



(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)



1-0030052

(51)⁷ G06F 17/10; H03M 7/30; G06T 9/00

(13) B

(21) 1-2017-02118

(22) 12/12/2012

(62) 1-2014-03478

(86) PCT/SE2012/051381 12/12/2012

(87) WO2013/147667 03/10/2013

(30) 61/617,151 29/03/2012 US

(45) 25/11/2021 404

(43) 25/07/2017 352A

(73) TELEFONAKTIEBOLAGET LM ERICSSON (PUBL) (SE)

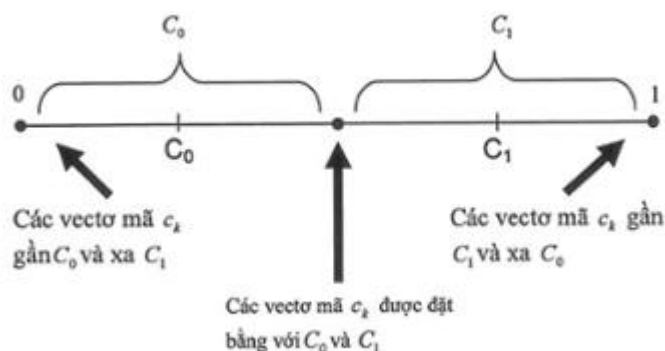
SE-164 83 Stockholm, Sweden

(72) GRANCHAROV, Volodya (BG); JANSSEN TOFTGÅRD, Tomas (SE).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP ĐƯỢC THỰC HIỆN BỞI BỘ MÃ HÓA-GIẢI MÃ CHUYÊN ĐỔI, BỘ MÃ HÓA-GIẢI MÃ CHUYÊN ĐỔI VÀ THIẾT BỊ ĐẦU CUỐI

(57) Sáng chế đề cập đến bộ lượng tử hóa vector và phương pháp dùng cho việc lượng tử hóa vector hiệu quả, ví dụ, trong bộ mã hóa-giải mã âm thanh chuyển đổi. Phương pháp này bao gồm bước so sánh vector đích đầu vào s với nhiều trọng tâm, mỗi trọng tâm biểu diễn lớp tương ứng của các vector mã trong bảng mã. Thêm nữa, điểm khởi đầu cho tìm kiếm liên quan đến vector đích đầu vào trong bảng mã được xác định, dựa trên kết quả so sánh. Các vector mã trong bảng mã được sắp xếp theo số đo độ méo phản ánh khoảng cách giữa mỗi vector mã và các trọng tâm của các lớp. Bộ lượng tử hóa vector và phương pháp cho phép lớp vector mã bao gồm các vector mã ứng viên có khả năng nhất liên quan đến các vector đầu vào có thể được tìm kiếm đầu tiên.



Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến việc lượng tử hóa vector (vector quantization-VQ), và cụ thể là đến độ chính xác và độ phức tạp tính toán của việc lượng tử hóa vector này.

Tình trạng kỹ thuật của sáng chế

Có hai lớp thuật toán lượng tử hóa chính, cụ thể là: các bộ lượng tử hóa vô hướng (scalar quantizer-SQ), sẽ xử lý vector đầu vào theo từng phần tử, và các bộ lượng tử hóa vector (vector quantizer - VQ), sẽ lượng tử hóa vector đầu vào như là một đơn vị (tất cả các chiều vector được lượng tử hóa kết hợp). Tại tốc độ bit (bitrate) đã cho, các bộ lượng tử hóa vector VQ vượt trội hơn các bộ lượng tử hóa vô hướng SQ, nhưng với phí tổn là việc tăng độ phức tạp tính toán và việc lưu trữ bộ nhớ.

Cho vector đích để lượng tử hóa là M chiều: $\mathbf{s} = [s(1) \ s(2) \ \dots \ s(M)]$. Thuật toán VQ thực hiện việc tìm kiếm trong bảng mã (CB - codebook) có kích thước K , $\{\mathbf{c}_k\}_{k=1}^K$ của các vector mã M chiều được lưu trữ từ trước $\mathbf{c}_k = [c_k(1) \ c_k(2) \ \dots \ c_k(M)]$. Việc tìm kiếm này trả về chỉ số vector của bảng mã tạo ra sự bắt cặp tốt nhất k^{opt} dựa trên số đo độ méo $d(\mathbf{s}, \mathbf{c}_k)$. Các phương trình (1-2) dưới đây mô tả phép toán này, giả sử rằng tiêu chí tìm kiếm này dựa trên sai số bình phương:

$$k^{opt} = \arg \min_k d(\mathbf{s}, \mathbf{c}_k) \quad (1)$$

$$d(\mathbf{s}, \mathbf{c}_k) = \sum_{m=1}^M (s(m) - c_k(m))^2 \quad (2)$$

Chỉ số tối ưu k^{opt} được truyền đến bộ giải mã, và vector mã tương ứng được tách khỏi bảng mã (các bảng mã giống nhau có sẵn ở cả bộ mã hóa và bộ giải mã) và được

sử dụng để tái cấu trúc vector đích. Bảng mã thường được luyện tập ngoại tuyến và giữ các thuộc tính thống kê của dữ liệu. Trong nhiều trường hợp, sai số bình phương đơn giản (cụ thể là phương trình (2)) được biến đổi với trọng số, sao cho:

$$d(\mathbf{s}, \mathbf{c}_k) = \sum_{m=1}^M w(m) \cdot (s(m) - c_k(m))^2 \quad (3)$$

trong đó, các trọng số $w(m)$ là phụ thuộc vào ứng dụng. Để làm đơn giản phương trình ở đây, chỉ có sai số bình phương, được xác định trong phương trình (2) sẽ được sử dụng trong phần mô tả tiếp theo. Tuy nhiên, nên lưu ý rằng các nguyên lý được mô tả trong bản mô tả này cũng có giá trị khi nhiều tiêu chí phức tạp, như một tiêu chí trong phương trình (3), được sử dụng.

Do có thể được loại bỏ ra khỏi phần mô tả trên đây, nên độ chính xác hoặc chất lượng của tín hiệu đích được tái cấu trúc, phụ thuộc vào kích thước K của bảng mã; trong đó bảng mã lớn dẫn đến độ chính xác cao hơn, và theo đó là chất lượng tốt hơn, so với bảng mã nhỏ hơn. Đồng thời, dựa vào phương trình (1) có thể kết luận rằng độ phức tạp tính toán chính cũng liên quan đến kích cỡ của bảng mã, giả sử rằng tính chất về chiều vector được cố định theo ứng dụng.

Thông thường, hệ thống truyền dẫn âm thanh được xây dựng trong các ràng buộc của độ phức tạp tính toán giới hạn. Cụ thể là, độ phức tạp của trường hợp xấu nhất không nên vượt quá mức được xác định trước cụ thể L_{MAX} . Ví dụ, độ phức tạp tính toán của bộ mã hóa-giải mã âm thanh thường được đo bằng hàng triệu thao tác được định trọng số trên mỗi giây (Weighted Millions of Operations per Second - WMOPS), nhưng do chúng ta xem xét mô đun lượng tử hóa vector, nên độ phức tạp chỉ liên quan trực tiếp đến kích thước không gian tìm kiếm (kích thước của CB). Bộ lượng tử hóa vector thường là mô đun phức tạp nhất trong bộ mã hóa-giải mã, và hơn nữa, việc tìm kiếm CB (số các so sánh với các vector CB) là những gì làm cho bộ lượng tử hóa vector trở nên quá phức tạp.

Nếu hệ bộ lượng tử hóa vector là để lượng tử hóa một vector đích $\mathbf{S}$ tại một thời điểm, thì không gian tìm kiếm K phải được tối ưu hoá sao cho độ phức tạp không vượt quá L_{MAX} . Một số kỹ thuật tối ưu ngoại tuyến, như bộ lượng tử hóa vector chia

tách và nhiều giai đoạn có thể tạo ra sự giảm độ phức tạp nhất định (và lưu trữ), thu được các thuộc tính của vector $\mathbf{S}$ và các yêu cầu về chất lượng với vector được tái cấu trúc.

Nếu hệ bộ lượng tử hóa vector là để lượng tử hóa nhiều vector đích đến (đầu vào) $\{\mathbf{s}_n\}_{n=1}^N$ tại một thời điểm, với số vector khác nhau N , thì các kỹ thuật tối ưu ngoại tuyến nêu trên không có khả năng duy trì các ràng buộc độ phức tạp và chất lượng. Trong các trường hợp này, sự tối ưu hoá ngoại tuyến phải tìm ra sự cân bằng giữa các yêu cầu mâu thuẫn của A) giới hạn độ phức tạp (= giới hạn việc tìm kiếm) khi số lượng lớn vector đầu vào cần được lượng tử hóa một cách đồng thời, và B) duy trì độ chính xác cao (= tìm kiếm trong bảng mã lớn) khi số lượng nhỏ vector cần được lượng tử hóa, đây vốn không phải là nhiệm vụ đơn giản.

Bản chất kỹ thuật của sáng chế

Công nghệ được mô tả trong bản mô tả này có thể áp dụng, ví dụ, cho các hệ nén/truyền dẫn âm thanh và video, thực hiện việc nén bị mất trên dòng vào, và có thể được mô tả với một số khía cạnh khác nhau. Công nghệ được mô tả trong bản mô tả này bao gồm bảng mã được chia thành các lớp và được sắp xếp, và việc phân loại các vector đích đầu vào $\mathbf{s}$ cần được lượng tử hóa vào một trong số các lớp này. Công nghệ được mô tả trong bản mô tả này cho phép lớp vector mã, trong bảng mã, bao gồm tập hợp các vector mã ứng viên có khả năng nhất liên quan đến các vector đầu vào $\mathbf{s}$ được tìm kiếm đầu tiên của các lớp trong bảng mã. Do đó, vector mã bắt cặp tốt nhất với các vector đầu vào $\mathbf{s}$ có thể được tìm thấy sớm trong việc tìm kiếm, và độ phức tạp tính toán có thể được giảm.

Theo khía cạnh thứ nhất, sáng chế đề xuất phương pháp trong bộ lượng tử hóa vector. Phương pháp này bao gồm bước so sánh vector đích đầu vào $\mathbf{s}$ với nhiều trọng tâm, nghĩa là các vector tham chiếu, trong đó mỗi trọng tâm là lớp vector mã tương ứng trong bảng mã. Phương pháp này còn bao gồm bước xác định điểm khởi đầu, trong bảng mã, với tìm kiếm liên quan đến vector đích đầu vào trong bảng mã, trong đó điểm khởi đầu được xác định dựa trên kết quả so sánh. Các vector mã trong bảng mã được sắp xếp theo số đo độ méo phản ánh khoảng cách giữa mỗi vector mã và các trọng tâm.

Phương pháp này cho phép lớp vector mã, trong bảng mã, bao gồm tập hợp các vector ứng viên có khả năng nhất liên quan đến vector đầu vào s được tìm kiếm đầu tiên trong các lớp trong bảng mã.

Theo khía cạnh thứ hai, sáng chế đề xuất bộ lượng tử hóa vector, bao gồm các đơn vị chức năng thích ứng để thực thi phương pháp theo khía cạnh thứ nhất. Bộ lượng tử hóa vector bao gồm đơn vị so sánh thích ứng để so sánh vector đích đầu vào s với nhiều trọng tâm, mỗi trọng tâm biểu diễn lớp tương ứng của các vector mã trong bảng mã. Bộ lượng tử hóa vector còn bao gồm đơn vị xác định được thích ứng để xác định điểm khởi đầu cho việc tìm kiếm trong bảng mã, dựa trên kết quả so sánh. Các vector mã trong bảng mã được sắp xếp theo số đo độ méo phản ánh khoảng cách giữa mỗi vector mã và các trọng tâm. Bộ lượng tử hóa vector cho phép lớp vector mã bao gồm các vector mã ứng viên có khả năng nhất liên quan đến vector đầu vào s được tìm kiếm đầu tiên trong các lớp trong bảng mã.

Theo khía cạnh thứ ba, sáng chế đề xuất bộ mã hóa-giải mã bao gồm bộ lượng tử hóa vector theo khía cạnh thứ hai.

Theo khía cạnh thứ tư, sáng chế đề xuất thiết bị đầu cuối di động bao gồm bộ lượng tử hóa vector theo khía cạnh thứ hai nêu trên.

Theo khía cạnh thứ năm, sáng chế đề xuất bảng mã cho việc lượng tử hóa vector mà bảng mã được sắp xếp sao cho các vector mã của bảng mã được chia thành nhiều lớp, mỗi lớp được biểu diễn bằng trọng tâm, và trong đó các vector mã còn được sắp xếp theo số đo độ méo phản ánh khoảng cách của chúng đến các trọng tâm của nhiều lớp. Các vector mã có thể được sắp xếp theo, ví dụ, giá trị méo giảm dần hoặc tăng dần.

Theo khía cạnh thứ sáu, sáng chế đề xuất việc sử dụng bảng mã cho việc lượng tử hóa vector, trong đó bảng mã được sắp xếp các vector mã của bảng mã được chia thành nhiều lớp, mỗi lớp được biểu diễn bằng trọng tâm, và trong đó các vector mã còn được sắp xếp theo số đo độ méo phản ánh khoảng cách của chúng đến các trọng tâm của nhiều lớp. Các vector mã có thể được sắp xếp theo, ví dụ, giá trị méo giảm dần hoặc tăng dần.

Theo khía cạnh thứ bảy, sáng chế đề xuất chương trình máy tính, bao gồm mã có thể đọc được bằng máy tính, trong đó khi chạy trong đơn vị xử lý, làm cho bộ lượng tử hóa vectơ thực hiện phương pháp theo khía cạnh thứ nhất.

Theo khía cạnh thứ tám, sáng chế đề xuất sản phẩm chương trình máy tính, bao gồm vật ghi đọc được bằng máy tính và chương trình máy tính theo khía cạnh thứ sáu, được lưu trữ trên vật ghi đọc được bằng máy tính.

Kích thước của vùng tìm kiếm trong bảng mã, trong đó thực hiện việc tìm kiếm, có thể thích ứng dựa trên số vectơ đích đầu vào và ràng buộc phức tạp tối đa. Số vectơ đích đầu vào trên mỗi đơn vị mã hóa có thể biến đổi. Hơn nữa, ràng buộc phức tạp tối đa có thể được thiết lập theo cách động. Hơn nữa, việc tìm kiếm có thể được thực hiện trong bảng mã trong không gian tìm kiếm được xác định, khởi đầu tại điểm khởi đầu được xác định, trong đó việc tìm kiếm đưa ra sự bắt cặp tốt nhất với vectơ đích đầu vào s . “Sự bắt cặp tốt nhất” ở đây nghĩa là sự bắt cặp gần nhất, khoảng cách ngắn nhất, đối với số đo khoảng cách giữa vectơ đích đầu vào và vectơ ứng viên trong bảng mã, nghĩa là sự bắt cặp tốt nhất là vectơ mã mà có khoảng cách ngắn nhất với vectơ đích đầu vào, theo số đo khoảng cách.

Khía cạnh thứ năm có thể là bộ mã hóa-giải mã bao gồm bộ lượng tử hóa vectơ theo khía cạnh thứ hai.

Khía cạnh thứ sáu có thể là thiết bị đầu cuối di động bao gồm bộ lượng tử hóa vectơ theo khía cạnh thứ hai.

Mô tả vắn tắt các hình vẽ

Công nghệ được đề xuất sẽ được mô tả chi tiết hơn bằng các phương án minh họa và có dựa vào các hình vẽ kèm theo, trong đó:

Fig.1a và Fig.1b thể hiện cấu trúc CB được sắp xếp theo giải pháp được mô tả ở đây. Việc tìm kiếm khởi đầu từ điểm 0 hoặc điểm 1 về phía đầu còn lại của CB.

Fig.2 thể hiện cấu trúc CB minh họa khai thác sự đối xứng. Chỉ các vectơ mã của C_0 và C_1 được sắp xếp trong bộ nhớ ROM.

Fig.3 minh họa đơn vị chức năng minh họa để xác định lớp tối ưu với vector đầu vào s bằng cách so sánh vector đầu vào s với một trong số các trọng tâm $\{C_0, C_1, C_{1,flip}, C_{0,flip}\}$, mỗi trọng tâm kết hợp với lớp trong bảng mã.

Fig.4 minh họa đơn vị chức năng để xác định kích thước của vùng tìm kiếm trong bảng mã dựa trên số đỉnh phổ được phát hiện trong khung hiện tại, và khả năng trên tốc độ bit của bộ mã hóa-giải mã.

Fig.5 là bảng minh họa vùng tìm kiếm làm tăng số đỉnh khi số đỉnh trên khung giảm. Trong ví dụ này; với 17 đỉnh (=17 vector đầu vào), việc tìm kiếm được thực hiện chỉ trong 7 bit CB (được xác định là không gian tìm kiếm tối thiểu trong ví dụ này), nhưng với 8 đỉnh hoặc nhỏ hơn, thì việc tìm kiếm được thực hiện trong bảng mã 8 bit (không gian tìm kiếm tối đa), do điều này có thể “đạt được” trong ràng buộc phức tạp tối đa.

Các hình vẽ từ Fig.6a đến Fig.6d thể hiện các ví dụ của các vùng tìm kiếm khác nhau.

Fig.7 minh họa độ phức tạp cho phép L_{MAX} có thể được truyền tín hiệu vào hệ thống từ thực thể bên ngoài. Tham số L_{MAX} có thể dựa trên, ví dụ, tải CPU, hoặc tình trạng pin.

Fig.8 là lưu đồ minh họa các hoạt động trong quy trình tạo bảng mã, CB, được sử dụng trong công nghệ được đề xuất.

Các hình vẽ từ Fig.9a đến Fig.9c là các lưu đồ minh họa các hoạt động trong các quy trình lượng tử hóa vector, VQ, theo các ví dụ của công nghệ được đề xuất ở đây.

Fig.10 là giản đồ khối minh họa bộ lượng tử hóa vector, theo ví dụ của công nghệ được đề xuất ở đây.

Fig.11 là giản đồ khối minh họa bộ mã hóa-giải mã bao gồm bộ lượng tử hóa vector, theo ví dụ của công nghệ được đề xuất ở đây.

Fig.12 là giản đồ khối minh họa bố trí lượng tử hóa vector, theo ví dụ của công nghệ được đề xuất ở đây.

Mô tả chi tiết sáng chế

Nói một cách ngắn gọn, giải pháp được mô tả trong bản mô tả này đề cập đến sự thích ứng theo cách động, không gian tìm kiếm của lượng tử hóa vector, sao cho, với số vector đích đến (đầu vào) (trên mỗi khối hoặc khoảng thời gian), độ chính xác cao, và do đó đạt được việc lượng tử hóa trong ràng buộc phức tạp được đưa ra. Cụ thể là, các yêu cầu của độ phức tạp tính toán (cụ thể là, $L_{\max}$) không bị vi phạm. Điều này đạt được bằng cách thực hiện việc tìm kiếm trong bảng mã được phân loại và sắp xếp đặc biệt. Điểm khởi đầu trong không gian tìm kiếm với mỗi vector đích đến dựa trên quy trình phân loại, và tăng hoặc giảm kích thước không gian tìm kiếm, dựa trên số vector đích đến. Thuật toán bộ lượng tử hóa vector được mô tả ở đây có thể được xem xét là “công cụ” để nén dữ liệu, độc lập với những gì dữ liệu có, nghĩa là dữ liệu có thể là, ví dụ video và/hoặc âm thanh. Trong phần mô tả này, lượng tử hóa vector được mô tả trong ngữ cảnh của bộ mã hóa-giải mã âm thanh, nhưng định nghĩa được mô tả ở đây không giới hạn đến các bộ mã hóa-giải mã âm thanh. Ví dụ, cũng có thể được thực thi trong các bộ mã hóa-giải mã video.

Thuật toán được mô tả ở đây dựa trên bảng mã được thiết kế một cách đặc biệt. Một số biến của bảng mã này sẽ được mô tả chi tiết dưới đây. Thứ nhất, trường hợp cơ bản sẽ được mô tả, và sau đây sơ đồ nâng cao hơn sẽ được thảo luận. Các vector mã của bảng mã có thể được sắp xếp theo giải pháp được mô tả ở đây theo kiểu ngoại tuyến.

Để tạo ra phiên bản cơ bản của bảng mã có lợi, được thiết kế một cách đặc biệt, các vector mã của bảng mã được chia thành hai lớp, trong bản mô tả này được ký hiệu là C_0 và C_1 (ký hiệu này được sử dụng cho cả hai tên của các lớp, cũng như các trọng tâm tương ứng, cụ thể là Fig.1). Để phân dữ liệu thành hai lớp, có thể sử dụng thuật toán còn được gọi là K trung bình (K-means) (thuật toán của Generalized Lloyd). Kỹ thuật đã biết mà lấy toàn bộ dữ liệu làm đầu vào, và số lớp mong muốn, và đưa ra các trọng tâm của số lớp mong muốn. Ví dụ, thuật toán đưa ra 2 vector trọng tâm nếu số lớp mong muốn được chỉ ra 2. Cần lưu ý rằng các trọng tâm này, khi sử dụng thuật toán K trung bình, là các vector có cùng kích thước làm vector từ tập hợp dữ liệu, nhưng chúng không thuộc về tập hợp dữ liệu. Cụ thể là, các vector trọng tâm nằm ngoài bảng

mã và không cần trùng với một số vector mã đang tồn tại. “Trọng tâm” ở đây thường có nghĩa là vector tham chiếu biểu diễn lớp vector.

Tất cả vector mã trong bảng mã tiếp đó được sắp xếp theo số đo độ méo, ví dụ, như được xác định trong phương trình (4)

$$d(\mathbf{c}_k, C_0, C_1) = \sum_{m=1}^M (s_k(m) - C_0(m))^2 - \sum_{m=1}^M (s_k(m) - C_1(m))^2 \quad (4)$$

Số đo độ méo trên đây dẫn đến, hoặc xem xét, các giá trị âm đối với các vector mã gần C_0 , và các giá trị dương lớn đối với các vector mã gần C_1 . Các vector mã có khoảng cách bằng nhau từ các trọng tâm (C_0 và C_1) của hai lớp tạo ra số đo độ méo d là gần đến 0. Trong bảng mã, các vector mã được sắp xếp, ví dụ, bằng cách tăng số đo độ méo, như được minh họa trên Fig.1a và b.

Mỗi vector đích đầu vào được so sánh với hai trọng tâm (trọng tâm tương ứng của hai lớp) và, tùy thuộc vào kết quả, được chuyển đến, nghĩa là được loại bỏ hoặc được xác định thuộc về, lớp C_0 hoặc lớp C_1 . Dựa trên phân loại đó, điểm khởi đầu của tìm kiếm được chọn là điểm phía trên cùng (Fig.1a) hoặc điểm phía tận cùng bên trái (Fig.1b) 0 (khi vector đích đến thuộc về lớp C_0) hoặc điểm thấp nhất (Fig.1a) hoặc điểm phía phải ngoài cùng (Fig.1b) 1 (khi vector đích đến thuộc về lớp C_1). Bây giờ, kích thước của không gian tìm kiếm nên được thực hiện tùy thuộc vào số vector đích đầu vào N trên mỗi khối hoặc khoảng/đoạn thời gian. Nếu chúng ta xác định lại không gian tìm kiếm K không phải là kích thước của toàn bộ bảng mã, nhưng có thể biến đổi, định nghĩa đằng sau sự thích hợp được mô tả ở đây có thể được xác định trong phương trình (5)

$$N \times K \approx \text{const} \quad (5)$$

Nói cách khác, #Các bộ lượng tử hóa (*Quantizers*) x #Các thao tác trên mỗi bộ lượng tử (*Operations_per_Quantizer*) $\approx$ hằng số (*const*), trong đó “bộ lượng tử hóa” có thể được xem là thuật toán ánh xạ vector đầu vào với một trong các vector mã.

Trong bản mô tả này, ví dụ, lượng tử hóa vector được mô tả trong ngữ cảnh của bộ mã hóa-giải mã chuyển đổi mã hóa các đỉnh phổ, hoặc nghiêm ngặt, các vùng quanh các đỉnh phổ. Trong ngữ cảnh của bộ mã hóa-giải mã này, vector đích đầu vào

có thể phản ánh đỉnh (vùng) phổ của đoạn tín hiệu âm thanh được xử lý. Số đỉnh phổ trong phổ tín hiệu của đoạn thời gian, ví dụ, 30ms, tín hiệu âm thanh tùy thuộc vào các thuộc tính phổ của tín hiệu âm thanh trong đoạn thời gian đó. Do các thuộc tính phổ của tín hiệu âm thanh có thể biến đổi theo thời gian và khác nhau, ví dụ, với các loại âm thanh khác nhau, số đỉnh phổ có thể khác nhau giữa các đoạn thời gian khác nhau và giữa các tín hiệu âm thanh khác nhau. Do đó, khi sử dụng bộ mã hóa chuyển đổi mà mã hóa các vùng đỉnh phổ, số vector đầu vào, trên mỗi khối hoặc đoạn thời gian, vào lượng tử hóa vector sẽ khác nhau. Trong các ví dụ ở đây, số vector đầu vào tối ưu, tương ứng với số đỉnh phổ trong đoạn thời gian của tín hiệu âm thanh, là 17. Tuy nhiên, con số này chỉ là ví dụ, và không nên được xem là giới hạn giải pháp nói chung.

Một cách hiệu quả là, sơ đồ được mô tả trên đây giữ số thao tác đối với lượng tử hóa vector trong khoảng hẹp (hoặc hầu như không đổi); cụ thể là, khi số lượng tử hóa vector tăng lên, nghĩa là số vector đích đầu vào tăng lên, số thao tác trên mỗi lượng tử hóa vector giảm đi (kích thước của không gian tìm kiếm giảm đi/chỉ có một phần bảng mã được tìm kiếm), sao cho các yêu cầu phức tạp (nghĩa là các điều kiện) không bị vi phạm. Với việc giảm N , không gian tìm kiếm K có thể tăng lên, với hầu hết lên đến kích thước của toàn bộ bảng mã, dẫn đến độ chính xác cao và theo đó là chất lượng của vector được tái cấu trúc cao. Độ chính xác của bộ lượng tử hóa vector có thể được đo là sai số bình phương giữa tín hiệu ban đầu và dữ liệu được tái cấu trúc tương ứng.

Theo cách này, bảng mã của lượng tử hóa vector không cần được thiết kế với trường hợp xấu nhất (nghĩa là số vector đích đầu vào tối đa). Thay vào đó, nó có thể được thiết kế, ví dụ, với trường hợp tốt nhất, theo đó chứa nhiều vector mã hơn có thể được tìm kiếm hơn với số vector đích đầu vào tối đa trong ràng buộc phức tạp tối đa L_{MAX} . Yêu cầu phức tạp tối đa sẽ được hoàn thành bằng cách mở rộng tìm kiếm, nghĩa là kích thước của không gian tìm kiếm, trong bảng mã, tùy thuộc vào số vector đích đầu vào. Tuy nhiên, nếu điều này được thực hiện “mù”, ví dụ, không có bảng mã được đề xuất ở đây, thì chất lượng lượng tử hóa sẽ bị ảnh hưởng lớn, do sẽ không có cách nào biết vector “bắt cặp tốt nhất” được đặt ở đâu trong bảng mã, hoặc liệu vector bắt cặp tốt nhất này được đặt trong một phần của bảng mã sẽ được tìm kiếm hay không, khi không gian tìm kiếm giảm đi. Vấn đề này được giải quyết bằng thiết kế đặc biệt của

bảng mã, được mô tả ở đây. Lưu ý rằng thiết kế bảng mã được mô tả ở đây cũng có lợi với các ứng dụng mà số vector đầu vào, trên mỗi đơn vị mã hóa, là không đổi.

Phương án minh họa 1: bộ lượng tử hóa vector bị ràng buộc, trên các vùng đỉnh phổ

Tập hợp các vector đích đến $\mathbf{S}$ biểu diễn các vùng đỉnh phổ trong mã hóa âm thanh miền biến đổi, ví dụ, các hệ số biến đổi trong vùng lân cận của các đỉnh MDCT. Do đó, trong trường hợp này, số vector đích đến biến đổi theo thời gian, do số đỉnh phổ thay đổi từ một khối thời gian tới khối thời gian khác.

Trong kiểu ứng dụng này (mã hóa vùng đỉnh), các vector đích đến $\mathbf{S}$ thể hiện các mức độ đối xứng nào đó, có thể được sử dụng để tối ưu hơn bảng mã. Ví dụ, các hệ số chuyển đổi trên cả hai phía của đỉnh phổ có các thống kê tương tự. Nếu giả sử rằng các vector đích đến $\mathbf{S}$ được đặt giữa vị trí đỉnh, thì đối xứng được mô tả trên đây cho phép bổ sung thêm cấu trúc khác trong bảng mã được sắp xếp trên Fig.1a và 1b. Cấu trúc của bảng mã mới, được cải thiện hơn này được minh họa trên Fig.2. Fig.2 minh họa bảng mã mà các vector mã của phần bên trái được lưu trữ trong bộ nhớ, như bộ nhớ chỉ đọc. Tuy nhiên, không có vector mã được lưu trữ trước ở mặt phải của bảng mã, nghĩa là với các lớp $C_{1,flip}$ và $C_{0,flip}$. Các vector mã của các lớp này là các phiên bản được quay của các vector mã của mặt phải của bảng mã. Do đó, khi thực hiện việc tìm kiếm trong các lớp $C_{0,flip}$ và $C_{1,flip}$, thì việc tìm kiếm được thực hiện trên các vector mã của phía trái, được lưu trữ trong bộ nhớ, nhưng với các phần tử vector mã quay quanh tâm, sao cho các vector mã $\mathbf{c}_{k,flip}$ được đưa ra bởi phương trình (6)

$$\mathbf{c}_{k,flip} = [c_k(M) \quad c_k(M-1) \quad \dots \quad c_k(1)], \quad (6)$$

trong đó, $c_k(m)$ là các phần tử vector của lớp tương ứng C_j trong bảng mã được lưu trữ (nghĩa là C_0 hoặc C_1). Cụ thể là, nếu các phần tử của vector mã nào đó trong C_0 là $\{C_{01} \ C_{02} \ C_{03} \ C_{04}\}$, thì các phần tử của vector mã tương ứng trong $C_{0,flip}$ là $\{C_{04} \ C_{03} \ C_{02} \ C_{01}\}$

Khi sử dụng bảng mã như bảng mã được minh họa trên Fig.2, thì vector đích đầu vào được so sánh với 4 trọng tâm và được truyền đến lớp để xác định điểm khởi đầu cho tìm kiếm, nghĩa là lớp tối ưu với vector đích đầu vào được xác định bằng cách so sánh vector đầu vào $\mathbf{S}$ với một trong các trọng tâm $\{C_0 \ C_1 \ C_{1,flip} \ C_{0,flip}\}$. Điều này

được minh họa trên Fig.3, trong đó vector đích đến $\mathbf{S}$ được đưa vào đơn vị chuyển đổi lớp 302, mà đưa bộ chỉ dẫn lớp, C_j , làm đầu ra. Các trọng tâm $C_{1,flip}$ và $C_{0,flip}$ được lưu trữ trong bảng, nhưng “được tạo ra” bằng cách quy các phần tử của trọng tâm C_0 và C_1 . Các phần tử không cần được quay theo nghĩa đen, thay vào đó thao tác tìm kiếm biến đổi có thể đọc các phần tử của C_0 và C_1 theo thứ tự ngược lại khi đọc $C_{0,flip}$ và $C_{1,flip}$, nghĩa là cả hai trọng tâm và các vector bảng mã. Theo cách này, bảng mã có thể được mở rộng để bao gồm hai lần số vector mã, như được so sánh với những gì được lưu trữ tự nhiên trong bảng mã, có nghĩa là tiết kiệm tài nguyên bộ nhớ. Đây là khả năng do sự đối xứng của các vùng đỉnh được khai thác, như được mô tả trên đây. Cụ thể hơn là, giải pháp này dựa trên sự quan sát sự đối xứng có thể được khai thác nhờ vector mã có hiệu lực được quay cũng là vector mã có hiệu lực.

Vùng tìm kiếm được thích ứng với số đỉnh phổ mà tương ứng với số vector đích đầu vào. Điều này được minh họa trên Fig.4 thể hiện đơn vị chức năng 402 đưa bộ chỉ dẫn của số đỉnh/vector làm đầu vào, và tạo ra bộ chỉ dẫn của vùng tìm kiếm làm đầu ra. Fig.4 còn minh họa rằng, ví dụ, tốc độ bit của bộ mã hóa-giải mã áp dụng lượng tử hóa vector có thể xem xét khi xác định vùng tìm kiếm (không gian tìm kiếm). Ví dụ, có thể có các yêu cầu chất lượng khác nhau đối với các tốc độ bit khác nhau, ví dụ, tốc độ bit càng cao, chất lượng mong đợi càng cao. Hơn nữa, độ phức tạp tối đa cho phép có thể thay đổi với tốc độ bit, ví dụ, do ở các tốc độ bit khác nhau thì các mô đun khác nhau trong bộ mã hóa-giải mã hoạt động mà không có độ phức tạp như nhau, nghĩa là giữ độ phức tạp cho phép với lượng tử hóa vector, từ ràng buộc phức tạp tối đa trên toàn bộ quy trình mã hóa, mà không thể là độ phức tạp tối đa này. Cụ thể là, thông tin tốc độ bit phản ánh sự thay đổi về yêu cầu chất lượng và độ phức tạp mà có thể xem xét lượng tử hóa vector.

Bảng trên Fig.5 minh họa cách mà vùng tìm kiếm thích ứng với số đỉnh. Trên Fig.5, vùng tìm kiếm được chỉ ra là số hệ số (vector mã) trong tìm kiếm trên mỗi vector đầu vào. Số hệ số này trong bảng trên Fig.5 thu được với giả thuyết rằng bảng mã từ Fig.2 bao gồm 4 đoạn 7 bit (4 đoạn với mỗi đoạn có 128 vector mã) mà hai đoạn trong số đó là “bình thường” hoặc “hữu hình” và hai đoạn là “được quay” hoặc “ảo”).

Logic phía sau bảng trên Fig.5 là logic khi có ít nhất một đỉnh để mã hóa, ví dụ, 16 thay vì 17, 128 so sánh “được tiết kiệm” có thể được phân bổ trong số các đỉnh còn lại. Tại một số điểm, chiều dài tìm kiếm là bão hòa, do nó đạt đến kích thước tự nhiên của bảng mã. Trong ví dụ được minh họa trên Fig.5, điểm này đạt tới khi số đỉnh là 8 hoặc nhỏ hơn. Cụ thể là, trong ví dụ được minh họa trên Fig.5, khi số đỉnh là 8 hoặc nhỏ hơn, thì việc tìm kiếm đầy đủ có thể được thực hiện với tất cả vectơ đích đầu vào (nghĩa là các đỉnh) mà không đạt đến độ phức tạp cho phép tối đa.

Các ví dụ về quy trình tìm kiếm được minh họa trên các hình vẽ từ Fig.6a đến Fig.6d. Trong thực tế, đó là cùng kiểu tìm kiếm như tìm kiếm được mô tả trước đây liên quan đến các hình vẽ từ Fig.1a đến Fig.1b, nhưng với phân loại bổ sung thành các đoạn/lớp bảng mã “bình thường” và “được quay”.

Trong ví dụ được thể hiện trên Fig.6a, vectơ đầu vào $\mathbf{S}$ thuộc về lớp C_1 (vị trí được chỉ ra với mũi tên đi xuống). Tiếp đó, không gian tìm kiếm được giới hạn kích thước chỉ giữa lớp C_1 , và không gian liên kết giữa các lớp C_0 và C_1 . Trên Fig.6a, điều này minh họa bằng ba mũi tên đứt chỉ ra các không gian tìm kiếm có kích thước khác nhau. Fig.6b minh họa trường hợp mà vectơ đầu vào thuộc về lớp C_0 (vị trí được chỉ ra bằng mũi tên đi xuống), trong đó trường hợp tìm kiếm có điểm khởi đầu khác nhau.

Tương tự, như được minh họa trên các hình vẽ từ Fig.6c đến Fig.6d, việc tìm kiếm có thể được thực hiện trong các lớp $C_{1,flip}$ và/hoặc $C_{0,flip}$ nếu vectơ đầu vào thuộc về một trong các lớp này. Không có tìm kiếm nào được thực hiện trong không gian tìm kiếm của C_1 và $C_{1,flip}$. Lý do là không gian kết hợp của lớp thông thường hoặc quay không tương ứng với tập hợp dữ liệu thật sự (thống kê của nó không tương ứng với thống kê dữ liệu thật sự). Do đó, rất không thể sự bắt cặp tốt có thể tìm thấy với vectơ đầu vào trong không gian này.

Phương án minh họa 2: Hệ giao tiếp với sự điều khiển bên ngoài của độ phức tạp cho phép tối ưu

Định nghĩa của bộ lượng tử hóa vectơ có độ phức tạp được điều chỉnh theo cách động tới số vectơ đích đến N , có thể mở rộng đến trường hợp khi giới hạn phức tạp không được xác định trước, nhưng có thể khác nhau, ví dụ, dựa trên một số tiêu chí, và

được truyền tín hiệu đến bộ lượng tử hóa vector và/hoặc thực thể mà bộ lượng tử hóa vector được áp dụng. Việc này được minh họa trong lưu đồ trên Fig.7, thể hiện đơn vị chức năng 702 lấy bộ chỉ dẫn hoặc số đỉnh/vector làm đầu vào, và tiếp tục lấy ràng buộc phức tạp L_{MAX} làm đầu vào. Đơn vị chức năng 702 thể hiện bộ chỉ dẫn của không gian tìm kiếm/vùng của bảng mã, cụ thể là các mũi tên đứt trong các hình vẽ từ Fig.6a đến Fig.6d.

Thuật toán bộ lượng tử hóa lượng tử hóa vector có độ phức tạp có thể điều chỉnh thu được sự cân bằng tối ưu giữa độ chính xác của việc lượng tử hóa (nghĩa là chất lượng) và duy trì độ phức tạp tính toán dưới ngưỡng được xác định trước.

Quy trình minh họa để thu được cấu trúc bảng mã

Quy trình minh họa để thiết kế hoặc tổ chức bảng mã để sử dụng trong lượng tử hóa vector sẽ được mô tả sau đây, có dựa vào Fig.8. Quy trình này để tạo ra bảng mã để sử dụng trong lượng tử hóa tạo ra lượng tử hóa vector trong bộ mã hóa âm thanh chuyển đổi, như, ví dụ, bộ mã hóa MDCT.

Quy trình được mô tả sau đây đề cập các phần của quy trình tạo bảng mã mà lệch khỏi, và/hoặc bổ sung vào việc tạo hoặc tổ chức bảng mã lượng tử hóa vector.

Bảng mã được chia thành các lớp, trong hoạt động 802, ví dụ, bằng sử dụng giải thuật, còn được gọi là K trung bình, như được mô tả trước đây. Các vector mã của bảng mã tiếp đó được sắp xếp trong bảng mã dựa trên số đo độ méo, ví dụ, như được mô tả trong phương trình (4). Số đo độ méo với mỗi vector mã tùy thuộc vào mối liên hệ giữa vector mã và các trọng tâm biểu diễn mỗi lớp của bảng mã, như được mô tả trước đây.

Tổ chức này của bảng mã cho phép thực hiện sự thích ứng của không gian tìm kiếm, và do đó là cho phép có độ phức tạp tìm kiếm trong bộ lượng tử hóa vector, tại chất lượng bộ lượng tử hóa vector được lưu giữ cao (ví dụ, chất lượng của các vector đích đến được tái cấu trúc).

Quy trình bộ lượng tử hóa vector làm ví dụ

Quy trình minh họa trong bộ lượng tử hóa vector (VQ) sẽ được mô tả sau đây, có dựa vào Fig.9a. Quy trình này thích hợp để sử dụng trong bộ mã hóa âm thanh chuyển

đổi, như, ví dụ, bộ mã hóa MDCT mã hóa, ví dụ, các vùng đỉnh phổ. Tín hiệu âm thanh có thể bao gồm, ví dụ, bài diễn thuyết và/hoặc âm nhạc.

Như được mô tả trước đây, số N vector đích đầu vào được nhận bởi bộ lượng tử hóa vector. Sau đây, các hoạt động liên quan đến một trong các vector đích đầu vào sẽ được mô tả, với các lý do đơn giản hóa.

Vector đích đầu vào s được so sánh với số vector mã mà mỗi vector biểu diễn lớp bảng mã (cụ thể là các lớp C_0 và C_1 , v.v được mô tả trước đây), tốt hơn là trọng tâm của mỗi lớp. Sự so sánh này được minh họa là hoạt động 902 trên các hình vẽ từ Fig.9a đến Fig.9c. Theo cách khác, hoạt động 902 có thể được xem xét là kết hợp với hoạt động 904, được minh họa trên Fig.9c. Tùy thuộc vào kết quả của các so sánh, vector đích đầu vào s được chuyển một trong các lớp, hoặc các bộ phận, của bảng mã, trong hoạt động 904. Vector đích đầu vào s được chuyển, hoặc loại bỏ thuộc về, lớp mà nó có khoảng cách ngắn nhất, nghĩa là nó tương tự nhất, theo một số số đo khoảng cách (số đo sai số). Điểm khởi đầu của tìm kiếm trong bảng mã được xác định trong hoạt động 906, dựa trên sự chuyển đổi lớp hoặc số đo khoảng cách.

Việc tìm kiếm có thể được thực hiện trong bảng mã trong hoạt động 910. Tìm kiếm khởi đầu trong điểm khởi đầu được chọn, và thực hiện trong không gian tìm kiếm, mà có thể có kích thước được xác định, bao gồm một hoặc nhiều lớp, hoặc các bộ phận của chúng. Do bảng mã được thiết kế và tổ chức có lợi, xác suất của sự bắt cặp tốt nhất, của tất cả các vector mã ứng viên nằm trong toàn bộ bảng mã, với vector đích đầu vào s sẽ được tìm kiếm trong không gian tìm kiếm là rất cao, thậm chí khi không gian tìm kiếm bị giới hạn, ví dụ, một nửa bảng mã. Trong trường hợp mà không gian tìm kiếm bao gồm toàn bộ bảng mã, thì vector mã bắt cặp tốt nhất sẽ được tìm kiếm đầu tiên trong việc tìm kiếm khi bắt việc tìm kiếm tại điểm khởi đầu được xác định.

Khi tìm thấy sự bắt cặp tốt nhất nằm trong không gian tìm kiếm được xác định, thì chỉ số vector mã bắt cặp tốt nhất được tạo ra, do lượng tử hóa vector, trong hoạt động 912, ví dụ, để sử dụng trong bộ giải mã âm thanh.

Hơn nữa, kích thước của không gian tìm kiếm có thể được xác định trong hoạt động 908 được minh họa trên Fig.9c. Không gian tìm kiếm có thể được mô tả là số

vector mã trong bảng mã mà nên được đánh giá trong tìm kiếm đối với sự bắt cặp tốt nhất với vector đích đầu vào s . Kích thước của không gian tìm kiếm được xác định dựa trên số vector đích đầu vào và điều kiện về độ phức tạp tính toán L_{MAX} . Do đó, kích thước của không gian tìm kiếm có thể được xác định, ví dụ, một lần với mọi khối mã hóa, hoặc một số khoảng thời gian khác, tùy thuộc vào, ví dụ, đặc điểm của tín hiệu được lượng tử hóa và/hoặc thuộc tính của bộ mã hóa-giải mã. Nếu số vector đích đầu vào và điều kiện L_{MAX} là không đổi và bán không đổi theo thời gian, kích thước của không gian tìm kiếm cũng có thể giữ không đổi trong thời gian tương ứng.

Tiếp đó, minh họa bố trí lượng tử hóa vector

Sau đây, bố trí lượng tử hóa vector thích hợp để sử dụng trong bộ mã hóa chuyển đổi/bộ mã hóa-giải mã sẽ được mô tả có dựa vào Fig.10. Bộ mã hóa-giải mã chuyển đổi có thể là, ví dụ, bộ mã hóa-giải mã MDCT. Lượng tử hóa vector được thích ứng để thực hiện các hoạt động của quy trình được mô tả trên đây.

Bộ lượng tử hóa vector 1001 được minh họa để giao tiếp với các thực thể khác (ví dụ, bộ mã hóa-giải mã âm thanh) thông qua đơn vị truyền thông 1002. Lượng tử hóa vector có thể còn bao gồm các đơn vị chức năng khác 1016, như, ví dụ, các đơn vị chức năng cung cấp các chức năng thông thường, và có thể còn bao gồm một hoặc nhiều đơn vị lưu trữ 1014.

Bộ lượng tử hóa vector 1001 có thể được thực hiện, ví dụ, bằng một hoặc nhiều: bộ xử lý hoặc vi xử lý và theo đó phần mềm đầy đủ với lưu trữ thích hợp, thiết bị logic có thể lập trình được (PLD - Programmable Logic Device) hoặc (các) thành phần điện tử và/hoặc mạch khác.

Đơn vị truyền thông 1002 được coi để bao gồm các đơn vị chức năng, để thu được các tham số đầy đủ, như các vector đích đầu vào và L_{MAX} , được tạo ra, ví dụ, từ thực thể mã hóa.

Bộ lượng tử hóa vector có thể bao gồm đơn vị so sánh 1004, được làm thích ứng để so sánh vector đích đầu vào s với các vector biểu diễn mỗi lớp của bảng mã, ví dụ, vector trọng tâm của mỗi lớp. Hơn nữa, lượng tử hóa vector có thể bao gồm đơn vị gán 1006, được thích ứng để chuyển lớp vào vector đích đầu vào s (hoặc chuyển vector s

vào lớp), nghĩa là loại bỏ lớp nào mà vector thuộc về đó, dựa trên so sánh. Hơn nữa, lượng tử hóa vector có thể bao gồm đơn vị xác định 1008, thích hợp để xác định điểm khởi đầu đầy đủ với tìm kiếm trong bảng mã, dựa trên lớp được gán cho vector s . Đơn vị xác định còn có thể được thích ứng để xác định kích thước của không gian tìm kiếm trong bảng mã, dựa trên, ví dụ, số vector đích đầu vào nhận được và ràng buộc phức tạp tính toán.

Thêm nữa, bộ lượng tử hóa vector có thể bao gồm đơn vị tìm kiếm 1010, được thích ứng để thực hiện tìm kiếm trong bảng mã, khởi đầu tại điểm khởi đầu được xác định và tìm kiếm không gian tìm kiếm được xác định. Việc tìm kiếm này nên tạo ra một hoặc nhiều chỉ số bảng mã chỉ ra vector mã mà bắt cặp tốt nhất với vector đích đầu vào s . Lượng tử hóa vector còn bao gồm đơn vị cung cấp 1012, được thích ứng để cung cấp chỉ số hoặc các chỉ số đến thực thể khác, ví dụ, đến (hoặc để sử dụng bởi) bộ mã hóa-giải mã chuyển đổi.

Minh họa bố trí

Fig.12 thể hiện sơ đồ phương án sắp xếp 1200 thích hợp để sử dụng trong, ví dụ, chuyển đổi bộ giải mã âm thanh, mà cũng có thể là cách bộc lộ thay thế phương án của bộ lượng tử hóa vector được minh họa trên Fig.5. Chứa trong thiết bị 1200 trong bản mô tả này là đơn vị xử lý 1206, ví dụ, với DSP (Digital Signal Processor - Bộ xử lý tín hiệu số). Đơn vị xử lý 1206 có thể là một đơn vị hoặc nhiều đơn vị để thực hiện các bước quy trình khác nhau được mô tả ở đây. Thiết bị 1200 cũng có thể bao gồm đơn vị đầu vào 1202 để nhận các tín hiệu, như các vector đích đầu vào và các bộ chỉ dẫn của, ví dụ, tốc độ bit và/hoặc ràng buộc phức tạp; tiếp theo đơn vị đầu ra 1204 cho tín hiệu đầu ra (các tín hiệu đầu ra), như các chỉ số bảng mã với các vector mã bắt cặp tốt nhất. Đơn vị đầu vào 1202 và đơn vị đầu ra 1204 có thể được sắp xếp như là một đơn vị trong phần cứng của bố trí này.

Hơn nữa, thiết bị 1200 bao gồm ít nhất một sản phẩm chương trình máy tính 1208 ở dạng bộ nhớ cố định, ví dụ, EEPROM, bộ nhớ tác động nhanh và ổ cứng. Sản phẩm chương trình máy tính 1208 bao gồm chương trình máy tính 1210, bao gồm các phương tiện mã, mà khi chạy trong đơn vị xử lý 1206 trong thiết bị 1200 sẽ làm cho bố

trí này thực hiện các hoạt động của quy trình được mô tả trước đây liên quan đến các hình vẽ từ Fig.9a đến Fig.9c.

Do đó, trong các phương án minh họa được mô tả, các phương tiện mã trong chương trình máy tính 1210 của thiết bị 1200 có thể bao gồm mô đun so sánh 1210a để so sánh vector đích đầu vào với các trọng tâm lớp của bảng mã. Chương trình máy tính có thể bao gồm mô đun chuyển 1210b để chuyển lớp vào vector đích đầu vào. Chương trình máy tính 1210 còn có thể bao gồm đơn vị xác định 1210c để xác định điểm khởi đầu với tìm kiếm trong bảng mã; và còn để xác định không gian tìm kiếm hoặc vùng dựa trên các tham số đầu vào. Chương trình máy tính 1210 còn có thể bao gồm đơn vị tìm kiếm 1210d để tìm kiếm bảng mã theo trên đây. Hơn nữa, chương trình máy tính 1210 có thể bao gồm mô đun cung cấp 1210e, để cung cấp các chỉ số lấy ra từ việc tìm kiếm, tới các thực thể khác.

Chương trình máy tính 1210 ở dạng mã chương trình máy tính được cấu trúc trong mô đun chương trình máy tính. Mô đun 1210a-e có thể thực hiện cần thiết các hoạt động của luồng được minh họa trong hình bất kỳ trong số các hình vẽ từ Fig.9a đến Fig.9c sao chép ít nhất một phần lượng tử hóa vector 1001 được minh họa trên Fig.10. Nói cách khác, khi các mô đun khác nhau 1210a-d chạy trên đơn vị xử lý 1206, thì chúng tương ứng với ít nhất các đơn vị 1004-1012 của Fig.10.

Mặc dù các phương tiện mã trong phương án được bộc lộ trên đây liên quan đến Fig.12 được thực hiện là các mô đun chương trình máy tính mà khi chạy trên đơn vị xử lý sẽ làm cho bố trí và/hoặc bộ mã hóa âm thanh chuyển đổi để thực hiện các bước nêu trên kết hợp với các hình vẽ nêu trên, nhưng ít nhất một trong các phương tiện mã có thể, theo các phương án thay thế, được thực hiện ít nhất một phần như là các mạch của phần cứng.

Mặc dù công nghệ được đề xuất được mô tả có dựa vào các phương án cụ thể, nhưng phần mô tả nói chung chỉ nhằm minh họa định nghĩa và không nên làm giới hạn phạm vi của công nghệ được mô tả ở đây. Các dấu hiệu khác nhau của các phương án minh họa trên đây có thể được kết hợp theo các cách khác nhau theo nhu cầu, yêu cầu hoặc ưu tiên.

Giải pháp được mô tả trên đây có thể được sử dụng ở nơi bất kỳ nơi nào mà các bộ lượng tử hóa được áp dụng, ví dụ, trong các bộ mã hóa-giải mã trong các thiết bị như thiết bị đầu cuối di động, máy tính bảng, máy tính, điện thoại thông minh, v.v.

Sẽ được hiểu rằng sự lựa chọn tương tác các đơn vị hoặc mô đun, cũng như tên các đơn vị chỉ với mục đích minh họa, và các nút thích hợp để thực thi các phương pháp bất kỳ nêu trên có thể được cấu hình theo nhiều cách thay thế để có thể thực thi các hoạt động xử lý đề xuất.

Các chức năng của các phần tử khác nhau bao gồm các khối chức năng, bao gồm nhưng không chỉ giới hạn ở, các khối được đặt tên hoặc được mô tả là “đơn vị chức năng”, “bộ xử lý” hoặc “bộ điều khiển”, có thể được đề xuất thông qua việc sử dụng phần cứng như phần cứng mạch và/hoặc phần cứng có khả năng thực thi phần mềm ở dạng các hướng dẫn được mã hóa được lưu trữ trên vật ghi đọc được bằng máy tính. Do đó, các chức năng này và các khối chức năng được minh họa được hiểu là được thực hiện bằng phần cứng và/hoặc được thực thi bằng máy tính và do đó được thực hiện bằng máy.

Theo nghĩa việc thực thi phần cứng, các khối chức năng có thể bao gồm hoặc chứa, mà không giới hạn ở, phần cứng của bộ xử lý tín hiệu số (digital signal processor - DSP), bộ xử lý tập hợp lệnh rút gọn, mạch phần cứng (ví dụ, số hoặc tương tự) bao gồm nhưng chỉ không giới hạn ở mạch (các mạch) tích hợp ứng dụng cụ thể (application specific integrated circuit - ASIC), và (nếu thích hợp) là các máy móc trạng thái có khả năng thực hiện các chức năng này.

Các chữ viết tắt

SQ	Lượng tử hóa vô hướng
VQ	Lượng tử hóa vector
CB	Bảng mã
WMOPS	Hàng triệu thao tác được định trọng số trên mỗi giây
MDCT	Chuyển đổi cosin rời rạc được biến đổi

YÊU CẦU BẢO HỘ

1. Phương pháp được thực hiện bởi bộ mã hóa-giải mã chuyển đổi chứa bộ lượng tử hóa vector, phương pháp bao gồm các bước:

thu số các vector đích đầu vào, trong đó số các vector đích đầu vào cho mỗi đoạn thời gian là thay đổi và các vector đích đầu vào nêu trên thể hiện các đoạn của tín hiệu audio hoặc tín hiệu video;

với từng vector trong số các vector đích đầu vào:

so sánh vector đích đầu vào với bốn trọng tâm C_0 , C_1 , $C_{0,flip}$ và $C_{1,flip}$, trong đó trọng tâm $C_{0,flip}$ là phiên bản được quay của trọng tâm C_0 và trọng tâm $C_{1,flip}$ là phiên bản được quay của trọng tâm C_1 , từng trọng tâm biểu diễn lớp tương ứng của các vector mã;

xác định điểm bắt đầu cho việc tìm kiếm có liên quan tới vector đích đầu vào trong bảng mã, dựa trên kết quả của so sánh vector đích đầu vào với các trọng tâm, trong đó không gian tìm kiếm cho việc tìm kiếm được điều chỉnh động tới số các vector đích đầu vào, và các vector mã trong bảng mã được xác định là khác với nhau theo số đo độ méo phản ánh khoảng cách giữa từng vector mã giữa các vector mã trong bảng mã và các trọng tâm C_0 và C_1 nêu trên, và các vector mã được sắp xếp theo số đo độ méo phản ánh khoảng cách giữa từng vector mã trong các vector mã trong bảng mã và các trọng tâm C_0 và C_1 nêu trên; và

thực hiện việc tìm kiếm trong bảng mã, bắt đầu tại điểm bắt đầu đã được xác định, và nhận diện vector mã để biểu diễn vector đích đầu vào; và

truyền, qua mạch liên lạc, chỉ thị của từng vector mã trong các vector mã đã được nhận diện.

2. Phương pháp theo điểm 1, trong đó không gian tìm kiếm được điều chỉnh động bằng cách tăng hoặc giảm kích cỡ của không gian tìm kiếm dựa trên số vector đích.

3. Phương pháp theo điểm 1, trong đó các vector mã trong bảng mã được sắp xếp sao cho các từ mã gần nhất với trọng tâm C_0 nêu trên và xa nhất từ trọng tâm C_1 nêu trên là trong một phía của bảng mã, trong khi các từ mã gần nhất với trọng tâm C_1 nêu trên và xa nhất từ trọng tâm C_0 nêu trên được kết cụm trong phía khác của bảng mã.

4. Phương pháp theo điểm 1, trong đó các vector mã cho lớp $C_{0,flip}$ là các phiên bản được quay của các vector mã cho lớp C_0 và các vector mã cho lớp $C_{1,flip}$ là các phiên bản được quay của các vector mã cho lớp C_1 .

5. Phương pháp theo điểm 4, trong đó các thành phần của các vector mã $c_{k,flip}$ trong lớp $C_{j,flip}$ được cho bởi phương trình:

$$c_{k,flip} = [c_k(M)c_k(M-1) \dots c_k(1)],$$

trong đó $c_k(m)$ là các thành phần vector của lớp C_j tương ứng và M là chiều dài của vector mã và k là chỉ số của vector mã.

6. Phương pháp theo điểm 1 trong đó bảng mã được tạo ra dựa trên các trọng tâm C_0 và C_1 nêu trên, trong đó trọng tâm $C_{0,flip}$ nêu trên và trọng tâm $C_{1,flip}$ nêu trên được tạo ra bằng cách đọc các trọng tâm C_0 và C_1 nêu trên theo thứ tự đảo ngược, các vector mã của lớp C_0 nêu trên và các vector mã cho lớp C_1 nêu trên được lưu trong bộ nhớ, và các vector mã cho lớp $C_{0,flip}$ và các vector mã cho lớp $C_{1,flip}$ được tạo ra bằng cách đọc động các thành phần của các vector mã nêu trên cho lớp C_0 , và các vector mã nêu trên cho lớp C_1 theo thứ tự đảo ngược khi đọc các vector mã cho lớp $C_{0,flip}$ và các vector mã cho lớp $C_{1,flip}$.

7. Bộ mã hóa-giải mã chuyển đổi bao gồm bộ lượng tử hóa vector bao gồm:

đơn vị liên lạc, được làm thích ứng để thu số các vector đích đầu vào thay đổi cho mỗi đoạn thời gian, số các vector đích đầu vào thay đổi thể hiện các đoạn của tín hiệu audio hoặc tín hiệu video;

đơn vị so sánh, được làm thích ứng để, với từng vector trong số các vector đích đầu vào thay đổi, so sánh vector đích đầu vào với bốn trọng tâm C_0 , C_1 , $C_{0,flip}$ và $C_{1,flip}$, trong đó trọng tâm $C_{0,flip}$ là phiên bản được quay của trọng tâm C_0 và trọng tâm $C_{1,flip}$

là phiên bản được quay của trọng tâm C_1 , từng trọng tâm biểu diễn lớp tương ứng của các vector mã;

đơn vị xác định được làm thích ứng để xác định điểm bắt đầu, cho từng vector trong số các vector đích đầu vào thay đổi, cho việc tìm kiếm trong bảng mã, dựa trên kết quả của kết quả từ việc so sánh của vector đích đầu vào với các trọng tâm, và để điều chỉnh động không gian tìm kiếm tới số các vector đích đầu vào thay đổi;

đơn vị tìm kiếm, được làm thích ứng để, với từng vector trong số các vector đích đầu vào thay đổi, thực hiện việc tìm kiếm trong bảng mã, bắt đầu tại điểm bắt đầu đã được xác định, và nhận diện vector mã để biểu diễn vector đích đầu vào, trong đó các vector mã trong bảng mã được xác định là khác với nhau theo số đo độ méo phản ánh khoảng cách giữa từng vector mã giữa các vector mã trong bảng mã và các trọng tâm C_0 và C_1 nêu trên, và các vector mã được sắp xếp theo số đo độ méo phản ánh khoảng cách giữa từng vector mã và các trọng tâm C_0 và C_1 nêu trên; và

đơn vị cung cấp, được làm thích ứng để truyền, qua mạch liên lạc, chỉ thị của từng vector mã trong các vector mã đã được nhận diện.

8. Bộ mã hóa-giải mã chuyển đổi theo điểm 7, trong đó đơn vị xác định được làm thích ứng để điều chỉnh động không gian tìm kiếm bằng cách tăng hoặc giảm kích cỡ của không gian tìm kiếm dựa trên số vector đích.

9. Bộ mã hóa-giải mã chuyển đổi theo điểm 7, trong đó các vector mã trong bảng mã được sắp xếp sao cho các từ mã gần nhất với trọng tâm C_0 nêu trên và xa nhất từ trọng tâm C_1 nêu trên là trong một phía của bảng mã, trong khi các từ mã gần nhất với trọng tâm C_1 nêu trên và xa nhất từ trọng tâm C_0 nêu trên được kết cụm trong phía khác của bảng mã.

10. Bộ mã hóa-giải mã chuyển đổi theo điểm 7, trong đó các vector mã cho lớp $C_{0,flip}$ là các phiên bản được quay của các vector mã cho lớp C_0 và các vector mã cho lớp $C_{1,flip}$ là các phiên bản được quay của các vector mã cho lớp C_1 .

11. Bộ mã hóa-giải mã chuyển đổi theo điểm 10, trong đó các thành phần của các vector mã $c_{k,flip}$ trong lớp $C_{j,flip}$ được cho bởi phương trình:

$$c_{k,flip} = [c_k(M)c_k(M-1) \dots c_k(1)],$$

trong đó $c_k(m)$ là các thành phần vector của lớp C_j tương ứng và M là chiều dài của vector mã và k là và chỉ số của vector mã.

12. Bộ mã hóa-giải mã chuyển đổi theo điểm 7 trong đó bảng mã được tạo ra dựa trên các trọng tâm C_0 và C_1 nêu trên, trong đó trọng tâm $C_{0,flip}$ nêu trên và trọng tâm $C_{1,flip}$ nêu trên được tạo ra bằng cách đọc các trọng tâm C_0 và C_1 nêu trên theo thứ tự đảo ngược, và các vector mã cho lớp C_0 nêu trên và các vector mã cho lớp C_1 nêu trên được lưu trong bộ nhớ, và các vector mã cho lớp $C_{0,flip}$ và các vector mã cho lớp $C_{1,flip}$ được tạo ra bằng cách đọc động các thành phần của các vector mã nêu trên cho lớp C_0 và các vector mã nêu trên cho lớp C_1 theo thứ tự đảo ngược khi đọc các vector mã cho lớp $C_{0,flip}$ và các vector mã cho lớp $C_{1,flip}$.

13. Thiết bị đầu cuối chứa bộ mã hóa-giải mã chuyển đổi theo điểm 7.

14. Vật ghi phi chuyển tiếp đọc được bởi máy tính lưu mã đọc được bởi máy tính mà mã này khi chạy trong đơn vị xử lý sẽ làm cho bộ lượng tử hóa vector thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 6.

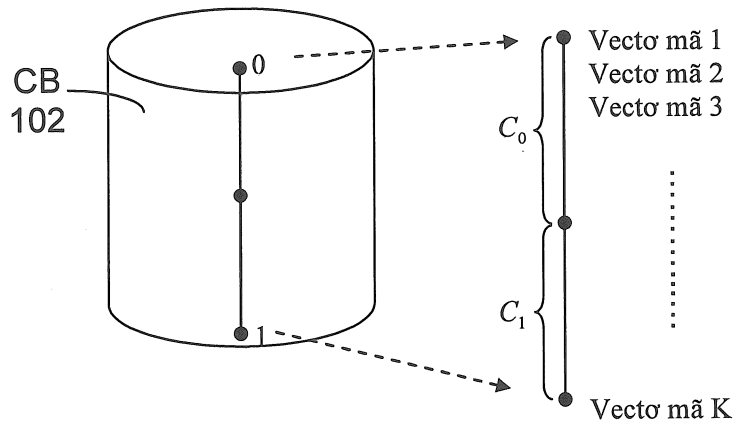


Fig. 1a

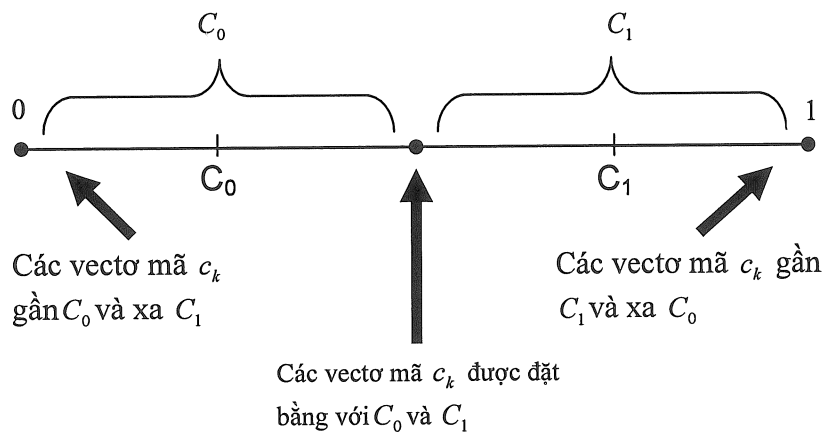


Fig. 1b

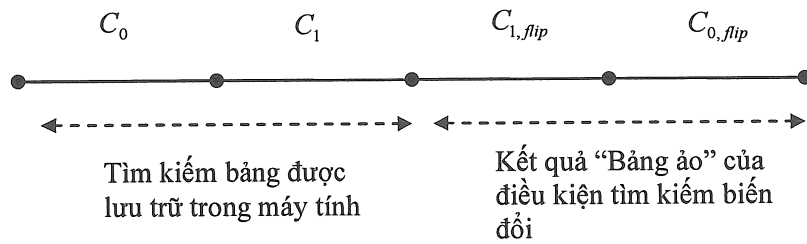


Fig.2

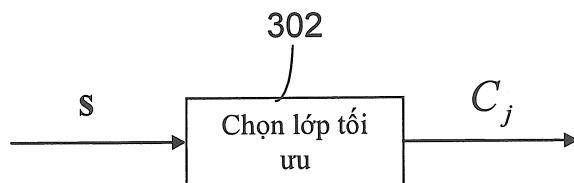


Fig.3

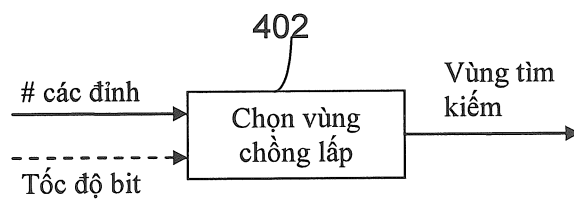


Fig.04

# các đỉnh	17	16	15	14	13	12	11	10	9	8	...	1
# các hệ số trong tìm kiếm	128	136	145	155	167	181	197	217	241	256	...	256

Fig.5

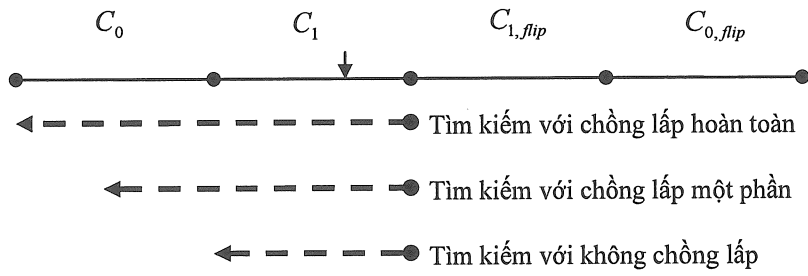


Fig.6a

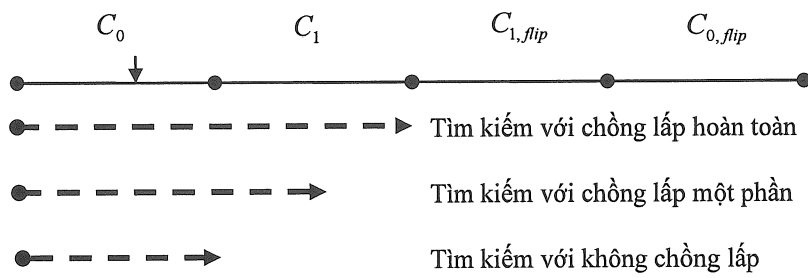


Fig.6b

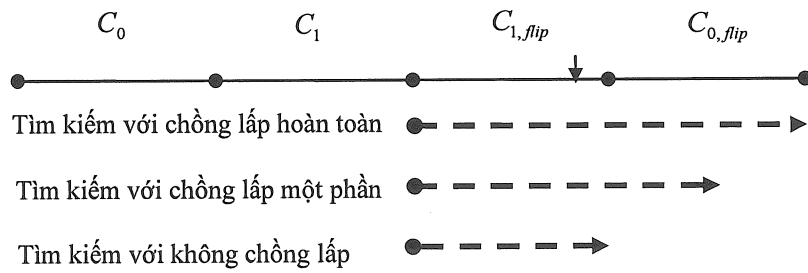


Fig.6c

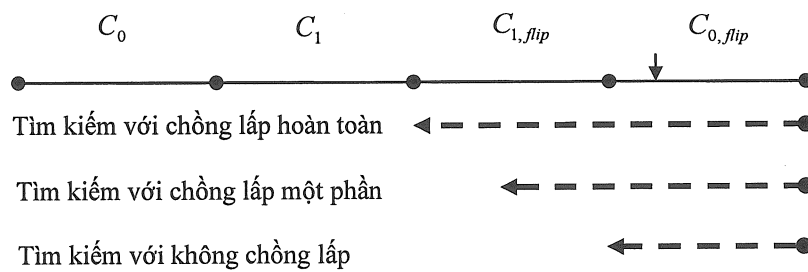


Fig.6d

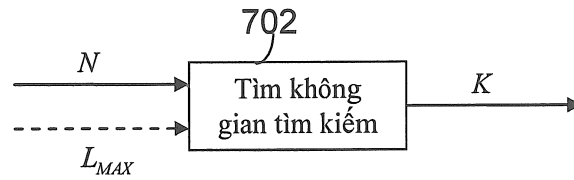


Fig.7

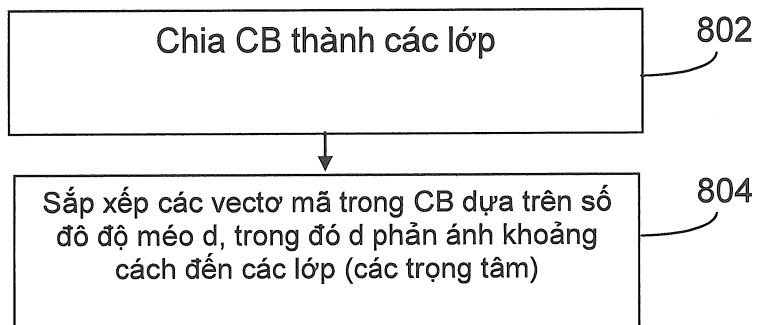


Fig.8

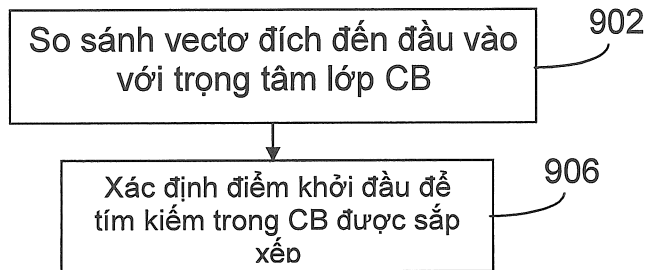


Fig.9a

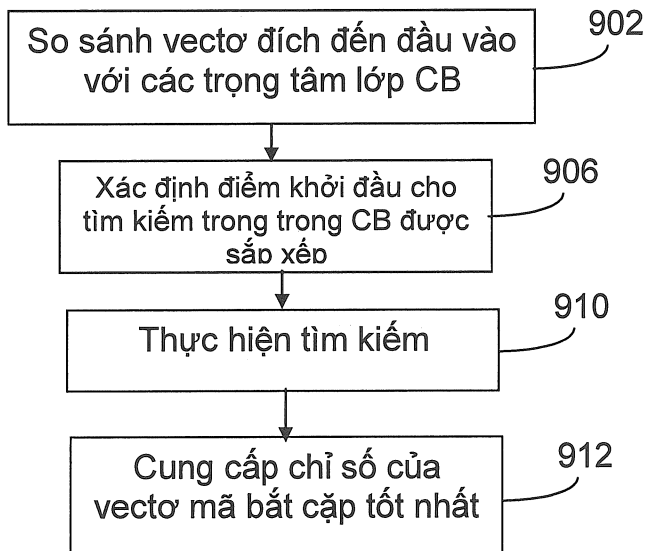


Fig.9b

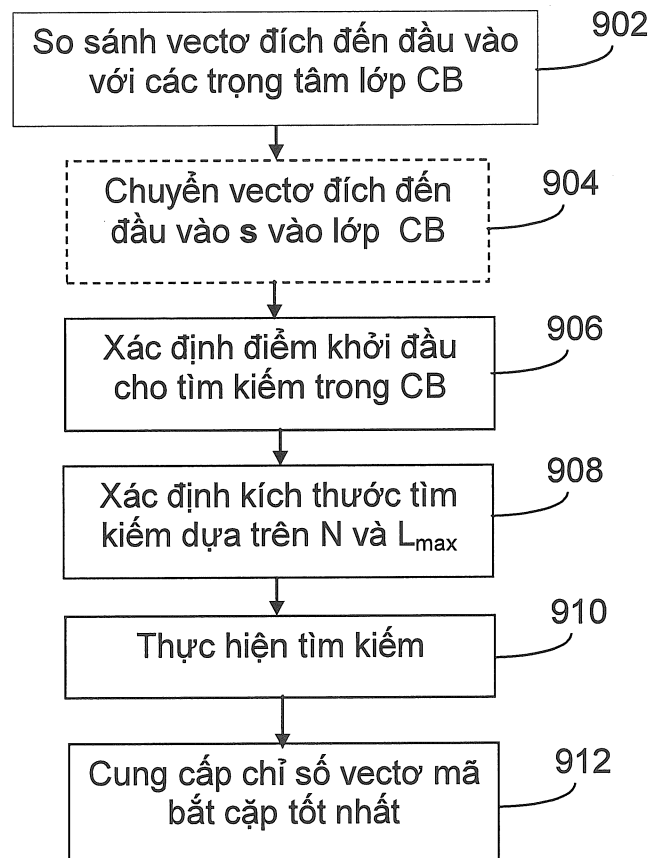


Fig.9c

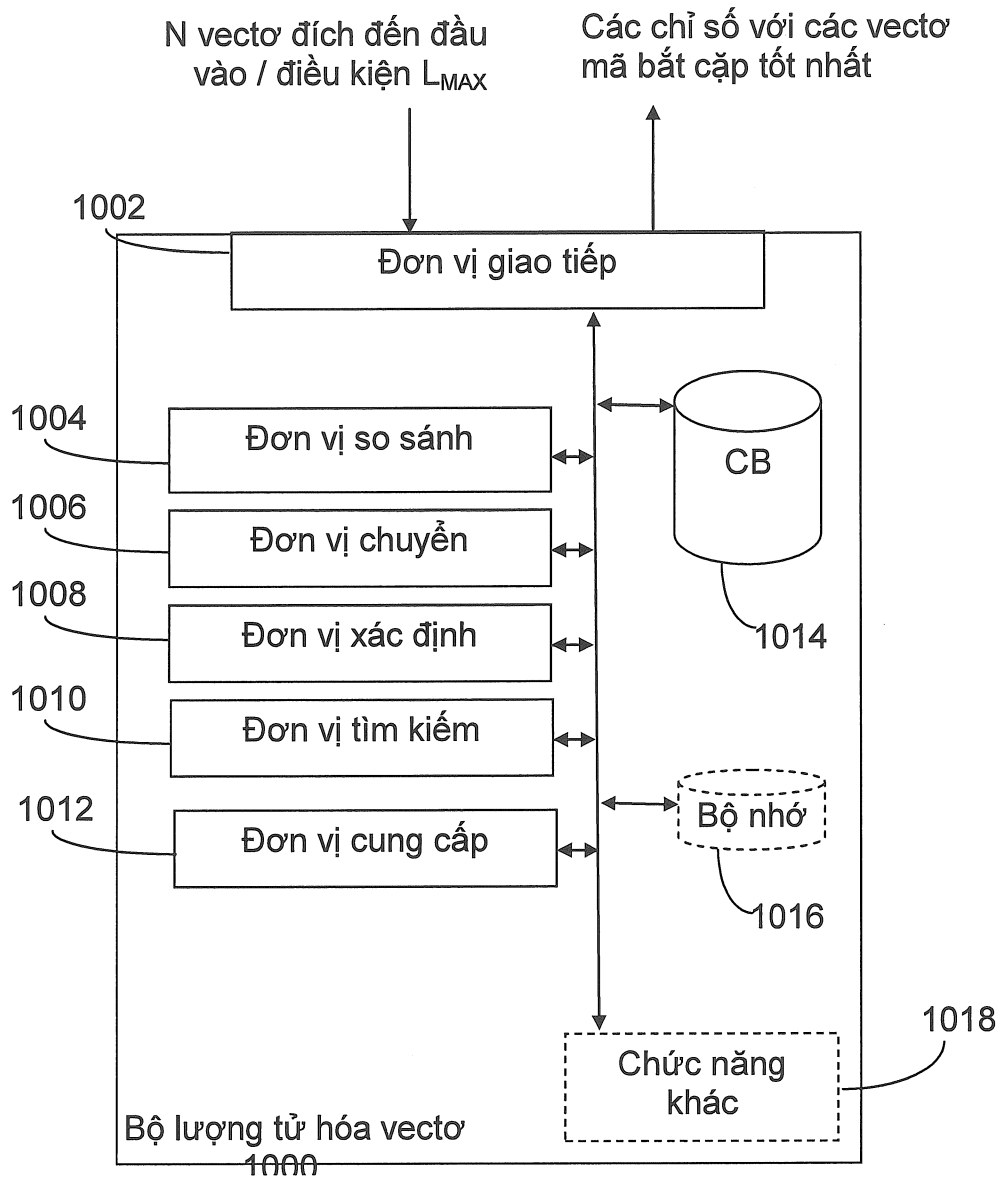


Fig.10

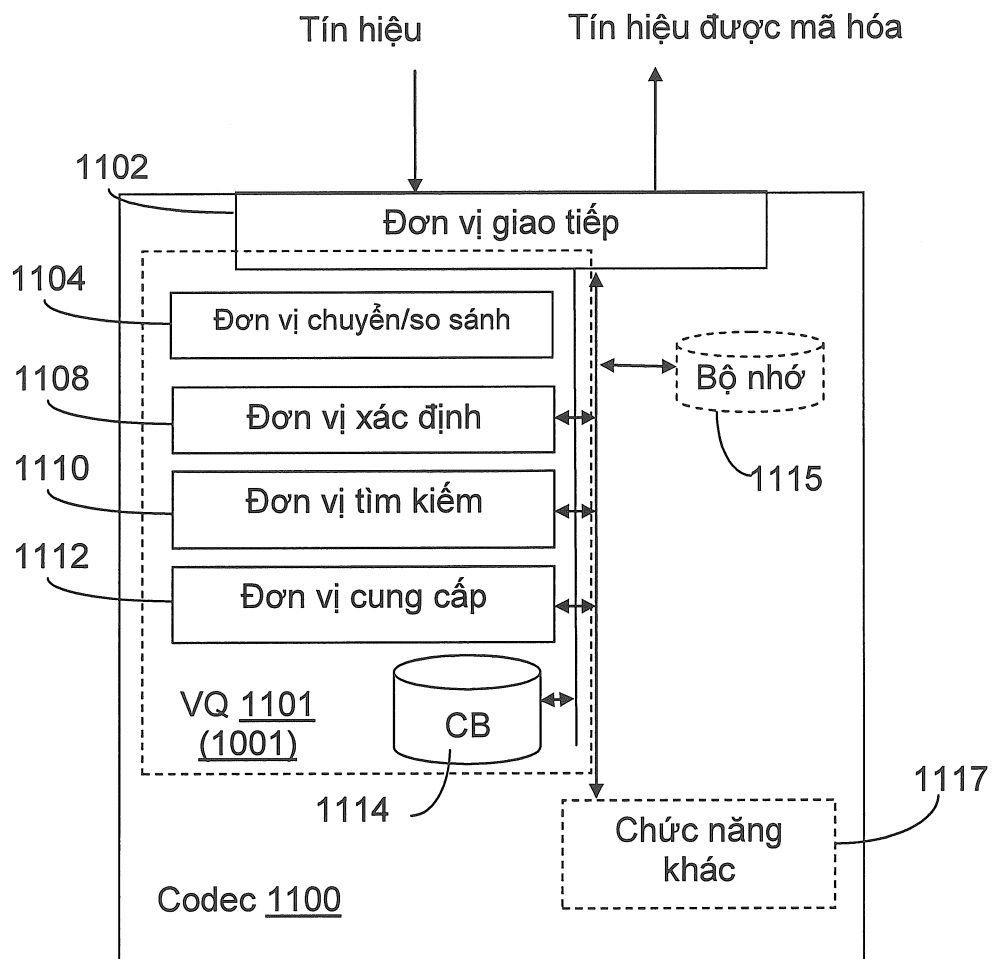


Fig.11

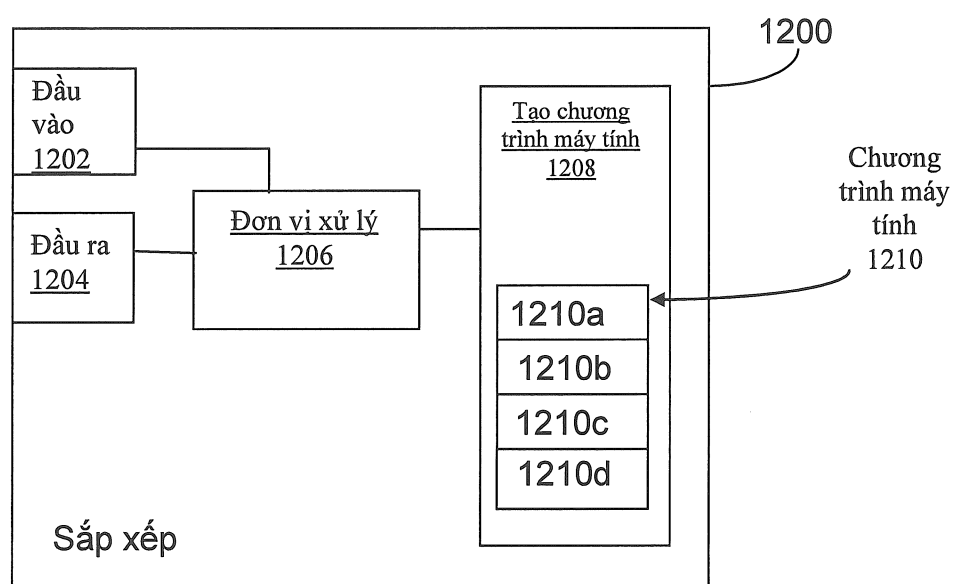


Fig.12