



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11) 
1-0021136

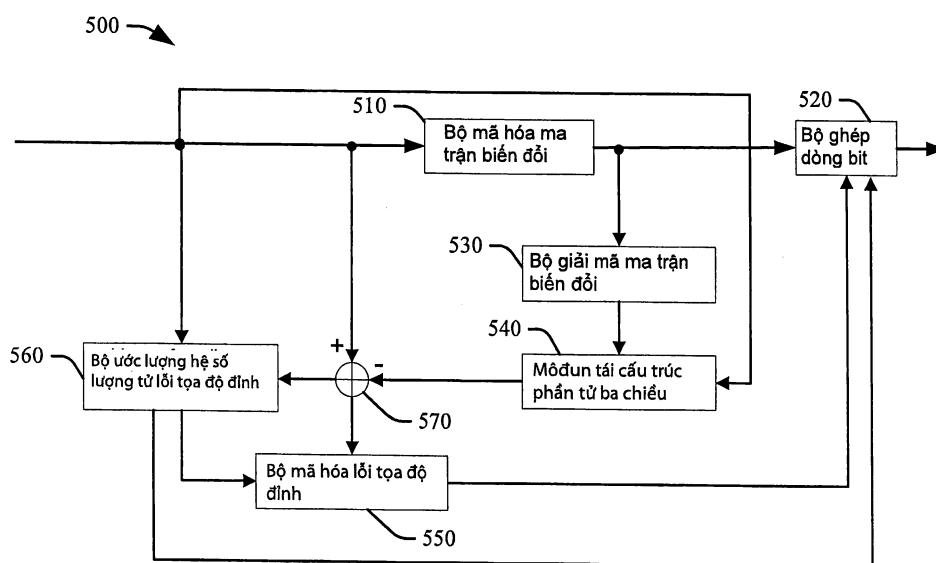
(51)⁷ G06T 9/00

(13) B

(21) 1-2015-02454 (22) 18.06.2013
(86) PCT/CN2013/077404 18.06.2013 (87) WO2014/107944 17.07.2014
(30) PCT/CN2013/070310 10.01.2013 CN
(45) 25.06.2019 375 (43) 25.11.2015 332
(73) THOMSON LICENSING (FR)
1-5 rue Jeanne d'Arc, F-92130 Issy-Les-Moulineaux, France
(72) JIANG, Wenfei (CN), CAI, Kangying (CN), TIAN, Jiang (CN)
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ TẠO RA DÒNG BIT THỂ HIỆN MÔ HÌNH BA CHIỀU, VÀ PHƯƠNG PHÁP VÀ THIẾT BỊ GIẢI MÃ DÒNG BIT NÀY

(57) Sáng chế đề cập đến phương pháp và thiết bị tạo ra dòng bit thể hiện mô hình ba chiều, và phương pháp và thiết bị giải mã dòng bit này. Mô hình ba chiều có thể được mô hình hóa bằng cách sử dụng "đối tượng mẫu", trong đó phần tử đối tượng có thể được thể hiện ở dạng biến thể (ví dụ, quay, dịch chuyển, và biến đổi tỷ lệ) của mẫu. Lỗi lượng tử hóa có thể phát sinh khi mã hóa thông tin quay, gây ra các lỗi tọa độ đỉnh khác nhau ở các đỉnh khác nhau của đối tượng. Để bù các lỗi tọa độ đỉnh một cách hữu hiệu, bộ mã hóa quyết định thông số lượng tử hóa để bù lỗi tọa độ đỉnh. Thông số lượng tử hóa được truyền tín hiệu trong dòng bit ở dạng chỉ số lượng tử hóa. Chỉ số lượng tử hóa, bảng lượng tử hóa xác định ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa, và các lỗi tọa độ đỉnh được mã hóa thành dòng bit. Bảng lượng tử hóa có thể được xây dựng dựa trên dữ liệu thống kê. Ở bộ giải mã, lỗi tọa độ đỉnh được giải mã dựa trên thông số lượng tử hóa, được xác định từ chỉ số lượng tử hóa thu được.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và thiết bị tạo ra dòng bit thể hiện mô hình ba chiều, và phương pháp và thiết bị giải mã dòng bit này.

Tình trạng kỹ thuật của sáng chế

Trong các ứng dụng thực tế, nhiều mô hình ba chiều chứa một lượng lớn các phần tử được nối với nhau. Các mô hình ba chiều nhiều phần tử này thường chứa nhiều cấu trúc lặp trong nhiều hình thái khác nhau, như được thể hiện trên Fig.1.

Đã biết các thuật toán nén dành cho mô hình ba chiều nhiều phần tử lợi dụng cấu trúc lặp trong các mô hình nhập vào. Người ta đã phát hiện ra các cấu trúc lặp của mô hình ba chiều ở nhiều vị trí, hướng và thông số tỷ lệ. Sau đó, mô hình ba chiều được sắp xếp để thể hiện dưới dạng "đối tượng mẫu". Mẫu được dùng để biểu thị hình dạng điển hình của cấu trúc lặp tương ứng. Các phần tử thuộc cấu trúc lặp được biểu thị dưới dạng các đối tượng của các mẫu tương ứng và có thể được thể hiện bằng mã số nhận dạng của mẫu và thông tin biến đổi, ví dụ, phản xạ, dịch chuyển, quay và có thể biến đổi tỷ lệ so với mẫu. Ví dụ, thông tin biến đổi của đối tượng có thể được sắp xếp thành phần phản xạ, phần dịch chuyển, phần quay, và phần có thể biến đổi tỷ lệ. Có thể có một số phần tử của các mô hình ba chiều không lặp lại, được gọi là các phần tử duy nhất.

Đơn PCT của các đồng Chủ đơn W. Jiang, K. Cai, và J. Tian (PCT/CN2012/074286, số tham chiếu của đại diện sở hữu trí tuệ PA120012, sau đây gọi là "Jiang") với tiêu đề "Vertex Correction for Rotated 3D Components", nội dung của đơn này được kết hợp ở đây bằng cách viện dẫn, bộc lộ phương pháp và thiết bị bù lỗi đỉnh khi mã hóa và giải mã mô hình ba chiều.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất phương pháp tạo ra dòng bit thể hiện mô hình ba chiều bao gồm các bước: truy cập đối tượng tái cấu trúc tương ứng với đối tượng; xác định thông số lượng tử hóa dựa trên lỗi tọa độ đỉnh giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc; xác định chỉ số lượng tử hóa để đáp lại thông số lượng tử hóa; và mã hóa chỉ số lượng tử hóa và lỗi tọa độ đỉnh vào dòng bit như được mô tả dưới đây. Sáng chế cũng đề xuất thiết bị để thực hiện các bước nêu trên.

Sáng chế đề xuất phương pháp giải mã dòng bit thể hiện mô hình ba chiều bao gồm các bước: truy cập đối tượng tái cấu trúc tương ứng với đối tượng; xác định chỉ số lượng tử hóa từ dòng bit; xác định thông số lượng tử hóa để đáp lại chỉ số lượng tử hóa; giải mã lỗi tọa độ đỉnh thể hiện lỗi giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc; và cải thiện đối tượng tái cấu trúc để đáp lại lỗi tọa độ đỉnh đã được mã hóa như được mô tả dưới đây. Sáng chế cũng đề xuất thiết bị để thực hiện các bước nêu trên.

Sáng chế cũng đề xuất phương tiện lưu trữ đọc được bằng máy tính đã lưu trữ các câu lệnh để tạo ra hoặc giải mã dòng bit thể hiện mô hình ba chiều theo các phương pháp được mô tả trên đây.

Sáng chế cũng đề xuất phương tiện lưu trữ đọc được bằng máy tính đã lưu trữ dòng bit thể hiện mô hình ba chiều được tạo ra theo các phương pháp được mô tả trên đây.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ thể hiện các mô hình ba chiều làm ví dụ với một lượng lớn các phân tử được nối với nhau và các cấu trúc lặp;

Fig.2A là hình vẽ thể hiện các ví dụ về hình dạng của các mẫu và Fig.2B là hình vẽ thể hiện các ví dụ về hình dạng của các đối tượng tương ứng và các đối tượng tái

cấu trúc;

Fig.3 là lưu đồ thể hiện một ví dụ về mã hóa đối tượng của mô hình ba chiều, theo một phương án của sáng chế;

Fig.4 là lưu đồ thể hiện một ví dụ về giải mã đối tượng của mô hình ba chiều, theo một phương án của sáng chế;

Fig.5 là hình vẽ thể hiện một ví dụ về bộ mã hóa đối tượng theo sáng chế; và

Fig.6 là hình vẽ thể hiện một ví dụ về bộ giải mã đối tượng theo sáng chế.

Mô tả chi tiết sáng chế

Như được thể hiện trên Fig.1, có thể có nhiều cấu trúc lặp trong các mô hình ba chiều. Để mã hóa một cách hữu hiệu các mô hình ba chiều, các cấu trúc lặp có thể được sắp xếp thành các mẫu và các đối tượng, trong đó đối tượng có thể được thể hiện dưới dạng biến thể của mẫu tương ứng, ví dụ, bằng cách sử dụng mã số nhận dạng mẫu và ma trận biến đổi chứa thông tin như dịch chuyển, quay và biến đổi tỷ lệ.

Khi đối tượng được thể hiện bằng mã số nhận dạng mẫu và ma trận biến đổi, mã số nhận dạng mẫu và ma trận biến đổi được nén khi nén đối tượng. Do đó, đối tượng có thể được tái cấu trúc thông qua mã số nhận dạng mẫu và ma trận biến đổi được giải mã, tức là đối tượng có thể được tái cấu trúc dưới dạng biến thể (từ ma trận biến đổi được giải mã) của mẫu được giải mã được ký hiệu bởi mã số nhận dạng mẫu. Theo một phương án, khi mã hóa ma trận biến đổi, phần quay của ma trận biến đổi có thể được lượng tử hóa, ví dụ, bằng cách sử dụng một số lượng bit cố định. Vì sự mất xảy ra ở bước lượng tử hóa, phần quay có thể khác so với phần quay ban đầu.

Các hình vẽ Fig.2A và Fig.2B thể hiện các phần tử ở dạng hai chiều, trong đó các phần tử 210 và 220 là các mẫu, các phần tử 250 và 270 (ở dạng các đường nét liền) là các đối tượng ban đầu được nén, và các phần tử 260 và 280 (ở dạng các đường đứt nét) là các thể tái cấu trúc. Cụ thể hơn, các đối tượng 250 và 270 có thể được thể hiện lần

lượt dưới dạng phiên bản đã dịch chuyển (tức là, quay và dịch chuyển) của các mẫu 210 và 220.

Trong các ví dụ được thể hiện trên Fig.2B, quá trình lượng tử và quay tạo ra lỗi khoảng 5° , do đó gây ra sự khác nhau giữa các đối tượng ban đầu và các đối tượng tái cấu trúc. Như được thể hiện trên Fig.2B, khi các lỗi quay (được đo bằng góc) là tương tự đối với các đối tượng 250 và 270, các lỗi tọa độ đỉnh (tức là, ví dụ, sự dịch chuyển đỉnh từ A đến A', từ B đến B' trên Fig.2B, giữa các đối tượng ban đầu và các đối tượng tái cấu trúc) gây ra bởi phép lượng tử quay thay đổi đáng kể giữa cả hai đối tượng, với đối tượng 270 có lỗi tọa độ đỉnh lớn hơn nhiều. Do đó, chất lượng của các phần tử tái cấu trúc có thể không đồng đều, ví dụ, đối tượng lớn có thể có chất lượng tái cấu trúc thấp hơn so với đối tượng nhỏ.

Theo đơn sáng chế của Jiang, để bù một cách hữu hiệu các lỗi tọa độ đỉnh, giới hạn trên có thể được ước lượng cho lỗi tọa độ đỉnh. Dựa trên giới hạn trên, bộ mã hóa - giải mã quyết định xem liệu lỗi tọa độ đỉnh có cần được bù hay không, và quyết định thông số lượng tử hóa để bù lỗi tọa độ đỉnh nếu việc bù là cần thiết. Giới hạn trên có thể được ước lượng ở cả bộ mã hóa và bộ giải mã, và do đó không cần truyền tín hiệu chi tiết để xác định xem việc bù lỗi tọa độ đỉnh đã được sử dụng chưa để xác định thông số lượng tử hóa cho lỗi tọa độ đỉnh.

Sáng chế cũng đề xuất phương pháp và thiết bị để bù một cách hữu hiệu lỗi tọa độ đỉnh gây ra bởi phép lượng tử quay. Để giảm tải cho việc tính toán ở bộ giải mã, thông số lượng tử hóa được gửi tín hiệu trong dòng bit. Theo một phương án, ký hiệu tương ứng với số lượng bit lượng tử được dùng để lượng tử hóa lỗi tọa độ đỉnh được truyền trong dòng bit thông qua bảng lượng tử hóa.

Fig.3 thể hiện một ví dụ về phương pháp 300 để mã hóa đối tượng của mô hình ba chiều. Phương pháp 300 bắt đầu từ bước 305. Ở bước 310, dữ liệu về mô hình ba chiều được nhập và việc khởi tạo được thực hiện. Dữ liệu bổ sung, như thông số chất

lượng, lỗi tọa độ đỉnh lớn nhất có thể bỏ qua được, các thông số lượng tử hóa cho phân dịch chuyển và phân quay của ma trận biến đổi cũng có thể được nhập hoặc được suy ra từ dữ liệu nhập. Theo một phương án làm ví dụ, bước khởi tạo có thể sắp xếp các cấu trúc lặp thành các mẫu và các đối tượng, tạo ra các ma trận biến đổi dành cho các đối tượng, và mã hóa các mẫu để tạo ra các mẫu tái cấu trúc. Đối với một đối tượng cần mã hóa cụ thể (được biểu thị bằng chữ C), mẫu ban đầu tương ứng, mẫu tái cấu trúc, và các ma trận biến đổi lần lượt được biểu thị bằng P, P', và T. Đối tượng cũng có thể được thể hiện một cách chính xác dưới dạng biến thể của mẫu, tức là, $C=TP$. Theo cách khác, phép biến thể mẫu có thể là lấy xấp xỉ đối tượng trong một số trường hợp, tức là, $C\approx TP$.

Ma trận biến đổi (T) được mã hóa ở bước 320. Sau đó, ở bước 330, ma trận biến đổi được mã hóa được giải mã thành T', và đối tượng được tái cấu trúc, ví dụ, bằng cách sử dụng mẫu tái cấu trúc tương ứng và ma trận biến đổi được giải mã ($C'=T'P'$).

Ở bước 340, lỗi tọa độ đỉnh (E_i) giữa đỉnh (V_i') trong đối tượng tái cấu trúc và đỉnh tương ứng (V_i) trong đối tượng ban đầu được tính toán, ví dụ, bằng công thức $E_i = V_i - V_i'$. Để mã hóa lỗi tọa độ đỉnh, thông số lượng tử hóa được ước lượng ở bước 350. Lỗi tọa độ đỉnh được lượng tử hóa và được mã hóa ở bước 360. Ngoài ra, thông số lượng tử hóa được truyền tín hiệu trong dòng bit. Ở bước 370, kiểm tra xem liệu có cần xử lý nhiều đỉnh hơn không. Nếu cần xử lý nhiều đỉnh hơn, thì quay trở lại bước 340. Nếu không thì quay trở lại bước 399.

Để truyền tín hiệu thông số lượng tử hóa một cách hữu hiệu, thì không phải là thông số lượng tử hóa thực tế, mà ký hiệu của thông số lượng tử hóa có thể được mã hóa thành dòng bit. Bằng cách sử dụng một số bit lượng tử làm ví dụ về thông số lượng tử hóa, quy trình lượng tử sẽ được đề cập chi tiết hơn. Sáng chế cũng có thể được áp dụng khi các thông số lượng tử hóa khác được sử dụng, ví dụ, nhưng không giới hạn ở, kích thước bước lượng tử.

Bảng 1

Chỉ số lượng tử hóa	Số lượng bit lượng tử hóa
0	2
1	4
2	5
3	6

Bảng 1 thể hiện một ví dụ về bảng lượng tử hóa, trong đó số lượng bit lượng tử được ánh xạ thành chỉ số lượng tử hóa. Cụ thể hơn,

$$QB_{\text{bảng}}[0] = 2;$$

$$QB_{\text{bảng}}[1] = 4;$$

$$QB_{\text{bảng}}[2] = 5;$$

$$QB_{\text{bảng}}[3] = 6.$$

MaxErrorAllow được biểu thị là yêu cầu về chất lượng (tức là, lỗi tọa độ đỉnh lớn nhất có thể bỏ qua được) được cung cấp bởi người dùng và Error là sự khác biệt giữa các tọa độ của đối tượng ban đầu và đối tượng tái cấu trúc đối với một đỉnh cụ thể. Đối với một đỉnh cụ thể, chúng ta ước lượng số lượng bit ban đầu cần để lượng tử hóa các lỗi tọa độ đỉnh là:

$$QB = \text{ceil}[\log_2(\text{Error} / \text{MaxErrorAllow})]. \quad (1)$$

Sau đó, chúng ta tìm kiếm bảng lượng tử hóa cho số lượng bit lượng tử gần nhất với QB. Ví dụ, khi QB=7, chỉ số lượng tử hóa 3 ($QB_{\text{bảng}}[3] = 6$) được lựa chọn làm chỉ số lượng tử hóa tương ứng và QB được thiết lập bằng 6. Sau đó, Error / MaxErrorAllow được lượng tử hóa thành mã nhị phân của các bit QB. Sau đó, chỉ số lượng tử hóa và mã nhị phân được mã hóa thành dòng bit. Như được thể hiện trên Bảng 1, giá trị của chỉ số lượng tử hóa thường nhỏ hơn số lượng bit lượng tử tương ứng, và có thể yêu cầu ít bit hơn để truyền. Do đó, việc gửi chỉ số lượng tử hóa thay vì gửi số lượng các bit lượng tử theo cách trực tiếp có thể giảm lưu lượng bit.

Ở bộ giải mã, lỗi lớn nhất có thể bỏ qua được (MaxErrorAllow) và bảng lượng tử hóa có thể được tạo ra từ dòng bit. Đối với đỉnh, chỉ số lượng tử hóa nhận được từ dòng bit, và số lượng bit lượng tử QB có thể được xác định từ chỉ số lượng tử hóa và bảng lượng tử hóa. Sau đó, lỗi tọa độ đỉnh được lượng tử hóa giá trị_Q có thể được đọc dưới dạng các bit QB từ dòng bit. Lỗi tọa độ đỉnh có thể được tính như sau:

$$\text{Error}' = \text{MaxErrorAllow} * \text{giá trị_Q}. \quad (2)$$

Ví dụ, giả sử chỉ số lượng tử hóa và mã nhị phân thể hiện lỗi tọa độ đỉnh được lượng tử hóa được mã hóa trong dòng bit một cách liên tiếp và là '1010100111...'. Nếu chỉ số lượng tử hóa được mã hóa bởi 2 bit, thì ta thu được chỉ số lượng tử hóa '10' = 2. Nếu bảng lượng tử hóa như được thể hiện trên Bảng 1 được sử dụng, thì số lượng bit lượng tử có thể được tính là QBbảng [2] = 5. Sau đó, 5 được biểu thị là '10100' từ dòng bit và xác định lỗi tọa độ đỉnh là giá trị_Q = '10100' = 20. Do đó, Error' = MaxErrorAllow * 20.

Như được đề cập ở trên, bảng lượng tử hóa được dùng để xác định thông số lượng tử hóa trong dòng bit. Theo một phương án, bảng lượng tử hóa có thể được xác định bằng siêu dữ liệu hoặc được người dùng nhập vào. Theo một phương án khác, bảng lượng tử hóa có thể được xây dựng dựa trên dữ liệu thống kê.

Ví dụ, chúng ta có thể tính giá trị cho QB, dựa trên phương trình. (1), bằng cách sử dụng các đỉnh khác nhau từ một lượng lớn các phần tử từ các mô hình ba chiều khác nhau. Sau khi chúng ta thu được một lượng lớn các giá trị QB, chúng ta có thể lựa chọn n giá trị QB hay xuất hiện nhất làm các phần tử của bảng QB. Nếu chúng ta biểu thị n giá trị hay xuất hiện nhất là QB₀, QB₁, ..., QB_{n-1}, thì bảng lượng tử hóa có thể được thể hiện như trong Bảng 2. Khi việc mã hóa với độ dài cố định được dùng để mã hóa chỉ số lượng tử hóa, chúng ta có thể xây dựng bảng sao cho QB₀ < QB₁ < ... < QB_{n-1}. Khi việc mã hóa với độ dài thay đổi được dùng để mã hóa chỉ số lượng tử hóa, để làm giảm lượng dữ liệu dùng để gửi các chỉ số lượng tử hóa, chúng ta có thể thiết

lập bảng lượng tử hóa sao cho $\text{Prob}(\text{QB}_0) > \text{Prob}(\text{QB}_1) > \dots > \text{Prob}(\text{QB}_{n-1})$. Tức là, số lượng bit lượng tử với xác suất xảy ra lớn hơn tương ứng với ký hiệu nhỏ hơn, thường cần ít bit hơn để mã hóa.

Bảng 2

Chỉ số lượng tử hóa	Số lượng bit lượng tử hóa
0	QB_0
1	QB_1
...	...
n-1	QB_{n-1}

Fig.4 thể hiện một ví dụ về phương pháp 400 để giải mã đối tượng của mô hình ba chiều. Đầu vào của phương pháp 400 có thể bao gồm dòng bit, ví dụ, dòng bit được tạo ra bằng cách sử dụng phương pháp 300. Dữ liệu bổ sung, ví dụ, mẫu tái cấu trúc (p') tương ứng với đối tượng cần được giải mã cũng có thể được coi là đầu vào. Phương pháp 400 bắt đầu từ bước 405. Ở bước 410, việc khởi tạo được thực hiện, ví dụ, thông số lượng tử hóa dành cho ma trận biến đổi và thông số chất lượng xuất phát từ dòng bit nhập vào và lỗi tọa độ đỉnh lớn nhất có thể bỏ qua được tính từ thông số chất lượng.

Ma trận biến đổi được giải mã là T' , và đối tượng được tái cấu trúc là C' ở bước 420, ví dụ, bằng cách sử dụng mẫu tái cấu trúc tương ứng và ma trận biến đổi được giải mã ($C' = T'P'$). Ở bước 430, thông số lượng tử hóa, ví dụ, số lượng bit lượng tử (QB), được xác định từ dòng bit. Lỗi tọa độ đỉnh được mã hóa sẽ được giải mã ở bước 440, ví dụ, các bit QB được đọc từ dòng bit và lỗi tọa độ đỉnh được tính toán bằng cách sử dụng phương trình. (2). Ở bước 450, lỗi tọa độ đỉnh được giải mã (E_i') được

dùng để bù đỉnh tương ứng (V_i') của đối tượng được tái cấu trúc từ đầu ở bước 420, ví dụ, là $V_i'' = V_i' + E_i'$. Tức là, đỉnh của đối tượng tái cấu trúc được cải thiện. Phương pháp 400 kết thúc ở bước 499.

Cờ hiệu bù lỗi tọa độ đỉnh có thể được dùng để xác định lỗi tọa độ đỉnh có được bù hay không. Cờ hiệu cần được biết ở cả bộ mã hóa và bộ giải mã. Khi cờ hiệu được thiết lập bằng 1, việc bù lỗi được sử dụng. Nếu không, việc bù lỗi tọa độ đỉnh không được sử dụng. Cụ thể hơn, khi các phương pháp 300 và 400 được sử dụng, các bước từ 340 đến 370 trong phương pháp 300 và các bước từ 430-460 là không cần thiết nếu cờ hiệu là 0.

Theo một phương án làm ví dụ, quy trình giải mã của dòng bit thể hiện mô hình ba chiều có thể được mô tả bằng cách sử dụng các mã giả sau.

```
void PB3DMC_Decoder()
{
    Đọc PB3DMC_stream_header;
    if (uni_part_bit == 0 && repeat_struc_bit == 0)
    {
        Giải mã mô hình ba chiều bằng cách sử dụng bộ giải mã được xác định bởi
        3d_model_compr_mode;
    }
    else
    {
        if (uni_part_bit == 1)
        {
            Giải mã phần đặc biệt bằng cách sử dụng bộ giải mã được xác định bởi
            3d_model_compr_mode;
            Tách các phần tử đặc biệt khác nhau bằng cách chuyển dựa trên độ liên kết;
            Giải mã các vector của các phần tử đặc biệt;
            Tái cấu trúc phần đặc biệt bằng cách dịch chuyển tất cả các phần tử đặc biệt tái
            cấu trúc tới các vị trí của chúng;
        }
        if (repeat_struc_bit == 1)
        {
            Repeat_Struc_Decoder();
        }
    }
}
```

```

    }
}
}

```

```
void Repeat_Struc_Decoder ()
```

```

{
    Giải mã tất cả các mẫu;
    if (sym_instance_num > 0)
    {
        Giải mã tất cả các đối tượng đối xứng;
        Giải mã tất cả thông tin gắn kết;
    }

```

Tái cấu trúc tất cả các mẫu có cấu trúc lặp không gắn kết và các phần tử đặc biệt có cấu trúc đối xứng, bằng cách sử dụng các mẫu khôi phục lại, các đối tượng đối xứng và thông tin gắn kết;

Giải mã các vector dịch chuyển của tất cả các mẫu có cấu trúc lặp không gắn kết và các phần tử đặc biệt có cấu trúc đối xứng;

Tái cấu trúc các phần tử tương ứng với tất cả các mẫu có cấu trúc lặp không gắn kết và các phần tử đặc biệt có cấu trúc đối xứng bằng cách sử dụng các vector dịch chuyển được giải mã;

```

    if (insta_trans_elem_bit == 1)
    {
        Instance_Elementary_Mode_Decoder();
    }
    else
    {
        Instance_Grouped_Mode_Decoder();
    }
}

```

```
void Instance_Elementary_Mode_Decoder()
```

```

{
    for (i = 0; i < numInstance; i++)
    {
        Đọc elem_insta_QP_translation_flag;
        Đọc elem_insta_QP_rotation_flag;
        if(elem_insta_QP_translation_flag == 1)
        {

```

```

    Giải mã elem_QP_translation;
}
else
{
    elem_QP_translation = QP_Translation;
}
if(elem_insta_QP_rotation_flag == 1)
{
    Giải mã elem_QP_rotation;
}
else
{
    elem_QP_rotation = QP_rotation;
}
Giải mã idPattern;
Đọc elem_insta_flip_flag;
Đọc elem_insta_reflection_flag;
Đọc elem_insta_attribute_header;

```

Giải mã vectơ dịch chuyển đối tượng nhờ bộ giải mã có chiều dài cố định có thông số là QB_translation;

Giải mã các góc O-le bằng bộ giải mã có chiều dài cố định có thông số là QB_translation;

Khôi phục lại ma trận quay bằng cách sử dụng các góc O-le được giải mã;

```

if (using_scaling_bit == 1)

```

```

    Giải mã thông số tỷ lệ;

```

```

if (error_compen_enable_bit == 1)

```

```

{

```

```

    Đọc elem_insta_error_compen_flag;

```

```

    if (elem_insta_error_compen_flag == 1)

```

```

        Giải mã dữ liệu bù lỗi

```

```

    }

```

Khôi phục lại hình dạng của đối tượng hiện tại bằng mẫu được xác định bằng idPattern, vectơ dịch chuyển khôi phục lại, ma trận quay khôi phục lại, chuyển hướng

dịch chuyển nếu có, thông số tỷ lệ nếu có, dữ liệu bù lỗi nếu có;

if(elem_insta_flip_flag == 1)

Lật tất cả các tam giác của đối tượng hiện tại;

Giải mã dữ liệu thuộc tính của đối tượng hiện tại;

}
}

void Instance_Grouped_Mode_Decoder()

{

Đọc elem_insta_QB_translation_flag của tất cả các đối tượng;

Đọc elem_insta_QB_rotation_flag của tất cả các đối tượng;

Giải mã elem_insta_QB_translation của các đối tượng có

elem_insta_QB_translation_flag là 1;

Giải mã elem_insta_QB_rotation của các đối tượng có elem_insta_QB_rotation_flag là 1;

Đọc compr_insta_patternID_header;

Giải mã mã số nhận dạng mẫu của tất cả các đối tượng;

Đọc elem_insta_QB_translation_flag của tất cả các đối tượng;

Đọc elem_insta_QB_translation_flag của tất cả các đối tượng;

Đọc compr_insta_transl_header;

Giải mã các vector dịch chuyển của tất cả các đối tượng bằng bộ giải mã phân tích phần tử hữu hạn dựa trên;

Đọc compr_insta_rotat_header;

Giải mã the Euler angles of all instances;

Khôi phục lại các ma trận quay của tất cả các đối tượng;

if (use_scaling_bit == 1)

{

Đọc compr_insta_scaling_header;

Giải mã các thông số tỷ lệ của tất cả các đối tượng;

}

if (error_compen_enable_bit == 1)

```

{
    Đọc elem_insta_error_compen_flag của tất cả các đối tượng;
    for (i = 0; i < numInstance; i++)
    {
        if (elem_insta_error_compen_flag tương ứng là 1)
            Giải mã dữ liệu bù lỗi cho các đối tượng hiện tại;
    }
}

Khôi phục lại hình dạng của tất cả các đối tượng bằng các mẫu khôi phục lại, các
vector dịch chuyển khôi phục lại, các ma trận quay khôi phục lại, các biến thể chuyển
hướng nếu có, các thông số tỷ lệ nếu có, dữ liệu bù lỗi nếu có;
Giải mã dữ liệu thuộc tính của tất cả các đối tượng nếu có;
}

```

Trong Bảng 3, các cú pháp và ngữ nghĩa học được thể hiện cho bảng lượng tử hóa, có thể có trong phần đầu dòng bit.

Bảng 3

if (error_compen_enable_bit==’1’){
error_compen_QB_table[0..3]
}

error_compen_enable_bit: Số nguyên không dấu 1-bit này xác định liệu có các trường dữ liệu của các dữ liệu bù lỗi được nén cho một số đối tượng trong dòng bit. 0 nghĩa là không có trường dữ liệu của dữ liệu bù lỗi mã hóa được nén của các đối tượng trong dòng bit và 1 nghĩa là có trường dữ liệu của dữ liệu bù lỗi mã hóa được nén của một số đối tượng trong dòng bit. Bit này tương ứng với cờ hiệu bù lỗi tọa độ đỉnh được đề cập ở trên cho các phương pháp 300 và 400.

error_compen_QB_table: Nếu chế độ bù lỗi được kích hoạt, số lượng bit lượng tử của giá trị bù cho mỗi đỉnh có thể được xác định một cách thích hợp ở bộ mã hóa. Bộ mã hóa truyền ký hiệu của số lượng bit lượng tử thay vì số lượng bit lượng tử tới dòng bit. Bộ giải mã tìm kiếm bảng lượng tử hóa để xác định số lượng bảng lượng tử hóa. Có 4 bit lượng tử định trước trong bảng, mỗi bit được thể hiện bằng một số

nguyên không dấu 5-bit.

Trong Bảng 4, cú pháp và ngữ nghĩa học cho dữ liệu bù lỗi cửa đỉnh được thể hiện. Trong ví dụ này, lớp `compr_elem_insta_error_compen_data` chứa dữ liệu bù lỗi đỉnh được nén của đối tượng thứ i^{th} .

Bảng 4

lớp <code>compr_elem_insta_error_compen_data</code>{
for ($j = 0; j < \text{numofvertex}; j++$) {
<code>elem_compen_err_QB_id</code>
<code>compr_ver_compen_err_data</code>
}

`elem_compen_err_QB_id`: Số nguyên không dấu 2-bit này xác định ký hiệu của số lượng bit lượng tử cho đỉnh thứ j^{th} của đối tượng trong `error_compen_QB_table`.

`compr_ver_compen_err_data`: Trường dữ liệu này chứa giá trị bù được nén của đỉnh thứ j^{th} của đối tượng.

Fig.5 thể hiện biểu đồ khối của một ví dụ về bộ mã hóa đối tượng 500. Đầu vào của thiết bị 500 có thể có đối tượng (C) cần được mã hóa, mẫu tương ứng (P) và mẫu tái cấu trúc (P'), ma trận biến đổi T, thông số chất lượng, và các thông số lượng tử hóa dành cho ma trận biến đổi.

Bộ mã hóa ma trận biến đổi 510 mã hóa ma trận biến đổi T, ví dụ, dựa trên thông số lượng tử hóa dành cho các phần khác nhau của ma trận biến đổi. Bộ giải mã ma trận biến đổi 530 giải mã đầu ra của bộ mã hóa 510 để thu được ma trận biến đổi tái cấu trúc T'. Bằng cách sử dụng mẫu tái cấu trúc tương ứng P' và T', đối tượng có thể được tái cấu trúc là $C' = T'P'$ và ở mô đun tái cấu trúc phần tử ba chiều 540. Bộ cộng 570 lấy sự khác biệt giữa đối tượng ban đầu và đối tượng tái cấu trúc, ví dụ, là $E = C - C'$.

Dựa trên lỗi tọa độ đỉnh E, bộ ước lượng thông số lượng tử hóa lỗi tọa độ đỉnh 560 ước lượng thông số lượng tử hóa cho việc lượng tử hóa lỗi tọa độ đỉnh ở bộ mã hóa lỗi

tọa độ đỉnh 550, ví dụ, bằng cách sử dụng phương trình. (1). Bộ ước lượng thông số lượng tử hóa 560 có thể tiếp tục thu ký hiệu tương ứng dành cho thông số lượng tử hóa đã ước lượng từ bảng lượng tử hóa, và thông số lượng tử hóa có thể được điều chỉnh dựa trên chỉ số lượng tử hóa. Đầu ra của bộ mã hóa ma trận biến đổi 510 và bộ mã hóa lỗi tọa độ đỉnh 550, và chỉ số lượng tử hóa được ghép bằng bộ ghép dòng bit 520 thành dòng bit, mà có thể được kết hợp với các dòng bit khác thể hiện mẫu hoặc các phần tử khác để tạo ra dòng bit tổng thể cho mô hình ba chiều.

Fig.6 thể hiện biểu đồ khối của một ví dụ về bộ giải mã đối tượng 600. Đầu vào của thiết bị 600 có thể có dòng bit tương ứng với đối tượng (C), ví dụ, dòng bit được tạo ra theo phương pháp 300 hoặc bằng bộ mã hóa 500, và mẫu tái cấu trúc tương ứng (P'). Bộ giải mã entropy 610 giải mã dòng bit, ví dụ, để lấy các lỗi tọa độ đỉnh được lượng tử hóa, các chỉ số lượng tử hóa được dùng cho ma trận biến đổi, và bảng lượng tử hóa và các chỉ số lượng tử hóa dùng cho việc bù lỗi tọa độ đỉnh.

Bộ giải mã ma trận biến đổi 620 tái cấu trúc ma trận biến đổi T, ví dụ, dựa trên thông số lượng tử hóa dành cho các phần khác nhau của ma trận biến đổi. Bằng cách sử dụng mẫu tái cấu trúc tương ứng P' và T', đối tượng có thể được tái cấu trúc là $C'=T'P'$ và ở mô đun tái cấu trúc phần tử ba chiều 630.

Bộ giải mã lỗi tọa độ đỉnh 640 tính ra thông số lượng tử hóa, ví dụ, dựa trên chỉ số lượng tử hóa và bảng lượng tử hóa. Sau đó, bộ giải mã này giải mã lỗi tọa độ đỉnh. Các lỗi tọa độ đỉnh được giải mã E' được dùng để cải thiện đối tượng được tái cấu trúc từ đầu ở mô đun tái cấu trúc phần tử ba chiều 630. Cụ thể hơn, bộ cộng 650 cộng các lỗi tọa độ được giải mã (E') và các đối tượng tái cấu trúc ban đầu (C'), ví dụ, là $C''=C'+E'$. C'' thường có sự thể hiện đối tượng ban đầu chính xác hơn so với đối tượng tái cấu trúc ban đầu C'.

Nhiều cách thực hiện và các dấu hiệu kỹ thuật được mô tả trong đơn sáng chế này có thể được dùng trong các tiêu chuẩn MPEG 3DGC và các dạng mở rộng của

chúng.

Phương pháp thực hiện được mô tả ở đây có thể được thực hiện trong, ví dụ, phương pháp hoặc quy trình, thiết bị, chương trình phần mềm, dòng dữ liệu, hoặc tín hiệu. Thậm chí nếu chỉ được đề cập trong trường hợp có một phương án thực hiện duy nhất (ví dụ, chỉ được đề cập trong phương pháp), thì việc thực hiện các dấu hiệu kỹ thuật được đề cập cũng có thể được thực hiện trong các dạng khác (ví dụ, thiết bị hoặc chương trình). Thiết bị có thể được thực hiện trong, ví dụ, phần cứng phù hợp, phần mềm, và phần sụn. Các phương pháp này có thể được thực hiện trong, ví dụ, thiết bị như, ví dụ, bộ xử lý, tức là các thiết bị xử lý nói chung, bao gồm, ví dụ máy tính, bộ vi xử lý, mạch tích hợp, thiết bị logic có thể lập trình được. Các bộ xử lý cũng bao gồm các thiết bị truyền thông, ví dụ như máy tính, điện thoại di động, các thiết bị số di động/cá nhân ("PDAs"), và các thiết bị khác mà tạo điều kiện thuận lợi cho việc truyền thông tin giữa các người dùng đầu cuối.

Theo sáng chế, các thuật ngữ "một phương án" hoặc "phương án" hoặc "một phương án thực hiện" hoặc "phương án thực hiện" cũng như các biến thể của chúng, có nghĩa là một dấu hiệu kỹ thuật, kết cấu, thuộc tính đặc biệt, và được mô tả liên quan tới phương án có trong ít nhất một phương án của sáng chế. Do đó, hình thức của các thuật ngữ "theo một phương án" hoặc "theo phương án" hoặc theo "một phương án thực hiện" hoặc theo "phương án thực hiện" cũng như các biến thể khác, xuất hiện ở nhiều vị trí khác nhau trong toàn bộ bản mô tả không nhất thiết là đều đề cập đến cùng một phương án.

Ngoài ra, sáng chế hoặc các điểm yêu cầu bảo hộ có thể đề cập để "xác định" nhiều loại thông tin khác nhau. Việc xác định thông tin có thể bao gồm một hoặc nhiều, ví dụ, ước lượng thông tin, tính toán thông tin, dự đoán thông tin, hoặc lấy lại thông tin từ bộ nhớ.

Ngoài ra, sáng chế hoặc các điểm yêu cầu bảo hộ có thể đề cập đến "truy cập"

nhiều loại thông tin khác nhau. Ngoài ra, việc "nhận" thông tin thường bao gồm một hoặc nhiều cách, ví dụ như nhận thông tin, lấy lại thông tin (ví dụ, từ bộ nhớ), lưu trữ thông tin, xử lý thông tin, truyền thông tin, di chuyển thông tin, sao chép thông tin, xóa thông tin, tính toán thông tin, xác định thông tin, dự đoán thông tin, hoặc ước lượng thông tin.

Ngoài ra, sáng chế hoặc các điểm yêu cầu bảo hộ có thể đề cập để "tiếp nhận" nhiều loại thông tin khác nhau. Tương tự như với thuật ngữ "truy cập", thuật ngữ nhận cũng được hiểu theo nghĩa rộng. Việc nhận thông tin có thể bao gồm một hoặc nhiều cách, ví dụ truy cập thông tin, hoặc lấy lại thông tin (ví dụ, từ bộ nhớ). Ngoài ra, việc "nhận" thông tin thường bao gồm một hoặc nhiều cách, ví dụ như lưu trữ thông tin, xử lý thông tin, truyền thông tin, di chuyển thông tin, sao chép thông tin, xóa thông tin, tính toán thông tin, xác định thông tin, dự đoán thông tin, hoặc ước lượng thông tin.

Sẽ là hiển nhiên với người có hiểu biết trung bình về lĩnh vực kỹ thuật này rằng việc thực hiện sáng chế có thể tạo ra nhiều loại tín hiệu được định dạng để mang thông tin mà có thể được lưu trữ hoặc truyền chẳng hạn. Thông tin có thể bao gồm, ví dụ, các lệnh để thực hiện một phương pháp, hoặc dữ liệu được tạo ra bởi một hoặc nhiều phương án thực hiện được mô tả ở trên. Ví dụ, tín hiệu có thể được định dạng để mang theo dòng bit của phương án được mô tả ở trên. Tín hiệu đó có thể được định dạng, ví dụ, là sóng điện từ (ví dụ, bằng cách sử dụng dải tần số radio của quang phổ) hoặc là tín hiệu dải tần cơ sở. Việc định dạng có thể bao gồm, ví dụ, mã hóa dòng dữ liệu và môđun hóa vật mang bằng dòng dữ liệu được mã hóa. Thông tin mà tín hiệu mang có thể là, ví dụ thông tin tương tự hoặc thông tin số. Tín hiệu có thể được truyền qua nhiều loại đường truyền có dây hoặc không dây như đã biết. Tín hiệu có thể được lưu trữ trên phương tiện đọc được bằng bộ xử lý.

YÊU CẦU BẢO HỘ

1. Phương pháp tạo ra dòng bit thể hiện mô hình ba chiều bao gồm các bước:
 - truy cập đối tượng tái cấu trúc tương ứng với đối tượng;
 - xác định thông số lượng tử hóa dựa trên lỗi tọa độ đỉnh giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc;
 - xác định chỉ số lượng tử hóa để đáp lại thông số lượng tử hóa; và
 - mã hóa chỉ số lượng tử hóa và lỗi tọa độ đỉnh vào dòng bit.
2. Phương pháp theo điểm 1, trong đó bước xác định chỉ số lượng tử hóa dựa trên cú pháp biểu thị ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng.
3. Phương pháp theo điểm 2, trong đó phương pháp này còn bao gồm bước:
 - xác định ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng dựa trên dữ liệu thống kê.
4. Phương pháp theo điểm 3, trong đó chỉ số lượng tử hóa nhỏ hơn tương ứng với thông số lượng tử hóa phổ biến hơn trong dữ liệu thống kê.
5. Phương pháp theo điểm 1, trong đó thông số lượng tử hóa tiếp tục được xác định để đáp lại lỗi lớn nhất có thể bỏ qua được.
6. Phương pháp theo điểm 1, trong đó thông số lượng tử hóa tương ứng với ít nhất một trong số các bit lượng tử và kích thước của bước lượng tử hóa.
7. Phương pháp giải mã dòng bit thể hiện mô hình ba chiều bao gồm các bước:
 - truy cập đối tượng tái cấu trúc tương ứng với đối tượng;
 - xác định chỉ số lượng tử hóa từ dòng bit;
 - xác định thông số lượng tử hóa để đáp lại chỉ số lượng tử;

giải mã lỗi tọa độ đỉnh thể hiện lỗi giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc, dựa trên thông số lượng tử hóa được xác định; và

cải thiện đối tượng tái cấu trúc để đáp lại lỗi tọa độ đỉnh đã được giải mã.

8. Phương pháp theo điểm 7, trong đó bước xác định thông số lượng tử hóa dựa trên cú pháp biểu thị ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng.

9. Phương pháp theo điểm 7, trong đó lỗi tọa độ đỉnh còn được giải mã để đáp lại lỗi lớn nhất có thể bỏ qua được.

10. Phương pháp theo điểm 7, trong đó thông số lượng tử hóa tương ứng với ít nhất một trong số các bit lượng tử hóa và kích thước của bước lượng tử hóa.

11. Thiết bị tạo ra dòng bit thể hiện mô hình ba chiều bao gồm:

mô đun tái cấu trúc phần tử ba chiều tái cấu trúc đối tượng tái cấu trúc tương ứng với một đối tượng;

bộ ước lượng thông số lượng tử hóa lỗi tọa độ đỉnh xác định thông số lượng tử hóa dựa trên lỗi tọa độ đỉnh giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc và xác định chỉ số lượng tử hóa để đáp lại thông số lượng tử hóa xác định được; và

bộ mã hóa lỗi tọa độ đỉnh mã hóa chỉ số lượng tử hóa và lỗi tọa độ đỉnh thành dòng bit.

12. Thiết bị theo điểm 11, trong đó bộ ước lượng thông số lượng tử hóa lỗi tọa độ đỉnh xác định thông số lượng tử hóa dựa trên cú pháp biểu thị ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng.

13. Thiết bị theo điểm 12, trong đó bộ ước lượng thông số lượng tử hóa lỗi tọa độ đỉnh xác định ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng

dựa trên dữ liệu thống kê.

14. Thiết bị theo điểm 13, trong đó chỉ số lượng tử hóa nhỏ hơn tương ứng với thông số lượng tử hóa phổ biến hơn trong dữ liệu thống kê.

15. Thiết bị theo điểm 11, trong đó thông số lượng tử hóa còn được xác định để đáp lại lỗi lớn nhất có thể bỏ qua được.

16. Thiết bị theo điểm 11, trong đó thông số lượng tử hóa tương ứng với ít nhất một trong số các bit lượng tử hóa và kích thước của bước lượng tử hóa.

17. Thiết bị giải mã dòng bit thể hiện mô hình ba chiều bao gồm:

môđun tái cấu trúc phần tử ba chiều để tái cấu trúc đối tượng tái cấu trúc tương ứng với một đối tượng;

bộ giải mã entropy xác định chỉ số lượng tử hóa từ dòng bit;

bộ giải mã lỗi tọa độ đỉnh (1) xác định thông số lượng tử hóa tương ứng với chỉ số lượng tử hóa và (2) giải mã lỗi tọa độ đỉnh tương ứng với lỗi giữa đỉnh của đối tượng và đỉnh tương ứng của đối tượng tái cấu trúc, dựa trên thông số lượng tử hóa được xác định; và

bộ cộng cải thiện đối tượng tái cấu trúc để đáp lại lỗi tọa độ đỉnh.

18. Thiết bị theo điểm 17, trong đó bộ giải mã lỗi tọa độ đỉnh xác định thông số lượng tử hóa dựa trên cú pháp biểu thị ánh xạ giữa các chỉ số lượng tử hóa và các thông số lượng tử hóa tương ứng.

19. Thiết bị theo điểm 17, trong đó bộ giải mã lỗi tọa độ đỉnh còn giải mã lỗi tọa độ đỉnh để đáp lại lỗi lớn nhất có thể bỏ qua được.

20. Thiết bị theo điểm 17, trong đó thông số lượng tử hóa tương ứng với ít nhất một trong số các bit lượng tử hóa và kích thước của bước lượng tử hóa.

21. Phương tiện lưu trữ đọc được bằng máy tính lưu trữ trên đó các câu lệnh mà, khi

được thực hiện bởi thiết bị xử lý, khiến cho thiết bị xử lý này thực hiện phương pháp theo điểm bất kỳ trong số các điểm 1-6.

22. Phương tiện lưu trữ đọc được bằng máy tính lưu trữ trên đó các câu lệnh mà, khi được thực hiện bởi thiết bị xử lý, khiến cho thiết bị xử lý này thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 7-10.

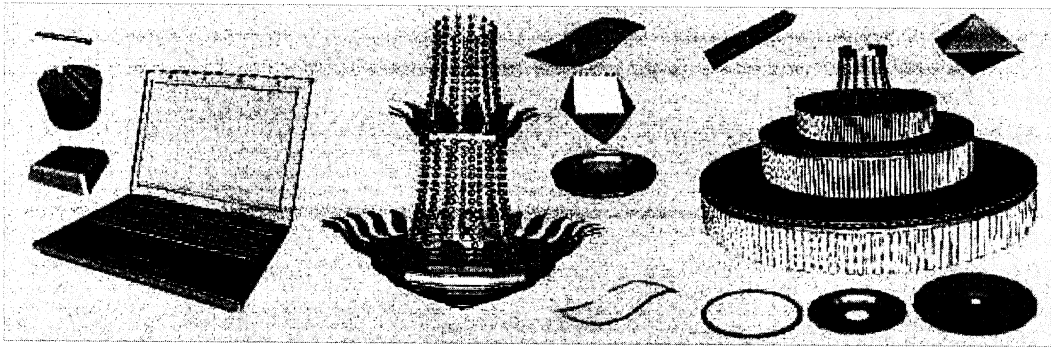


FIG. 1

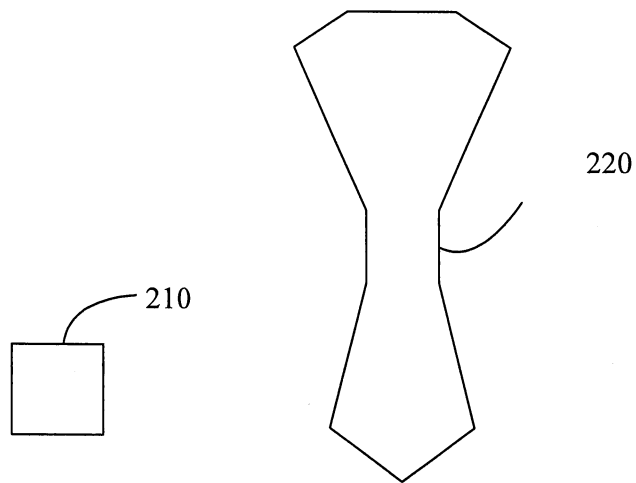


FIG. 2A

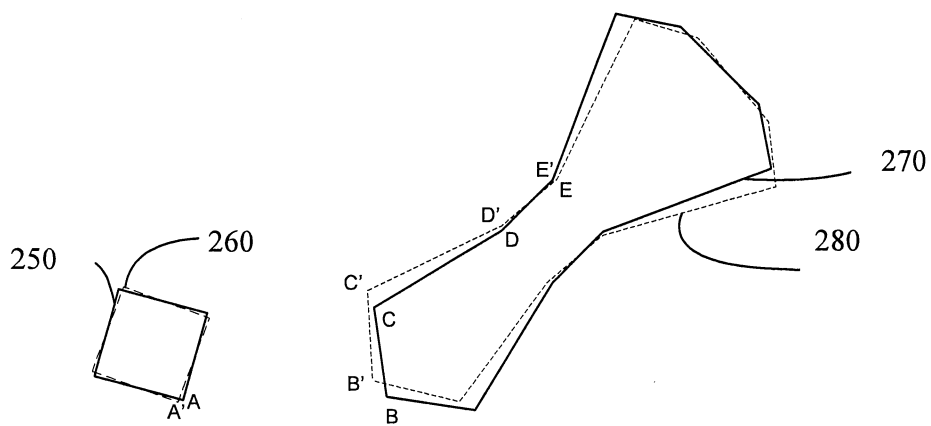


FIG. 2B

300 →

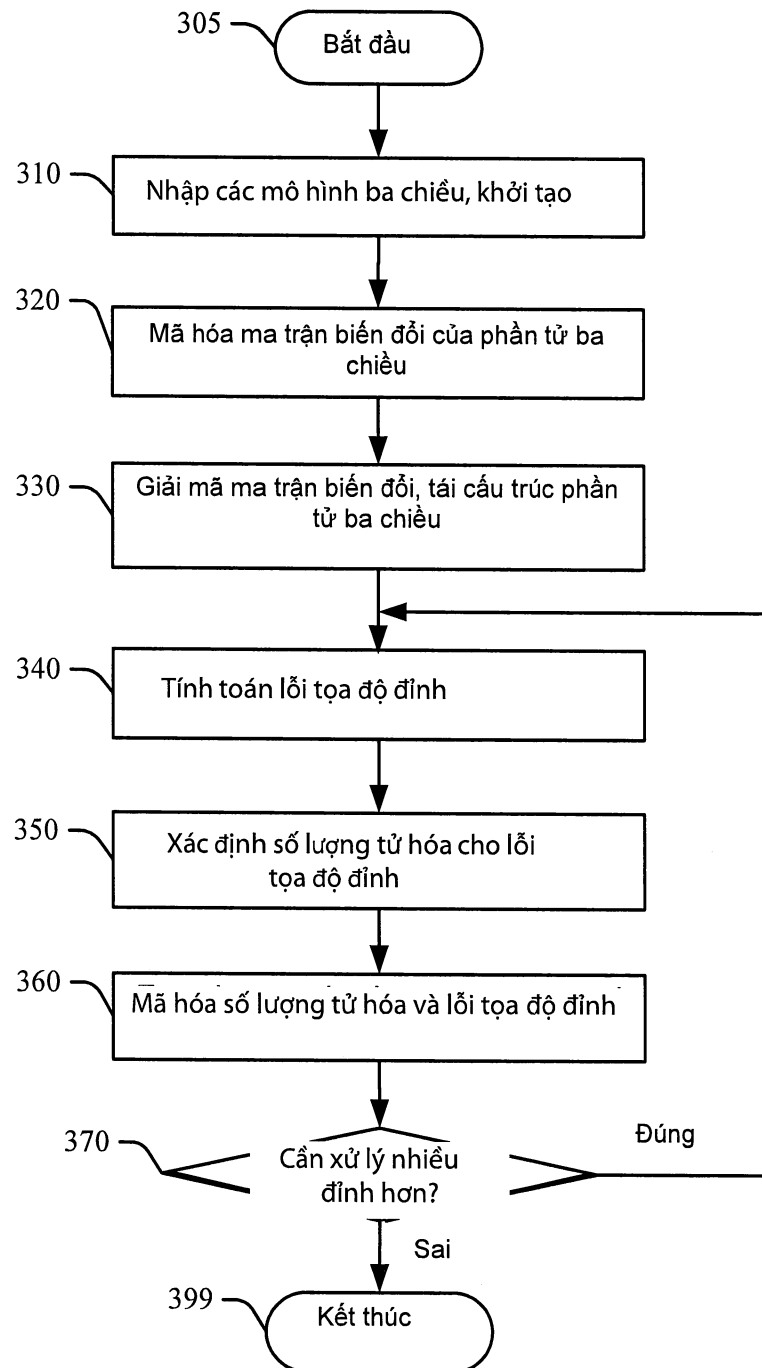


FIG. 3

400 →

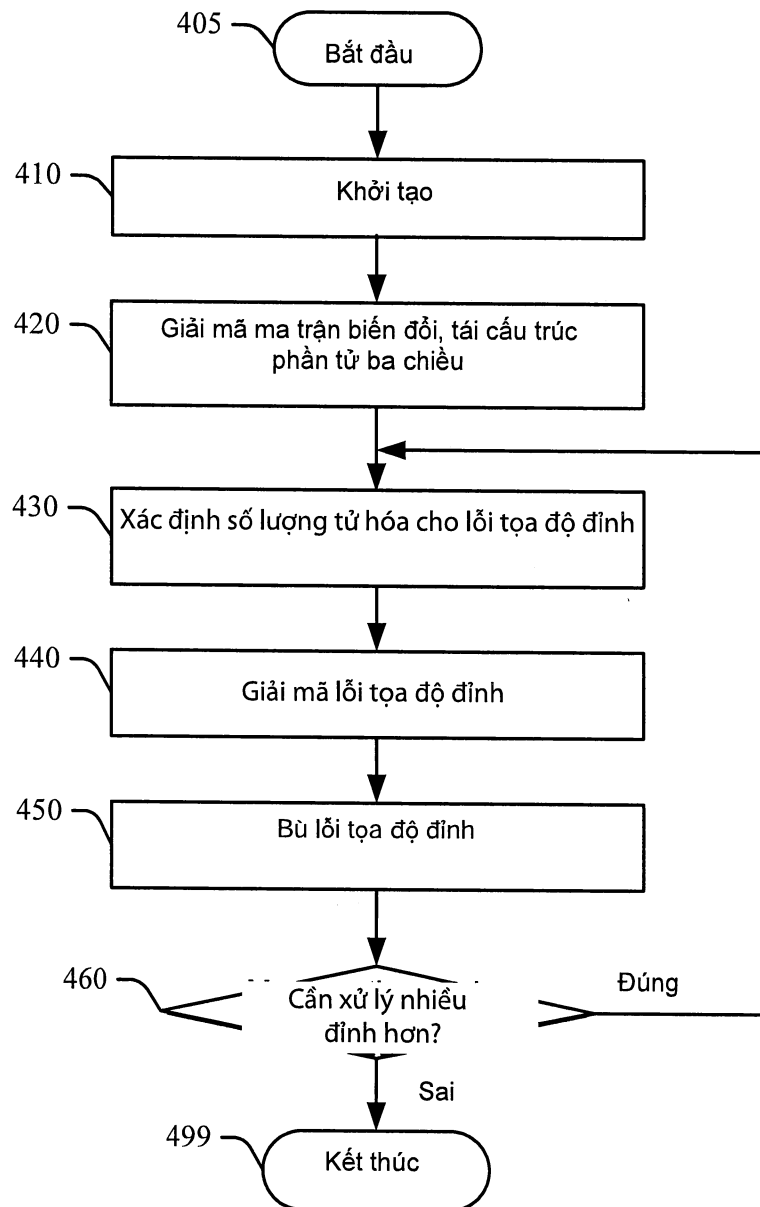


FIG. 4

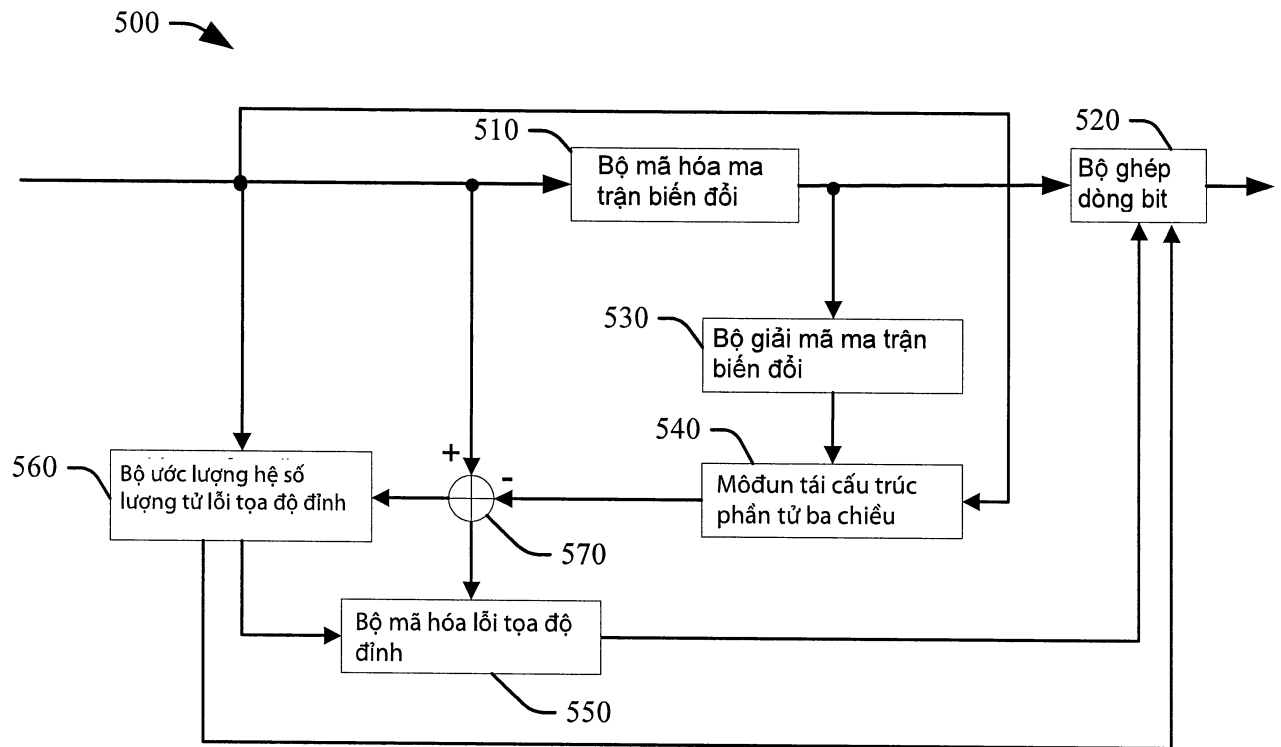


FIG. 5

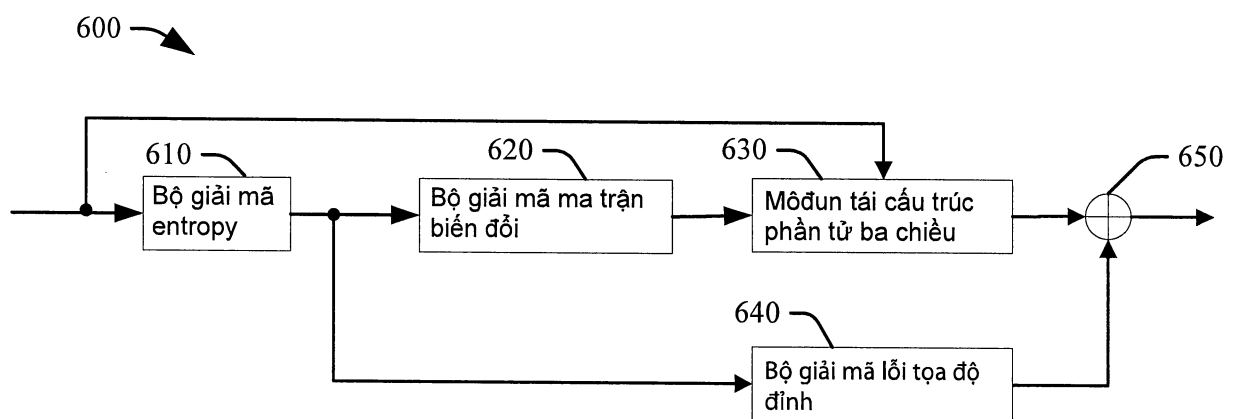


FIG. 6