường constrained_QTBT_flag là 0, quy trình diễn tiến đến hoạt động 55003.

Khi giá trị của trường constrained_QTBT_flag là 1, nó có thể chỉ ra rằng có ràng buộc trong việc tạo ra cấu trúc cây dạng hình học. Ở đây, ràng buộc này có thể là điều kiện đối với việc lập mã thuộc tính có thể định tỷ lệ được, và có thể chỉ ra rằng việc tạo ra LoD dựa trên OT có thể được sử dụng đối với độ sâu cụ thể hoặc độ sâu thấp hơn. Ngoài ra, giá trị của trường constrained_QTBT_flag bằng 1 có thể được sử dụng để chỉ

ra rằng thiết bị nhận có thể sử dụng việc tạo ra LoD dựa trên OT. Giá trị của trường `constrained_QTBT_flag` bằng 0 có thể được sử dụng để chỉ ra rằng thiết bị nhận cần phải sử dụng việc tạo ra LoD dựa trên cây dạng hình học.

Trong hoạt động 55002, các nút lập phương được tạo ra bởi việc phân vùng dựa trên OT được tạo cấu hình cho việc tìm kiếm lân cận tại độ sâu tương ứng.

Trong hoạt động 55003, quy trình diễn tiến đến hoạt động 55004 hoặc hoạt động 55005 tùy thuộc vào các giá trị của trường `cubic_node_flag` và trường `node_direction` được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD.

`cubic_node_flag` bằng 1 có thể chỉ ra rằng nút đơn vị được sử dụng cho việc tìm kiếm lân cận trong việc tạo ra LoD là nút có hình dạng khối lập phương có cùng chiều dài trên các cạnh xyz. `cubic_node_flag` bằng 0 có thể chỉ ra rằng nút đơn vị được sử dụng cho việc tìm kiếm lân cận trong việc tạo ra LoD là ở hình dạng phỏng lập phương có các chiều dài khác nhau trên các cạnh xyz. Trường `node_direction` có thể chỉ ra sự định hướng (cụ thể là, hướng của việc phân vùng) của nút có hình dạng phỏng lập phương. Theo cách khác, kích thước của nút theo các hướng trục xyz có thể được chỉ ra thông qua các trường `node_size_x`, `node_size_y`, và `node_size_z`.

Theo các phương án, các giá trị của trường `node_direction`, 0 đến 5, có thể lần lượt chỉ ra những sự định hướng z, y, x, yz, xz, và xy của các nút. Ví dụ, khi kích thước của nút trên các trục được biểu diễn dưới dạng 2^{dx} , 2^{dy} , và 2^{dz} , (k, k, 2k), (k, 2k, k), (2k, k, k), (k, 2k, 2k), (2k, k, 2k), và (2k, 2k, k) có thể được thu, ở đây, k biểu thị kích thước đơn vị.

Theo đó, khi giá trị của trường `cubic_node_flag` là 1, quy trình diễn tiến đến hoạt động 55002 dành cho việc tạo ra LoD dựa trên OT. Khi giá trị của trường `cubic_node_flag` là 0 và giá trị của trường `node_direction` là một giá trị trong số 3, 4, và 5, quy trình diễn tiến đến hoạt động 55004 dành cho việc tạo ra LoD dựa trên QT. Thêm vào đó, khi giá trị của trường `cubic_node_flag` là 0 và giá trị của trường `node_direction` là một giá trị trong số 0, 1, và 2, quy trình diễn tiến đến hoạt động 55005 dành cho việc tạo ra LoD dựa trên BT.

Trong hoạt động 55004, các nút phỏng lập phương được tạo ra bởi việc phân vùng dựa trên QT được tạo cấu hình cho việc tìm kiếm lân cận tại độ sâu tương ứng.

Trong hoạt động 55005, các nút phỏng lập phương được tạo ra bởi việc phân vùng

dựa trên BT được tạo cấu hình cho việc tìm kiếm lân cận tại độ sâu tương ứng.

Trong hoạt động 55006, điểm thuộc về LoD được chọn từ các nút lập phương hoặc các nút phỏng lập phương, được cung cấp từ một hoạt động trong số các hoạt động 55002, 55004, và 55005, dựa trên giá trị của trường `LoD_sampling_method` được chứa trong thông tin liên quan đến việc nén thuộc tính dựa trên LoD.

Ví dụ, trong hoạt động 55006, khi giá trị của trường `LoD_sampling_method` là 0, điểm thứ nhất có thể được chọn. Khi giá trị này là 1, các điểm thứ nhất và cuối cùng có thể được chọn luân phiên trong các LoD. Khi giá trị này là 2, điểm có khoảng cách Euclide nhỏ so với tâm lân cận có thể được chọn.

Trong hoạt động 55007, độ sâu hiện tại được kiểm tra. Khi có độ sâu còn lại cho việc tạo ra LoD, biến số độ sâu N được giảm đi 1 (hoạt động 55008), quy trình diễn tiến đến hoạt động 55001. Khi việc tạo ra LoD được hoàn thành đối với tất cả các độ sâu, việc tạo ra LoD được kết thúc.

Như vậy, khi giá trị của trường `constrained_QTBT_flag` là 0 (hoặc giá trị 1 có thể được sử dụng tùy theo việc chọn của thiết bị nhận), hình dạng của nút tùy theo độ sâu của cây dạng hình học được thay đổi thành khối lập phương hoặc phỏng lập phương. Một khi hình dạng nút (việc tìm kiếm lân cận) được xác định qua việc báo hiệu thông tin liên quan đến việc nén thuộc tính dựa trên LoD, điểm tương ứng với LoD được chọn cho mỗi nút. Trong hoạt động này, phương pháp được báo hiệu trong `LOD_sampling_method` có thể được sử dụng. Thông qua việc thực hiện quy trình này lên đến mức gốc hoặc độ sâu tương ứng với $N - \text{max_LoD_depth_fr_octree_based_LoD_generation}$, các LoD có thể được tạo ra.

Fig.48 minh họa một ví dụ khác về thiết bị nhận đám mây điểm theo các phương án. Các phần tử của thiết bị nhận đám mây điểm được thể hiện trên Fig.48 có thể được thi hành dưới dạng phần cứng, phần mềm, bộ xử lý và/hoặc sự kết hợp của chúng.

Theo các phương án, thiết bị nhận đám mây điểm có thể bao gồm bộ xử lý nhận 61001, bộ xử lý báo hiệu 61002, bộ giải mã dạng hình học 61003, bộ giải mã thuộc tính 61004, và bộ hậu xử lý 61005.

Bộ xử lý nhận 61001 theo các phương án có thể nhận một luồng bit, hoặc có thể nhận lần lượt luồng bit dạng hình học, luồng bit thuộc tính, và luồng bit báo hiệu. Khi tệp tin và/hoặc phân đoạn được nhận, bộ xử lý nhận 61001 có thể khử bao bọc tệp tin

và/hoặc phân đoạn nhận được và xuất ra luồng bit.

Khi một luồng bit được nhận (hoặc được khử bao bọc), bộ xử lý nhận 61001 theo các phương án có thể phân kênh luồng bit dạng hình học, luồng bit thuộc tính, và/hoặc luồng bit báo hiệu từ một luồng bit, và xuất ra luồng bit báo hiệu được phân kênh đến bộ xử lý báo hiệu 61002, xuất ra luồng bit dạng hình học được phân kênh đến bộ giải mã dạng hình học 61003, và xuất ra luồng bit thuộc tính được phân kênh đến bộ giải mã thuộc tính 61004.

Khi luồng bit dạng hình học, luồng bit thuộc tính, và/hoặc luồng bit báo hiệu được nhận (hoặc được khử bao bọc), tương ứng, bộ xử lý nhận 61001 theo các phương án có thể truyền luồng bit báo hiệu đến bộ xử lý báo hiệu 61002, truyền luồng bit dạng hình học đến bộ giải mã dạng hình học 61003, và truyền luồng bit thuộc tính đến bộ giải mã thuộc tính 61004.

Bộ xử lý báo hiệu 61002 có thể phân tích cú pháp và xử lý thông tin được chứa trong thông tin báo hiệu, ví dụ, SPS, GPS, APS, TPS, lý lịch dữ liệu, v.v., từ luồng bit báo hiệu đầu vào và cung cấp chúng cho bộ giải mã dạng hình học 61003, bộ giải mã thuộc tính 61004, và bộ hậu xử lý 61005. Trong một phương án khác, thông tin báo hiệu được chứa trong tiêu đề lát dạng hình học và/hoặc tiêu đề lát thuộc tính cũng có thể được phân tích cú pháp bởi bộ xử lý báo hiệu 61002 trước khi giải mã dữ liệu lát tương ứng.

Theo các phương án, bộ xử lý báo hiệu 61002 có thể phân tích cú pháp và xử lý ít nhất là thông tin liên quan đến việc nén thuộc tính dựa trên LoD được báo hiệu trong APS hoặc tiêu đề lát thuộc tính, và cung cấp thông tin này cho bộ giải mã dạng hình học 61003, bộ giải mã thuộc tính 61004, và bộ hậu xử lý 61005.

Theo các phương án, bộ giải mã dạng hình học 61003 có thể xây dựng lại dạng hình học bằng cách thực hiện quy trình ngược của hoạt động của bộ mã hóa dạng hình học 51003 trên Fig.45 đối với luồng bit dạng hình học được nén dựa trên thông tin báo hiệu. Thông tin dạng hình học được xây dựng lại (hoặc được khôi phục) bởi bộ giải mã dạng hình học 61003 được cung cấp cho bộ giải mã thuộc tính 61004. Bộ giải mã thuộc tính 61004 có thể xây dựng lại các thuộc tính bằng cách thực hiện quy trình ngược của hoạt động của bộ mã hóa thuộc tính 51004 trên Fig.45 đối với luồng bit thuộc tính được nén dựa trên thông tin báo hiệu và thông tin dạng hình học được xây dựng lại này.

Theo các phương án, bộ hậu xử lý 61005 có thể so khớp thông tin dạng hình học

(cụ thể là, các vị trí) được xây dựng lại và được xuất ra bởi bộ giải mã dạng hình học 61003 với thông tin thuộc tính được xây dựng lại và được xuất ra bởi bộ giải mã thuộc tính 61004 để xây dựng lại và hiển thị/kết xuất dữ liệu đám mây điểm.

Fig.49 là sơ đồ khối chi tiết mang tính ví dụ minh họa bộ giải mã thuộc tính 61004 theo các phương án. Các phần tử của bộ giải mã thuộc tính được thể hiện trên Fig.49 có thể được thi hành dưới dạng phần cứng, phần mềm, bộ xử lý và/hoặc sự kết hợp của chúng.

Theo các phương án, bộ giải mã thuộc tính 61004 có thể bao gồm bộ giải lượng tử hóa 63001, bộ biến đổi ngược 63002, bộ tạo ra LoD 63003, và bộ xây dựng lại thuộc tính 63004.

Theo các phương án, bộ giải mã thuộc tính 61004 xây dựng lại thông tin thuộc tính bằng cách thực hiện quy trình ngược của hoạt động của bộ mã hóa thuộc tính 51004 của thiết bị truyền. Nói cách khác, bộ giải lượng tử hóa 63001 có thể giải mã entropy trên thông tin thuộc tính phần dư được chứa trong luồng bit thuộc tính, và sau đó giải lượng tử hóa thông tin này. Sau đó, thông tin thuộc tính phần dư được giải lượng tử hóa được xử lý bởi bộ biến đổi ngược 63002 qua quy trình ngược của hoạt động của bộ biến đổi thuộc tính 53003, và sau đó được xuất ra đến bộ xây dựng lại thuộc tính 63004.

Bộ xây dựng lại thuộc tính 63004 thực hiện việc dự đoán thuộc tính dựa trên các LoD được tạo ra bởi bộ tạo ra LoD 63003, và xây dựng lại thông tin thuộc tính dựa trên thông tin thuộc tính được dự đoán và thông tin thuộc tính phần dư được xuất ra từ bộ biến đổi ngược 63002.

Bộ tạo ra LoD 63003 tạo ra các LoD theo cách giống hoặc tương tự với phương pháp tạo ra LoD cho bộ tạo ra LoD 53001 của thiết bị truyền. Ví dụ, bộ tạo ra LoD 63003 có thể tạo ra các LoD dựa trên cấu trúc cây dạng hình học được mô tả với tham chiếu đến Fig.16 đến Fig.30.

Theo các phương án, bộ tạo ra LoD 63003 có thể sử dụng các loại việc tạo ra LoD khác nhau dựa trên giá trị của trường `constrained_QTBT_flag`. Ví dụ, khi giá trị của trường `constrained_QTBT_flag` là 1, bộ tạo ra LoD 63003 có thể thực hiện việc tạo ra LoD dựa trên OT hoặc việc tạo ra LoD dựa trên OT/QT/BT. Mức độ sâu cho phép phép biểu diễn có thể định tỷ lệ được có thể được xác định qua trường `max_LoD_depth_for_octree_based_LoD_generation`. Ví dụ, khi giá trị của trường

constrained_QTBT_flag là 0, bộ tạo ra LoD 63003 có thể thực hiện chỉ việc tạo ra LoD dựa trên cây dạng hình học được tạo ra bởi sự kết hợp của OT/QT/BT, và xây dựng cây dạng hình học. Thông tin kích thước nút theo lớp LoD có thể được xác định dựa trên các trường num_LoD_minus1, cubic_node_flag, node_direction, node_size_x, node_size_y, và node_size_z. Theo các phương án, việc liệu LoD có thể được sử dụng trong phép biểu diễn có thể định tỷ lệ được hay không (tức là, việc liệu lớp dạng hình học có khớp với lớp thuộc tính hay không) có thể được xác định dựa trên trường no_representation_flag. Ví dụ, LoD có thể được sử dụng trong phép biểu diễn có thể định tỷ lệ được chỉ khi giá trị của trường no_representation_flag là 0.

Như vậy, khi việc nén dạng hình học được thực hiện bởi thiết bị truyền bằng cách tạo ra cây dạng hình học dựa trên OT/QT/BT, thông tin thuộc tính cũng có thể được nén bằng cách tạo ra các LoD dựa trên OT/QT/BT. Qua đó, số lượng điểm đối với dạng hình học có thể được khớp với số lượng điểm đối với thuộc tính.

Fig.50 là lưu đồ minh họa phương pháp truyền dữ liệu đám mây điểm theo các phương án.

Phương pháp truyền dữ liệu đám mây điểm theo các phương án có thể bao gồm các bước thu nhận dữ liệu đám mây điểm (71001), mã hóa dữ liệu đám mây điểm này (71002), và truyền dữ liệu đám mây điểm được mã hóa này và thông tin báo hiệu (71003). Trong trường hợp này, luồng bit chứa dữ liệu đám mây điểm được mã hóa và thông tin báo hiệu này có thể được bao bọc vào trong tệp tin và được truyền.

Trong hoạt động 71001 gồm việc thu nhận dữ liệu đám mây điểm, một phần hoặc toàn bộ các hoạt động của đơn vị thu nhận video đám mây điểm 10001 trên Fig.1 có thể được thực hiện, hoặc một phần hoặc toàn bộ các hoạt động của đơn vị đầu vào dữ liệu 12000 trên Fig.12 có thể được thực hiện.

Trong hoạt động 71002 gồm việc mã hóa dữ liệu đám mây điểm, một phần hoặc toàn bộ các hoạt động của bộ mã hóa video đám mây điểm 10002 trên Fig.1, việc mã hóa 20001 trên Fig.2, bộ mã hóa video đám mây điểm trên Fig.4, bộ mã hóa video đám mây điểm trên Fig.12, việc mã hóa đám mây điểm trên Fig.14, việc mã hóa đám mây điểm trên Fig.15, và bộ mã hóa video đám mây điểm trên Fig.18 có thể được thực hiện.

Hoạt động 71002 gồm việc mã hóa dữ liệu đám mây điểm 71002 theo các phương án có thể bao gồm việc mã hóa dữ liệu dạng hình học (hoặc được gọi là thông tin dạng

hình học) của dữ liệu đám mây điểm và mã hóa dữ liệu thuộc tính (hoặc được gọi là thông tin thuộc tính) của dữ liệu đám mây điểm.

Trong hoạt động gồm việc mã hóa thông tin dạng hình học, dữ liệu đám mây điểm được thu nhận được phân vùng dựa trên OT hoặc OT/QT/BT để nén thông tin dạng hình học. Để biết những chi tiết về việc tạo ra cây dạng hình học bằng cách phân vùng dữ liệu đám mây điểm dựa trên OT hoặc OT/QT/BT trong hoạt động gồm việc mã hóa thông tin dạng hình học này, hãy tham khảo phần mô tả về Fig.16 đến Fig.30. Phần mô tả của chúng sẽ được lược bỏ ở đây. Trong hoạt động gồm việc mã hóa thông tin dạng hình học, thông tin dạng hình học được xây dựng lại dựa trên các vị trí được thay đổi qua việc nén, và thông tin dạng hình học được xây dựng lại (hoặc được giải mã) được xuất ra đến hoạt động gồm việc mã hóa thông tin thuộc tính. Trong hoạt động gồm việc mã hóa thông tin thuộc tính, thông tin thuộc tính được nén dựa trên thông tin dạng hình học được xây dựng lại và/hoặc các vị trí mà việc mã hóa dạng hình học chưa được thực hiện trên chúng. Trong hoạt động gồm việc mã hóa thông tin thuộc tính, việc mã hóa thông tin thuộc tính hoặc việc mã hóa thuộc tính có thể định tỷ lệ được có thể được thực hiện qua một trong số hoặc sự kết hợp của hai hay hơn hai phương án trong số các phương án 1 đến 3 được mô tả trên đây. Việc mã hóa thông tin thuộc tính theo các phương án có thể được thực hiện theo lưu đồ trên Fig.47.

Trong sáng chế này, thông tin báo hiệu có thể là SPS, GPS, APS, TPS, lý lịch dữ liệu, hoặc dạng tương tự, và tiêu đề lát dạng hình học và/hoặc tiêu đề lát thuộc tính cũng có thể được gọi là thông tin báo hiệu. Thông tin liên quan đến việc nén thuộc tính dựa trên LoD cần thiết cho việc nén dạng hình học và nén thông tin thuộc tính có thể được báo hiệu ít nhất là trong APS hoặc tiêu đề lát thuộc tính. Thông tin liên quan đến việc nén thuộc tính dựa trên LoD đã được mô tả chi tiết trên đây, và do đó phần mô tả của nó sẽ được lược bỏ.

Fig.51 là lưu đồ minh họa phương pháp nhận dữ liệu đám mây điểm theo các phương án.

Phương pháp nhận dữ liệu đám mây điểm theo các phương án có thể bao gồm các bước nhận dữ liệu đám mây điểm và thông tin báo hiệu (81001), giải mã dữ liệu đám mây điểm này dựa trên thông tin báo hiệu này (81002), và kết xuất dữ liệu đám mây điểm được giải mã này (81003).

Hoạt động 81001 gồm việc nhận dữ liệu đám mây điểm và thông tin báo hiệu theo các phương án có thể được thực hiện bởi bộ nhận 10005 trên Fig.1, việc truyền 20002 hoặc việc giải mã 20003 trên Fig.2, hoặc bộ nhận 13000 hoặc bộ xử lý nhận 13001 trên Fig.13.

Hoạt động 81002 gồm việc giải mã dữ liệu đám mây điểm dựa trên thông tin báo hiệu có thể bao gồm việc giải mã thông tin dạng hình học và giải mã thông tin thuộc tính dựa trên dạng hình học được giải mã/được xây dựng lại và thông tin báo hiệu. Thông tin báo hiệu có thể là SPS, GPS, APS, TPS, lý lịch dữ liệu, hoặc dạng tương tự, và tiêu đề lát dạng hình học và/hoặc tiêu đề lát thuộc tính cũng có thể được gọi là thông tin báo hiệu. Ít nhất là APS hoặc tiêu đề lát thuộc tính có thể bao gồm thông tin liên quan đến việc nén thuộc tính dựa trên LoD. Thông tin liên quan đến việc nén thuộc tính dựa trên LoD đã được mô tả chi tiết trên đây, và do đó phần mô tả của nó sẽ được lược bỏ.

Trong hoạt động gồm việc giải mã thông tin dạng hình học theo các phương án, một phần hoặc toàn bộ các hoạt động của bộ giải mã video đám mây điểm 10006 trên Fig.1, việc giải mã 20003 trên Fig.2, bộ giải mã video đám mây điểm trên Fig.11, bộ giải mã video đám mây điểm trên Fig.13, và bộ giải mã dạng hình học trên Fig.48 có thể được thực hiện.

Trong hoạt động gồm việc giải mã thông tin thuộc tính theo các phương án, một phần hoặc toàn bộ các hoạt động của bộ giải mã video đám mây điểm 10006 trên Fig.1, việc giải mã 20003 trên Fig.2, bộ giải mã video đám mây điểm trên Fig.11, bộ giải mã video đám mây điểm trên Fig.13, và bộ giải mã thuộc tính trên Fig.48 có thể được thực hiện.

Trong hoạt động kết xuất 81003 theo các phương án, dữ liệu đám mây điểm được xây dựng lại dựa trên thông tin dạng hình học được giải mã và thông tin thuộc tính được giải mã có thể được kết xuất theo các phương pháp kết xuất khác nhau. Ví dụ, các điểm của nội dung đám mây điểm có thể được kết xuất dưới dạng đỉnh có độ dày nhất định, khối lập phương có kích thước tối thiểu cụ thể được định tâm trên vị trí của đỉnh này, hoặc hình tròn được định tâm trên vị trí của đỉnh này. Toàn bộ hoặc một phần các vùng của nội dung đám mây điểm được kết xuất có thể được cung cấp cho người dùng qua bộ hiển thị (ví dụ, bộ hiển thị VR/AR, bộ hiển thị thông thường, v.v.). Hoạt động 81003 gồm việc kết xuất dữ liệu đám mây điểm có thể được thực hiện bởi bộ kết xuất 10007

trên Fig.1, việc kết xuất 20004 trên Fig.2, hoặc bộ kết xuất 13011 trên Fig.13.

Phương pháp tạo ra LoD dựa trên cây dạng hình học được mô tả trong sáng chế này là phương pháp phân chia dữ liệu đám mây điểm và có thể được sử dụng cho việc lập mã thuộc tính. Cụ thể, khi việc lập mã dạng hình học dựa trên QTBT được thực hiện trong các ứng dụng lập mã có thể định tỷ lệ được, thông tin thuộc tính có thể được so khớp với các điểm/nút tương ứng theo kiểu phép tương ứng một-một theo độ sâu của cây dạng hình học. Qua đó, phép biểu diễn có thể định tỷ lệ được có thể được cho phép đối với tất cả các độ sâu hoặc một số độ sâu trùng khớp.

Sáng chế đề xuất phương pháp tạo ra LOD để gia tăng hiệu quả nén khi dữ liệu đám mây điểm có phân bố bất đối xứng. Theo các hoạt động của các phương án được mô tả trong sáng chế này, việc lập mã thuộc tính có thể định tỷ lệ được có thể được thực hiện trên dữ liệu đám mây điểm được phân bố bất đối xứng. Cụ thể hơn, LOD có thể được tạo ra có xem xét đến các nút phỏng lập phương bất đối xứng, và việc lập mã thuộc tính có thể được thực hiện dựa trên sự thay đổi về nút tùy theo lớp có xem xét đến việc nén dạng hình học dựa trên QTBT. Hơn nữa, phương pháp/thiết bị truyền/nhận theo các phương án có thể thực hiện việc nén thuộc tính có thể định tỷ lệ được dựa trên OT. Kết quả là, hiệu quả nén dữ liệu đám mây điểm có thể được gia tăng.

Mỗi bộ phận, môđun, hoặc đơn vị được mô tả trên đây có thể là bộ phận phần mềm, bộ xử lý, hoặc phần cứng mà chúng thực thi các thủ tục liên tiếp được lưu trữ trong bộ nhớ (hoặc đơn vị lưu trữ). Mỗi bước trong số các bước được mô tả trong các phương án trên đây có thể được thực hiện bởi các bộ phận của bộ xử lý, phần mềm, hoặc phần cứng. Mỗi môđun/khối/đơn vị được mô tả trong các phương án trên đây có thể vận hành dưới dạng bộ xử lý, phần mềm, hoặc phần cứng. Thêm vào đó, các phương pháp được trình bày bởi các phương án có thể được thực thi dưới dạng mã. Mã này có thể được viết trên phương tiện lưu trữ đọc được bằng bộ xử lý và do đó được đọc bằng bộ xử lý được trang bị bởi thiết bị.

Trong bản mô tả này, khi một bộ phận “chứa” hoặc “bao gồm” một phần tử, điều này có nghĩa là bộ phận này còn chứa hoặc còn bao gồm phần tử khác trừ khi được nêu theo cách khác. Ngoài ra, thuật ngữ “... môđun (hoặc đơn vị)” được bộc lộ trong bản mô tả này nghĩa là đơn vị cho việc xử lý ít nhất một chức năng hoặc hoạt động, và có thể được thi hành bởi phần cứng, phần mềm hoặc sự kết hợp của phần cứng và phần mềm.

Mặc dù các phương án đã được giải thích với tham chiếu đến mỗi hình vẽ trong số các hình vẽ kèm theo để cho đơn giản, có thể thiết kế các phương án mới bằng cách hợp nhất các phương án được minh họa trên các hình vẽ kèm theo. Nếu phương tiện ghi có thể đọc được bằng máy tính, mà các chương trình để thực thi các phương án được nêu trong phần mô tả ở trên được ghi nhận trong đó, được thiết kế bởi những người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng thì nó có thể nằm trong phạm vi của các điểm yêu cầu bảo hộ kèm theo và các nội dung tương đương của chúng.

Các thiết bị và các phương pháp có thể không bị giới hạn ở các cấu hình và các phương pháp của các phương án được mô tả trên đây. Các phương án được mô tả trên đây có thể được tạo cấu hình bằng cách được kết hợp một cách có chọn lọc toàn bộ hoặc một phần với nhau để cho phép các cải biến khác nhau.

Mặc dù các phương án ưu tiên đã được mô tả với tham chiếu đến các hình vẽ, những người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng sẽ hiểu rằng có các cải biến và biến thể khác nhau có thể được thực hiện trong các phương án mà không tách rời khỏi nguyên lý hoặc phạm vi của sáng chế được mô tả trong các yêu cầu bảo hộ kèm theo. Các cải biến như vậy không thể được hiểu theo cách riêng lẻ tách khỏi nguyên lý hoặc quan điểm kỹ thuật của các phương án.

Các phần tử khác nhau của các thiết bị trong các phương án có thể được thi hành bằng phần cứng, phần mềm, phần sụn, hoặc sự kết hợp của chúng. Các phần tử khác nhau trong các phương án có thể được thi hành bởi chip đơn, ví dụ, mạch phần cứng đơn. Theo các phương án, các thành phần theo các phương án có thể được thi hành dưới dạng các chip riêng rẽ, một cách tương ứng. Theo các phương án, ít nhất một hoặc nhiều thành phần trong số các thành phần của thiết bị theo các phương án có thể bao gồm một hoặc nhiều bộ xử lý có khả năng thực thi một hoặc nhiều chương trình. Một hoặc nhiều chương trình này có thể thực hiện một hoặc nhiều hoạt động/phương pháp bất kỳ trong số các hoạt động/các phương pháp theo các phương án hoặc bao gồm các lệnh để thực hiện các hoạt động/các phương pháp đó. Các lệnh có thể thực thi được để thực hiện phương pháp/các hoạt động của thiết bị theo các phương án có thể được lưu trữ trong CRM phi chuyển tiếp hoặc các sản phẩm chương trình máy tính khác được tạo cấu hình để được thực thi bởi một hoặc nhiều bộ xử lý, hoặc có thể được lưu trữ trong CRM chuyển tiếp hoặc các sản phẩm chương trình máy tính khác được tạo cấu hình để được

thực thi bởi một hoặc nhiều bộ xử lý. Thêm vào đó, bộ nhớ theo các phương án có thể được sử dụng như là khái niệm bao hàm không chỉ các bộ nhớ khả biến (ví dụ, RAM) mà cả các bộ nhớ bất khả biến, các bộ nhớ tốc độ nhanh, và các PROM. Thêm vào đó, nó cũng có thể được thi hành dưới dạng sóng mang, như là việc truyền qua Internet. Thêm vào đó, phương tiện ghi có thể đọc được bằng bộ xử lý có thể được phân tán cho các hệ thống máy tính được kết nối qua mạng sao cho mã đọc được bằng bộ xử lý có thể được lưu trữ và được thực thi theo kiểu phân tán. Trong sáng chế này, thuật ngữ “/” và “,” nên được diễn giải là biểu thị “và/hoặc.” Ví dụ, cách thể hiện “A/B” có thể nghĩa là “A và/hoặc B.” Hơn nữa, “A, B” có thể có nghĩa là “A và/hoặc B”. Hơn nữa, “A/B/C” có thể có nghĩa là “ít nhất một thành phần trong số A, B, và/hoặc C.” “A, B, C” cũng có thể nghĩa là “ít nhất một thành phần trong số A, B, và/hoặc C.”

Hơn nữa, trong sáng chế này, thuật ngữ “hoặc” cần được diễn giải là “và/hoặc.” Ví dụ, cách thể hiện “A hoặc B” có thể nghĩa là 1) chỉ A, 2) chỉ B, và/hoặc 3) cả A và B. Nói cách khác, thuật ngữ “hoặc” trong sáng chế này nên được diễn giải là “thêm vào hoặc thay vào đó.”

Các phần tử khác nhau của các phương án có thể được thi hành bằng phần cứng, phần mềm, phần sụn, hoặc sự kết hợp của chúng. Các phần tử khác nhau trong các phương án có thể được thực thi bởi chip đơn chẳng hạn như mạch phần cứng đơn. Theo các phương án, phần tử có thể được thực thi có chọn lọc tương ứng bởi các chip riêng rẽ. Theo các phương án, ít nhất một phần tử trong số các phần tử của các phương án có thể được thực thi trong một hoặc nhiều bộ xử lý chứa các lệnh để thực hiện các hoạt động theo các phương án.

Các thuật ngữ chẳng hạn như thứ nhất và thứ hai có thể được sử dụng để mô tả các phần tử khác nhau của các phương án. Tuy nhiên, các thành phần khác nhau theo các phương án không nên bị giới hạn ở các thuật ngữ trên. Các thuật ngữ này chỉ được sử dụng để phân biệt một phần tử với một phần tử khác. Ví dụ, tín hiệu đầu vào người dùng thứ nhất có thể được gọi là tín hiệu đầu vào người dùng thứ hai. Tương tự, tín hiệu đầu vào người dùng thứ hai có thể được gọi là tín hiệu đầu vào người dùng thứ nhất. Việc sử dụng các thuật ngữ này nên được hiểu là không tách rời khỏi phạm vi của các phương án khác nhau. Tín hiệu đầu vào người dùng thứ nhất và tín hiệu đầu vào người dùng thứ

hai đều là các tín hiệu đầu vào người dùng, nhưng không có nghĩa là cùng tín hiệu đầu vào người dùng trừ khi ngữ cảnh quy định một cách rõ ràng theo cách khác.

Hệ thuật ngữ được sử dụng để mô tả các phương án này chỉ được sử dụng cho mục đích mô tả các phương án cụ thể và không được dự tính là để giới hạn các phương án này. Như được sử dụng trong phần mô tả của các phương án và trong các điểm yêu cầu bảo hộ, các dạng số ít “a”, “an”, và “the” trong tiếng Anh bao gồm cả các tham chiếu dạng số nhiều trừ khi ngữ cảnh quy định một cách rõ ràng theo cách khác. Cách thể hiện “và/hoặc” được sử dụng để bao gồm tất cả các tổ hợp khả thi của các thuật ngữ. Các thuật ngữ như là “bao gồm” hoặc “có” được dự tính là để chỉ ra sự tồn tại của các hình, các số, các bước, các phần tử, và/hoặc các thành phần và cần phải được hiểu là không loại trừ khả năng có sự tồn tại của các hình, các số, các bước, các phần tử và/hoặc các thành phần bổ sung. Như được sử dụng ở đây, các cách thể hiện điều kiện chẳng hạn như “nếu” và “khi” không bị giới hạn ở trường hợp tùy chọn và được dự tính là để được diễn giải, khi một điều kiện cụ thể được thỏa mãn, nhằm thực hiện hoạt động liên quan hoặc diễn giải định nghĩa liên quan tùy theo điều kiện cụ thể này.

Chế độ để thực hiện sáng chế

Như được mô tả trên đây, các nội dung liên quan đã được mô tả tốt nhất để có thể tiến hành các phương án.

Khả năng ứng dụng công nghiệp

Như được mô tả trên đây, các phương án có thể được áp dụng một cách đầy đủ hoặc một phần cho thiết bị và hệ thống truyền/nhận dữ liệu đám mây điểm. Những người có hiểu biết số trung bình trong lĩnh vực kỹ thuật tương ứng sẽ hiểu rõ rằng những thay đổi hoặc cải biến khác nhau có thể được thực hiện đối với các phương án trong phạm vi của các phương án này. Vì thế, các phương án được nhằm để bao trùm các sửa đổi và cải biến đó đối với sáng chế này, miễn là chúng nằm trong phạm vi của các điểm yêu cầu bảo hộ kèm theo và các nội dung tương đương của chúng.

YÊU CẦU BẢO HỘ

1. Phương pháp truyền dữ liệu đám mây điểm, phương pháp này bao gồm các bước:

thu nhận dữ liệu đám mây điểm;

mã hóa thông tin dạng hình học bao gồm các vị trí của các điểm của dữ liệu đám mây điểm này bằng cách áp dụng cây bát phân;

mã hóa thông tin thuộc tính bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm này dựa trên cây bát phân này; và

truyền thông tin dạng hình học được mã hóa này, thông tin thuộc tính được mã hóa này, và thông tin báo hiệu,

trong đó bước mã hóa thông tin thuộc tính bao gồm các bước:

tạo ra các mức độ chi tiết (Level of Detail, LoD) dựa trên cây bát phân này; và

nén thông tin thuộc tính dựa trên các LoD này,

trong đó cây bát phân này được tạo ra bởi các độ sâu,

trong đó một độ sâu trong số các độ sâu này bao gồm một hoặc nhiều nút được chiếm giữ,

trong đó ít nhất một nút trong số một hoặc nhiều nút được chiếm giữ này bao gồm hai hay hơn hai nút con được chiếm giữ,

trong đó một nút con trong số hai hay hơn hai nút con được chiếm giữ này được chọn để tạo ra LoD,

trong đó việc chọn này được thực hiện trên mỗi nút trong số ít nhất một nút được chiếm giữ có hai hay hơn hai nút con được chiếm giữ này để tạo ra LoD tương ứng với độ sâu này, và

trong đó nút con được chiếm giữ được chọn đối với mỗi nút trong số ít nhất một nút được chiếm giữ này là nút thứ nhất hoặc nút cuối cùng trong số hai hay hơn hai nút con được chiếm giữ dựa trên việc liệu LoD là chẵn hay lẻ.

2. Phương pháp theo điểm 1, trong đó hai hay hơn hai nút con được chiếm giữ đã nêu, có cùng nút mẹ, được sắp xếp dựa trên thứ tự mã Morton.

3. Thiết bị truyền dữ liệu đám mây điểm, thiết bị này bao gồm:

bộ thu nhận được tạo cấu hình để thu nhận dữ liệu đám mây điểm;

bộ mã hóa dạng hình học được tạo cấu hình để mã hóa thông tin dạng hình học bao gồm các vị trí của các điểm của dữ liệu đám mây điểm này bằng cách áp dụng cây bát phân; và

bộ mã hóa thuộc tính được tạo cấu hình để mã hóa thông tin thuộc tính bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm này dựa trên cây bát phân này; và

bộ truyền được tạo cấu hình để truyền thông tin dạng hình học được mã hóa này, thông tin thuộc tính được mã hóa này, và thông tin báo hiệu,

trong đó bộ mã hóa thuộc tính bao gồm:

bộ tạo ra LoD được tạo cấu hình để tạo ra các mức độ chi tiết (LoD) dựa trên cây bát phân này; và

bộ nén được tạo cấu hình để nén thông tin thuộc tính dựa trên các LoD này,

trong đó cây bát phân này được tạo ra bởi các độ sâu,

trong đó một độ sâu trong số các độ sâu này bao gồm một hoặc nhiều nút được chiếm giữ,

trong đó ít nhất một nút trong số một hoặc nhiều nút được chiếm giữ này bao gồm hai hay hơn hai nút con được chiếm giữ,

trong đó một nút con trong số hai hay hơn hai nút con được chiếm giữ này được chọn để tạo ra LoD,

trong đó việc chọn này được thực hiện trên mỗi nút trong số ít nhất một nút được chiếm giữ có hai hay hơn hai nút con được chiếm giữ này để tạo ra LoD tương ứng với độ sâu này, và

trong đó nút con được chiếm giữ được chọn đối với mỗi nút trong số ít nhất một nút được chiếm giữ này là nút thứ nhất hoặc nút cuối cùng trong số hai hay hơn hai nút con được chiếm giữ dựa trên việc liệu LoD là chẵn hay lẻ.

4. Thiết bị theo điểm 3, trong đó hai hay hơn hai nút con được chiếm giữ đã nêu, có cùng nút mẹ, được sắp xếp dựa trên thứ tự mã Morton.

5. Phương pháp nhận dữ liệu đám mây điểm, phương pháp này bao gồm các bước:

nhận thông tin dạng hình học, thông tin thuộc tính, và thông tin báo hiệu;

giải mã thông tin dạng hình học này dựa trên thông tin báo hiệu này và cây bát phân;

giải mã thông tin thuộc tính dựa trên thông tin báo hiệu này và cây bát phân này;
và

kết xuất dữ liệu đám mây điểm được xây dựng lại dựa trên thông tin dạng hình học được giải mã và thông tin thuộc tính được giải mã,

trong đó thông tin dạng hình học được giải mã bao gồm các vị trí của các điểm của dữ liệu đám mây điểm được xây dựng lại,

trong đó thông tin thuộc tính được giải mã bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm được xây dựng lại,

trong đó bước giải mã thông tin thuộc tính bao gồm các bước:

tạo ra các mức độ chi tiết (Level of Detail, LoD) dựa trên cây bát phân này; và

giải mã thông tin thuộc tính dựa trên các LoD này,

trong đó cây bát phân này được tạo ra bởi các độ sâu,

trong đó một độ sâu trong số các độ sâu này bao gồm một hoặc nhiều nút được chiếm giữ,

trong đó ít nhất một nút trong số một hoặc nhiều nút được chiếm giữ này bao gồm hai hay hơn hai nút con được chiếm giữ,

trong đó một nút con trong số hai hay hơn hai nút con được chiếm giữ này được chọn để tạo ra LoD,

trong đó việc chọn này được thực hiện trên mỗi nút trong số ít nhất một nút được chiếm giữ có hai hay hơn hai nút con được chiếm giữ này để tạo ra LoD tương ứng với độ sâu này, và

trong đó nút con được chiếm giữ được chọn đối với mỗi nút trong số ít nhất một nút được chiếm giữ này là nút thứ nhất hoặc nút cuối cùng trong số hai hay hơn hai nút con được chiếm giữ dựa trên việc liệu LoD là chẵn hay lẻ.

6. Phương pháp theo điểm 5, trong đó hai hay hơn hai nút con được chiếm giữ đã nêu, có cùng nút mẹ, được sắp xếp dựa trên thứ tự mã Morton.

7. Thiết bị nhận dữ liệu đám mây điểm, thiết bị này bao gồm:

bộ nhận được tạo cấu hình để nhận thông tin dạng hình học, thông tin thuộc tính, và thông tin báo hiệu;

bộ giải mã dạng hình học được tạo cấu hình để giải mã thông tin dạng hình học này dựa trên thông tin báo hiệu này và cây bát phân;

bộ giải mã thuộc tính được tạo cấu hình để giải mã thông tin thuộc tính dựa trên thông tin báo hiệu này và cây bát phân này; và

bộ kết xuất được tạo cấu hình để kết xuất dữ liệu đám mây điểm được xây dựng lại dựa trên thông tin dạng hình học được giải mã và thông tin thuộc tính được giải mã, trong đó thông tin dạng hình học được giải mã bao gồm các vị trí của các điểm của dữ liệu đám mây điểm được xây dựng lại,

trong đó thông tin thuộc tính được giải mã bao gồm các giá trị thuộc tính của các điểm của dữ liệu đám mây điểm được xây dựng lại,

trong đó bộ giải mã thuộc tính bao gồm:

bộ tạo ra LoD được tạo cấu hình để tạo ra các mức độ chi tiết (LoD) dựa trên cây bát phân này; và

bộ giải mã được tạo cấu hình để giải mã thông tin thuộc tính dựa trên các LoD này, trong đó cây bát phân này được tạo ra bởi các độ sâu,

trong đó một độ sâu trong số các độ sâu này bao gồm một hoặc nhiều nút được chiếm giữ,

trong đó ít nhất một nút trong số một hoặc nhiều nút được chiếm giữ này bao gồm hai hay hơn hai nút con được chiếm giữ,

trong đó một nút con trong số hai hay hơn hai nút con được chiếm giữ này được chọn để tạo ra LoD,

trong đó việc chọn này được thực hiện trên mỗi nút trong số ít nhất một nút được chiếm giữ có hai hay hơn hai nút con được chiếm giữ này để tạo ra LoD tương ứng với độ sâu này, và

trong đó nút con được chiếm giữ được chọn đối với mỗi nút trong số ít nhất một nút được chiếm giữ này là nút thứ nhất hoặc nút cuối cùng trong số hai hay hơn hai nút con được chiếm giữ dựa trên việc liệu LoD là chẵn hay lẻ.

8. Thiết bị theo điểm 7, trong đó hai hay hơn hai nút con được chiếm giữ đã nêu, có cùng nút mẹ, được sắp xếp dựa trên thứ tự mã Morton.

FIG. 1

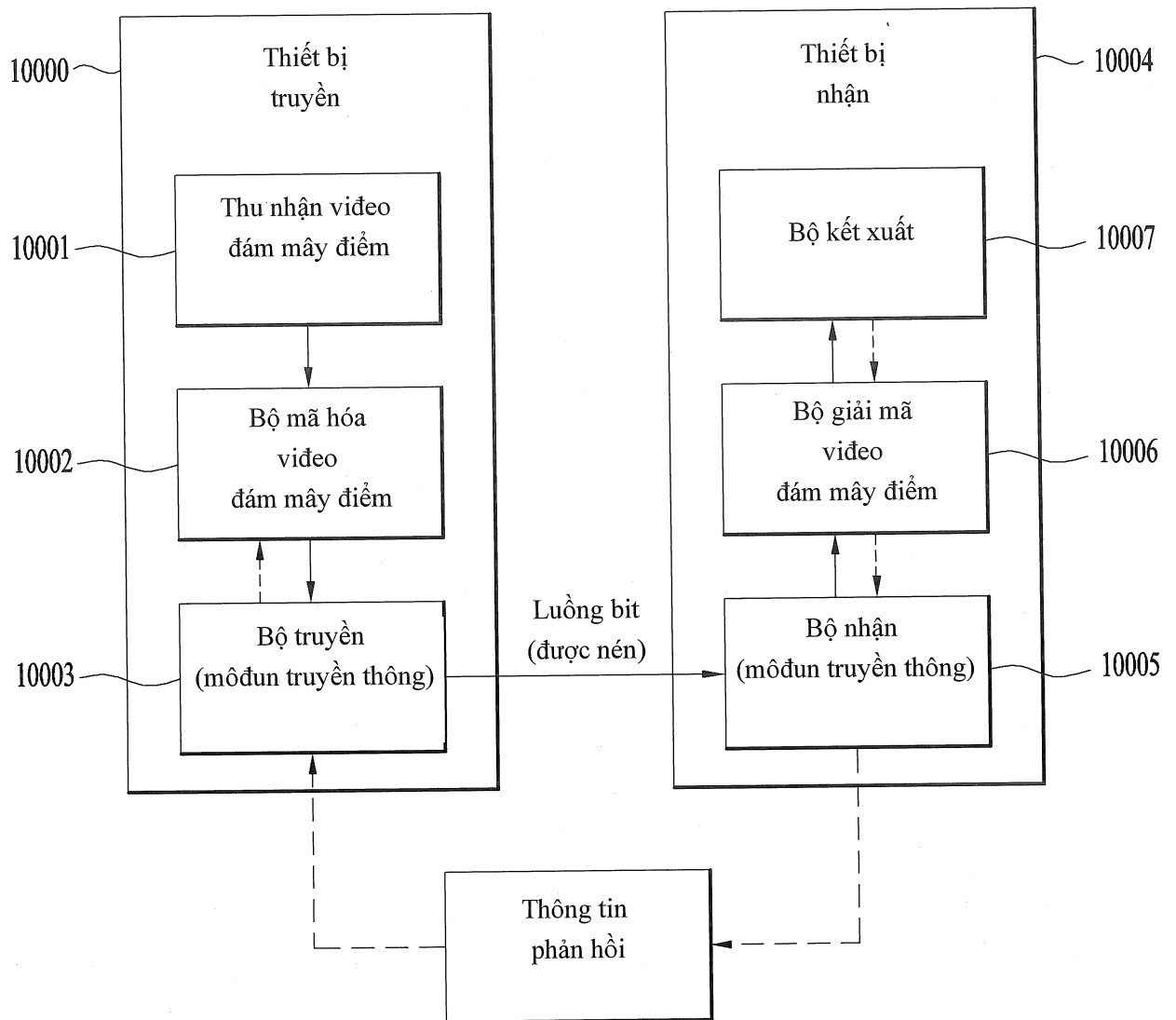


FIG. 2

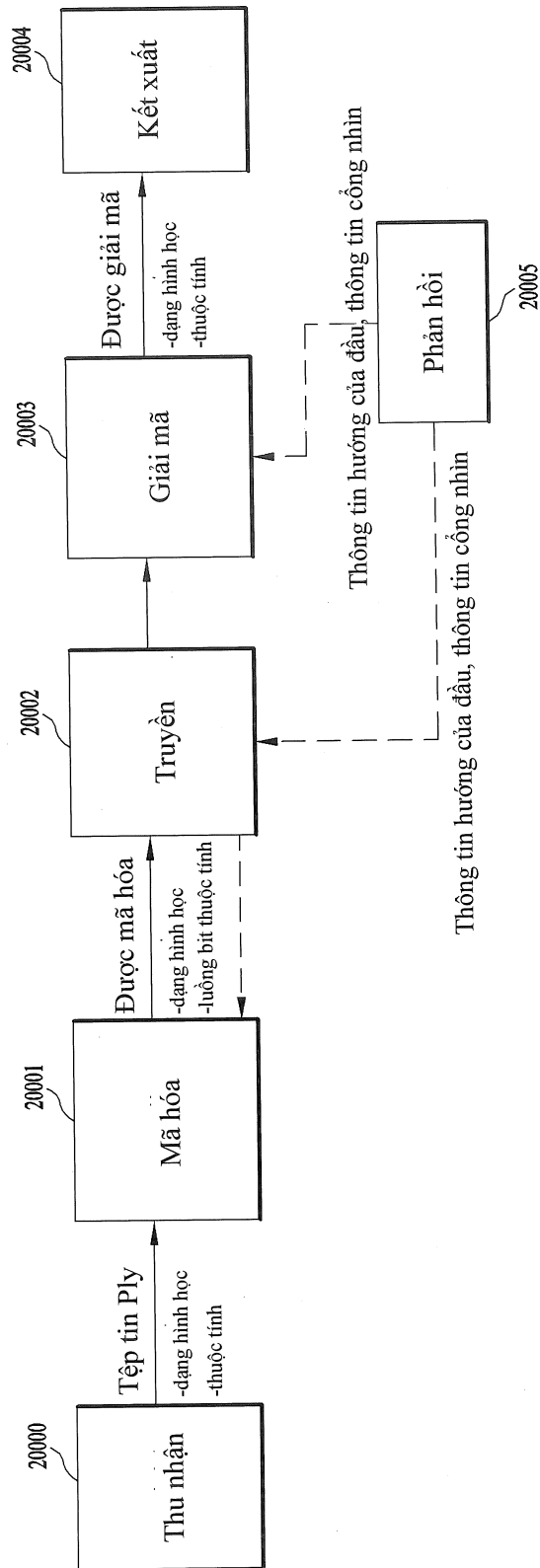


FIG. 3

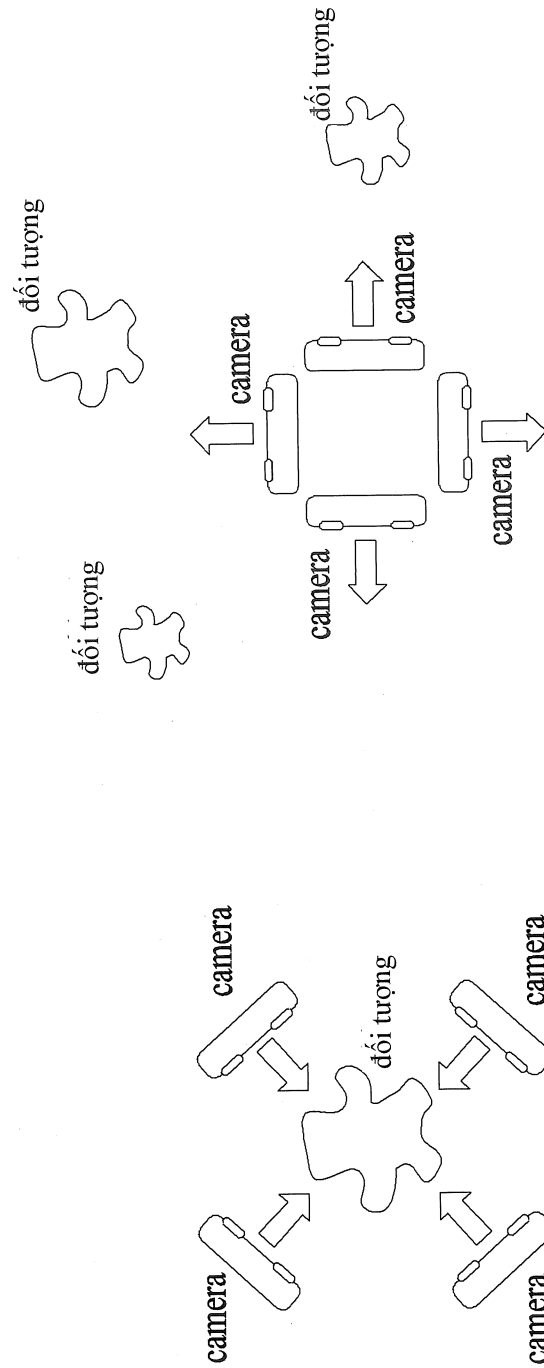


FIG. 4

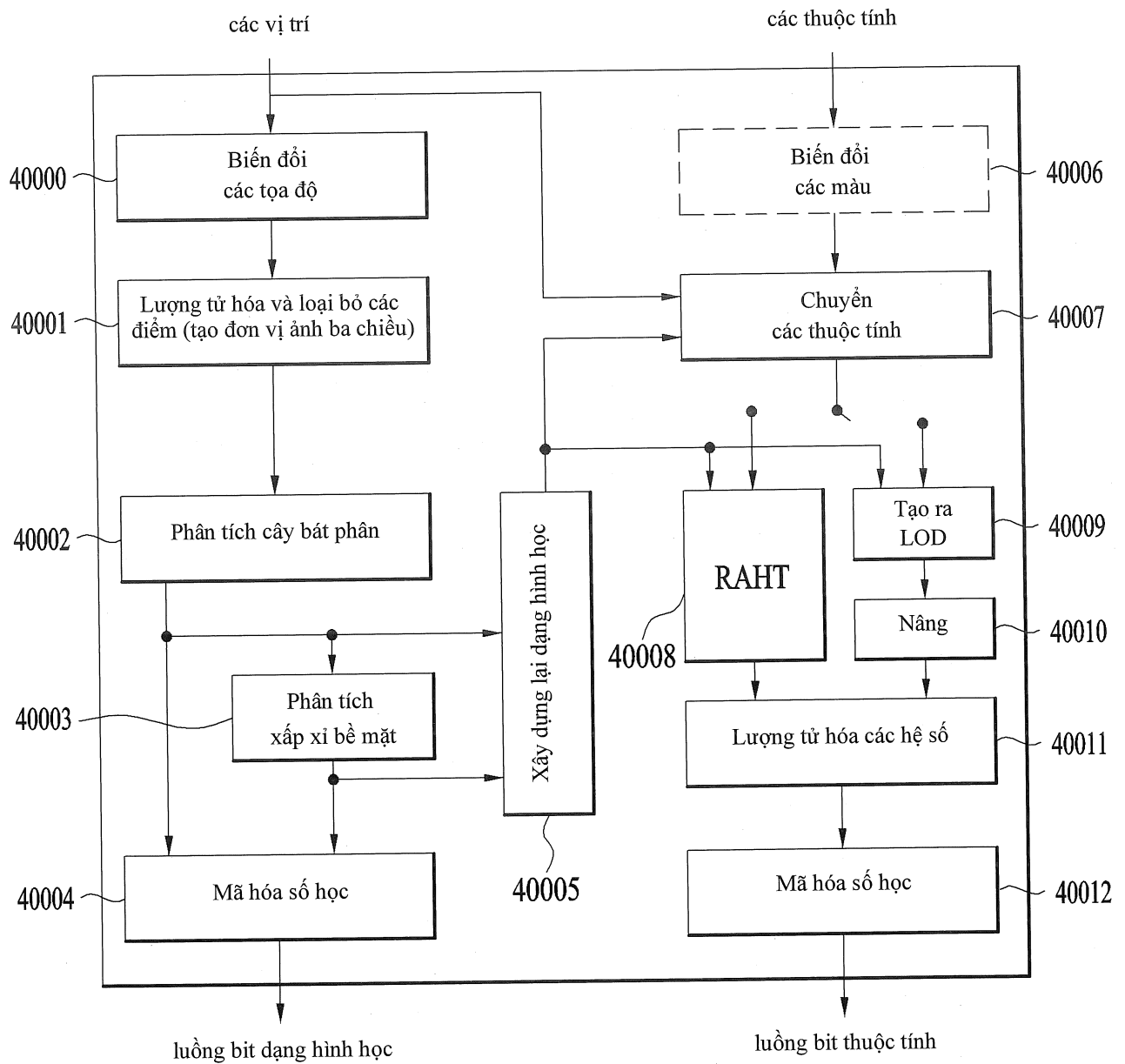


FIG. 5

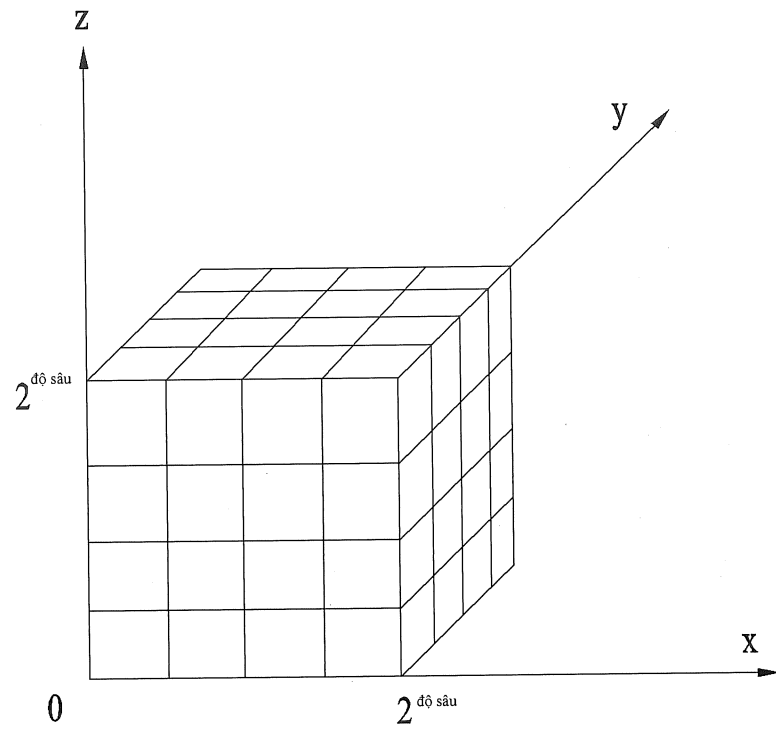


FIG. 6

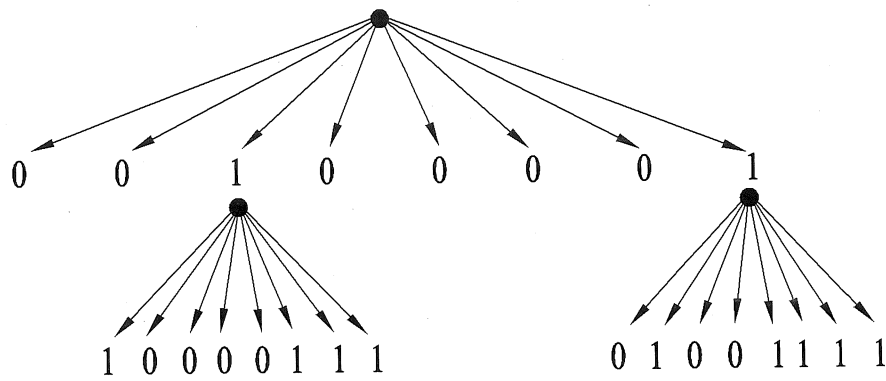
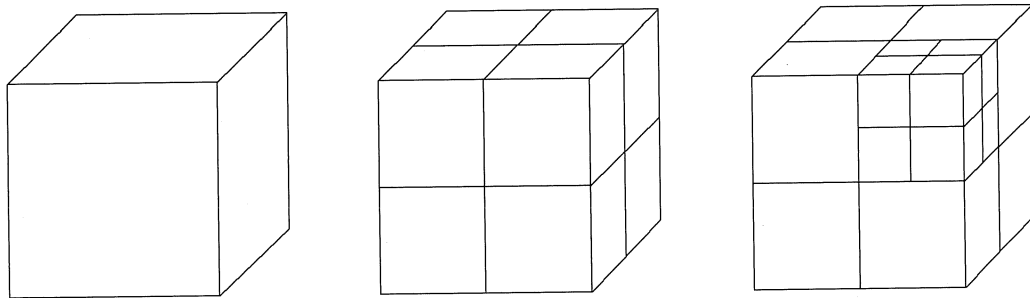


FIG. 7

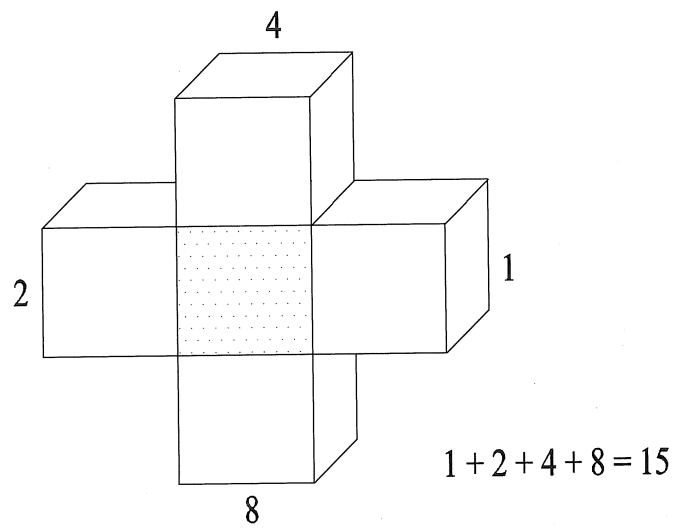
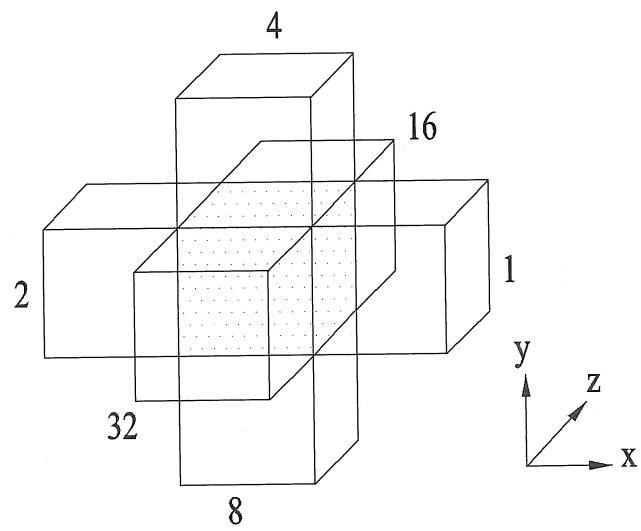


FIG. 8

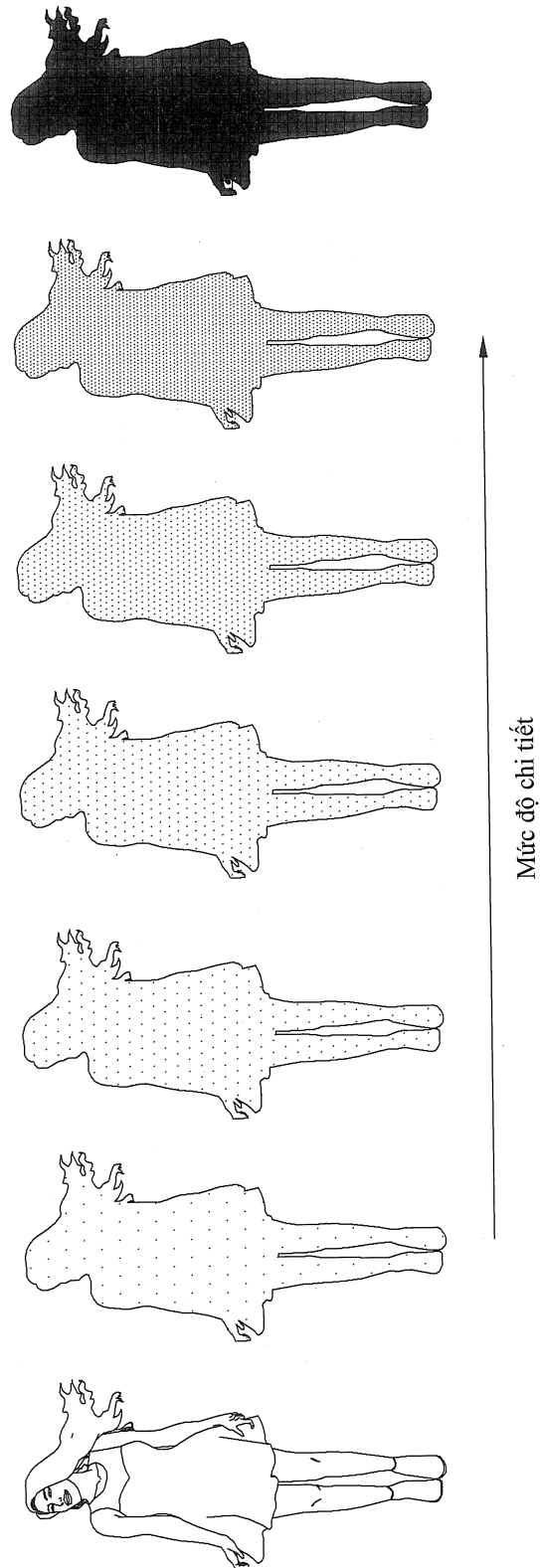


FIG. 9

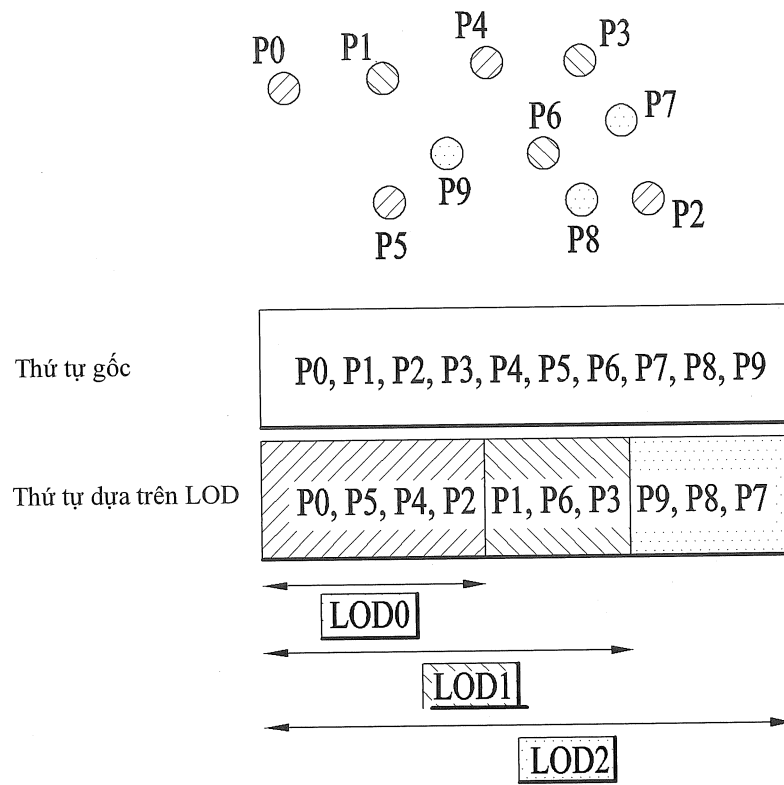


FIG. 10

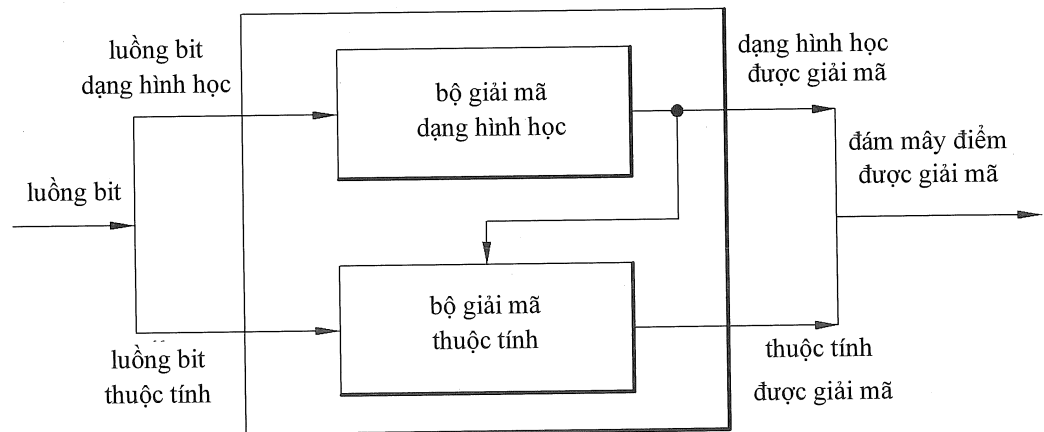


FIG. 11

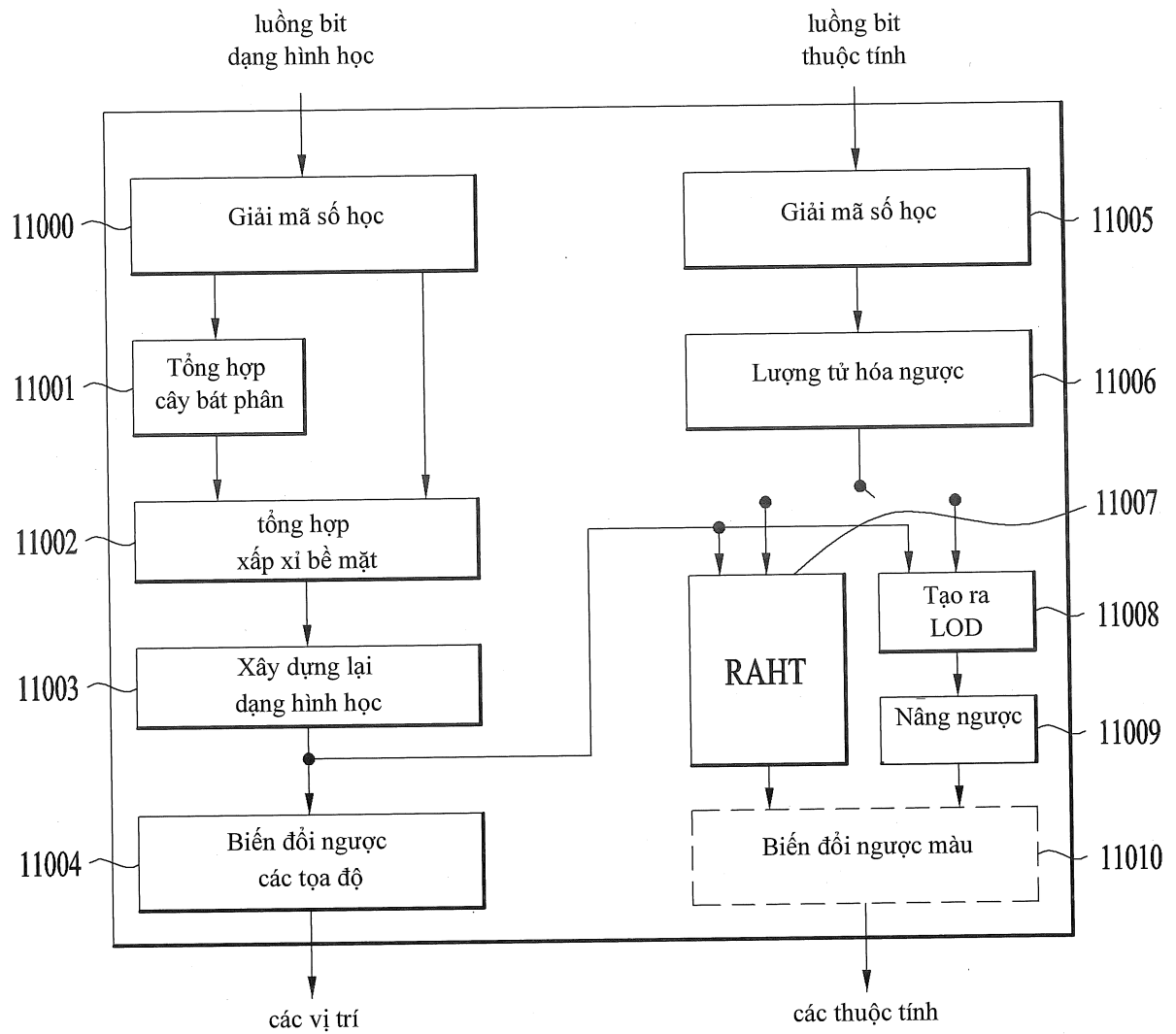


FIG. 12

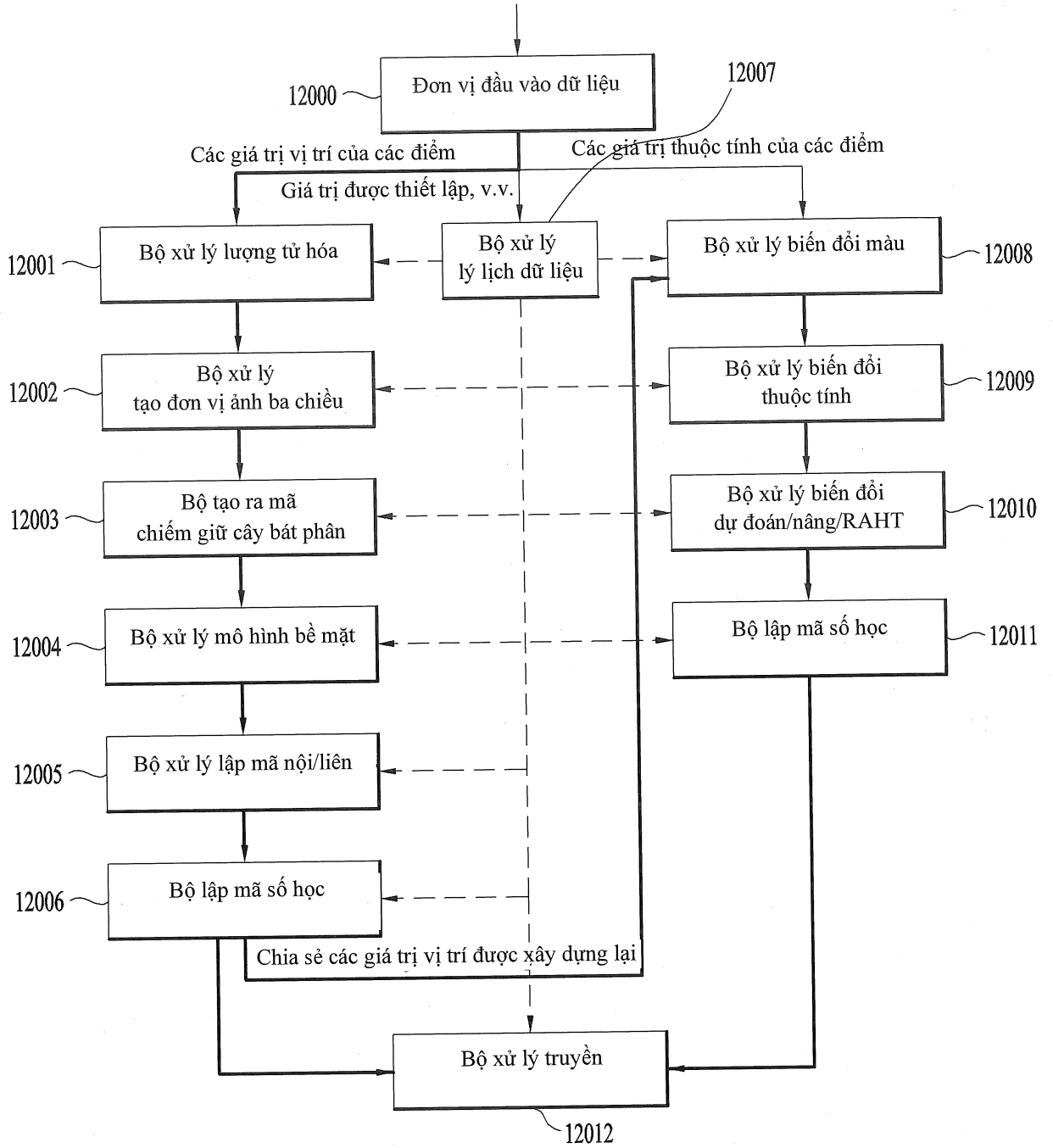


FIG. 13

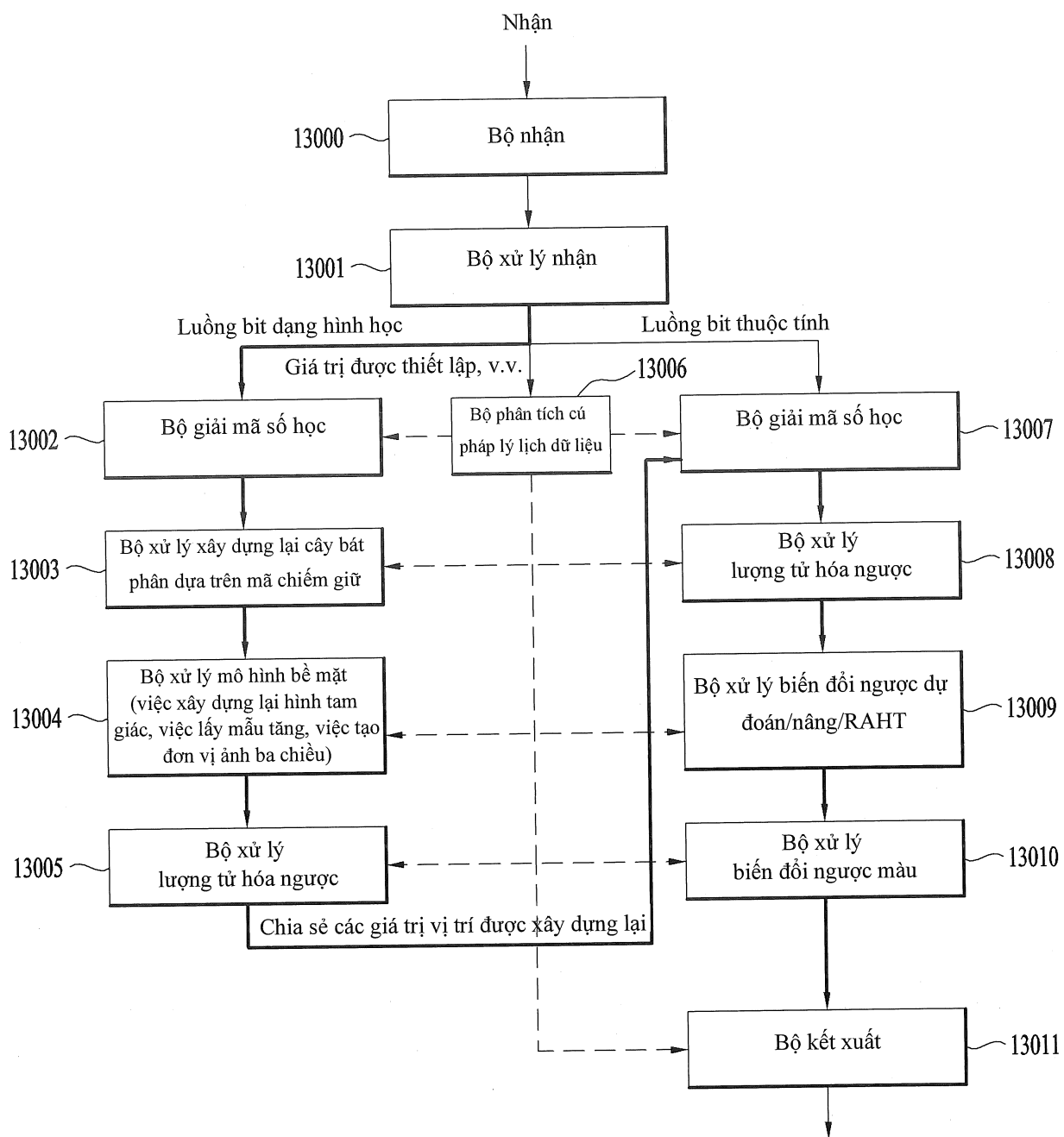


FIG. 14

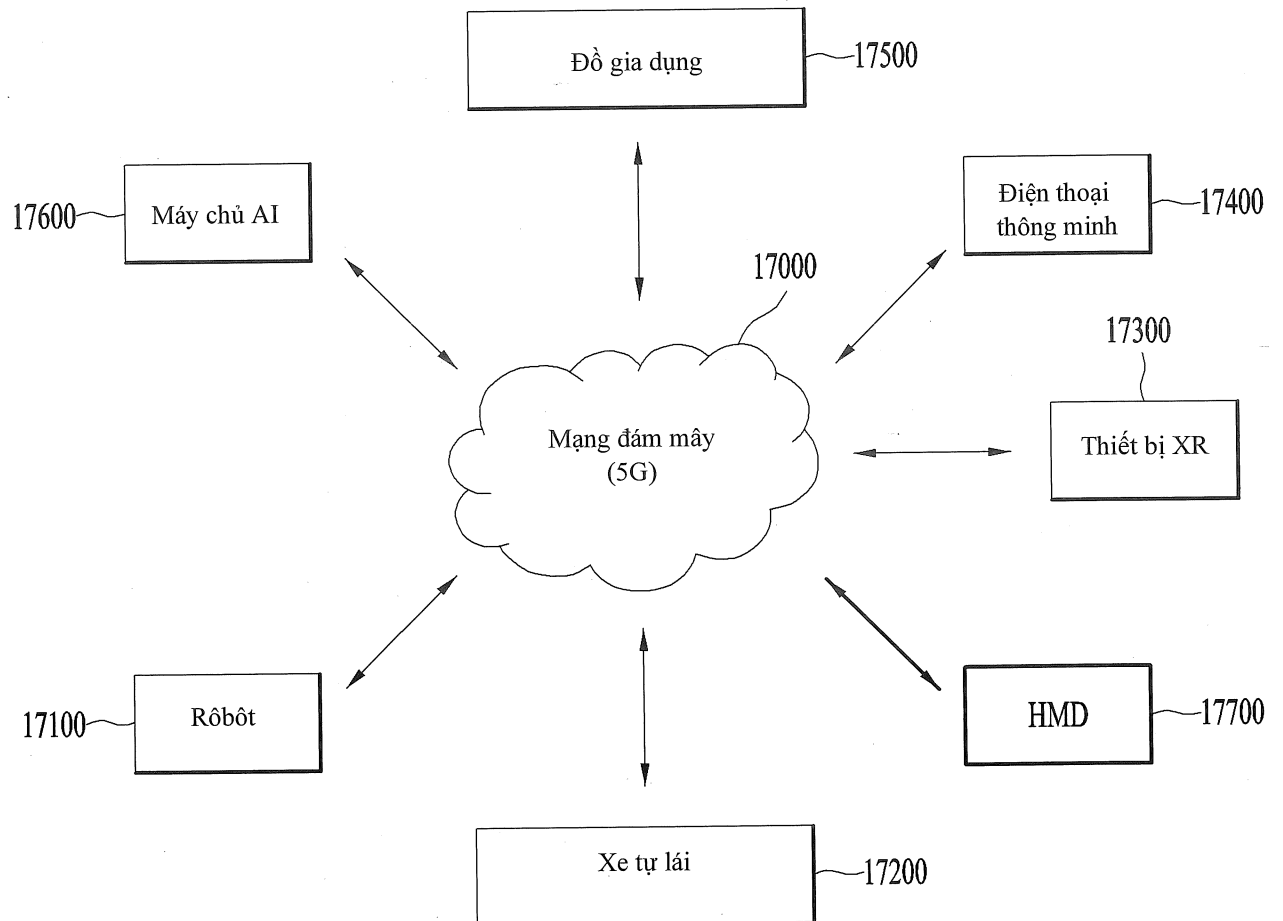


FIG. 15

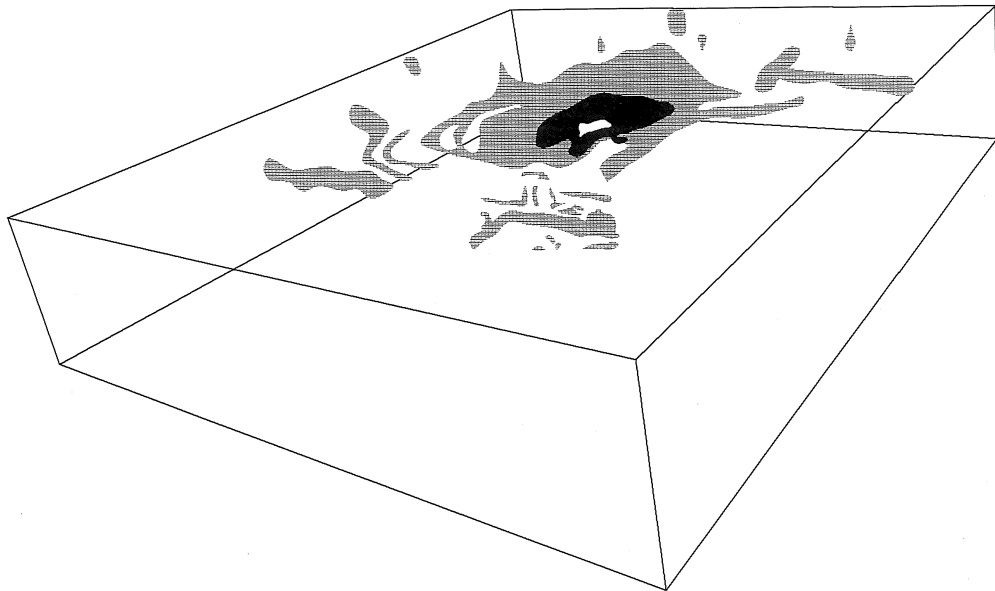


FIG. 16

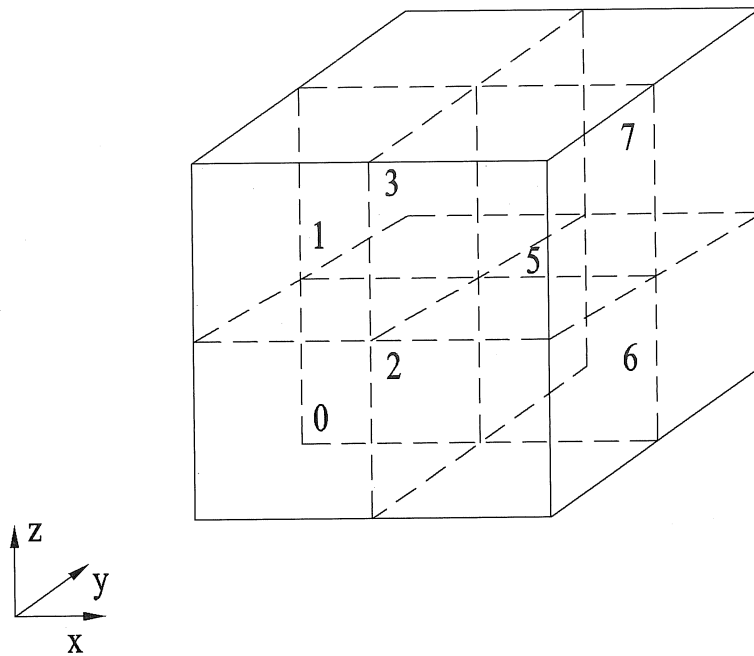


FIG. 17

Phân vùng cây tứ phân của khối lập phương 3D, lần lượt dọc theo các trục x-y, x-z, y-z

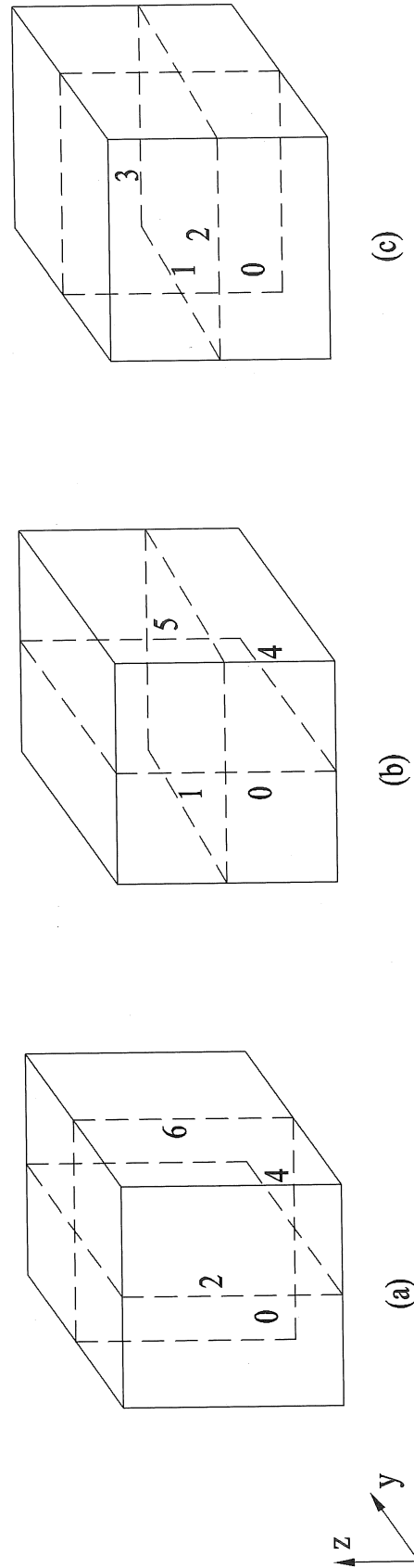


FIG. 18

Phân vùng cây nhị phân của khối lập phương 3D, lần lượt dọc theo trục x , y , z

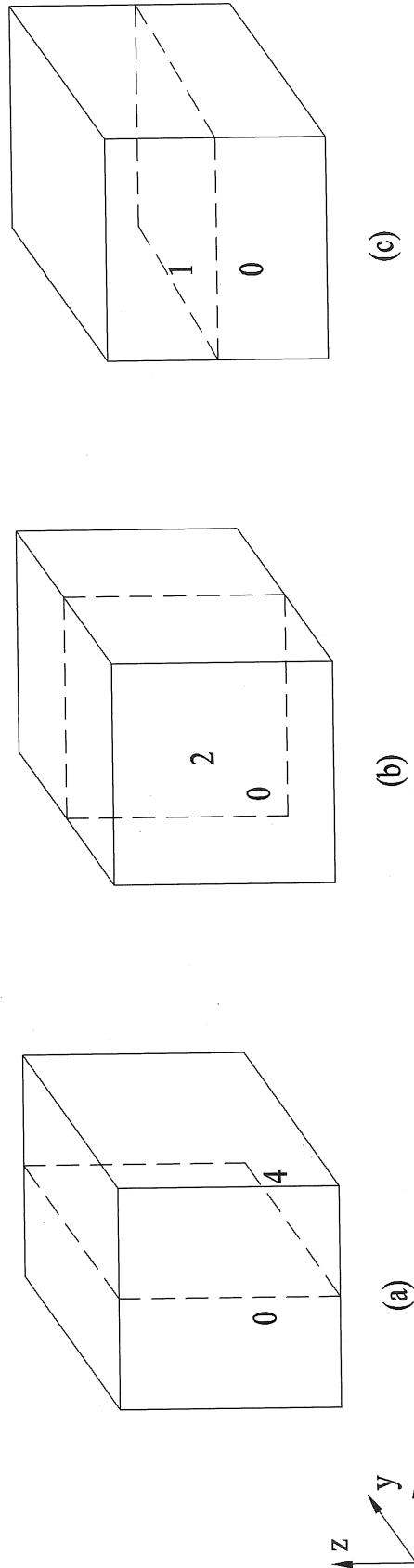
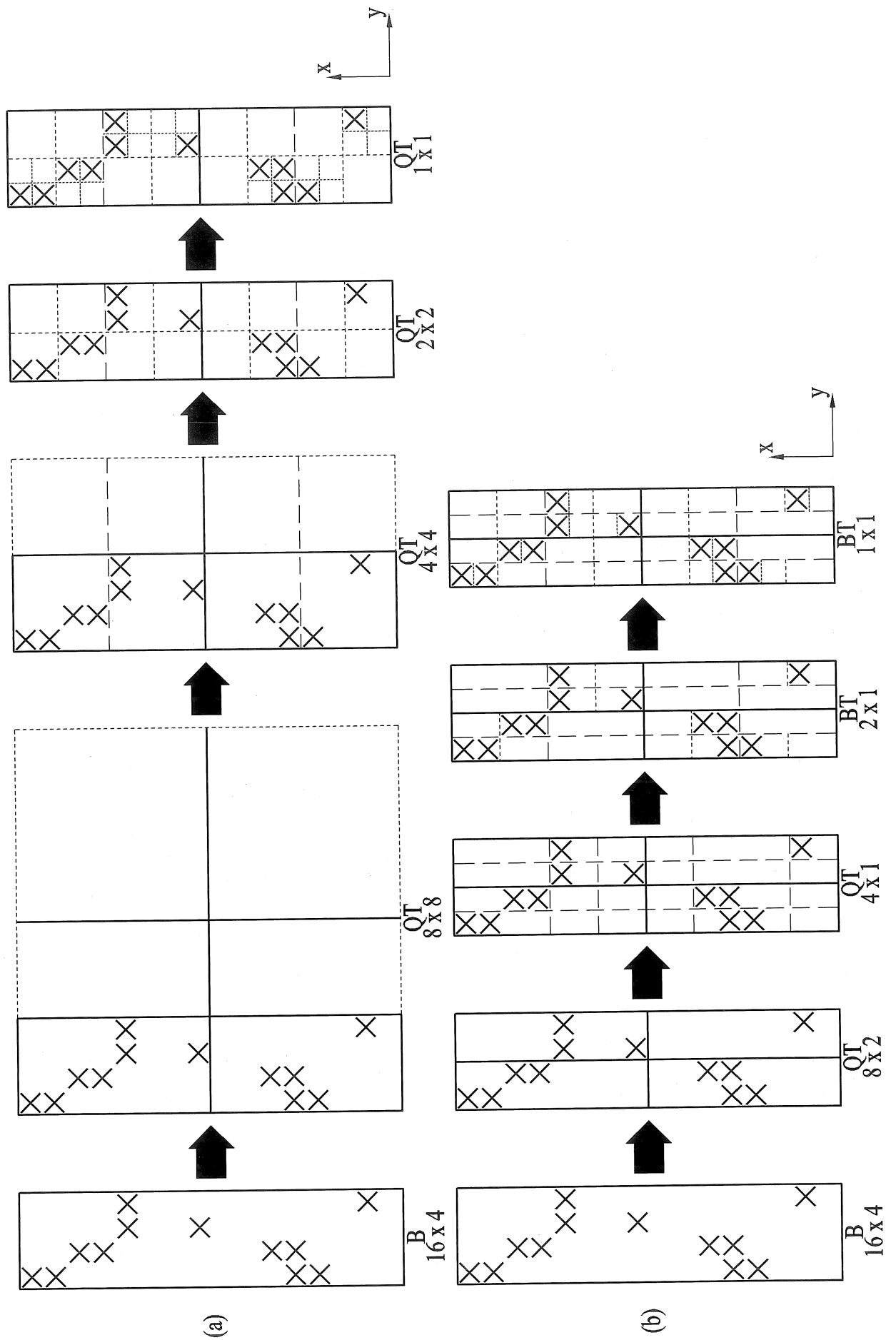


FIG. 19



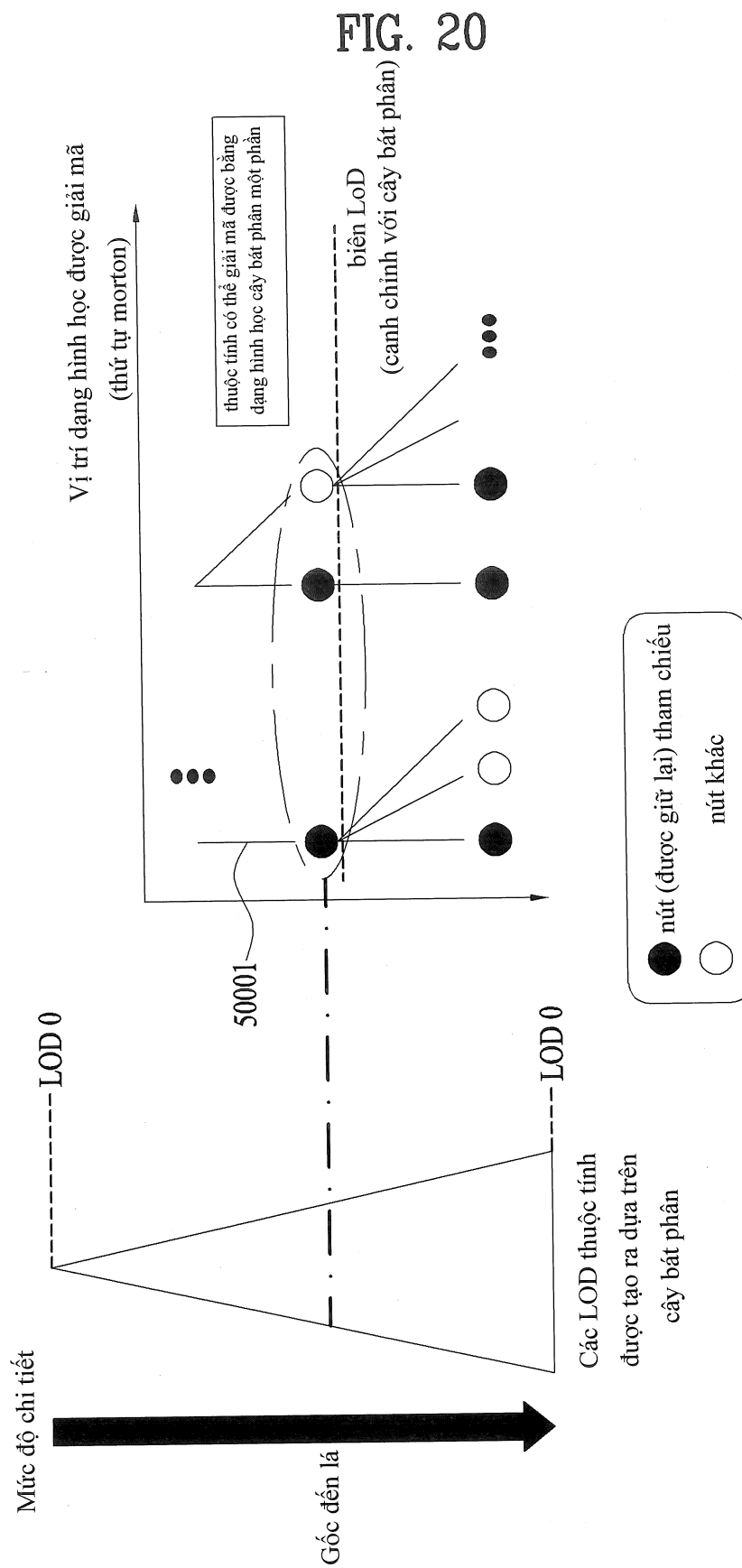
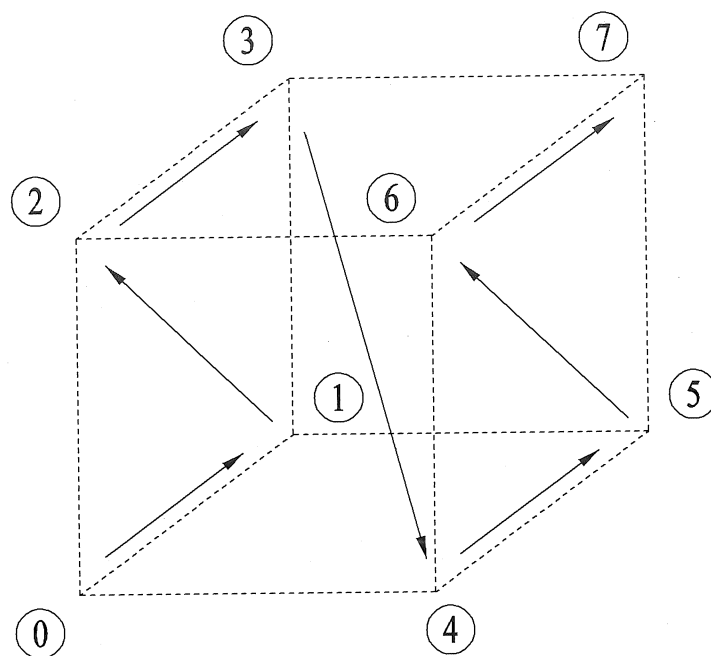


FIG. 21



Thứ tự Morton

FIG. 22

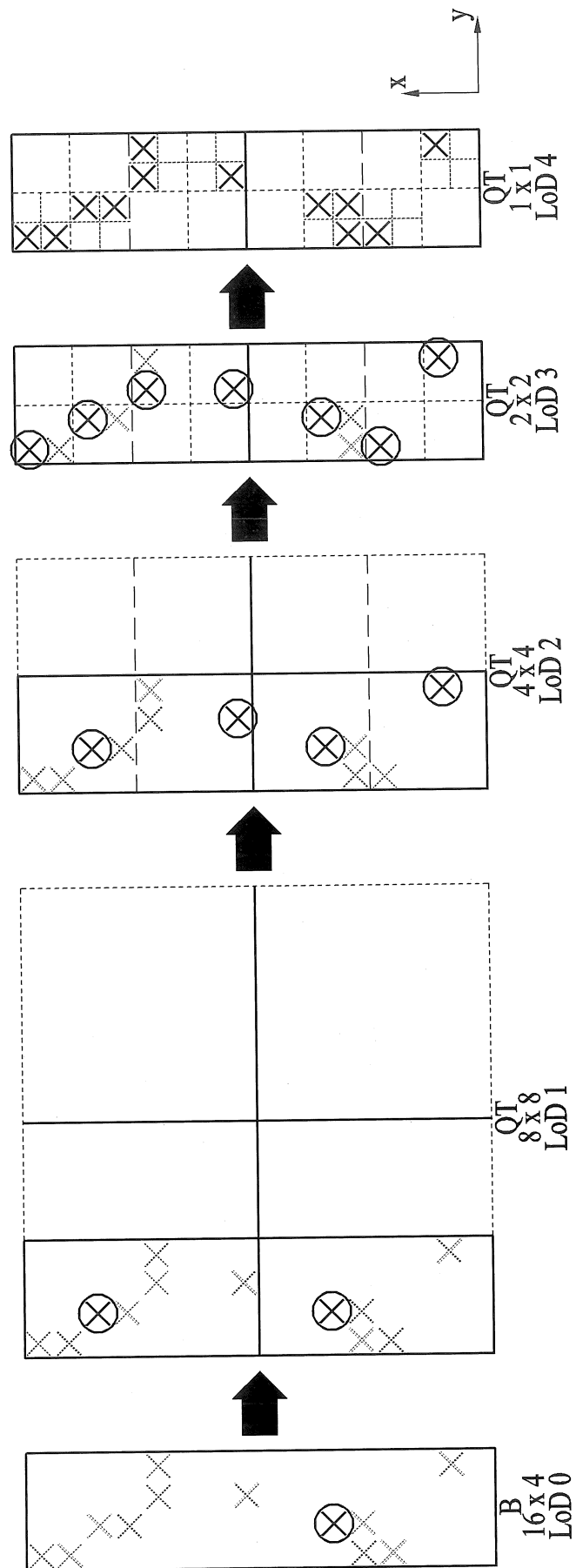


FIG. 23

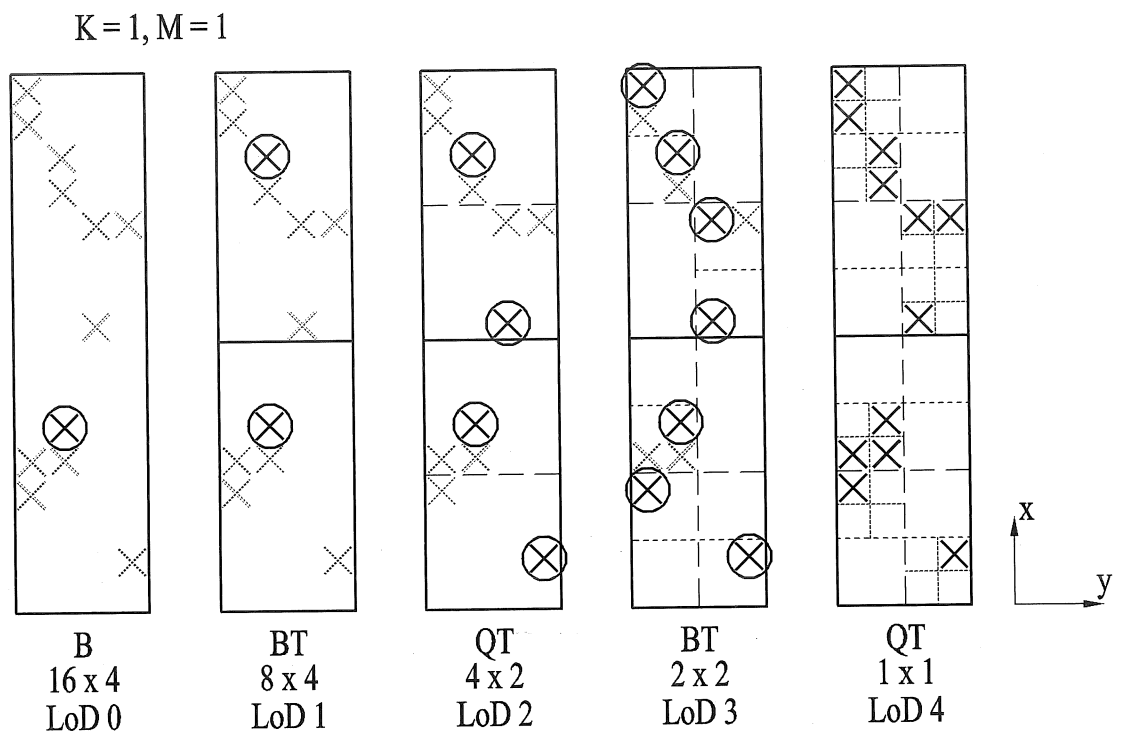


FIG. 24

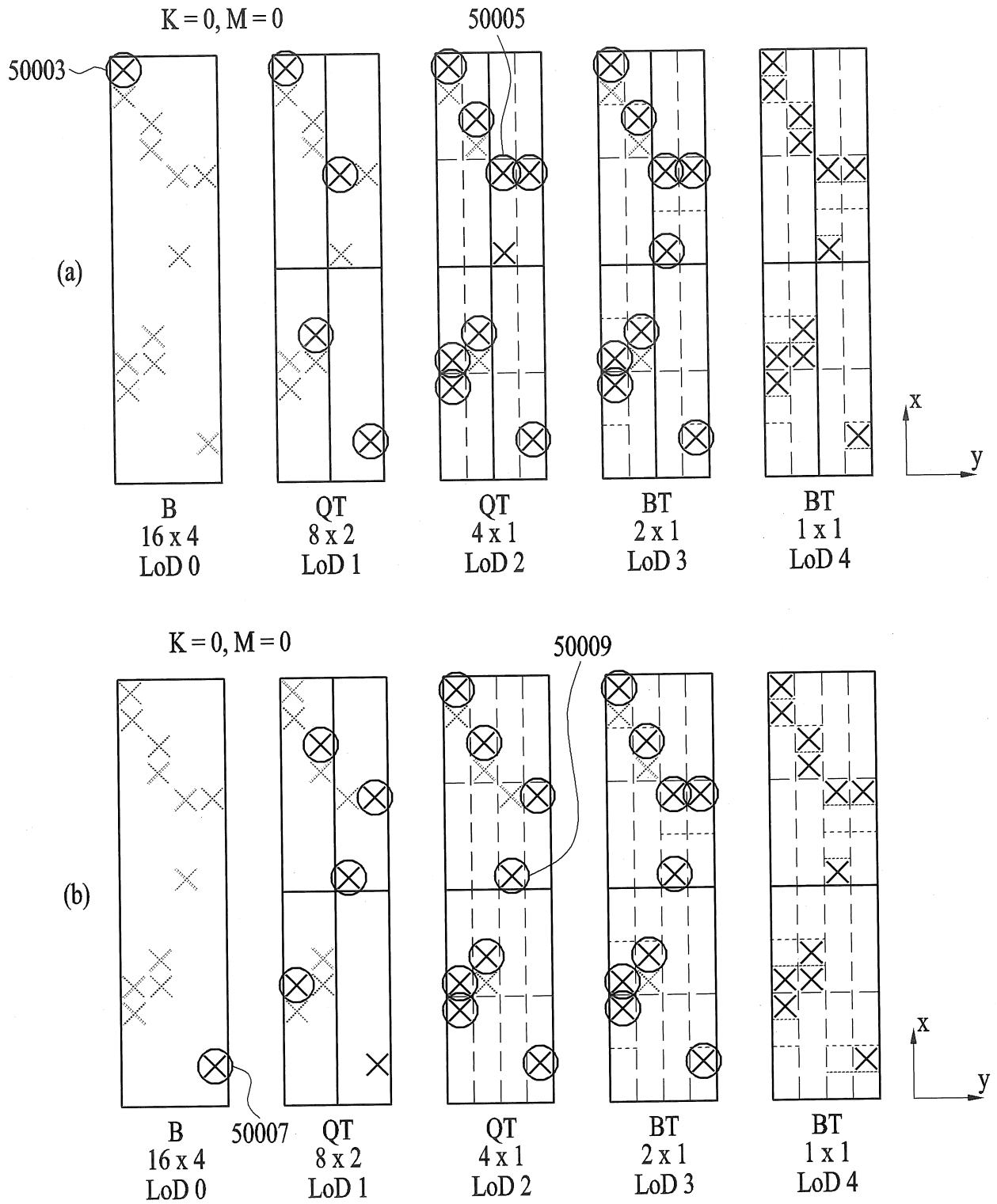


FIG. 25

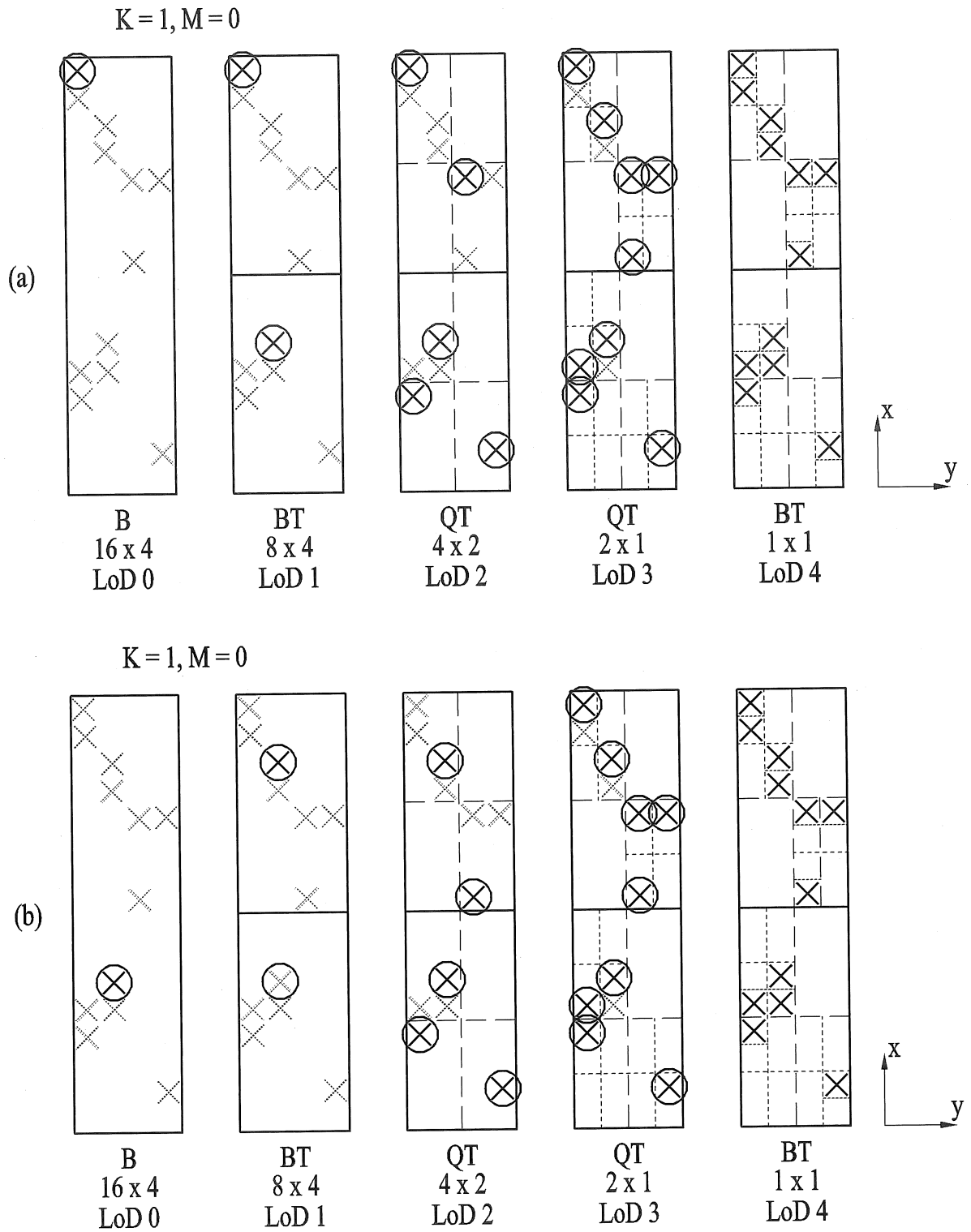


FIG. 26

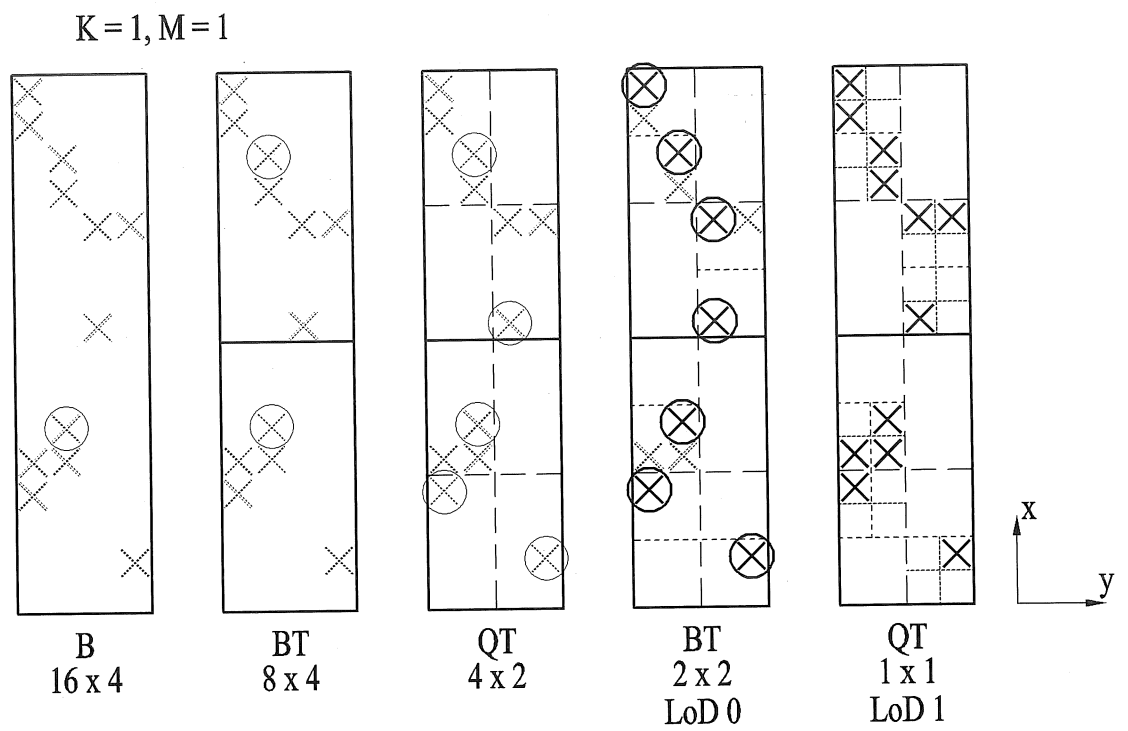


FIG. 27

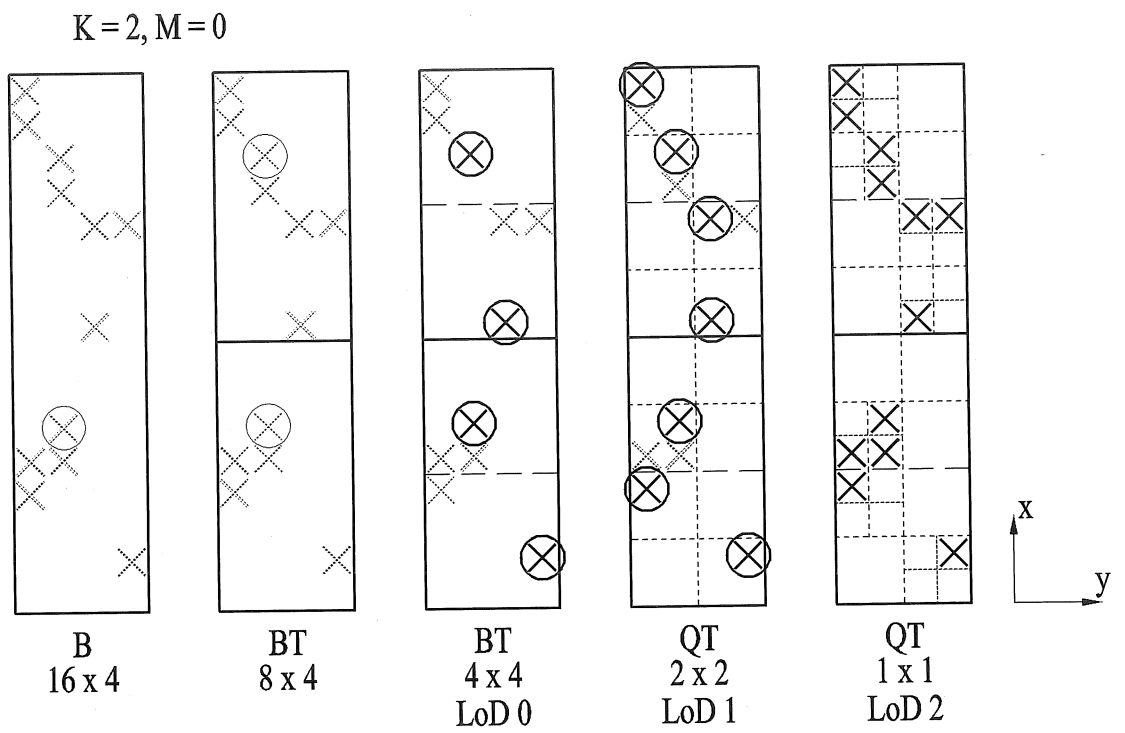


FIG. 28

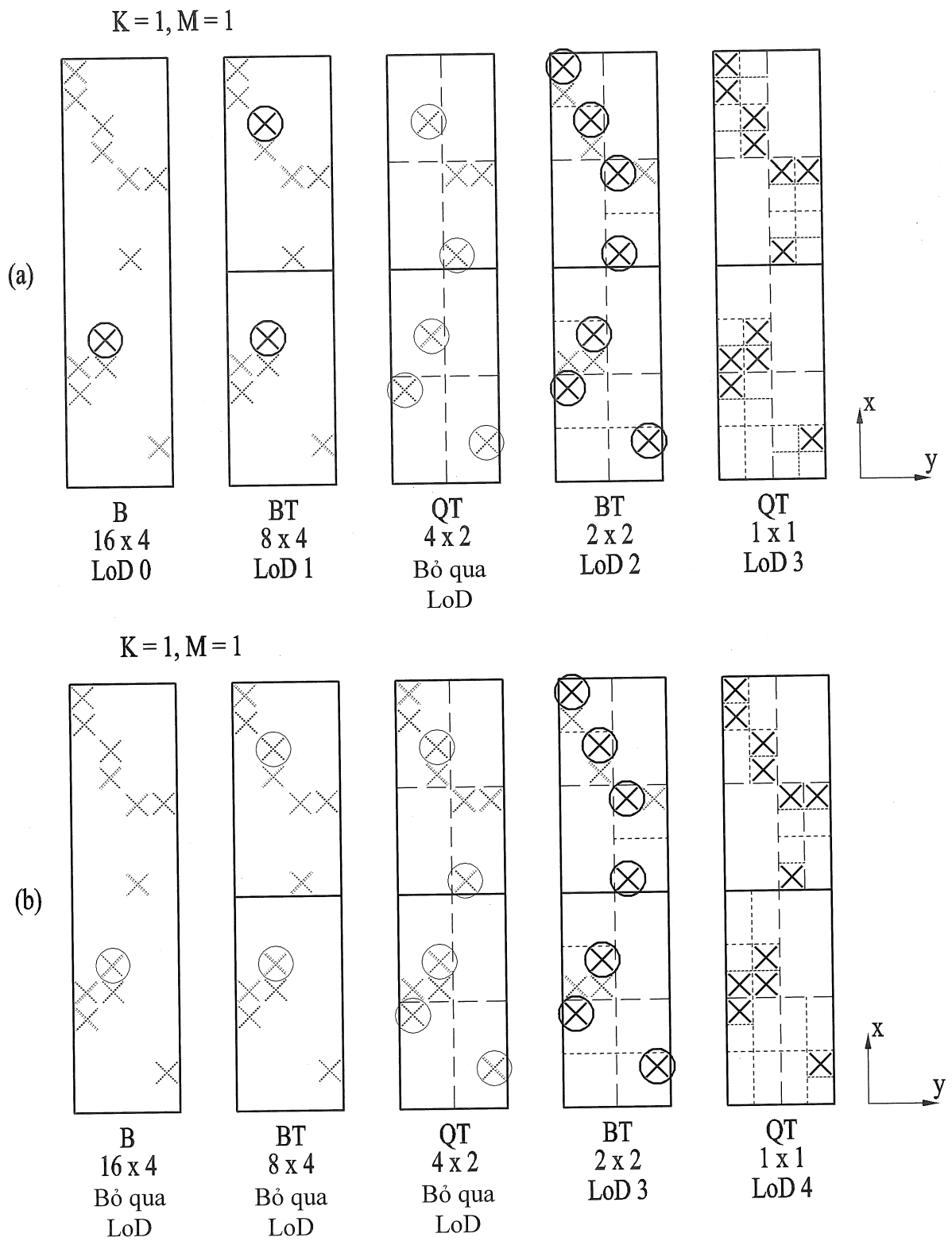


FIG. 29

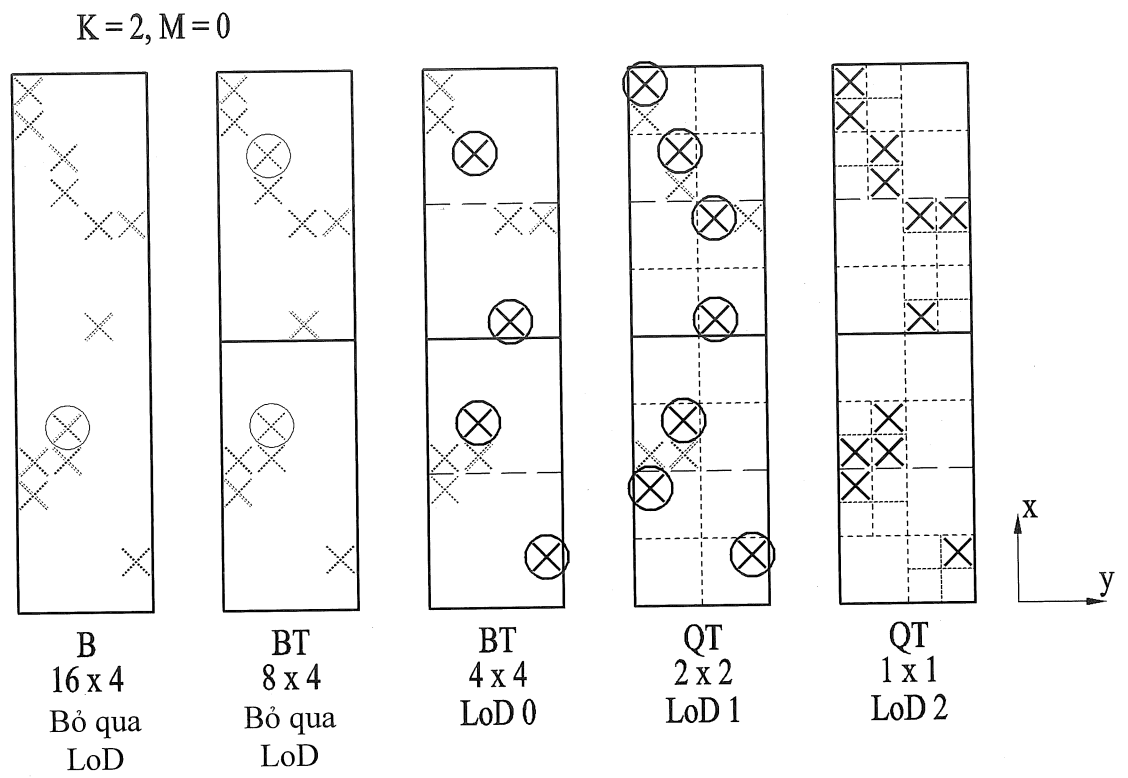


FIG. 30

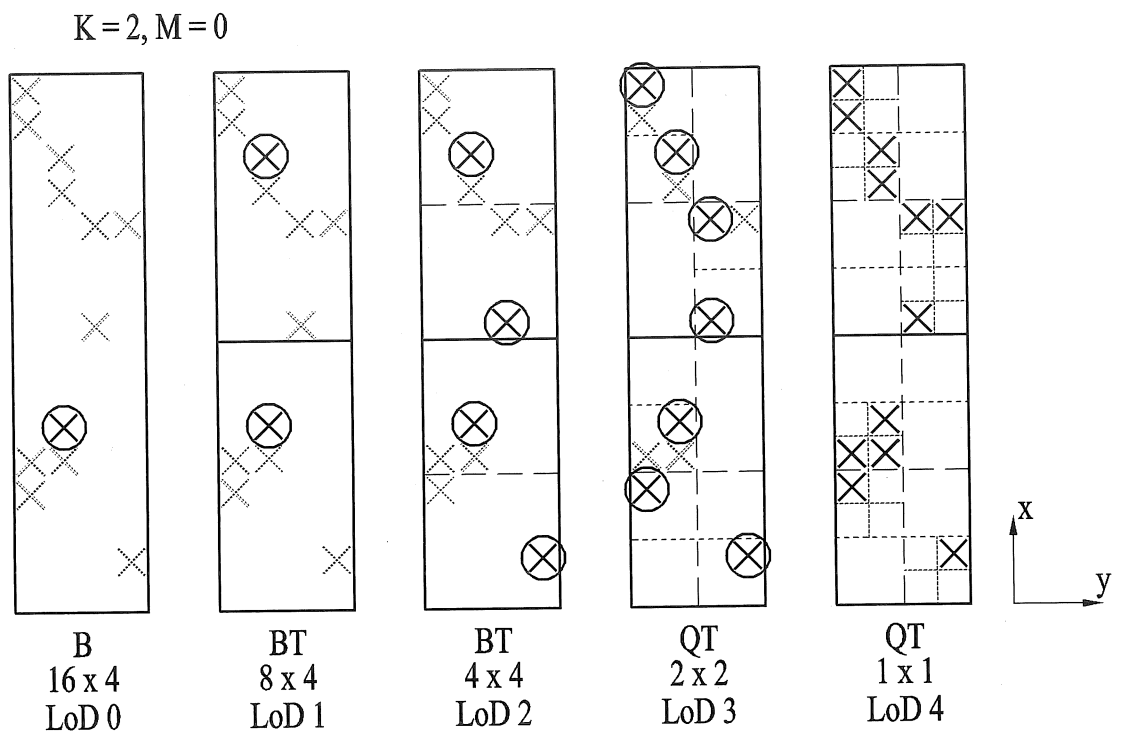


FIG. 31

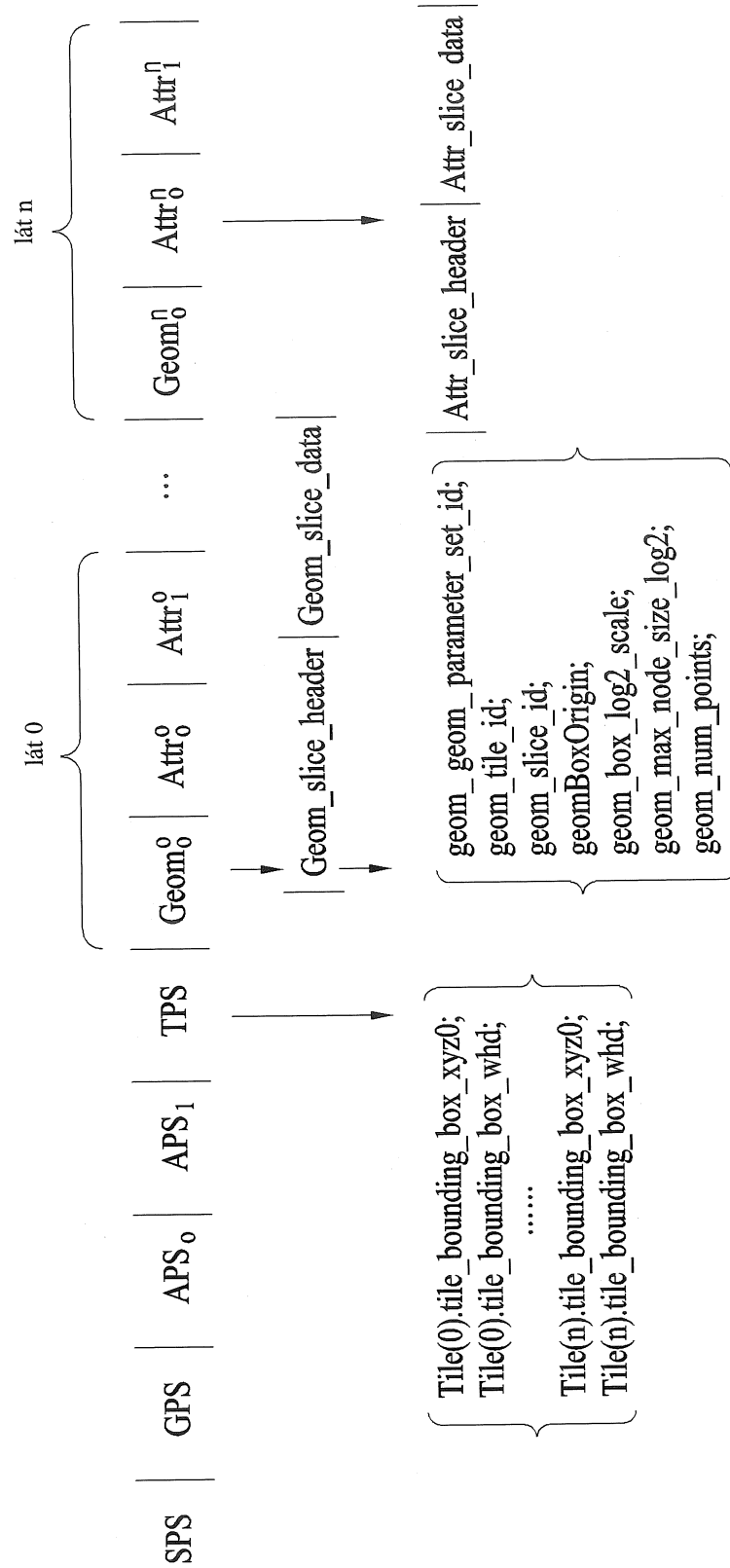


FIG. 32

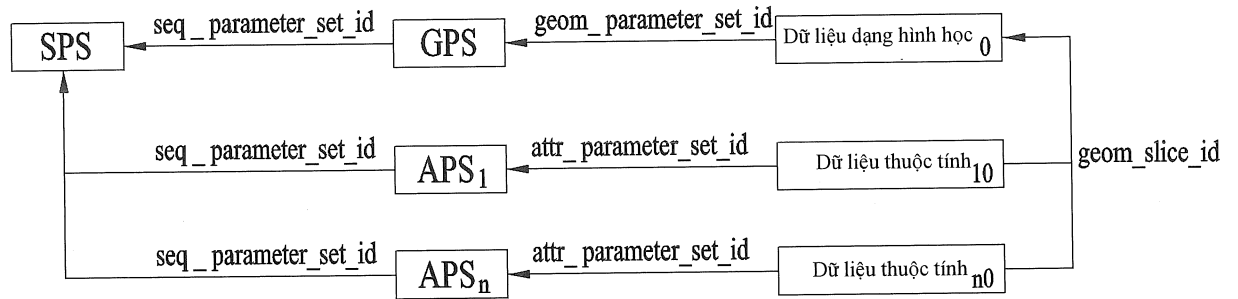


FIG. 33

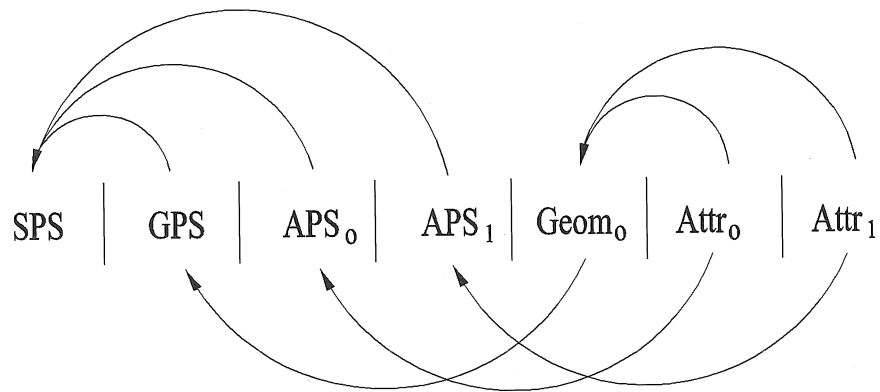


FIG. 34

seq_parameter_set() {	Descriptor
main_profile_compatibility_flag	u(1)
unique_point_positions_constraint_flag	u(1)
level_idc	u(8)
sps_seq_parameter_set_id	ue(v)
sps_bounding_box_present_flag	u(1)
if(sps_bounding_box_present_flag) {	
sps_bounding_box_offset_x	se(v)
sps_bounding_box_offset_y	se(v)
sps_bounding_box_offset_z	se(v)
sps_bounding_box_offset_log2_scale	ue(v)
sps_bounding_box_size_width	ue(v)
sps_bounding_box_size_height	ue(v)
sps_bounding_box_size_depth	ue(v)
}	
sps_source_scale_factor_numerator_minus1	ue(v)
sps_source_scale_factor_denominator_minus1	ue(v)
sps_num_attribute_sets	ue(v)
for(i = 0; i < sps_num_attribute_sets; i++) {	
attribute_dimension_minus1[i]	ue(v)
attribute_instance_id[i]	ue(v)
if(attribute_dimension_minus1[i] > 0)	
attribute_secondary_bitdepth_minus1[i]	ue(v)
attribute_cicp_colour_primaries[i]	ue(v)
attribute_cicp_transfer_characteristics[i]	ue(v)
attribute_cicp_matrix_coeffs[i]	ue(v)
attribute_cicp_video_full_range_flag[i]	u(1)
known_attribute_label_flag[i]	u(1)
if(known_attribute_label_flag[i])	
known_attribute_label[i]	ue(v)
else	
attribute_label_four_bytes[i]	u(32)
}	
log2_max_frame_idx	u(5)
axis_coding_order	u(3)
sps_bypass_stream_enabled_flag	u(1)
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_data_in_byte_stream())	
sps_extension_data_flag	u(1)
byte_alignment()	
}	

FIG. 35

geometry_parameter_set() {	Descriptor
gps_geom_parameter_set_id	ue(v)
gps_seq_parameter_set_id	ue(v)
gps_box_present_flag	u(1)
if(gps_box_present_flag){	
gps_gsh_box_log2_scale_present_flag	u(1)
if(gps_gsh_box_log2_scale_present_flag == 0)	
gps_gsh_box_log2_scale	ue(v)
}	
unique_geometry_points_flag	u(1)
geometry_planar_mode_flag	u(1)
if(geometry_planar_mode_flag){	
geom_planar_mode_th_idcm	ue(v)
geom_planar_mode_th[1]	ue(v)
geom_planar_mode_th[2]	ue(v)
}	
geometry_angular_mode_flag	u(1)
if(geometry_angular_mode_flag){	
lidar_head_position[0]	se(v)
lidar_head_position[1]	se(v)
lidar_head_position[2]	se(v)
number_lasers	ue(v)
for(i = 0; i < number_lasers; i++) {	
laser_angle[i]	se(v)
laser_correction[i]	se(v)
}	
planar_buffer_disabled	u(1)
implicit_qtbt_angular_max_node_min_dim_log2_to_split_z	se(v)
implicit_qtbt_angular_max_diff_to_split_z	se(v)
}	
neighbour_context_restriction_flag	u(1)
inferred_direct_coding_mode_enabled_flag	u(1)
bitwise_occupancy_coding_flag	u(1)
adjacent_child_contextualization_enabled_flag	u(1)
log2_neighbour_avail_boundary	ue(v)
log2_intra_pred_max_node_size	ue(v)
log2_trisoup_node_size	ue(v)
geom_scaling_enabled_flag	u(1)
if(geom_scaling_enabled_flag)	
geom_base_qp	ue(v)
gps_implicit_geom_partition_flag	u(1)
if(gps_implicit_geom_partition_flag) {	
gps_max_num_implicit_qtbt_before_ot	ue(v)
gps_min_size_implicit_qtbt	ue(v)
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_data_in_byte_stream())	
gps_extension_data_flag	u(1)
byte_alignment()	
}	

FIG. 36

attribute_parameter_set () {	Descriptor
aps_attr_parameter_set_id	ue(v)
aps_seq_parameter_set_id	ue(v)
attr_coding_type	ue(v)
aps_attr_initial_qp	ue(v)
aps_attr_chroma_qp_offset	se(v)
aps_slice_qp_delta_present_flag	u(1)
LodParametersPresent = (attr_coding_type == 0 attr_coding_type == 2) ? 1 : 0	
if (LodParametersPresent) {	
lifting_num_pred_nearest_neighbours_minus1	ue(v)
lifting_search_range_minus1	ue(v)
for (k = 0; k < 3; k++)	
lifting_neighbour_bias[k]	ue(v)
if (attr_coding_type == 2)	
lifting_scalability_enabled_flag	u(1)
if (! lifting_scalability_enabled_flag) {	
lifting_num_detail_levels_minus1	ue(v)
[Ed. The V7.0 code use the variable without minus1. It should be aligned]	
if (lifting_num_detail_levels_minus1 > 0) {	
lifting_lod_regular_sampling_enabled_flag	u(1)
for (idx = 0; idx < num_detail_levels_minus1; idx++) {	
if (lifting_lod_regular_sampling_enabled_flag)	
lifting_sampling_period_minus2[idx]	ue(v)
else	
lifting_sampling_distance_squared_scale_minus1[idx]	ue(v)
if (idx != 0)	
lifting_sampling_distance_squared_offset[idx]	ue(v)
}	
}	
}	
if (attr_coding_type == 0) {	
lifting_adaptive_prediction_threshold	ue(v)
lifting_intra_lod_prediction_num_layers	ue(v)
lifting_max_num_direct_predictors	ue(v)
inter_component_prediction_enabled_flag	u(1)
}	
}	
if (attribute_coding_type == 1) { //RAHT	
raht_prediction_enabled_flag	u(1)
if (raht_prediction_enabled_flag) {	
raht_prediction_threshold0	ue(v)
raht_prediction_threshold1	ue(v)
}	
}	
aps_extension_flag	u(1)
if (aps_extension_flag)	
while (more_data_in_byte_stream ())	
aps_extension_data_flag	u(1)
byte_alignment ()	
}	

FIG. 37

attribute_parameter_set() {	Descriptor
....	
constrained_QTBT_flag	u(1)
if(constrained_QTBT_flag)	
max_LoD_depth_for_octree_based_LoD_generation	u(8)
num_LoD_minus1	u(8)
for(i=0; i< num_LoD_minus1+1; i++) {	
cubic_node_flag	u(1)
if(!cubic_node_flag)	
node_direction // z, y, x, yz, xz, xy	u(3)
if(constrained_QTBT_flag)	
no_representation_flag	u(1)
node_size_x	
node_size_y	
node_size_z	
}	
LoD_sampling_method	u(4)
....	

FIG. 38

geometry_slice_bitstream() {	Descriptor
geometry_slice_header()	
geometry_slice_data()	
}	

FIG. 39

geometry_slice_header() {	Descriptor
gsh_geometry_parameter_set_id	ue(v)
gsh_tile_id	ue(v)
gsh_slice_id	ue(v)
frame_idx	u(n)
gsh_num_points	u(24)
if(gps_box_present_flag) {	
if(gps_gsh_box_log2_scale_present_flag)	
gsh_box_log2_scale	ue(v)
gsh_box_origin_x	ue(v)
gsh_box_origin_y	ue(v)
gsh_box_origin_z	ue(v)
}	
if(gps_implicit_geom_partition_flag) {	
gsh_log2_max_nodesize_x	ue(v)
gsh_log2_max_nodesize_y_minus_x	se(v)
gsh_log2_max_nodesize_z_minus_y	se(v)
} else {	
gsh_log2_max_nodesize	ue(v)
}	
if(geom_scaling_enabled_flag) {	
geom_slice_qp_offset	se(v)
geom_octree_qp_offsets_enabled_flag	u(1)
if(geom_octree_qp_offsets_enabled_flag)	
geom_octree_qp_offsets_depth	ue(v)
}	
byte_alignment()	
}	

FIG. 40

geometry_slice_data() {	Descriptor
for(depth = 0; depth < MaxGeometryOctreeDepth; depth++) {	
for(nodeId = 0; nodeId < NumNodesAtDepth[depth]; nodeId++) {	
xN = NodeX[depth][nodeId]	
yN = NodeY[depth][nodeId]	
zN = NodeZ[depth][nodeId]	
geometry_node(depth, nodeId, xN, yN, zN)	
}	
}	
if (log2_trisoup_node_size > 0)	
geometry_trisoup_data()	
}	

FIG. 41

attribute_slice_bitstream() {	Descriptor
attribute_slice_header()	
attribute_slice_data()	
}	

FIG. 42

attribute_slice_header() {	Descriptor
ash_attr_parameter_set_id	ue(v)
ash_attr_sps_attr_idx	ue(v)
ash_attr_geom_slice_id	ue(v)
if (aps_slice_qp_delta_present_flag) {	
ash_attr_qp_delta_luma	se(v)
if(attribute_dimension_minus1[ash_attr_sps_attr_idx] > 0)	
ash_attr_qp_delta_chroma	se(v)
}	
ash_attr_layer_qp_delta_present_flag	u(1)
if (ash_attr_layer_qp_delta_present_flag) {	
ash_attr_num_layer_qp_minus1	ue(v)
for(i = 0; i < NumLayerQp; i++){	
ash_attr_layer_qp_delta_luma[i]	se(v)
if(attribute_dimension_minus1[ash_attr_sps_attr_idx] > 0)	
ash_attr_layer_qp_delta_chroma[i]	se(v)
}	
}	
ash_attr_region_qp_delta_present_flag	u(1)
if (ash_attr_region_qp_delta_present_flag) {	
ash_attr_qp_region_box_origin_x	ue(v)
ash_attr_qp_region_box_origin_y	ue(v)
ash_attr_qp_region_box_origin_z	ue(v)
ash_attr_qp_region_box_width	ue(v)
ash_attr_qp_region_box_height	ue(v)
ash_attr_qp_region_box_depth	ue(v)
ash_attr_region_qp_delta	se(v)
}	
byte_alignment()	
}	

FIG. 43

attribute_slice_header() {	Descriptor
....	
constrained_QTBT_flag	u(1)
if(constrained_QTBT_flag)	
max_LoD_depth_for_octree_based_LoD_generation	u(8)
num_LoD_minus1	u(8)
for(i=0; i< num_LoD_minus1+1; i++) {	
cubic_node_flag	u(1)
if(!cubic_node_flag)	
node_direction // z, y, x, yz, xz, xy	u(3)
if(constrained_QTBT_flag)	
no_representation_flag	u(1)
node_size_x	
node_size_y	
node_size_z	
}	
LoD_sampling_method	u(4)
....	

FIG. 44

attribute_slice_data() {	Descriptor
dimension = attribute_dimension[ash_attr_sps_attr_idx]	
zerorun	ae(v)
for(i = 0; i < pointCount; i++) {	
if(attr_coding_type == 0 && maxPredDiff[i] > lifting_adaptive_prediction_threshold && MaxNumPredictors > 1) {	
predIndex[i]	ae(v)
}	
if(zerorun > 0) {	
for(k = 0; k < dimension ; k++)	
values[k][i] = 0	
zerorun -= 1	
}	
else {	
attribute_coding(dimension, i)	ae(v)
zerorun	ae(v)
}	
}	
byte_alignment()	
}	

FIG. 45

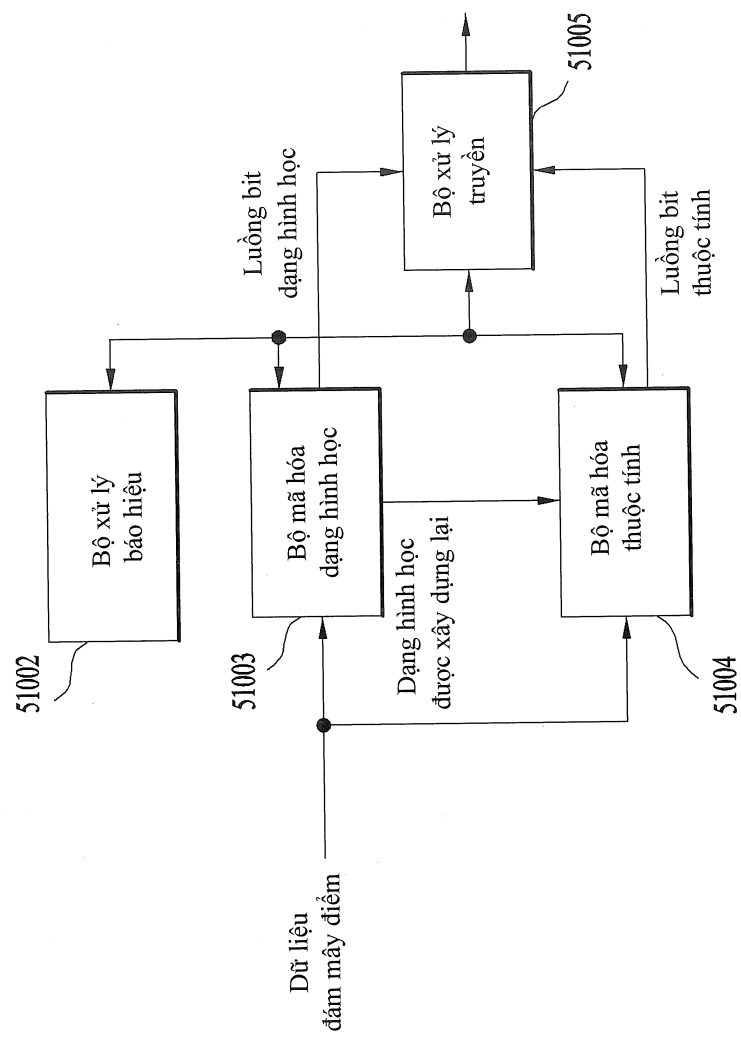


FIG. 46

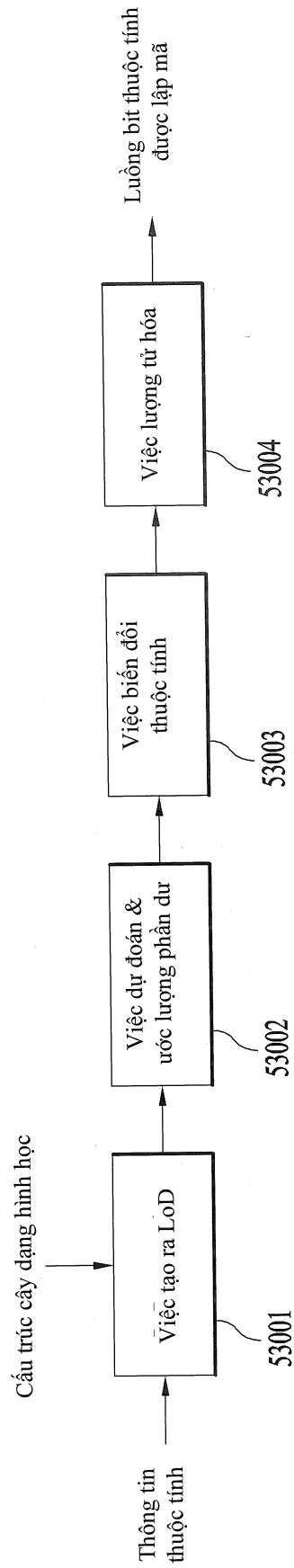


FIG. 47

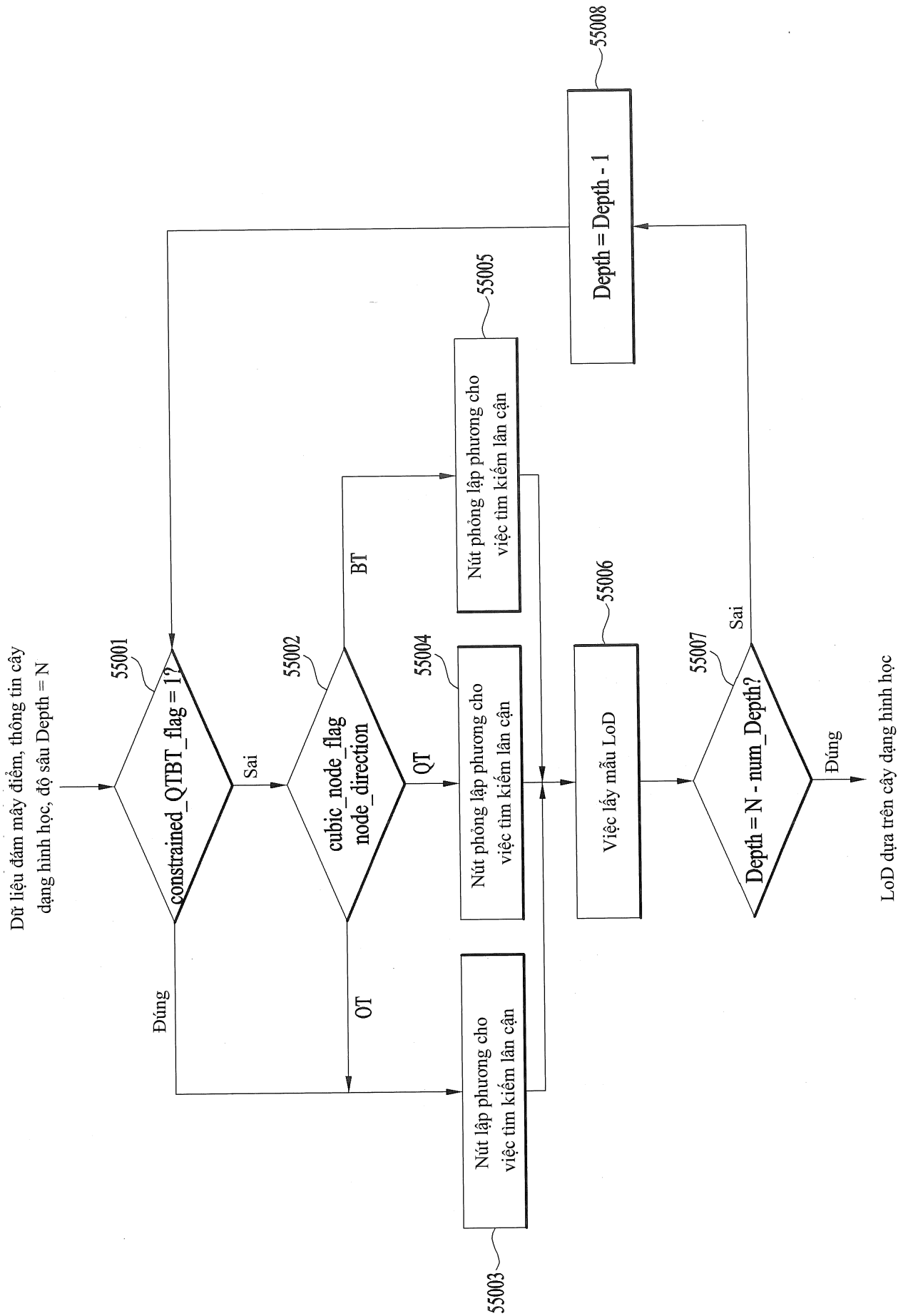


FIG. 48

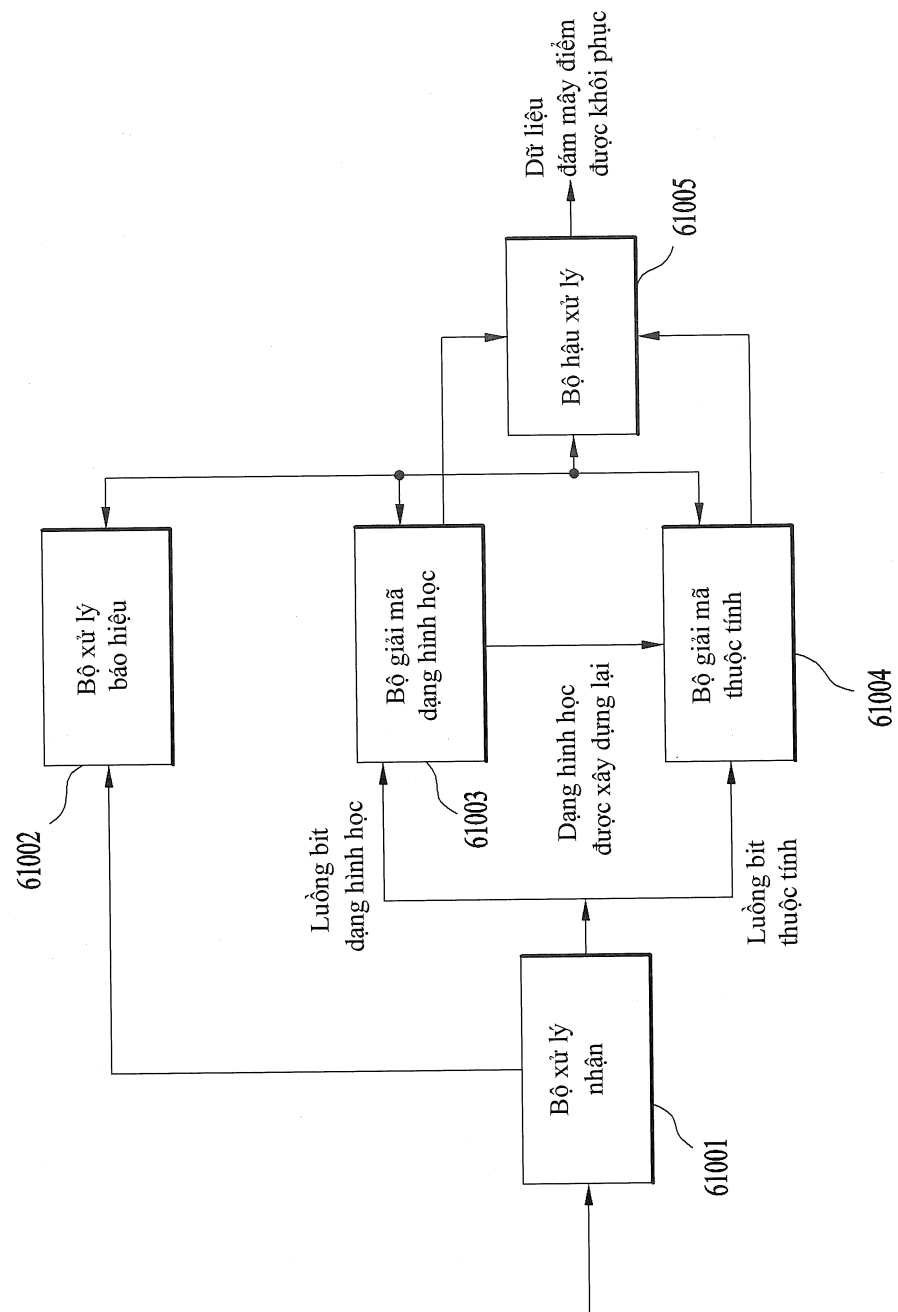


FIG. 49

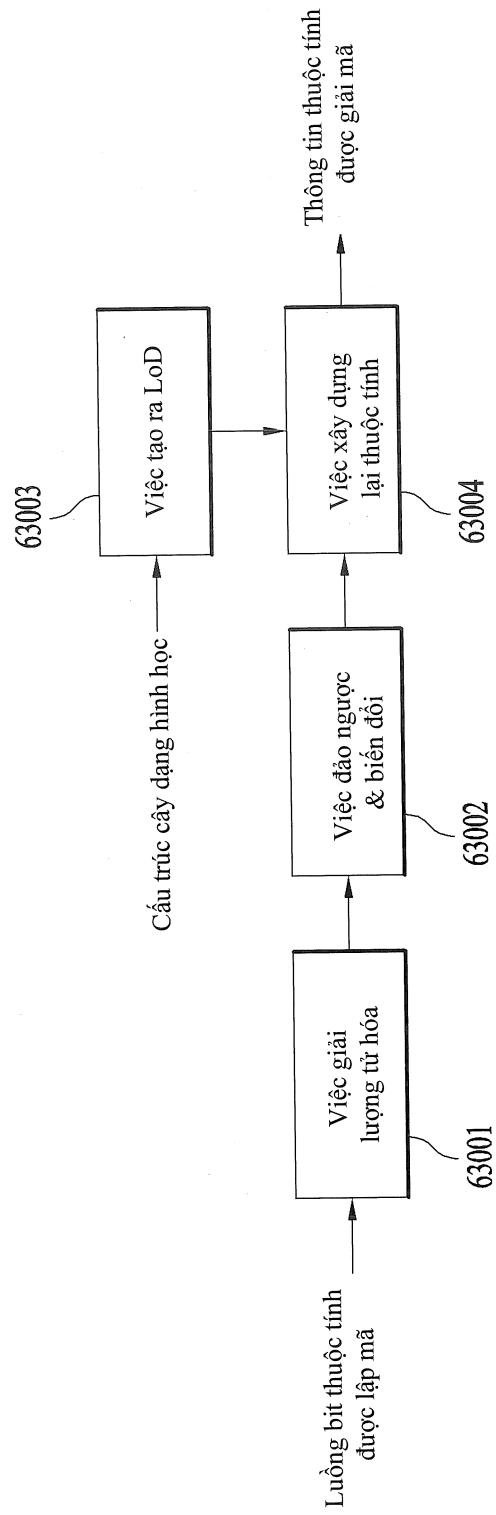


FIG. 50

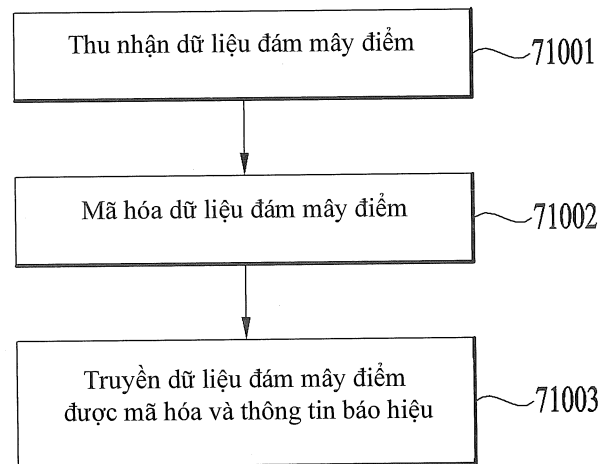


FIG. 51

