



(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)



1-0048166

(51)^{2020.01} H04N 19/117

(13) B

(21) 1-2021-04451

(22) 23/12/2019

(86) PCT/CN2019/127535 23/12/2019

(87) WO2020/135346 A1 02/07/2020

(30) 62/784,535 23/12/2018 US; 62/787,735 02/01/2019 US; 62/822,979 24/03/2019 US

(45) 25/07/2025 448

(43) 25/11/2021 404A

(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, P. R. China

(72) KOTRA, Anand Meher (IN); ESENLİK, Semih (TR); CHEN, Jianle (CN); GAO,
Han (CN); WANG, Biao (CN); ZHAO, Zhijie (CN).

(74) Công ty TNHH một thành viên Sở hữu trí tuệ VCCI (VCCI-IP CO.,LTD)

(54) THIẾT BỊ VÀ PHƯƠNG PHÁP LỌC TRỊ SỐ MẪU CỦA HÌNH ẢNH VÀ
PHƯƠNG TIỆN BẤT BIẾN ĐỌC ĐƯỢC BỞI MÁY TÍNH

(21) 1-2021-04451

(57) Sáng chế đề cập đến phương pháp lọc trị số mẫu của hình ảnh, bộ mã hóa, bộ giải mã, và phương tiện lưu trữ đọc được bởi máy tính, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối, trong đó hình ảnh bao gồm ít nhất $n \times m$ (n lần m) mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , trong đó phương pháp này bao gồm các bước, với t nguyên nhỏ hơn k và với j nguyên trong khoảng từ 1 đến m : tạo ra, với i nguyên trong khoảng từ 1 đến $k-t-1$, trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-t-1$, và/hoặc tạo ra, với i nguyên trong khoảng từ $k-t$ đến k , trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-t$.

Fig.9. Phân loại ALF được cải biến để tránh khỏi các bộ đệm dòng độ chói ALF (Biến thể không đối xứng)

* Chúng tôi cần việc phân loại khối ALF được cải biến cho các khối 4×4 ở ranh giới ảo

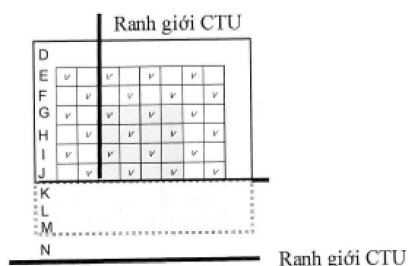


Fig. 9a

Giải pháp 1:

Đối với việc phân loại khối của khối 4×4 trung gian ở trên VB (4a):

* Chỉ sử dụng các mẫu ở trên VB trong việc phân loại (các mẫu ở các dòng E tới I)

Đối với việc phân loại khối của khối 4×4 trung gian ở dưới VB (4b):

* Chỉ sử dụng các mẫu ở dưới VB trong việc phân loại (các mẫu ở các dòng P tới L)

Lĩnh vực kỹ thuật được đề cập

Sáng chế nói chung đề cập đến lĩnh vực xử lý hình ảnh và cụ thể hơn là đến việc lọc các mẫu của các khối liền kề trong hình ảnh.

Tình trạng kỹ thuật của sáng chế

Việc mã video (mã hóa và giải mã video) được sử dụng trong khoảng rộng của các ứng dụng video số, ví dụ TV kỹ thuật số phát rộng, truyền video qua internet và các mạng di động, các ứng dụng hội thoại thời gian thực chẳng hạn như trò chuyện video, hội nghị video, DVD và các đĩa Blu-ray, hệ thống thu nhận và chỉnh sửa nội dung video, và các máy quay của các ứng dụng bảo mật.

Lượng dữ liệu video được cần để mô tả thậm chí video tương đối ngắn có thể là đáng kể, mà có thể dẫn đến các khó khăn khi dữ liệu được tạo dòng hoặc theo cách khác được truyền thông qua mạng truyền thông với dung lượng độ rộng dải giới hạn. Vì vậy, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ bởi vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để mã hóa dữ liệu video trước khi truyền hoặc lưu trữ, nhờ vậy làm giảm chất lượng của dữ liệu được cần để biểu diễn các ảnh video kỹ thuật số. Dữ liệu được nén sau đó được thu ở đích bởi thiết bị giải nén video mà giải mã dữ liệu video. Với các tài nguyên mạng giới hạn và các nhu cầu ngày càng tăng về chất lượng video cao hơn, mong muốn các kỹ thuật nén và giải nén được nâng cao để nâng cao tỷ lệ nén với việc suy giảm ít hoặc không có chất lượng hình ảnh.

Bản chất kỹ thuật của sáng chế

Các phương án của sáng chế đề xuất các máy và các phương pháp để mã hóa và giải mã theo các điểm yêu cầu độc lập.

Các mục đích nêu trên và các mục đích khác đạt được bởi chủ đề của các điểm yêu cầu bảo hộ độc lập. Các cách thực hiện khác rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Sáng chế đề xuất phương pháp lọc trị số mẫu của hình ảnh, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối, trong đó hình ảnh bao gồm ít nhất $n \times m$ (n lần m) mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , trong đó phương pháp này bao gồm các bước, với t nguyên nhỏ hơn k và với j nguyên trong khoảng từ 1 đến m : tạo ra, với i nguyên trong khoảng từ 1 đến $k-t-1$, trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-t-1$, và/hoặc tạo ra, với i nguyên trong khoảng từ $k-t$ đến k , trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-t$.

Các mẫu hỗ trợ bộ lọc là các mẫu mà các trị số của nó xác định trị số được lọc của mẫu trong đó bộ lọc được áp dụng. Ví dụ, trị số mẫu được lọc có thể là tổng trọng số của các trị số của các mẫu hỗ trợ bộ lọc. Bộ lọc có thể là bộ lọc trong cảm biến toán học và vì vậy có thể được thực hiện hoàn toàn ở phần mềm. Ranh giới khối có thể được bố trí giữa hàng k và hàng $k+1$.

Sáng chế còn đề xuất phương pháp lọc các trị số mẫu của hình ảnh, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối, trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột

mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , trong đó phương pháp này bao gồm các bước, với t nguyên nhỏ hơn k và đối với mỗi i từ 1 đến k , và mỗi j từ 1 đến m : tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó nếu i trong khoảng từ 1 đến $k-t-1$, tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-t-1$, và trong đó nếu i trong khoảng từ $k-t$ đến k , tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-t$.

Theo phương án, khối thứ nhất có thể là đơn vị cây mã hóa (coding tree unit - CTU) thứ nhất, và khối thứ hai có thể là CTU thứ hai.

Trị số của t có thể là 2, 3, 4, 5 hoặc 6. Theo phương án làm ví dụ, t có thể là 4.

Theo phương án khác, đối với mỗi i trong khoảng từ 1 đến $k-t-1$ và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc có thể bao gồm nhiều mẫu $p(i',j')$ với i' trong khoảng từ 1 đến $k-t-1$; và đối với mỗi i trong khoảng từ $k-t$ đến k và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc có thể bao gồm nhiều mẫu $p(i',j')$ với i' bằng hoặc lớn hơn $k-t$.

Điều này còn có thể được phát triển thêm ở chỗ đối với mỗi i trong khoảng từ $k-t$ đến k và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc còn có thể bao gồm một hoặc nhiều mẫu $p(i',j')$ với i' lớn hơn k . Theo đó, theo sự phát triển này tập hợp của các mẫu hỗ trợ bộ lọc đối với điểm ảnh $p(i,j)$ có thể bao gồm một hoặc nhiều mẫu từ khối thứ hai, nếu i trong khoảng từ $k-t$ đến k .

Bộ lọc có thể là bộ lọc trong vòng lặp. Theo cách khác hoặc ngoài ra bộ lọc có thể bao gồm bộ lọc giải khối.

Theo phương án khác, bộ lọc có thể bao gồm bộ lọc giải khối, bộ lọc độ dịch thích nghi mẫu (sample adaptive offset - SAO), và bộ lọc vòng lặp thích nghi (adaptive loop filter - ALF), trong đó SAO được áp dụng sau bộ lọc giải khối, và

ALF được áp dụng sau SAO.

Theo phương án, phương pháp có thể bao gồm các bước: xác định các trị số mẫu của khối thứ nhất; áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất ở các hàng 1 đến $k-t-1$; lưu trữ các trị số mẫu của khối thứ nhất từ các hàng $k-t$ đến k ở bộ đệm; xác định các trị số mẫu của khối thứ hai; và áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất ở các hàng $k-t$ mà đã được lưu trữ bộ đệm.

Theo phương án khác, bộ lọc có thể là bộ lọc vòng lặp thích nghi, và bộ lọc thích nghi có thể được xác định nhờ sử dụng bước phân loại mà truy cập các mẫu của khối hiện tại và cả các mẫu xung quanh khối. Vì vậy, bộ lọc có thể không là bộ lọc cố định. Trung bình của tất cả các mẫu trong khối hiện tại và cả trong vùng lân cận của khối đã cho có thể được sử dụng cho bước phân loại; và theo ví dụ, trong trường hợp một số mẫu xung quanh khối là không khả dụng, sau đó trị số chuẩn hóa khác có thể được sử dụng để phân loại khối hiện tại như được mô tả dưới đây.

Theo phương án làm ví dụ, khi các mẫu xung quanh khối không khả dụng, bộ lọc thích nghi có thể được xác định nhờ sử dụng trị số chuẩn hóa, trong đó trị số chuẩn hóa có thể là trị số nguyên dương và có thể phụ thuộc vào số lượng các mẫu có thể truy cập được xung quanh khối đã cho. Trị số chuẩn hóa có thể có mối tương quan ngược với số lượng các mẫu có thể truy cập được.

Theo ví dụ, các mẫu lân cận được yêu cầu cho việc phân loại. Theo cách khác, nếu một vài trong số các mẫu lân cận không khả dụng, thì chuẩn hóa khác có thể được sử dụng. Theo ví dụ, việc chuẩn hóa được thực hiện mọi lúc: Trường hợp (1) khi tất cả các mẫu khả dụng xung quanh khối khả dụng. Trường hợp (2) khi một số mẫu không khả dụng. Trong trường hợp (1): Trong trường hợp này một chuẩn hóa có thể được sử dụng. Điều này có thể tương ứng với việc tính trung bình. Nếu nhiều mẫu hơn khả dụng, sau đó tổng của tất cả các mẫu có thể được lấy và được chia bởi tổng số lượng của các mẫu. Trong trường hợp (2): Nếu ít mẫu khả dụng hơn, sau đó tổng có thể được thực hiện chỉ cho các mẫu khả dụng, và sau đó chỉ có thể được chia bởi số lượng các mẫu khả dụng. Theo đó, hệ số chuẩn hóa có thể thay đổi dựa vào số lượng các mẫu khả dụng. Theo ví dụ, trị số chuẩn hóa có thể tăng lên khi số lượng

các mẫu khả dụng giảm.

Theo phương án làm ví dụ, trị số chuẩn hóa có thể là 96, khi không phải tất cả các mẫu xung quanh khối khả dụng. Vì vậy, điều này có thể tương ứng với trường hợp khi ít nhất một mẫu xung quanh khối không khả dụng. Theo cách khác hoặc ngoài ra, trị số chuẩn hóa có thể là 64, khi tất cả các mẫu xung quanh khối khả dụng.

Theo phương án, bộ lọc có thể là bộ lọc vòng lặp thích nghi.

Theo phương án khác, thao tác đệm đối xứng có thể được áp dụng khi các mẫu được yêu cầu không thể truy cập được để lọc mẫu đã cho. Theo phương án làm ví dụ, thao tác đệm đối xứng được định rõ khi thay thế các mẫu không khả dụng bởi các mẫu lân cận trung gian mà khả dụng đối với thao tác lọc. Theo cách khác hoặc ngoài ra, thao tác đệm đối xứng có thể được định rõ là phương pháp mà bộ lọc được làm cho đối xứng bằng cách áp dụng thao tác đệm ở cả hai phía của bộ lọc mặc dù ở một phía, các mẫu bộ lọc vẫn có thể truy cập được.

Theo phương án, hình ảnh có thể được bao gồm trong khung của chuỗi video.

Theo phương án khác, phương pháp có thể được thực hiện để mã hóa hoặc giải mã hình ảnh, hoặc được thực hiện để mã hóa hoặc giải mã chuỗi video mà bao gồm hình ảnh.

Sáng chế còn đề xuất bộ mã hóa bao gồm hệ mạch xử lý để tiến hành phương pháp theo sáng chế và bộ giải mã bao gồm hệ mạch xử lý để tiến hành phương pháp theo sáng chế.

Sáng chế cũng đề xuất chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo sáng chế.

Sáng chế còn đề xuất bộ giải mã, bao gồm: một hoặc nhiều bộ xử lý; và phương tiện lưu trữ đọc được bởi máy tính không nhất thời được ghép nối với các bộ xử lý và lưu trữ việc lập trình cho việc thực thi bởi các bộ xử lý, trong đó việc lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để tiến hành phương pháp theo sáng chế.

Sáng chế còn đề xuất bộ mã hóa, bao gồm: một hoặc nhiều bộ xử lý; và phương

tiện lưu trữ đọc được bởi máy tính không nhất thời được ghép nối với các bộ xử lý và lưu trữ việc lập trình cho việc thực thi bởi các bộ xử lý, trong đó việc lập trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ mã hóa để tiến hành phương pháp theo sáng chế.

Chi tiết về một hoặc nhiều phương án được đưa ra trên các hình vẽ kèm theo và phần mô tả dưới đây. Các dấu hiệu, các mục đích, và các ưu điểm khác sẽ rõ ràng từ phần mô tả, các hình vẽ, và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Trong phần sau đây các phương án của sáng chế được mô tả chi tiết hơn dựa vào các hình vẽ và các bản vẽ kèm theo, trong đó:

Fig.1A là sơ đồ khối thể hiện ví dụ về hệ thống mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.1B là sơ đồ khối thể hiện ví dụ khác về hệ thống mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.2 là sơ đồ khối thể hiện ví dụ về bộ mã hóa video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.3 là sơ đồ khối thể hiện cấu trúc ví dụ của bộ giải mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.4 là sơ đồ khối minh họa ví dụ về máy mã hóa hoặc máy giải mã;

Fig.5 là sơ đồ khối minh họa ví dụ khác về máy mã hóa hoặc máy giải mã;

Fig.6 đến Fig.22 minh họa giản lược các ví dụ về các phương án của sáng chế.

Trong phần sau đây các số chỉ dẫn giống nhau đề cập đến các dấu hiệu giống nhau hoặc ít nhất tương đương về mặt chức năng trừ khi được quy định cụ thể khác.

Mô tả chi tiết sáng chế

Trong phần mô tả sau đây, tham chiếu được thực hiện tới các hình vẽ kèm theo, mà tạo nên một phần của sáng chế, và thể hiện, bằng minh họa, các khía cạnh cụ thể của các phương án của sáng chế hoặc các khía cạnh cụ thể trong đó các phương

án của sáng chế có thể được sử dụng. Được hiểu rằng các phương án của sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các sự thay đổi cấu trúc hoặc logic không được mô tả trên các hình vẽ. Phần mô tả chi tiết sau đây, do đó, không được hiểu theo nghĩa giới hạn, và phạm vi của sáng chế được định rõ bởi các điểm yêu cầu bảo hộ kèm theo.

Chẳng hạn, được hiểu rằng sáng chế kết hợp với phương pháp được mô tả cũng có thể đúng đối với thiết bị hoặc hệ thống tương ứng được tạo cấu hình để thực hiện phương pháp và ngược lại. Ví dụ, nếu một hoặc nhiều bước về phương pháp cụ thể được mô tả, thì thiết bị tương ứng có thể bao gồm một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, để thực hiện một hoặc nhiều bước về phương pháp được mô tả (ví dụ, một bộ phận thực hiện một hoặc nhiều bước, hoặc nhiều bộ phận đều thực hiện một hoặc nhiều trong số các bước), ngay cả khi một hoặc nhiều bộ phận như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Mặt khác, ví dụ, nếu máy cụ thể được mô tả dựa vào một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, thì phương pháp tương ứng có thể bao gồm một bước để thực hiện tính năng của một hoặc nhiều bộ phận (ví dụ, một bước thực hiện tính năng của một hoặc nhiều bộ phận, hoặc nhiều bước đều thực hiện tính năng của một hoặc nhiều trong số các bộ phận), ngay cả khi một hoặc nhiều bước như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Hơn nữa, được hiểu rằng các dấu hiệu của các phương án và/hoặc các khía cạnh làm ví dụ khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi được lưu ý cụ thể khác.

Việc mã video thường đề cập đến việc xử lý của chuỗi các hình ảnh, mà tạo nên video hoặc chuỗi video. Thay vì thuật ngữ “hình ảnh”, thuật ngữ “khung” hoặc “ảnh” có thể được sử dụng như các từ đồng nghĩa trong lĩnh vực mã video. Việc mã video (hoặc mã nói chung) bao gồm hai phần mã hóa video và giải mã video. Mã hóa video được thực hiện ở phía nguồn, thường bao gồm việc xử lý (ví dụ, bằng cách nén) các hình ảnh video gốc để làm giảm lượng dữ liệu được yêu cầu để biểu diễn các hình ảnh video (nhằm lưu trữ và/hoặc truyền hữu hiệu hơn). Giải mã video được thực hiện ở phía đích và thường bao gồm quy trình xử lý ngược so với bộ mã hóa để tái cấu trúc các hình ảnh video. Các phương án đề cập đến việc “mã” của các hình

ảnh video (hoặc các hình ảnh nói chung) sẽ được hiểu liên quan đến việc “mã hóa” hoặc “giải mã” của các hình ảnh video hoặc các chuỗi video tương ứng. Sự kết hợp của phần mã hóa và phần giải mã cũng được gọi là bộ mã hóa/giải mã (CODEC) (mã hóa và giải mã).

Trong trường hợp mã hóa video tổn hao, các hình ảnh video gốc có thể được tái cấu trúc, nghĩa là, các hình ảnh video được tái cấu trúc có cùng chất lượng như các hình ảnh video gốc (giả sử không có sự mất mát truyền hoặc sự mất mát dữ liệu khác trong suốt thời gian lưu trữ hoặc truyền). Trong trường hợp mã hóa video tổn hao, việc nén thêm, ví dụ, bởi việc lượng tử hóa, được thực hiện, để làm giảm lượng dữ liệu biểu diễn các hình ảnh video, mà không thể được tái cấu trúc hoàn toàn ở bộ giải mã, nghĩa là, chất lượng của các hình ảnh video được tái cấu trúc thấp hơn hoặc kém hơn so với chất lượng của các hình ảnh video gốc.

Nhiều chuẩn mã hóa video thuộc về nhóm “các bộ mã hóa/giải mã video lai mất mát” (nghĩa là, kết hợp việc dự báo không gian và thời gian trong miền mẫu và việc mã hóa biến đổi 2D để áp dụng việc lượng tử hóa trong miền biến đổi). Mỗi hình ảnh của chuỗi video thường được phân chia thành tập hợp của các khối không chồng lấp, và việc mã hóa thường được thực hiện trên mức khối. Nói cách khác, ở bộ mã hóa video thường được xử lý, nghĩa là, được mã hóa, trên mức khối (khối video), ví dụ, nhờ sử dụng việc dự báo không gian (hình ảnh bên trong) và/hoặc việc dự báo thời gian (hình ảnh liên kết) để tạo ra khối dự báo, trừ khối dự báo khỏi khối hiện tại (khối được xử lý hiện tại/cần được xử lý) để thu nhận khối dư, biến đổi khối dư và lượng tử hóa khối dư trong miền biến đổi để làm giảm lượng dữ liệu được truyền (nén), trong khi ở bộ giải mã quy trình xử lý ngược so với bộ mã hóa được áp dụng tới khối được mã hóa hoặc được nén để tái cấu trúc khối hiện tại nhằm biểu diễn. Hơn nữa, bộ mã hóa nhân đôi vòng lặp xử lý bộ giải mã sao cho cả hai sẽ tạo ra các việc dự báo giống nhau (ví dụ, các việc dự báo bên trong và liên kết) và/hoặc các việc tái cấu trúc để xử lý, ví dụ, mã hóa, các khối tiếp theo.

Trong phần sau đây các phương án về hệ thống mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa vào Fig.1 đến Fig.3.

Fig.1A là sơ đồ khối giản lược minh họa hệ thống mã 10 ví dụ, ví dụ, hệ thống mã video 10 (hoặc ngắn gọn là hệ thống mã 10) mà có thể sử dụng các kỹ thuật của sáng chế. Bộ mã hóa video 20 (hoặc ngắn gọn là bộ mã hóa 20) và bộ giải mã video 30 (hoặc ngắn gọn là bộ giải mã 30) của hệ thống mã video 10 biểu diễn các ví dụ về các thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật phù hợp với các ví dụ khác nhau được mô tả theo sáng chế.

Như được thể hiện trên Fig.1A, hệ thống mã 10 bao gồm thiết bị nguồn 12 được tạo cấu hình để cung cấp dữ liệu hình ảnh được mã hóa 21 ví dụ, tới thiết bị đích 14 để giải mã dữ liệu hình ảnh được mã hóa 13.

Thiết bị nguồn 12 bao gồm bộ mã hóa 20, và có thể bao gồm bổ sung, nghĩa là, một cách tùy chọn, nguồn hình ảnh 16, bộ xử lý trước (hoặc bộ phận xử lý trước) 18, ví dụ, bộ phận xử lý trước hình ảnh 18, và giao diện truyền thông hoặc bộ phận truyền thông 22.

Nguồn hình ảnh 16 có thể bao gồm hoặc là loại thiết bị chụp hình ảnh bất kỳ, ví dụ camera để chụp hình ảnh thế giới thực, và/hoặc loại thiết bị tạo hình ảnh bất kỳ, ví dụ bộ xử lý đồ họa máy tính để tạo ra hình ảnh hoạt hình máy tính, hoặc loại thiết bị khác bất kỳ để thu nhận và/hoặc cung cấp hình ảnh thế giới thực, hình ảnh được tạo ra bởi máy tính (ví dụ, nội dung màn hình, hình ảnh thực tế ảo (virtual reality - VR)) và/hoặc sự kết hợp bất kỳ của chúng (ví dụ, hình ảnh thực tế tăng cường (augmented reality - AR)). Nguồn hình ảnh có thể là loại bộ nhớ bất kỳ hoặc thiết bị lưu trữ bất kỳ trong số các hình ảnh nêu trên.

Phân biệt với bộ xử lý trước 18 và quy trình xử lý được thực hiện bởi bộ phận xử lý trước 18, hình ảnh hoặc dữ liệu hình ảnh 17 cũng có thể được gọi là hình ảnh thô hoặc dữ liệu hình ảnh thô 17.

Bộ xử lý trước 18 được tạo cấu hình để thu dữ liệu hình ảnh (thô) 17 và để thực hiện việc xử lý trước trên dữ liệu hình ảnh 17 để thu nhận hình ảnh được xử lý trước 19 hoặc dữ liệu hình ảnh được xử lý trước 19. Việc xử lý trước được thực hiện bởi bộ xử lý trước 18 có thể, ví dụ, bao gồm việc sắp xếp, chuyển đổi định dạng màu

(ví dụ, từ RGB đến YCbCr), điều chỉnh màu sắc, hoặc khử nhiễu. Có thể hiểu rằng bộ phận xử lý trước 18 có thể là bộ phận tùy chọn.

Bộ mã hóa video 20 được tạo cấu hình để thu dữ liệu hình ảnh được xử lý trước 19 và cung cấp dữ liệu hình ảnh được mã hóa 21 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để thu dữ liệu hình ảnh được mã hóa 21 và để truyền dữ liệu hình ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng) qua kênh truyền thông 13 tới thiết bị khác, ví dụ, thiết bị đích 14 hoặc thiết bị khác bất kỳ, nhằm lưu trữ hoặc tái cấu trúc trực tiếp.

Thiết bị đích 14 bao gồm bộ giải mã 30 (ví dụ, bộ giải mã video 30), và có thể bao gồm bổ sung, nghĩa là, một cách tùy chọn, giao diện truyền thông hoặc bộ phận truyền thông 28, bộ xử lý sau 32 (hoặc bộ phận xử lý sau 32) và thiết bị hiển thị 34.

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình để thu dữ liệu hình ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng), ví dụ, trực tiếp từ thiết bị nguồn 12 hoặc từ nguồn khác bất kỳ, ví dụ, thiết bị lưu trữ, ví dụ, thiết bị lưu trữ dữ liệu hình ảnh được mã hóa, và cung cấp dữ liệu hình ảnh được mã hóa 21 tới bộ giải mã 30.

Giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình để truyền hoặc thu dữ liệu hình ảnh được mã hóa 21 hoặc dữ liệu được mã hóa 13 qua liên kết truyền thông trực tiếp giữa thiết bị nguồn 12 và thiết bị đích 14, ví dụ, sự kết nối nối dây hoặc không dây trực tiếp, hoặc qua loại mạng bất kỳ, ví dụ, mạng nối dây hoặc không dây hoặc sự kết hợp bất kỳ của chúng, hoặc loại mạng riêng và công cộng bất kỳ, hoặc loại kết hợp bất kỳ của chúng.

Giao diện truyền thông 22 có thể, ví dụ, được tạo cấu hình để đóng gói dữ liệu hình ảnh được mã hóa 21 vào định dạng thích hợp, ví dụ, các gói, và/hoặc xử lý dữ liệu hình ảnh được mã hóa nhờ sử dụng loại mã hóa hoặc xử lý truyền bất kỳ đối với việc truyền qua liên kết truyền thông hoặc mạng truyền thông.

Giao diện truyền thông 28, tạo nên bản sao của giao diện truyền thông 22, có thể, ví dụ, được tạo cấu hình để thu dữ liệu được truyền và xử lý dữ liệu truyền nhờ sử dụng loại giải mã hoặc xử lý và/hoặc mở gói truyền tương ứng bất kỳ để thu nhận dữ liệu hình ảnh được mã hóa 21.

Cả, giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình là các giao diện truyền thông một chiều như được chỉ báo bởi mũi tên cho kênh truyền thông 13 trên Fig.1A chỉ từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai chiều, và có thể được tạo cấu hình, ví dụ, để gửi và thu các tin nhắn, ví dụ, để thiết lập sự kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc việc truyền dữ liệu, ví dụ, việc truyền dữ liệu hình ảnh được mã hóa.

Bộ giải mã 30 được tạo cấu hình để thu dữ liệu hình ảnh được mã hóa 21 và cung cấp dữ liệu hình ảnh được giải mã 31 hoặc hình ảnh được giải mã 31 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.3 hoặc Fig.5).

Bộ xử lý sau 32 của thiết bị đích 14 được tạo cấu hình để xử lý sau dữ liệu hình ảnh được giải mã 31 (cũng được gọi là dữ liệu hình ảnh được tái cấu trúc), ví dụ, hình ảnh được giải mã 31, để thu nhận dữ liệu hình ảnh được xử lý sau 33, ví dụ, hình ảnh được xử lý sau 33. Quy trình xử lý sau được thực hiện bởi bộ phận xử lý sau 32 có thể bao gồm, ví dụ, chuyển đổi định dạng màu (ví dụ, từ YCbCr đến RGB), điều chỉnh màu sắc, sắp xếp, hoặc lấy mẫu lại, hoặc quy trình xử lý khác bất kỳ, ví dụ, để chuẩn bị dữ liệu hình ảnh được giải mã 31 nhằm hiển thị, ví dụ, bởi thiết bị hiển thị 34.

Thiết bị hiển thị 34 của thiết bị đích 14 được tạo cấu hình để thu dữ liệu hình ảnh được xử lý sau 33 để hiển thị hình ảnh, ví dụ, tới người dùng hoặc người xem. Thiết bị hiển thị 34 có thể là hoặc bao gồm loại màn hình bất kỳ để biểu diễn hình ảnh được tái cấu trúc, ví dụ, màn hình hoặc màn hiển thị bên ngoài hoặc được tích hợp. Các màn hình có thể, ví dụ, bao gồm các màn hình tinh thể lỏng (liquid crystal display - LCD), các màn hình điốt phát sáng hữu cơ (organic light emitting diodes - OLED), các màn hình plasma, các máy chiếu, các màn hình LED nhỏ, tinh thể lỏng

trên silicon (liquid crystal on silicon - LCoS), bộ xử lý ánh sáng kỹ thuật số (digital light processor - DLP) hoặc loại màn hình khác bất kỳ.

Mặc dù Fig.1A mô tả thiết bị nguồn 12 và thiết bị đích 14 là các thiết bị riêng biệt, nhưng các phương án về các thiết bị cũng có thể bao gồm cả hai hoặc cả hai tính năng, thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng. Theo các phương án như vậy thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng có thể được thực hiện nhờ sử dụng cùng phần cứng và/hoặc phần mềm hoặc bởi phần cứng và/hoặc phần mềm riêng biệt hoặc sự kết hợp bất kỳ của chúng.

Như sẽ rõ ràng với người có trình độ dựa vào phần mô tả, sự tồn tại và việc tách (chính xác) các tính năng của các bộ phận khác nhau hoặc các tính năng nằm trong thiết bị nguồn 12 và/hoặc thiết bị đích 14 như được thể hiện trên Fig.1A có thể thay đổi tùy thuộc vào thiết bị và ứng dụng thực tế.

Bộ mã hóa 20 (ví dụ, bộ mã hóa video 20) và bộ giải mã 30 (ví dụ, bộ giải mã video 30) hoặc cả bộ mã hóa 20 và bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý như được thể hiện trên Fig.1B, chẳng hạn như một hoặc nhiều bộ vi xử lý, nhiều bộ xử lý tín hiệu kỹ thuật số (digital signal processor - DSP), các mạch tích hợp chuyên dụng (application-specific integrated circuit - ASIC), các mảng cổng lập trình được dạng trường (field-programmable gate array - FPGA), logic rời rạc, phần cứng, hoặc các sự kết hợp bất kỳ của chúng. Bộ mã hóa 20 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống con bộ mã hóa khác bất kỳ được mô tả ở đây. Bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống con bộ giải mã khác bất kỳ được mô tả ở đây. Hệ mạch xử lý có thể được tạo cấu hình để thực hiện các thao tác khác nhau như được thảo luận dưới đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được thực hiện một phần ở phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong phương tiện lưu trữ có thể đọc được bởi máy tính không nhất thời, thích hợp và có thể thực

thi các lệnh ở phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Một trong số bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp như phần của bộ mã hóa/bộ giải mã (CODEC) được kết hợp trong thiết bị đơn, ví dụ, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm bất kỳ trong số khoảng rộng của các thiết bị, bao gồm loại bất kỳ của các thiết bị cầm tay hoặc văn phòng, ví dụ, các máy tính cầm tay hoặc xách tay, các điện thoại di động, các điện thoại thông minh, các máy tính bảng (tablet) hoặc các máy tính bảng (tablet computer), các camera, các máy tính để bàn, các thiết bị giải mã tín hiệu (set-top box), các tivi, các thiết bị hiển thị, các trình phát phương tiện kỹ thuật số, các máy chơi trò chơi video, các thiết bị tạo dòng video (chẳng hạn như các máy chủ dịch vụ nội dung hoặc các máy chủ phân bổ nội dung), thiết bị thu phát sóng, thiết bị truyền phát sóng, hoặc tương tự và có thể sử dụng không hoặc loại bất kỳ của hệ điều hành. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị cho việc truyền thông không dây. Vì vậy, thiết bị nguồn 12 và thiết bị đích 14 có thể là các thiết bị truyền thông không dây.

Trong một số trường hợp, hệ thống mã video 10 được minh họa trên Fig.1A chỉ là ví dụ và các kỹ thuật của sáng chế có thể áp dụng tới các sự thiết đặt mã video (ví dụ, mã hóa video hoặc giải mã video) mà không cần thiết bao gồm việc truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Theo các ví dụ khác, dữ liệu được truy xuất từ bộ nhớ cục bộ, được tạo dòng qua mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu tới bộ nhớ, và/hoặc thiết bị giải mã video có thể truy xuất và giải mã dữ liệu từ bộ nhớ. Theo một số ví dụ, việc mã hóa và giải mã được thực hiện bởi các thiết bị mà không truyền thông với thiết bị khác, mà mã hóa một cách đơn giản dữ liệu tới bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Nhằm thuận tiện cho việc mô tả, các phương án của sáng chế được mô tả ở đây, ví dụ, bằng cách tham chiếu tới việc mã hóa video hiệu quả cao (High-Efficiency Video Coding - HEVC) hoặc phần mềm tham chiếu của việc mã hóa video đa năng

(Versatile Video Coding - VVC), chuẩn mã hóa video thế hệ tiếp theo được phát triển bởi nhóm hợp tác mã hóa video liên kết (Joint Collaboration Team on Video coding - JCT-VC) của nhóm các chuyên gia mã hóa video (Video Coding Experts Group - VCEG) ITU-T và nhóm các chuyên gia hình ảnh động (Motion Picture Experts Group - MPEG) ISO/IEC. Người có trình độ trung bình trong lĩnh vực kỹ thuật tương ứng sẽ hiểu rằng các phương án của sáng chế không giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

Fig.2 thể hiện sơ đồ khối giản lược của bộ mã hóa video 20 ví dụ mà được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Theo ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đầu vào 201 (hoặc giao diện đầu vào 201), bộ phận tính toán phần dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lượng tử hóa ngược 210, và bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ phận lọc vòng lặp 220, bộ đệm hình ảnh được giải mã (decoded picture buffer - DPB) 230, bộ phận lựa chọn chế độ 260, bộ phận mã hóa entropi 270 và đầu ra 272 (hoặc giao diện đầu ra 272). Bộ phận lựa chọn chế độ 260 có thể bao gồm bộ phận dự báo liên kết 244, bộ phận dự báo bên trong 254 và bộ phận phân chia 262. Bộ phận dự báo liên kết 244 có thể bao gồm bộ phận đánh giá chuyển động và bộ phận bù chuyển động (không được thể hiện). Bộ mã hóa video 20 như được thể hiện trên Fig.2 cũng có thể được gọi là bộ mã hóa video lai hoặc bộ mã hóa video theo bộ mã hóa/giải mã video lai.

Bộ phận tính toán phần dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lựa chọn chế độ 260 có thể được đề cập đến là tạo nên đường truyền tín hiệu xuôi của bộ mã hóa 20, trong khi bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ đệm 216, bộ lọc vòng lặp 220, bộ đệm hình ảnh được giải mã (DPB) 230, bộ phận dự báo liên kết 244 và bộ phận dự báo bên trong 254 có thể được đề cập đến là tạo nên đường truyền tín hiệu ngược của bộ mã hóa video 20, trong đó đường truyền tín hiệu ngược của bộ mã hóa video 20 tương ứng với đường truyền tín hiệu của bộ giải mã (xem bộ giải mã video 30 trên Fig.3). Bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ

phận tái cấu trúc 214, bộ lọc vòng lặp 220, bộ đệm hình ảnh được giải mã (DPB) 230, bộ phận dự báo liên kết 244 và bộ phận dự báo bên trong 254 cũng được đề cập là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20.

Các hình ảnh & việc phân chia hình ảnh (các hình ảnh & các khối)

Bộ mã hóa 20 có thể được tạo cấu hình để thu, ví dụ, qua đầu vào 201, hình ảnh 17 (hoặc dữ liệu hình ảnh 17), ví dụ, hình ảnh của chuỗi các hình ảnh tạo nên video hoặc chuỗi video. Hình ảnh hoặc dữ liệu hình ảnh thu được cũng có thể là hình ảnh được xử lý trước 19 (hoặc dữ liệu hình ảnh được xử lý trước 19). Nhằm đơn giản hóa phân mô tả sau đây đề cập đến hình ảnh 17. Hình ảnh 17 cũng có thể được gọi là hình ảnh hiện tại hoặc hình ảnh được mã hóa (cụ thể trong việc mã hóa video để phân biệt hình ảnh hiện tại với các hình ảnh khác, ví dụ, các hình ảnh được mã hóa và/hoặc được giải mã trước đó của cùng chuỗi video, nghĩa là, chuỗi video mà cũng bao gồm hình ảnh hiện tại).

Hình ảnh (kỹ thuật số) là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ. Mẫu trong mảng cũng có thể được gọi là điểm ảnh (dạng ngắn của thành phần hình ảnh) hoặc điểm ảnh (pel). Số lượng các mẫu theo hướng chiều ngang và chiều thẳng đứng (hoặc trục) của mảng hoặc hình ảnh định rõ kích thước và/hoặc độ phân giải của hình ảnh. Nhằm biểu diễn màu sắc, thường ba thành phần màu sắc được sử dụng, nghĩa là, hình ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Theo định dạng RGB hoặc không gian màu hình ảnh bao gồm mảng mẫu đỏ, xanh lá cây và xanh lam tương ứng. Tuy nhiên, trong mã hóa video mỗi điểm ảnh thường được biểu diễn theo định dạng độ chói và sắc độ hoặc không gian màu, ví dụ, YCbCr, mà bao gồm thành phần độ chói được chỉ báo bởi Y (thỉnh thoảng L cũng được sử dụng thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần sắc độ (hoặc viết tắt là luma) Y biểu diễn độ sáng hoặc cường độ mức xám (ví dụ, như trong hình ảnh thước xám), trong khi hai thành phần sắc độ (hoặc viết tắt là chroma) Cb và Cr biểu diễn sắc độ hoặc các thành phần thông tin màu sắc. Theo đó, hình ảnh theo định dạng YCbCr bao gồm mảng mẫu độ chói của các trị số mẫu độ chói (Y), và hai mảng mẫu sắc độ của các trị số sắc độ

(Cb và Cr). Các hình ảnh theo định dạng RGB có thể được chuyển đổi hoặc được biến đổi thành định dạng YCbCr và ngược lại, quy trình xử lý cũng được biết đến là biến đổi hoặc chuyển đổi màu sắc. Nếu hình ảnh là đơn sắc, thì hình ảnh có thể bao gồm chỉ mảng mẫu độ chói. Theo đó, hình ảnh có thể là, ví dụ, mảng của các mẫu độ chói theo định dạng đơn sắc hoặc mảng của các mẫu độ chói và hai mảng tương ứng của các mẫu sắc độ theo định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Các phương án về bộ mã hóa video 20 có thể bao gồm bộ phận phân chia hình ảnh (không được mô tả trên Fig.2) được tạo cấu hình để phân chia hình ảnh 17 thành các khối hình ảnh 203 (thường không chồng lấp). Các khối này cũng có thể được gọi là các khối gốc, các khối lớn (H.264/AVC) hoặc các khối cây mã hóa (coding tree block - CTB) hoặc các đơn vị cây mã hóa (CTU) (H.265/HEVC và VVC). Bộ phận phân chia hình ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối cho tất cả các hình ảnh của chuỗi video và lưới tương ứng định rõ kích thước khối, hoặc để thay đổi kích thước khối giữa các hình ảnh hoặc các tập hợp con hoặc các nhóm của các hình ảnh, và phân chia mỗi hình ảnh thành các khối tương ứng.

Theo các phương án khác, bộ mã hóa video có thể được tạo cấu hình để thu trực tiếp khối 203 của hình ảnh 17, ví dụ, một, nhiều hoặc tất cả các khối tạo nên hình ảnh 17. Khối hình ảnh 203 cũng có thể được gọi là khối hình ảnh hiện tại hoặc khối hình ảnh được mã hóa.

Giống hình ảnh 17, khối hình ảnh 203 lại là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ (các trị số mẫu), mặc dù kích thước nhỏ hơn hình ảnh 17. Nói cách khác, khối 203 có thể bao gồm, ví dụ, một mảng mẫu (ví dụ, mảng độ chói trong trường hợp hình ảnh đơn sắc 17, hoặc mảng độ chói hoặc sắc độ trong trường hợp hình ảnh màu) hoặc ba mảng mẫu (ví dụ, độ chói và hai mảng sắc độ trong trường hợp hình ảnh màu 17) hoặc số lượng và/hoặc loại khác bất kỳ của các mảng tùy thuộc vào định dạng màu được áp dụng. Số lượng các mẫu theo hướng chiều ngang và chiều thẳng đứng (hoặc trục) của khối 203 định rõ kích thước của khối 203. Theo đó, khối có thể là, ví dụ, mảng $M \times N$ (cột M nhân hàng N) của các mẫu, hoặc mảng $M \times N$ của các hệ số biến đổi.

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể được tạo cấu hình để mã hóa hình ảnh 17 theo từng khối, ví dụ, việc mã hóa và dự báo được thực hiện trên mỗi khối 203.

Tính toán phần dư

Bộ phận tính toán phần dư 204 có thể được tạo cấu hình để tính toán khối dư 205 (cũng được gọi là phần dư 205) dựa vào khối hình ảnh 203 và khối dự báo 265 (chi tiết thêm về khối dự báo 265 được đưa ra dưới đây), ví dụ, bằng cách trừ các trị số mẫu của khối dự báo 265 khỏi các trị số mẫu của khối hình ảnh 203, theo từng mẫu (theo từng điểm ảnh) để thu nhận khối dư 205 trong miền mẫu.

Biến đổi

Bộ phận xử lý biến đổi 206 được tạo cấu hình để áp dụng việc biến đổi, ví dụ, biến đổi cosin rời rạc (discrete cosine transform, viết tắt là DCT) hoặc biến đổi sin rời rạc (discrete sine transform, viết tắt là DST), trên các trị số mẫu của khối dư 205 để thu nhận các hệ số biến đổi 207 trong miền biến đổi. Các hệ số biến đổi 207 cũng có thể được gọi là các hệ số dư biến đổi và biểu diễn khối dư 205 trong miền biến đổi.

Bộ phận xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng các phép xấp xỉ số nguyên của DCT/DST, chẳng hạn như các sự biến đổi được quy định cho HEVC/H.265. So với việc biến đổi DCT trực giao, các phép xấp xỉ số nguyên như vậy thường được chia tỷ lệ bởi hệ số nhất định. Để giữ lại định mức của khối dư mà được xử lý bởi các sự biến đổi xuôi và ngược, các hệ số tỷ lệ bổ sung được áp dụng như phần của quy trình biến đổi. Các hệ số tỷ lệ thường được chọn dựa vào các sự ràng buộc nhất định như các hệ số tỷ lệ là lũy thừa của hai đối với các phép toán dịch chuyển, độ sâu bit của các hệ số biến đổi, sự cân bằng giữa độ chính xác và các chi phí thực hiện, v.v. Các hệ số tỷ lệ cụ thể, ví dụ, được quy định đối với việc biến đổi ngược, ví dụ, bởi bộ phận xử lý biến đổi ngược 212 (và việc biến đổi ngược tương ứng, ví dụ, bởi bộ phận xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các hệ số tỷ lệ tương ứng đối với việc biến đổi xuôi, ví dụ, bởi bộ phận xử lý biến đổi 206, ở bộ mã hóa 20 có thể được quy định theo đó.

Các phương án về bộ mã hóa video 20 (một cách tương ứng bộ phận xử lý biến đổi 206) có thể được tạo cấu hình để đưa ra các thông số biến đổi, ví dụ, loại biến đổi hoặc các phép biến đổi, ví dụ, trực tiếp hoặc được mã hóa hoặc được nén qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số biến đổi để giải mã.

Lượng tử hóa

Bộ phận lượng tử hóa 208 có thể được tạo cấu hình để lượng tử hóa các hệ số biến đổi 207 để thu nhận các hệ số được lượng tử hóa 209, ví dụ, bằng cách áp dụng việc lượng tử hóa vô hướng hoặc việc lượng tử hóa vectơ. Các hệ số được lượng tử hóa 209 cũng có thể được gọi là các hệ số biến đổi được lượng tử hóa 209 hoặc các hệ số dư được lượng tử hóa 209.

Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số biến đổi 207. Ví dụ, hệ số biến đổi n bit có thể được làm tròn xuống tới hệ số biến đổi m bit trong suốt thời gian lượng tử hóa, trong đó n lớn hơn m . Mức lượng tử hóa có thể được cải biến bằng cách điều chỉnh thông số lượng tử hóa (quantization parameter - QP). Ví dụ đối với việc lượng tử hóa vô hướng, việc chia tỷ lệ khác có thể được áp dụng để đạt được việc lượng tử hóa mịn hơn hoặc thô hơn. Các kích thước bước lượng tử hóa nhỏ hơn tương ứng với việc lượng tử hóa mịn hơn, trong khi các kích thước bước lượng tử hóa lớn hơn tương ứng với việc lượng tử hóa thô hơn. Kích thước bước lượng tử hóa có thể áp dụng được có thể được chỉ báo bởi thông số lượng tử hóa (QP). Thông số lượng tử hóa có thể ví dụ là chỉ số tối tập hợp được định rõ trước của các kích thước bước lượng tử hóa có thể áp dụng được. Ví dụ, các thông số lượng tử hóa nhỏ có thể tương ứng với việc lượng tử hóa mịn (các kích thước bước lượng tử hóa nhỏ) và các thông số lượng tử hóa lớn có thể tương ứng với việc lượng tử hóa thô (các kích thước bước lượng tử hóa lớn) hoặc ngược lại. Lượng tử hóa có thể bao gồm phép chia cho kích thước bước lượng tử hóa và tương ứng và/hoặc việc giải lượng tử hóa ngược, ví dụ, bởi bộ phận lượng tử hóa ngược 210, có thể bao gồm phép nhân với kích thước bước lượng tử hóa. Các phương án theo một số chuẩn, ví dụ, HEVC, có thể được tạo cấu hình để sử dụng

thông số lượng tử hóa để xác định kích thước bước lượng tử hóa. Nói chung, kích thước bước lượng tử hóa có thể được tính toán dựa vào thông số lượng tử hóa nhờ sử dụng phép xấp xỉ điểm cố định của phương trình bao gồm phép chia. Các hệ số chia tỷ lệ bổ sung có thể được giới thiệu cho việc lượng tử hóa và giải lượng tử hóa để khôi phục tiêu chuẩn của khôi dư, mà có thể được cải biến bởi vì việc chia tỷ lệ được sử dụng trong phép xấp xỉ điểm cố định của phương trình đối với kích thước bước lượng tử hóa và thông số lượng tử hóa. Theo một cách thực hiện ví dụ, việc chia tỷ lệ của việc biến đổi ngược và giải lượng tử hóa có thể được kết hợp. Theo cách khác, các bảng lượng tử hóa được tùy chỉnh có thể được sử dụng và được truyền tín hiệu từ bộ mã hóa tới bộ giải mã, ví dụ, ở dòng bit. Lượng tử hóa là phép toán mất mát, trong đó sự mất mát tăng lên với các kích thước bước lượng tử hóa tăng lên.

Các phương án về bộ mã hóa video 20 (một cách tương ứng bộ phận lượng tử hóa 208) có thể được tạo cấu hình để đưa ra các thông số lượng tử hóa (QP), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và áp dụng các thông số lượng tử hóa để giải mã.

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 210 được tạo cấu hình để áp dụng việc lượng tử hóa ngược của bộ phận lượng tử hóa 208 trên các hệ số được lượng tử hóa để thu nhận các hệ số được giải lượng tử hóa 211, ví dụ, bằng cách áp dụng ngược sơ đồ lượng tử hóa được áp dụng bởi bộ phận lượng tử hóa 208 dựa vào hoặc nhờ sử dụng kích thước bước lượng tử hóa giống như bộ phận lượng tử hóa 208. Các hệ số được giải lượng tử hóa 211 cũng có thể được gọi là các hệ số dư được giải lượng tử hóa 211 và tương ứng - mặc dù thường không giống với các hệ số biến đổi do sự mất mát bởi việc lượng tử hóa – với các hệ số biến đổi 207.

Biến đổi ngược

Bộ phận xử lý biến đổi ngược 212 được tạo cấu hình để áp dụng việc biến đổi ngược của việc biến đổi được áp dụng bởi bộ phận xử lý biến đổi 206, ví dụ, biến đổi cosin rời rạc (DCT) ngược hoặc biến đổi sin rời rạc (DST) ngược hoặc các phép

biến đổi ngược khác, để thu nhận khối dư được tái cấu trúc 213 (hoặc các hệ số được giải lượng tử hóa 213 tương ứng) trong miền mẫu. Khối dư được tái cấu trúc 213 cũng có thể được gọi là khối biến đổi 213.

Tái cấu trúc

Bộ phận tái cấu trúc 214 (ví dụ, bộ bổ sung hoặc bộ cộng 214) được tạo cấu hình để bổ sung khối biến đổi 213 (nghĩa là, khối dư được tái cấu trúc 213) vào khối dự báo 265 để thu nhận khối được tái cấu trúc 215 trong miền mẫu, ví dụ, bằng cách bổ sung – theo từng mẫu - các trị số mẫu của khối dư được tái cấu trúc 213 và các trị số mẫu của khối dự báo 265.

Lọc

Bộ phận lọc vòng lặp 220 (hoặc viết tắt là “bộ lọc vòng lặp” 220), được tạo cấu hình để lọc khối được tái cấu trúc 215 để thu nhận khối được lọc 221, hoặc nói chung, để lọc các mẫu được tái cấu trúc để thu nhận các mẫu được lọc. Bộ phận lọc vòng lặp, ví dụ, được tạo cấu hình để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận lọc vòng lặp 220 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc giải khối, bộ lọc độ dịch thích nghi mẫu (SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc hai chiều, bộ lọc vòng lặp thích nghi (ALF), làm sắc, các bộ lọc làm mịn hoặc các bộ lọc phối hợp, hoặc sự kết hợp bất kỳ của chúng. Mặc dù bộ phận lọc vòng lặp 220 được thể hiện trên Fig.2 là bộ lọc trong vòng lặp, theo các cấu hình khác, bộ phận lọc vòng lặp 220 có thể được thực hiện như bộ lọc sau vòng lặp. Khối được lọc 221 cũng có thể được gọi là khối được tái cấu trúc được lọc 221.

Các phương án về bộ mã hóa video 20 (một cách tương ứng bộ phận lọc vòng lặp 220) có thể được tạo cấu hình để đưa ra các thông số bộ lọc vòng lặp (chẳng hạn như thông tin độ dịch thích nghi mẫu), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã 30 có thể thu và áp dụng cùng các thông số bộ lọc vòng lặp hoặc các bộ lọc vòng lặp tương ứng để giải mã.

Bộ đệm hình ảnh được giải mã

Bộ đệm hình ảnh được giải mã (DPB) 230 có thể là bộ nhớ mà lưu trữ các hình ảnh tham chiếu, hoặc dữ liệu hình ảnh tham chiếu nói chung, để mã hóa dữ liệu video bởi bộ mã hóa video 20. DPB 230 có thể được tạo nên bởi bất kỳ trong số các thiết bị nhớ, chẳng hạn như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM), bao gồm DRAM đồng bộ (synchronous DRAM - SDRAM), RAM điện từ (magnetoresistive RAM - MRAM), RAM điện trở (resistive RAM - RRAM), hoặc các loại khác của các thiết bị nhớ. Bộ đệm hình ảnh được giải mã (DPB) 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. Bộ đệm hình ảnh được giải mã 230 còn có thể được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, ví dụ, các khối được lọc và được tái cấu trúc trước đó 221, của cùng hình ảnh hiện tại hoặc của các hình ảnh khác nhau, ví dụ, các hình ảnh được tái cấu trúc trước đó, và có thể cung cấp các hình ảnh được tái cấu trúc, nghĩa là, được giải mã trước đó hoàn toàn (và các khối và các mẫu tham chiếu tương ứng) và/hoặc một phần hình ảnh hiện tại được tái cấu trúc (và các khối và các mẫu tham chiếu tương ứng), ví dụ đối với việc dự báo liên kết. Bộ đệm hình ảnh được giải mã (DPB) 230 cũng có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái cấu trúc không được lọc 215, hoặc các mẫu được tái cấu trúc không được lọc nói chung, ví dụ, nếu khối được tái cấu trúc 215 không được lọc bởi bộ phận lọc vòng lặp 220, hoặc phiên bản được xử lý thêm khác bất kỳ của các khối hoặc các mẫu được tái cấu trúc.

Lựa chọn chế độ (phân chia & dự báo)

Bộ phận lựa chọn chế độ 260 bao gồm bộ phận phân chia 262, bộ phận dự báo liên kết 244 và bộ phận dự báo bên trong 254, và được tạo cấu hình để thu hoặc thu nhận dữ liệu hình ảnh gốc, ví dụ, khối gốc 203 (khối hiện tại 203 của hình ảnh hiện tại 17), và dữ liệu hình ảnh được tái cấu trúc, ví dụ, các mẫu hoặc các khối được tái cấu trúc được lọc và/hoặc không được lọc của cùng hình ảnh (hiện tại) và/hoặc từ một hoặc nhiều hình ảnh được giải mã trước đó, ví dụ, từ bộ đệm hình ảnh được giải mã 230 hoặc các bộ đệm khác (ví dụ, bộ đệm dòng, không được thể hiện). Dữ liệu hình ảnh được tái cấu trúc được sử dụng như dữ liệu hình ảnh tham chiếu cho việc

dự báo, ví dụ, dự báo liên kết hoặc dự báo bên trong, để thu nhận khối dự báo 265 hoặc bộ dự báo 265.

Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc lựa chọn việc phân chia cho chế độ dự báo khối hiện tại (bao gồm việc không phân chia) và chế độ dự báo (ví dụ, chế độ dự báo bên trong hoặc liên kết) và tạo ra khối dự báo 265 tương ứng, mà được sử dụng cho việc tính toán của khối dư 205 và cho việc tái cấu trúc của khối được tái cấu trúc 215.

Các phương án về bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để lựa chọn việc phân chia và chế độ dự báo (ví dụ, từ các việc phân chia và chế độ dự báo được hỗ trợ bởi hoặc khả dụng đối với bộ phận lựa chọn chế độ 260), mà đưa ra sự phù hợp nhất hoặc nói cách khác khối dư nhỏ nhất (khối dư nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc phí tổn truyền tín hiệu nhỏ nhất (phí tổn truyền tín hiệu nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định việc phân chia và chế độ dự báo dựa vào sự tối ưu hóa tốc độ biến dạng (rate distortion optimization - RDO), nghĩa là, lựa chọn chế độ dự báo mà đưa ra tốc độ biến dạng nhỏ nhất. Các thuật ngữ như “tốt nhất”, “nhỏ nhất”, “tối ưu” v.v. theo ngữ cảnh này không cần thiết đề cập đến tổng thể “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. mà cũng có thể đề cập đến việc hoàn tất của tiêu chuẩn chấm dứt hoặc lựa chọn như trị số vượt quá hoặc giảm dưới ngưỡng hoặc các sự hạn chế khác có khả năng dẫn tới “sự lựa chọn dưới điều kiện tốt nhất” nhưng làm giảm độ phức tạp và thời gian xử lý.

Nói cách khác, bộ phận phân chia 262 có thể được tạo cấu hình để phân chia khối 203 thành các phần chia khối nhỏ hơn hoặc các khối con (mà tạo nên lại các khối), ví dụ, nhờ sử dụng lặp lại việc phân chia cây tứ phân (quad-tree - QT), việc phân chia cây nhị phân (binary-tree - BT), việc phân chia cây tam phân (triple-tree - TT) hoặc sự kết hợp bất kỳ của chúng, và để thực hiện, ví dụ, việc dự báo cho mỗi trong số các phần chia khối hoặc các khối con, trong đó việc lựa chọn chế độ bao

gồm việc lựa chọn cấu trúc cây của khối được phân chia 203 và các chế độ dự báo được áp dụng tới mỗi trong số các phần chia khối hoặc các khối con.

Trong phần sau đây việc phân chia (ví dụ, bởi bộ phận phân chia 260) và việc xử lý dự báo (bởi bộ phận dự báo liên kết 244 và bộ phận dự báo bên trong 254) được thực hiện bởi bộ mã hóa video 20 ví dụ sẽ được giải thích chi tiết hơn.

Phân chia

Bộ phận phân chia 262 có thể phân chia (hoặc phân tách) khối hiện tại 203 thành các phần chia nhỏ hơn, ví dụ, các khối nhỏ hơn có kích thước hình vuông hoặc hình chữ nhật. Các khối nhỏ hơn này (mà cũng có thể được gọi là các khối con) còn có thể được phân chia thậm chí thành các phần chia nhỏ hơn. Điều này cũng được đề cập đến là việc phân chia cây hoặc việc phân chia cây theo thứ bậc, trong đó khối gốc, ví dụ, ở mức cây gốc 0 (mức thứ bậc 0, độ sâu 0), có thể được phân chia đệ quy, ví dụ, được phân chia thành hai hoặc nhiều khối của mức cây thấp hơn tiếp theo, ví dụ, các nút ở mức cây 1 (mức thứ bậc 1, độ sâu 1), trong đó các khối này có thể được phân chia lại thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, ví dụ, mức cây 2 (mức thứ bậc 2, độ sâu 2), v.v. cho đến khi việc phân chia được chấm dứt, ví dụ, bởi vì tiêu chuẩn chấm dứt được hoàn tất, ví dụ, độ sâu cây lớn nhất hoặc kích thước khối nhỏ nhất đạt được. Các khối mà không còn được phân chia cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng việc phân chia thành hai phần chia được gọi là cây nhị phân (BT), cây sử dụng việc phân chia thành ba phần chia được gọi là cây tam phân (TT), và cây sử dụng việc phân chia thành bốn phần chia được gọi là cây tứ phân (QT).

Như được nêu trên, thuật ngữ “khối” như được sử dụng ở đây có thể là phần, cụ thể là phần hình vuông hoặc hình chữ nhật, của hình ảnh. Ví dụ, dựa vào HEVC và VVC, khối có thể là hoặc tương ứng với đơn vị cây mã hóa (CTU), đơn vị mã hóa (coding unit - CU), đơn vị dự đoán (prediction unit - PU), và đơn vị biến đổi (transform unit - TU) và/hoặc với các khối tương ứng, ví dụ, khối cây mã hóa (CTB), khối mã hóa (coding block - CB), khối biến đổi (transform block - TB) hoặc khối dự báo (prediction block - PB).

Ví dụ, đơn vị cây mã hóa (CTU) có thể là hoặc bao gồm CTB của các mẫu độ chói, hai CTB tương ứng của các mẫu sắc độ của hình ảnh mà có ba mảng mẫu, hoặc CTB của các mẫu của hình ảnh đơn sắc hoặc hình ảnh mà được mã hóa nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu. Một cách tương ứng, khối cây mã hóa (CTB) có thể là khối $N \times N$ của các mẫu đối với một số trị số của N sao cho việc chia bộ phận thành các CTB là phần chia. Đơn vị mã hóa (CU) có thể là hoặc bao gồm khối mã hóa của các mẫu độ chói, hai khối mã hóa tương ứng của các mẫu sắc độ của hình ảnh mà có ba mảng mẫu, hoặc khối mã hóa của các mẫu của hình ảnh đơn sắc hoặc hình ảnh mà được mã hóa nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu. Một cách tương ứng khối mã hóa (CB) có thể là khối $M \times N$ của các mẫu đối với một số trị số của M và N sao cho việc chia CTB thành các khối mã hóa là phần chia.

Theo các phương án, ví dụ, theo HEVC, đơn vị cây mã hóa (CTU) có thể được phân tách thành các CU nhờ sử dụng cấu trúc cây tứ phân được biểu thị là cây mã hóa. Quyết định xem có mã hóa vùng hình ảnh nhờ sử dụng việc dự báo hình ảnh liên kết (thời gian) hoặc hình ảnh bên trong (không gian) được thực hiện ở mức CU hay không. Mỗi CU còn có thể được phân tách thành một, hai hoặc bốn PU theo loại phân tách PU. Bên trong một PU, quy trình dự báo giống nhau được áp dụng và thông tin thích hợp được truyền tới bộ giải mã trên cơ sở PU. Sau khi thu nhận khối dư bằng cách áp dụng quy trình dự báo dựa vào loại phân tách PU, CU có thể được phân chia thành các đơn vị biến đổi (TU) theo cấu trúc cây tứ phân khác giống với cây mã hóa cho CU.

Theo các phương án, ví dụ, theo chuẩn mã hóa video mới nhất hiện nay đang phát triển, mà được gọi là mã hóa video đa năng (VVC), việc phân chia cây tứ phân và cây nhị phân (Quad-tree and binary tree - QTBT) được sử dụng để phân chia khối mã hóa. Trong cấu trúc khối QTBT, CU có thể có hoặc dạng hình vuông hoặc hình chữ nhật. Ví dụ, đơn vị cây mã hóa (CTU) đầu tiên được phân chia bởi cấu trúc cây tứ phân. Các nút lá tứ phân còn được phân chia bởi cấu trúc cây nhị phân hoặc bậc ba (tam phân (triple)). Các nút lá cây phân chia được gọi là các đơn vị mã hóa (các

CU), và sự phân đoạn đó được sử dụng cho việc dự báo và xử lý biến đổi mà không phân chia thêm bất kỳ. Điều này nghĩa là CU, PU và TU có cùng kích thước khối trong cấu trúc khối mã hóa QTBT. Song song, nhiều phần chia, ví dụ, phần chia cây tam phân cũng đã được đề xuất được sử dụng cùng với cấu trúc khối QTBT.

Theo một ví dụ, bộ phận lựa chọn chế độ 260 của bộ mã hóa video 20 có thể được tạo cấu hình để thực hiện sự kết hợp bất kỳ của các kỹ thuật phân chia được mô tả ở đây.

Như được mô tả ở trên, bộ mã hóa video 20 được tạo cấu hình để xác định hoặc lựa chọn chế độ dự báo tốt nhất hoặc tối ưu từ tập hợp của các chế độ dự báo (được xác định trước). Tập hợp của các chế độ dự báo có thể bao gồm, ví dụ, các chế độ dự báo bên trong và/hoặc các chế độ dự báo liên kết.

Dự báo bên trong

Tập hợp của các chế độ dự báo bên trong có thể bao gồm 35 chế độ dự báo bên trong khác nhau, ví dụ, các chế độ không định hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ định hướng, ví dụ, như được định rõ trong HEVC, hoặc có thể bao gồm 67 chế độ dự báo bên trong khác nhau, ví dụ, các chế độ không định hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ định hướng, ví dụ, như được định rõ đối với VVC.

Bộ phận dự báo bên trong 254 được tạo cấu hình để sử dụng các mẫu được tái cấu trúc của các khối lân cận của cùng hình ảnh hiện tại để tạo ra khối dự báo bên trong 265 theo chế độ dự báo bên trong của tập hợp của các chế độ dự báo bên trong.

Bộ phận dự báo bên trong 254 (hoặc bộ phận lựa chọn chế độ 260 nói chung) còn được tạo cấu hình để đưa ra các thông số dự báo bên trong (hoặc thông tin biểu thị của chế độ dự báo bên trong được lựa chọn cho khối nói chung) tới bộ phận mã hóa entropi 270 dưới dạng của các thành phần cú pháp 266 cho việc đưa vào dữ liệu hình ảnh được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số dự báo để giải mã.

Dự báo liên kết

Tập hợp của các chế độ dự báo liên kết (hoặc khả thi) tùy thuộc vào các hình ảnh tham chiếu khả dụng (nghĩa là, các hình ảnh được giải mã ít nhất một phần trước đó, ví dụ, được lưu trữ trong DBP 230) và các thông số dự báo liên kết khác, ví dụ, xem toàn bộ hình ảnh tham chiếu hay chỉ một phần, ví dụ, vùng cửa sổ tìm kiếm xung quanh vùng của khối hiện tại, của hình ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu phù hợp nhất, và/hoặc ví dụ, xem phép nội suy điểm ảnh được áp dụng hay không, ví dụ, phép nội suy nửa điểm ảnh và/hoặc một phần tư điểm ảnh.

Ngoài các chế độ dự báo nêu trên, chế độ bỏ qua và/hoặc chế độ trực tiếp có thể được áp dụng.

Bộ phận dự báo liên kết 244 có thể bao gồm bộ phận đánh giá chuyển động (motion estimation - ME) và bộ phận bù chuyển động (motion compensation - MC) (cả hai không được thể hiện trên Fig.2). Bộ phận đánh giá chuyển động có thể được tạo cấu hình để thu hoặc thu nhận khối hình ảnh 203 (khối hình ảnh hiện tại 203 của hình ảnh hiện tại 17) và hình ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái cấu trúc trước đó, ví dụ, các khối được tái cấu trúc của một hoặc nhiều hình ảnh được giải mã trước đó khác/khác nhau 231, dùng cho việc đánh giá chuyển động. Ví dụ, chuỗi video có thể bao gồm hình ảnh hiện tại và các hình ảnh được giải mã trước đó 231, hoặc nói cách khác, hình ảnh hiện tại và các hình ảnh được giải mã trước đó 231 có thể là phần của hoặc tạo nên chuỗi các hình ảnh tạo nên chuỗi video.

Bộ mã hóa 20 có thể, ví dụ, được tạo cấu hình để lựa chọn khối tham chiếu từ các khối tham chiếu của cùng hoặc các hình ảnh khác nhau của các hình ảnh khác và cung cấp hình ảnh tham chiếu (hoặc chỉ số hình ảnh tham chiếu) và/hoặc độ dịch (độ dịch không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện tại là các thông số dự báo liên kết tới bộ phận đánh giá chuyển động. Độ dịch này cũng được gọi là vector chuyển động (motion vector - MV).

Bộ phận bù chuyển động được tạo cấu hình để thu nhận, ví dụ, thu, thông số dự báo liên kết và để thực hiện việc dự báo liên kết dựa vào hoặc nhờ sử dụng thông số dự báo liên kết để thu nhận khối dự báo liên kết 265. Việc bù chuyển động, được thực hiện bởi bộ phận bù chuyển động, có thể bao gồm việc nạp hoặc tạo ra khối dự

báo dựa vào vectơ chuyển động/khối được xác định bởi việc đánh giá chuyển động, có thể thực hiện các phép nội suy với độ chính xác điểm ảnh phụ. Việc lọc nội suy có thể tạo ra các mẫu điểm ảnh bổ sung từ các mẫu điểm ảnh đã biết, vì vậy có khả năng làm tăng số lượng các khối dự báo ứng viên mà có thể được sử dụng để mã hóa khối hình ảnh. Khi thu vectơ chuyển động cho PU của khối hình ảnh hiện tại, bộ phận bù chuyển động có thể bố trí khối dự báo trong đó vectơ chuyển động chỉ trong một trong số các danh mục hình ảnh tham chiếu.

Bộ phận bù chuyển động cũng có thể tạo ra các thành phần cú pháp được kết hợp với các khối và lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối hình ảnh của lát video.

Mã hóa entropi

Bộ phận mã hóa entropi 270 được tạo cấu hình để áp dụng, ví dụ, thuật toán hoặc sơ đồ mã hóa entropi (ví dụ, sơ đồ mã hóa độ dài biến đổi (variable length coding, viết tắt là VLC), sơ đồ VLC thích nghi ngữ cảnh (context adaptive VLC - CAVLC), sơ đồ mã hóa thuật toán, nhị phân hóa, mã hóa thuật toán nhị phân thích nghi ngữ cảnh (context adaptive binary arithmetic coding - CABAC), mã hóa thuật toán nhị phân thích nghi ngữ cảnh dựa vào cú pháp (syntax-based context-adaptive binary arithmetic coding - SBAC), mã hóa entropi phân chia khoảng xác suất (probability interval partitioning entropy - PIPE) hoặc phương pháp hoặc kỹ thuật mã hóa entropi khác) hoặc tránh qua (không nén) trên các hệ số được lượng tử hóa 209, các thông số dự báo liên kết, các thông số dự báo bên trong, các thông số bộ lọc vòng lặp và/hoặc các thành phần cú pháp khác để thu nhận dữ liệu hình ảnh được mã hóa 21 mà có thể được đưa ra qua đầu ra 272, ví dụ, dưới dạng của dòng bit được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số để giải mã. Dòng bit được mã hóa 21 có thể được truyền tới bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ dùng cho việc truyền hoặc truy xuất sau đó bởi bộ giải mã video 30.

Các biến thể cấu trúc khác của bộ mã hóa video 20 có thể được sử dụng để mã hóa dòng video. Ví dụ, bộ mã hóa dựa vào việc không biến đổi 20 có thể lượng tử

hóa trực tiếp tín hiệu dư mà không có bộ phận xử lý biến đổi 206 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ mã hóa 20 có thể có bộ phận lượng tử hóa 208 và bộ phận lượng tử hóa ngược 210 được kết hợp thành bộ phận đơn.

Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện ví dụ về bộ giải mã video 30 mà được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Bộ giải mã video 30 được tạo cấu hình để thu dữ liệu hình ảnh được mã hóa 21 (ví dụ, dòng bit được mã hóa 21), ví dụ, được mã hóa bởi bộ mã hóa 20, để thu nhận hình ảnh được giải mã 331. Dữ liệu hình ảnh được mã hóa hoặc dòng bit bao gồm thông tin để giải mã dữ liệu hình ảnh được mã hóa, ví dụ, dữ liệu mà biểu diễn các khối hình ảnh của lát video được mã hóa và các thành phần cú pháp được kết hợp.

Theo ví dụ trên Fig.3, bộ giải mã 30 bao gồm bộ phận giải mã entropi 304, bộ phận lượng tử hóa ngược 310, bộ phận xử lý biến đổi ngược 312, bộ phận tái cấu trúc 314 (ví dụ, bộ cộng 314), bộ lọc vòng lặp 320, bộ đệm hình ảnh được giải mã (DBP) 330, bộ phận dự báo liên kết 344 và bộ phận dự báo bên trong 354. Bộ phận dự báo liên kết 344 có thể là hoặc bao gồm bộ phận bù chuyển động. Bộ giải mã video 30 có thể, theo một số ví dụ, thực hiện quy trình giải mã thường ngược với quy trình mã hóa được mô tả dựa vào bộ mã hóa video 100 từ Fig.2.

Như được giải thích liên quan đến bộ mã hóa 20, bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214 bộ lọc vòng lặp 220, bộ đệm hình ảnh được giải mã (DPB) 230, bộ phận dự báo liên kết 344 và bộ phận dự báo bên trong 354 cũng được đề cập đến là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20. Theo đó, bộ phận lượng tử hóa ngược 310 có thể giống nhau về chức năng với bộ phận lượng tử hóa ngược 110, bộ phận xử lý biến đổi ngược 312 có thể giống nhau về chức năng với bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 314 có thể giống nhau về chức năng với bộ phận tái cấu trúc 214, bộ lọc vòng lặp 320 có thể giống nhau về chức năng với bộ lọc vòng lặp 220, và bộ đệm hình ảnh được giải mã 330 có thể giống nhau về chức năng với bộ đệm hình ảnh

được giải mã 230. Do đó, các phần giải thích được đưa ra đối với các bộ phận và các chức năng tương ứng của bộ mã hóa video 20 áp dụng một cách tương ứng với các bộ phận và các chức năng tương ứng của bộ giải mã video 30.

Giải mã entropi

Bộ phận giải mã entropi 304 được tạo cấu hình để tách dòng bit 21 (hoặc dữ liệu hình ảnh được mã hóa 21 nói chung) và thực hiện, ví dụ, việc giải mã entropi tới dữ liệu hình ảnh được mã hóa 21 để thu nhận, ví dụ, các hệ số được lượng tử hóa 309 và/hoặc các thông số mã hóa được giải mã (không được thể hiện trên Fig.3), ví dụ, bất kỳ hoặc tất cả các thông số dự báo liên kết (ví dụ, chỉ số hình ảnh tham chiếu và vectơ chuyển động), thông số dự báo bên trong (ví dụ, chế độ hoặc chỉ số dự báo bên trong), các thông số biến đổi, các thông số lượng tử hóa, các thông số bộ lọc vòng lặp, và/hoặc các thành phần cú pháp khác. Bộ phận giải mã entropi 304 có thể được tạo cấu hình để áp dụng các thuật toán hoặc các sơ đồ giải mã tương ứng với các sơ đồ mã hóa như được mô tả liên quan đến bộ phận mã hóa entropi 270 của bộ mã hóa 20. Bộ phận giải mã entropi 304 còn có thể được tạo cấu hình để cung cấp các thông số dự báo liên kết, thông số dự báo bên trong và/hoặc các thành phần cú pháp khác tới bộ phận lựa chọn chế độ 360 và các thông số khác tới các bộ phận khác của bộ giải mã 30. Bộ giải mã video 30 có thể thu các thành phần cú pháp ở mức lát video và/hoặc mức khối video.

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 310 có thể được tạo cấu hình để thu các thông số lượng tử hóa (QP) (hoặc thông tin liên quan đến việc lượng tử hóa ngược nói chung) và các hệ số được lượng tử hóa từ dữ liệu hình ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) và để áp dụng dựa vào các thông số lượng tử hóa việc lượng tử hóa ngược trên các hệ số được lượng tử hóa được giải mã 309 để thu nhận các hệ số được giải lượng tử hóa 311, mà cũng có thể được gọi là các hệ số biến đổi 311. Quy trình lượng tử hóa ngược có thể bao gồm việc sử dụng của thông số lượng tử hóa được xác định bởi bộ mã hóa video 20 cho

mỗi khối video trong lát video để xác định mức lượng tử hóa và, tương tự, mức lượng tử hóa ngược mà sẽ được áp dụng.

Biến đổi ngược

Bộ phận xử lý biến đổi ngược 312 có thể được tạo cấu hình để thu các hệ số được giải lượng tử hóa 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng việc biến đổi tới các hệ số được giải lượng tử hóa 311 để thu nhận các khối dư được tái cấu trúc 213 trong miền mẫu. Các khối dư được tái cấu trúc 213 cũng có thể được gọi là các khối biến đổi 313. Việc biến đổi có thể là biến đổi ngược, ví dụ, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc quy trình biến đổi ngược giống nhau về mặt khái niệm. Bộ phận xử lý biến đổi ngược 312 còn có thể được tạo cấu hình để thu các thông số biến đổi hoặc thông tin tương ứng từ dữ liệu hình ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entrôpi 304) để xác định việc biến đổi được áp dụng tới các hệ số được giải lượng tử hóa 311.

Tái cấu trúc

Bộ phận tái cấu trúc 314 (ví dụ, bộ bổ sung hoặc bộ cộng 314) có thể được tạo cấu hình để bổ sung khối dư được tái cấu trúc 313, vào khối dự báo 365 để thu nhận khối được tái cấu trúc 315 trong miền mẫu, ví dụ, bằng cách bổ sung các trị số mẫu của khối dư được tái cấu trúc 313 và các trị số mẫu của khối dự báo 365.

Lọc

Bộ phận lọc vòng lặp 320 (hoặc trong vòng lặp mã hóa hoặc sau vòng lặp mã hóa) được tạo cấu hình để lọc khối được tái cấu trúc 315 để thu nhận khối được lọc 321, ví dụ, để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận lọc vòng lặp 320 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc giải khối, bộ lọc độ dịch thích nghi mẫu (SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc hai chiều, bộ lọc vòng lặp thích nghi (ALF), làm sắc, các bộ lọc làm mịn hoặc các bộ lọc phối hợp, hoặc sự kết hợp bất kỳ của chúng. Mặc dù bộ phận lọc vòng lặp 320 được thể hiện trên Fig.3 là bộ lọc trong

vòng lặp, nhưng theo các cấu hình khác, bộ phận lọc vòng lặp 320 có thể được thực hiện như bộ lọc sau vòng lặp.

Bộ đệm hình ảnh được giải mã

Các khối video được giải mã 321 của hình ảnh sau đó được lưu trữ trong bộ đệm hình ảnh được giải mã 330, mà lưu trữ các hình ảnh được giải mã 331 như các hình ảnh tham chiếu cho việc bù chuyển động tiếp theo cho các hình ảnh khác và/hoặc cho việc đưa ra màn hình tương ứng.

Bộ giải mã 30 được tạo cấu hình để đưa ra hình ảnh được giải mã 311, ví dụ, qua đầu ra 312, nhằm biểu diễn hoặc xem tới người dùng.

Dự báo

Bộ phận dự báo liên kết 344 có thể giống với bộ phận dự báo liên kết 244 (cụ thể là bộ phận bù chuyển động) và bộ phận dự báo bên trong 354 có thể giống với bộ phận dự báo liên kết 254 về chức năng, và thực hiện các quyết định phân tách hoặc phân chia và dự báo dựa vào các thông số dự báo và/hoặc phân chia hoặc thông tin tương ứng thu được từ dữ liệu hình ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304). Bộ phận lựa chọn chế độ 360 có thể được tạo cấu hình để thực hiện việc dự báo (dự báo bên trong hoặc liên kết) trên mỗi khối dựa vào các hình ảnh được tái cấu trúc, các khối hoặc các mẫu tương ứng (được lọc hoặc không được lọc) để thu nhận khối dự báo 365.

Khi lát video được mã hóa là lát (I) được mã hóa bên trong, bộ phận dự báo bên trong 354 của bộ phận lựa chọn chế độ 360 được tạo cấu hình để tạo ra khối dự báo 365 cho khối hình ảnh của lát video hiện tại dựa vào chế độ dự báo bên trong được truyền tín hiệu và dữ liệu từ các khối được giải mã trước đó của hình ảnh hiện tại. Khi hình ảnh video được mã hóa là lát (nghĩa là, B hoặc P) được mã hóa liên kết, bộ phận dự báo liên kết 344 (ví dụ, bộ phận bù chuyển động) của bộ phận lựa chọn chế độ 360 được tạo cấu hình để tạo ra các khối dự báo 365 cho khối video của lát video hiện tại dựa vào các vectơ chuyển động và các thành phần cú pháp khác thu được từ bộ phận giải mã entropi 304. Đối với việc dự báo liên kết, các khối dự báo có thể được tạo ra từ một trong số các hình ảnh tham chiếu nằm trong một trong số

các danh mục hình ảnh tham chiếu. Bộ giải mã video 30 có thể cấu trúc các danh mục khung tham chiếu, danh mục 0 và danh mục 1, nhờ sử dụng các kỹ thuật cấu trúc mặc định dựa vào các hình ảnh tham chiếu được lưu trữ trong DPB 330.

Bộ phận lựa chọn chế độ 360 được tạo cấu hình để xác định thông tin dự báo cho khối video của lát video hiện tại bằng cách tách các vectơ chuyển động và các thành phần cú pháp khác, và sử dụng thông tin dự báo để tạo ra các khối dự báo cho khối video hiện tại được giải mã. Ví dụ, bộ phận lựa chọn chế độ 360 sử dụng một vài trong số các thành phần cú pháp thu được để xác định chế độ dự báo (ví dụ, dự báo bên trong hoặc liên kết) được sử dụng để mã hóa các khối video của lát video, loại lát dự báo liên kết (ví dụ, lát B, lát P, hoặc lát GPB), thông tin cấu trúc cho một hoặc nhiều trong số các danh mục hình ảnh tham chiếu cho lát, các vectơ chuyển động cho mỗi khối video được mã hóa liên kết của lát, trạng thái dự báo liên kết cho mỗi khối video được mã hóa liên kết của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại.

Các biến thể khác của bộ giải mã video 30 có thể được sử dụng để giải mã dữ liệu hình ảnh được mã hóa 21. Ví dụ, bộ giải mã 30 có thể tạo ra dòng video đầu ra mà không có bộ phận lọc vòng lặp 320. Ví dụ, bộ giải mã dựa vào việc không biến đổi 30 có thể lượng tử hóa ngược trực tiếp tín hiệu dư mà không có bộ phận xử lý biến đổi ngược 312 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ giải mã video 30 có thể có bộ phận lượng tử hóa ngược 310 và bộ phận xử lý biến đổi ngược 312 được kết hợp thành bộ phận đơn.

Sẽ được hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại còn có thể được xử lý và sau đó được đưa ra tới bước tiếp theo. Ví dụ, sau khi lọc nội suy, việc dẫn ra vectơ chuyển động hoặc lọc vòng lặp, phép toán khác, chẳng hạn như cắt hoặc dịch chuyển, có thể được thực hiện trên kết quả xử lý của việc lọc nội suy, việc dẫn ra vectơ chuyển động hoặc lọc vòng lặp.

Cần lưu ý rằng các phép toán khác có thể được áp dụng tới các vectơ chuyển động được dẫn ra của khối hiện tại (bao gồm nhưng không giới hạn ở các vectơ chuyển động điểm điều khiển của chế độ afin, các vectơ chuyển động khối con ở các

chế độ afin, hai chiều, ATMVP, các vectơ chuyển động thời gian, và v.v.). Ví dụ, trị số của vectơ chuyển động được hạn chế ở khoảng được định rõ trước theo bit biểu diễn của nó. Nếu bit biểu diễn của vectơ chuyển động là bitDepth, thì sau đó khoảng là $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)-1}$, trong đó “^” nghĩa là lũy thừa. Ví dụ, nếu bitDepth được thiết đặt bằng 16, thì khoảng là $-32768 \sim 32767$; nếu bitDepth được thiết đặt bằng 18, thì khoảng là $-131072 \sim 131071$. Ví dụ, trị số của vectơ chuyển động được dẫn ra (ví dụ, các MV của bốn khối con 4x4 nằm trong một khối 8x8) được hạn chế sao cho độ chênh lệch lớn nhất giữa các phần số nguyên của bốn MV khối con 4x4 không nhiều hơn N điểm ảnh, chẳng hạn như không nhiều hơn 1 điểm ảnh. Ở đây đề xuất hai phương pháp hạn chế vectơ chuyển động theo bitDepth.

Phương pháp 1: loại bỏ MSB tràn (bit quan trọng nhất) bởi các phép toán sau đây

$$ux = (mvx + 2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

$$uy = (mvy + 2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

$$mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

trong đó mvx là thành phần theo chiều ngang của vectơ chuyển động của khối ảnh hoặc khối con, mvy là thành phần theo chiều thẳng đứng của vectơ chuyển động của khối ảnh hoặc khối con, và ux và uy chỉ báo trị số trung gian.

Ví dụ, nếu trị số của mvx là -32769, sau khi áp dụng công thức (1) và (2), trị số kết quả là 32767. Trong hệ thống máy tính, các số thập phân được lưu trữ là bù hai. Bù hai của -32769 là 1,0111,1111,1111,1111 (17 bit), sau đó MSB được loại bỏ, vì vậy bù hai kết quả là 0111,1111,1111,1111 (số thập phân là 32767), mà giống như đầu ra bằng cách áp dụng công thức (1) và (2).

$$ux = (mvp_x + mvd_x + 2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

Các phép toán có thể được áp dụng trong suốt thời gian tổng của.mvp và.mvd, như được thể hiện trong công thức (5) đến (8).

Phương pháp 2: loại bỏ MSB tràn bằng cách cắt trị số

$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vx)$$

$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vy)$$

trong đó vx là thành phần theo chiều ngang của vectơ chuyển động của khối ảnh hoặc khối con, vy là thành phần theo chiều thẳng đứng của vectơ chuyển động của khối ảnh hoặc khối con; x, y và z lần lượt tương ứng với ba trị số đầu vào của quy trình cắt MV, và định nghĩa của phương trình Clip3 như sau:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{theo cách khác} \end{cases}$$

Fig.4 là hình vẽ giản lược của thiết bị mã video 400 theo phương án của sáng chế. Thiết bị mã video 400 thích hợp cho việc thực hiện các phương án được bộc lộ như được mô tả ở đây. Theo phương án, thiết bị mã video 400 có thể là bộ giải mã chẳng hạn như bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa chẳng hạn như bộ mã hóa video 20 trên Fig.1A.

Thiết bị mã video 400 bao gồm các cổng vào 410 (hoặc các cổng đầu vào 410) và các bộ phận thu (Rx) 420 để thu dữ liệu; bộ xử lý, bộ phận logic, hoặc bộ xử lý trung tâm (central processing unit - CPU) 430 để xử lý dữ liệu; các bộ phận truyền (Tx) 440 và các cổng ra 450 (hoặc các cổng đầu ra 450) để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị mã video 400 cũng có thể bao gồm các thành phần quang-điện (optical-to-electrical - OE) và các thành phần điện-quang (electrical-to-optical - EO) được ghép nối với các cổng vào 410, các bộ phận thu 420, các bộ phận truyền 440, và các cổng ra 450 dùng cho việc ra hoặc vào của các tín hiệu quang hoặc điện.

Bộ xử lý 430 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 430 có thể được thực hiện như một hoặc nhiều chip CPU, các lõi (ví dụ, như bộ xử lý nhiều

lỗi), các FPGA, các ASIC, và các DSP. Bộ xử lý 430 truyền thông với các cổng vào 410, các bộ phận thu 420, các bộ phận truyền 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun mã 470. Môđun mã 470 thực hiện các phương án được bộc lộ nêu trên. Chẳng hạn, môđun mã 470 thực hiện, xử lý, chuẩn bị, hoặc đưa ra các thao tác mã khác nhau. Việc bao gồm môđun mã 470 do đó cung cấp sự nâng cao đáng kể tới tính năng của thiết bị mã video 400 và ảnh hưởng việc biến đổi của thiết bị mã video 400 tới trạng thái khác. Theo cách khác, môđun mã 470 được thực hiện như các lệnh được lưu trữ trong bộ nhớ 460 và được thực thi bởi bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, nhiều ổ đĩa băng, và nhiều ổ đĩa trạng thái rắn và có thể được sử dụng như thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình như vậy được lựa chọn cho việc thực thi, và để lưu trữ các lệnh và dữ liệu mà được đọc trong suốt thời gian thực thi chương trình. Bộ nhớ 460 có thể, ví dụ, khả biến và/hoặc bất khả biến và có thể là bộ nhớ chỉ đọc (read-only memory - ROM), bộ nhớ truy cập ngẫu nhiên (random access memory - RAM), bộ nhớ nội dung địa chỉ bậc ba (ternary content-addressable memory - TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (static random-access memory - SRAM).

Fig.5 là sơ đồ khối được đơn giản hóa của máy 500 mà có thể được sử dụng như một trong hai hoặc cả thiết bị nguồn 12 và thiết bị đích 14 từ Fig.1A theo phương án làm ví dụ.

Bộ xử lý 502 trong máy 500 có thể là bộ xử lý trung tâm. Theo cách khác, bộ xử lý 502 có thể là loại thiết bị khác bất kỳ, hoặc nhiều thiết bị, có khả năng điều khiển hoặc xử lý thông tin tồn tại hiện thời hoặc được phát triển sau này. Mặc dù các cách thực hiện được bộc lộ có thể được thực hiện với bộ xử lý đơn như được thể hiện, ví dụ, bộ xử lý 502, các ưu điểm về tốc độ và hiệu quả có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong máy 500 có thể là thiết bị bộ nhớ chỉ đọc (ROM) hoặc thiết bị bộ nhớ truy cập ngẫu nhiên (RAM) theo cách thực hiện. Loại thiết bị lưu trữ thích hợp khác bất kỳ có thể được sử dụng như bộ nhớ 504. Bộ nhớ 504 có thể bao gồm

mã và dữ liệu 506 mà được truy cập bởi bộ xử lý 502 nhờ sử dụng bus 512. Bộ nhớ 504 còn có thể bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình mà cho phép bộ xử lý 502 thực hiện các phương pháp được mô tả ở đây. Ví dụ, các chương trình ứng dụng 510 có thể bao gồm ứng dụng 1 qua N, mà bao gồm ứng dụng mã hóa video mà thực hiện các phương pháp được mô tả ở đây.

Máy 500 cũng có thể bao gồm một hoặc nhiều thiết bị đầu ra, chẳng hạn như màn hình 518. Màn hình 518 có thể là, theo một ví dụ, màn hình nhạy chạm mà kết hợp màn hình với thành phần nhạy chạm mà có thể thao tác được để cảm biến các đầu vào chạm. Màn hình 518 có thể được ghép nối với bộ xử lý 502 qua bus 512.

Mặc dù được mô tả ở đây là bus đơn, nhưng bus 512 của máy 500 có thể bao gồm nhiều bus. Hơn nữa, thiết bị lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các bộ phận còn lại của máy 500 hoặc có thể được truy cập qua mạng và có thể bao gồm bộ phận tích hợp đơn chẳng hạn như thẻ nhớ hoặc nhiều bộ phận chẳng hạn như nhiều thẻ nhớ. Máy 500 vì vậy có thể được thực hiện theo nhiều cấu hình khác nhau.

Khái quát bộ lọc trong vòng lặp

Có tổng cộng ba bộ lọc trong vòng lặp trong VTM3 (Mô hình thử nghiệm VVC 3). Ngoài bộ lọc giải khối và SAO (hai bộ lọc vòng lặp trong HEVC), bộ lọc vòng lặp thích nghi (ALF) được áp dụng trong VTM3. Thứ tự của quy trình lọc trong VTM3 là bộ lọc giải khối, SAO và ALF.

Khái quát ALF

Trong VTM3, bộ lọc vòng lặp thích nghi (ALF) với việc thích nghi bộ lọc dựa vào khối được áp dụng. Đối với thành phần độ chói, một trong số 25 bộ lọc được lựa chọn cho mỗi khối 4×4 , dựa vào hướng và hoạt động của các độ dốc cục bộ.

Hình dạng bộ lọc:

Trong JEM, hai hình dạng bộ lọc kim cương (như được thể hiện trên Fig.6) được sử dụng cho thành phần độ chói. Hình dạng kim cương 7×7 được áp dụng cho

thành phần độ chói và hình dạng kim cương 5×5 được áp dụng cho thành phần sắc độ.

Phân loại khối:

Đối với thành phần độ chói, mỗi khối 4×4 được phân loại thành một trong số 25 loại. Chỉ số phân loại C được dẫn ra dựa vào định hướng của nó D và trị số được lượng tử hóa của hoạt động \hat{A} , như sau:

$$C = 5D + \hat{A} \quad (9)$$

Để tính toán D và \hat{A} , các độ dốc của hướng theo chiều ngang, theo chiều thẳng đứng và theo hai đường chéo đầu tiên được tính toán nhờ sử dụng 1-D Laplacian:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad (10)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad (11)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (12)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (13)$$

Trong đó các chỉ số i và j đề cập đến các tọa độ của mẫu trên cùng bên trái nằm trong khối 4×4 và $R(i,j)$ chỉ báo mẫu được tái cấu trúc ở tọa độ (i,j) .

Để làm giảm độ phức tạp của việc phân loại khối, việc tính toán 1-D Laplacian được lấy mẫu phụ được áp dụng. Như được thể hiện trên Fig.7, các vị trí được lấy mẫu phụ giống nhau được sử dụng cho việc tính toán độ dốc của tất cả các hướng.

Sau đó các trị số lớn nhất và nhỏ nhất D của các độ dốc của các hướng theo

chiều thẳng đứng và theo chiều ngang được thiết đặt là:

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v) \quad (14)$$

Các trị số lớn nhất và nhỏ nhất của độ dốc của hai hướng theo đường chéo được thiết đặt là:

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}) \quad (15)$$

Để dẫn ra trị số của định hướng D , các trị số này được so với nhau và với hai ngưỡng t_1 và t_2 :

Bước 1. Nếu cả $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ và $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ là đúng, D được thiết đặt tới 0.

Bước 2. Nếu $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, tiếp tục từ bước 3; theo cách khác tiếp tục từ bước 4.

Bước 3. Nếu $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D được thiết đặt tới 2; theo cách khác D được thiết đặt tới 1.

Bước 4. Nếu $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D được thiết đặt tới 4; theo cách khác D được thiết đặt tới 3.

Trị số hoạt động A được tính toán là:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \quad (16)$$

A được lượng tử hóa thêm tới khoảng từ 0 đến 4, theo cách bao hàm, và trị số được lượng tử hóa được biểu thị là \hat{A} .

Đối với các thành phần sắc độ trong hình ảnh, không có phương pháp phân loại nào được áp dụng, nghĩa là, tập hợp đơn của các hệ số ALF được áp dụng cho mỗi thành phần sắc độ.

Các phép biến đổi hình học của các hệ số bộ lọc

Trước khi lọc mỗi khối độ chói 4×4 , các phép biến đổi hình học chẳng hạn như quay hoặc lật theo đường chéo hoặc theo chiều thẳng đứng được áp dụng tới các hệ

số bộ lọc $f(k, l)$ tùy thuộc vào các trị số độ dốc được tính toán cho khối đó. Điều này tương đương với việc áp dụng các phép biến đổi này tới các mẫu trong vùng hỗ trợ bộ lọc. Ý tưởng là làm cho các khối khác nhau trong đó ALF được áp dụng giống nhau hơn bằng cách căn chỉnh định hướng của chúng.

Ba phép biến đổi hình học, bao gồm việc quay và lật theo chiều thẳng đứng, theo đường chéo được giới thiệu:

$$\text{Đường chéo: } f_D(k, l) = f(l, k) \quad (17)$$

$$\text{Lật theo chiều thẳng đứng: } f_V(k, l) = f(k, K - l - 1) \quad (18)$$

$$\text{Quay: } f_R(k, l) = f(K - l - 1, k) \quad (19)$$

trong đó K là kích thước của bộ lọc và $0 \leq k, l \leq K - 1$ là các tọa độ các hệ số, sao cho vị trí $(0,0)$ ở góc trên cùng bên trái và vị trí $(K - 1, K - 1)$ ở góc dưới cùng bên phải. Các phép biến đổi được áp dụng tới các hệ số bộ lọc $f(k, l)$ tùy thuộc vào các trị số độ dốc được tính toán cho khối đó. Mối tương quan giữa phép biến đổi và bốn độ dốc của bốn hướng được tóm tắt trong bảng sau đây.

Bảng 1 Việc ánh xạ của độ dốc được tính toán cho một khối và các phép biến đổi

| Các trị số độ dốc | Phép biến đổi |
|----------------------------------|---------------------------|
| $g_{d2} < g_{d1}$ và $g_h < g_v$ | Không có phép biến đổi |
| $g_{d2} < g_{d1}$ và $g_v < g_h$ | Đường chéo |
| $g_{d1} < g_{d2}$ và $g_h < g_v$ | Lật theo chiều thẳng đứng |
| $g_{d1} < g_{d2}$ và $g_v < g_h$ | Quay |

Truyền tín hiệu các thông số bộ lọc

Trong VTM3, các thông số bộ lọc ALF được truyền tín hiệu ở đoạn đầu lát. Lên tới 25 tập hợp của các hệ số bộ lọc độ chói có thể được truyền tín hiệu. Để làm giảm phí tổn các bit, các hệ số bộ lọc của sự phân loại khác có thể được hợp nhất.

Quy trình lọc có thể được điều khiển ở mức CTB. Cờ luôn được truyền tín

hiệu để chỉ báo xem ALF được áp dụng tới CTB độ chói hay không. Đối với mỗi CTB sắc độ, còn có thể được truyền tín hiệu để chỉ báo xem ALF được áp dụng tới CTB sắc độ tùy thuộc vào trị số của `alf_chroma_ctb_present_flag` hay không.

Các hệ số bộ lọc được lượng tử hóa với định mức bằng 128. Để giới hạn thêm độ phức tạp phép nhân, sự phù hợp dòng bit được áp dụng mà trị số hệ số của vị trí trung tâm sẽ trong khoảng từ 0 đến 2^8 và các trị số hệ số của các vị trí còn lại sẽ trong khoảng từ -2^7 đến $2^7 - 1$, theo cách bao hàm.

Quy trình lọc

Ở phía bộ giải mã, khi ALF được phép cho CTB, mỗi mẫu $R(i, j)$ nằm trong CU được lọc, dẫn đến trị số mẫu $R'(i, j)$ như được thể hiện dưới đây, trong đó L biểu thị độ dài bộ lọc, $f_{m,n}$ biểu diễn hệ số bộ lọc, và $f(k, l)$ biểu thị các hệ số bộ lọc được giải mã.

$$R'(i, j) = \left(\sum_{k=-\frac{L}{2}}^{\frac{L}{2}} \sum_{l=-\frac{L}{2}}^{\frac{L}{2}} f(k, l) \times R(i+k, j+l) + 64 \right) \gg 7 \quad (20)$$

Đặc tả cú pháp ALF theo bản mô tả VVC

Quy trình lọc vòng lặp thích nghi

1.1 Khái quát

Các đầu vào của quy trình xử lý này là các mảng mẫu hình ảnh được tái cấu trúc trước bộ lọc vòng lặp thích nghi `recPictureL`, `recPictureCb` và `recPictureCr`.

Các đầu ra của quy trình xử lý này là các mảng mẫu hình ảnh được tái cấu trúc được cải biến sau bộ lọc vòng lặp thích nghi `alfPictureL`, `alfPictureCb` và `alfPictureCr`.

Các trị số mẫu trong các mảng mẫu hình ảnh được tái cấu trúc được cải biến sau bộ lọc vòng lặp thích nghi `alfPictureL`, `alfPictureCb` và `alfPictureCr`, ban đầu được thiết đặt lần lượt bằng các trị số mẫu trong các mảng mẫu hình ảnh được tái cấu trúc trước bộ lọc vòng lặp thích nghi `recPictureL`, `recPictureCb` và `recPictureCr`.

Khi trị số của `tile_group_alf_enabled_flag` bằng 1, đối với mọi đơn vị cây mã

hóa với vị trí khối cây mã hóa độ chói (rx, ry), trong đó $rx = 0..PicWidthInCtbs - 1$ và $ry = 0..PicHeightInCtbs - 1$, quy trình xử lý sau đây được áp dụng:

Khi trị số của $alf_ctb_flag[0][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu độ chói như được quy định trong khoản 1.2 được truy xuất với $recPictureL$, $alfPictureL$, và vị trí khối cây mã hóa độ chói ($xCtb, yCtb$) được thiết đặt bằng ($rx \ll CtbLog2SizeY, ry \ll CtbLog2SizeY$) là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureL$.

Khi trị số của $alf_ctb_flag[1][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu sắc độ như được quy định trong khoản 1.1 được truy xuất với $recPicture$ được thiết đặt bằng $recPictureCb$, $alfPicture$ được thiết đặt bằng $alfPictureCb$, và vị trí khối cây mã hóa sắc độ ($xCtbC, yCtbC$) được thiết đặt bằng ($rx \ll (CtbLog2SizeY - 1), ry \ll (CtbLog2SizeY - 1)$) là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureCb$.

Khi trị số của $alf_ctb_flag[2][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu sắc độ như được quy định trong khoản 1.4 được truy xuất với $recPicture$ được thiết đặt bằng $recPictureCr$, $alfPicture$ được thiết đặt bằng $alfPictureCr$, và vị trí khối cây mã hóa sắc độ ($xCtbC, yCtbC$) được thiết đặt bằng ($rx \ll (CtbLog2SizeY - 1), ry \ll (CtbLog2SizeY - 1)$) là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureCr$.

1.2 Quy trình lọc khối cây mã hóa đối với các mẫu độ chói

Các đầu vào của quy trình xử lý này là:

mảng mẫu hình ảnh độ chói được tái cấu trúc $recPictureL$ trước quy trình lọc vòng lặp thích nghi,

mảng mẫu hình ảnh độ chói được tái cấu trúc được lọc $alfPictureL$,

vị trí độ chói ($xCtb, yCtb$) xác định mẫu trên cùng bên trái của khối cây mã hóa độ chói hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu hình ảnh độ chói được tái cấu trúc được lọc được cải biến $alfPictureL$.

Quy trình dẫn ra đối với chỉ số bộ lọc khoản 1.3 được truy xuất với vị trí (x_{Ctb} , y_{Ctb}) và mảng mẫu hình ảnh độ chói được tái cấu trúc $recPictureL$ là các đầu vào, và $filtIdx[x][y]$ và $transposeIdx[x][y]$ với $x, y = 0..CtbSizeY - 1$ là các đầu ra.

Đối với việc dẫn ra của các mẫu độ chói được tái cấu trúc được lọc $alfPictureL[x][y]$, mỗi mẫu độ chói được tái cấu trúc bên trong khối cây mã hóa độ chói hiện tại $recPictureL[x][y]$ được lọc như sau với $x, y = 0..CtbSizeY - 1$:

Mảng của các hệ số bộ lọc độ chói $f[j]$ tương ứng với bộ lọc được quy định bởi $filtIdx[x][y]$ được dẫn ra như sau với $j = 0..12$:

$$f[j] = AlfCoeffL[filtIdx[x][y]][j].$$

Các hệ số bộ lọc độ chói $filterCoeff$ được dẫn ra tùy thuộc vào $transposeIdx[x][y]$ như sau:

Nếu $transposeIndex[x][y] = 1$,

$$filterCoeff[] = \{ f[9], f[4], f[10], f[8], f[1], f[5], f[11], f[7], f[3], f[0], f[2], f[6], f[12] \}$$

Theo cách khác, nếu $transposeIndex[x][y] = 2$,

$$filterCoeff[] = \{ f[0], f[3], f[2], f[1], f[8], f[7], f[6], f[5], f[4], f[9], f[10], f[11], f[12] \}$$

Theo cách khác, nếu $transposeIndex[x][y] = 3$,

$$filterCoeff[] = \{ f[9], f[8], f[10], f[4], f[3], f[7], f[11], f[5], f[1], f[0], f[2], f[6], f[12] \}$$

Theo cách khác,

$$filterCoeff[] = \{ f[0], f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9], f[10], f[11], f[12] \}$$

Các vị trí (hx , vy) đối với mỗi trong số các mẫu độ chói tương ứng (x , y) bên trong mảng đã cho $recPicture$ của các mẫu độ chói được dẫn ra như sau:

$$hx = Clip3(0, pic_width_in_luma_samples - 1, x_{Ctb} + x)$$

$$vy = Clip3(0, pic_height_in_luma_samples - 1, y_{Ctb} + y)$$

Tổng biến được dẫn ra như sau:

$$\begin{aligned}
 \text{tổng} = & \text{filterCoeff}[0] * (\text{recPictureL}[\text{hx}, \text{vy} + 3] + \text{recPictureL}[\text{hx}, \text{vy} - 3]) + \\
 & \text{filterCoeff}[1] * (\text{recPictureL}[\text{hx} + 1, \text{vy} + 2] + \text{recPictureL}[\text{hx} - 1, \text{vy} - \\
 & 2]) + \\
 & \text{filterCoeff}[2] * (\text{recPictureL}[\text{hx}, \text{vy} + 2] + \text{recPictureL}[\text{hx}, \text{vy} - 2]) + \\
 & \text{filterCoeff}[3] * (\text{recPictureL}[\text{hx} - 1, \text{vy} + 2] + \text{recPictureL}[\text{hx} + 1, \text{vy} - \\
 & 2]) + \\
 & \text{filterCoeff}[4] * (\text{recPictureL}[\text{hx} + 2, \text{vy} + 1] + \text{recPictureL}[\text{hx} - 2, \text{vy} - \\
 & 1]) + \\
 & \text{filterCoeff}[5] * (\text{recPictureL}[\text{hx} + 1, \text{vy} + 1] + \text{recPictureL}[\text{hx} - 1, \text{vy} - \\
 & 1]) + \\
 & \text{filterCoeff}[6] * (\text{recPictureL}[\text{hx}, \text{vy} + 1] + \text{recPictureL}[\text{hx}, \text{vy} - 1]) + \\
 & \text{filterCoeff}[7] * (\text{recPictureL}[\text{hx} - 1, \text{vy} + 1] + \text{recPictureL}[\text{hx} + 1, \text{vy} - \\
 & 1]) + \\
 & \text{filterCoeff}[8] * (\text{recPictureL}[\text{hx} - 2, \text{vy} + 1] + \text{recPictureL}[\text{hx} + 2, \text{vy} - \\
 & 1]) + \\
 & \text{filterCoeff}[9] * (\text{recPictureL}[\text{hx} + 3, \text{vy}] + \text{recPictureL}[\text{hx} - 3, \text{vy}]) + \\
 & \text{filterCoeff}[10] * (\text{recPictureL}[\text{hx} + 2, \text{vy}] + \text{recPictureL}[\text{hx} - 2, \text{vy}]) + \\
 & \text{filterCoeff}[11] * (\text{recPictureL}[\text{hx} + 1, \text{vy}] + \text{recPictureL}[\text{hx} - 1, \text{vy}]) + \\
 & \text{filterCoeff}[12] * \text{recPictureL}[\text{hx}, \text{vy}]
 \end{aligned}$$

$$\text{tổng} = (\text{tổng} + 64) \gg 7$$

Mẫu hình ảnh độ chói được tái cấu trúc được lọc được cải biến $\text{alfPictureL}[\text{xCtb} + \text{x}][\text{yCtb} + \text{y}]$ được dẫn ra như sau:

$$\text{alfPictureL}[\text{xCtb} + \text{x}][\text{yCtb} + \text{y}] = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, \text{sum}).$$

1.3 Quy trình dẫn ra đối với chỉ số bộ lọc và chuyển vị ALF cho các mẫu độ chói

Các đầu vào của quy trình xử lý này là:

vị trí độ chói $(\text{xCtb}, \text{yCtb})$ xác định mẫu trên cùng bên trái của khối cây mã hóa độ

chói hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại, mảng mẫu hình ảnh độ chói được tái cấu trúc `recPictureL` trước quy trình lọc vòng lặp thích nghi.

Các đầu ra của quy trình xử lý này là

mảng chỉ số bộ lọc phân loại `filtIdx[x][y]` với $x, y = 0..CtbSizeY - 1$,

mảng chỉ số chuyển vị `transposeIdx[x][y]` với $x, y = 0..CtbSizeY - 1$.

Các vị trí (hx, vy) đối với mỗi trong số các mẫu độ chói tương ứng (x, y) bên trong mảng đã cho `recPicture` của các mẫu độ chói được dẫn ra như sau:

$$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x)$$

$$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y)$$

Mảng chỉ số bộ lọc phân loại `filtIdx` và mảng chỉ số chuyển vị `transposeIdx` được dẫn ra bởi các bước theo thứ tự sau đây:

Các biến `filtH[x][y]`, `filtV[x][y]`, `filtD0[x][y]` và `filtD1[x][y]` với $x, y = -2..CtbSizeY + 1$ được dẫn ra như sau:

Nếu cả x và y là các số chẵn hoặc cả x và y là các số không chẵn, thì phần sau đây áp dụng:

$$\text{filtH}[x][y] = \text{Abs}((\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y] - \text{recPicture}[hxCtb+x+1, vyCtb+y])$$

$$\text{filtV}[x][y] = \text{Abs}((\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x, vyCtb+y-1] - \text{recPicture}[hxCtb+x, vyCtb+y+1])$$

$$\text{filtD0}[x][y] = \text{Abs}((\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y-1] - \text{recPicture}[hxCtb+x+1, vyCtb+y+1])$$

$$\text{filtD1}[x][y] = \text{Abs}((\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x+1, vyCtb+y-1] - \text{recPicture}[hxCtb+x-1, vyCtb+y+1])$$

Theo cách khác, `filtH[x][y]`, `filtV[x][y]`, `filtD0[x][y]` và `filtD1[x][y]` được thiết đặt bằng 0.

Các biến $\text{varTempH1}[x][y]$, $\text{varTempV1}[x][y]$, $\text{varTempD01}[x][y]$, $\text{varTempD11}[x][y]$ và $\text{varTemp}[x][y]$ với $x, y = 0..(\text{CtbSizeY} - 1) \gg 2$ được dẫn ra như sau:

$$\text{sumH}[x][y] = \sum_i \sum_j \text{filtH}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumV}[x][y] = \sum_i \sum_j \text{filtV}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD0}[x][y] = \sum_i \sum_j \text{filtD0}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD1}[x][y] = \sum_i \sum_j \text{filtD1}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumOfHV}[x][y] = \text{sumH}[x][y] + \text{sumV}[x][y]$$

Các biến $\text{dir1}[x][y]$, $\text{dir2}[x][y]$ và $\text{dirS}[x][y]$ với $x, y = 0..\text{CtbSizeY} - 1$ được dẫn ra như sau:

Các biến hv1 , hv0 và dirHV được dẫn ra như sau:

Nếu $\text{sumV}[x \gg 2][y \gg 2]$ lớn hơn $\text{sumH}[x \gg 2][y \gg 2]$, phần sau đây áp dụng:

$$\text{hv1} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 1$$

Theo cách khác, phần sau đây áp dụng:

$$\text{hv1} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 3$$

Các biến d1 , d0 và dirD được dẫn ra như sau:

Nếu $\text{sumD0}[x \gg 2][y \gg 2]$ lớn hơn $\text{sumD1}[x \gg 2][y \gg 2]$, phần sau đây áp dụng:

$$\text{d1} = \text{sumD0}[x \gg 2][y \gg 2]$$

$$\text{d0} = \text{sumD1}[x \gg 2][y \gg 2]$$

dirD = 0

Theo cách khác, phần sau đây áp dụng:

d1 = sumD1[x >> 2][y >> 2]

d0 = sumD0[x >> 2][y >> 2]

dirD = 2

Các biến hvd1, hvd0, được dẫn ra như sau:

hvd1 = (d1 * hv0 > hv1 * d0) ? d1 : hv1

hvd0 = (d1 * hv0 > hv1 * d0) ? d0 : hv0

Các biến dirS[x][y], dir1[x][y] và dir2[x][y] được dẫn ra như sau:

dir1[x][y] = (d1 * hv0 > hv1 * d0) ? dirD : dirHV

dir2[x][y] = (d1 * hv0 > hv1 * d0) ? dirHV : dirD

dirS[x][y] = (hvd1 > 2 * hvd0) ? 1 : ((hvd1 * 2 > 9 * hvd0) ? 2 : 0)

Biến avgVar[x][y] với x, y = 0..CtbSizeY - 1 được dẫn ra như sau:

varTab[] = { 0, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4 }

avgVar[x][y] = varTab[Clip3(0, 15, (sumOfHV[x >> 2][y >> 2] * 64) >> (3 + BitDepthY))]

Mảng chỉ số bộ lọc phân loại filtIdx[x][y] và mảng chỉ số chuyển vị transposeIdx[x][y] với x = y = 0..CtbSizeY - 1 được dẫn ra như sau:

transposeTable[] = { 0, 1, 0, 2, 2, 3, 1, 3 }

transposeIdx[x][y] = transposeTable[dir1[x][y] * 2 + (dir2[x][y] >> 1)]

filtIdx[x][y] = avgVar[x][y]

Khi dirS[x][y] không bằng 0, filtIdx[x][y] được cải biến như sau:

filtIdx[x][y] += (((dir1[x][y] & 0x1) << 1) + dirS[x][y]) * 5.

1.4 Quy trình lọc khối cây mã hóa đối với các mẫu sắc độ

Các đầu vào của quy trình xử lý này là:

mảng mẫu hình ảnh sắc độ được tái cấu trúc recPicture trước quy trình lọc vòng lặp thích nghi,

mảng mẫu hình ảnh sắc độ được tái cấu trúc được lọc alfPicture,

vị trí sắc độ (xCtbC, yCtbC) xác định mẫu trên cùng bên trái của khối cây mã hóa sắc độ hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu hình ảnh sắc độ được tái cấu trúc được lọc được cải biến alfPicture.

Kích thước của khối cây mã hóa sắc độ hiện tại ctbSizeC được dẫn ra như sau:

$$ctbSizeC = CtbSizeY / SubWidthC$$

Đối với việc dẫn ra của các mẫu sắc độ được tái cấu trúc được lọc alfPicture[x][y], mỗi mẫu sắc độ được tái cấu trúc bên trong khối cây mã hóa sắc độ hiện tại recPicture[x][y] được lọc như sau với $x, y = 0..ctbSizeC - 1$:

Các vị trí (hx, vy) đối với mỗi trong số các mẫu sắc độ tương ứng (x, y) bên trong mảng đã cho recPicture của các mẫu sắc độ được dẫn ra như sau:

$$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, xCtbC + x)$$

$$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, yCtbC + y)$$

Tổng biến được dẫn ra như sau:

$$\begin{aligned} \text{tổng} &= \text{AlfCoeffC}[0] * (\text{recPicture}[hx, vy + 2] + \text{recPicture}[hx, vy - 2]) + \\ &\quad \text{AlfCoeffC}[1] * (\text{recPicture}[hx + 1, vy + 1] + \text{recPicture}[hx - 1, vy - 1]) \\ &+ \\ &\quad \text{AlfCoeffC}[2] * (\text{recPicture}[hx, vy + 1] + \text{recPicture}[hx, vy - 1]) + \\ &\quad \text{AlfCoeffC}[3] * (\text{recPicture}[hx - 1, vy + 1] + \text{recPicture}[hx + 1, vy - 1]) \\ &+ \\ &\quad \text{AlfCoeffC}[4] * (\text{recPicture}[hx + 2, vy] + \text{recPicture}[hx - 2, vy]) + \\ &\quad \text{AlfCoeffC}[5] * (\text{recPicture}[hx + 1, vy] + \text{recPicture}[hx - 1, vy]) + \\ &\quad \text{AlfCoeffC}[6] * \text{recPicture}[hx, vy] \end{aligned}$$

$$\text{tổng} = (\text{tổng} + 64) \gg 7$$

Mẫu hình ảnh sắc độ được tái cấu trúc được lọc được cải biến $alfPicture[xCtbC + x][yCtbC + y]$ được dẫn ra như sau:

$$alfPicture[xCtbC + x][yCtbC + y] = Clip3(0, (1 \ll BitDepthC) - 1, sum)$$

Yêu cầu bộ đệm dòng hiện tại của ba bộ lọc vòng lặp được xếp tầng (Giải khối, SAO, ALF) của VTM-3.0 đối với thành phần độ chói được dẫn ra là 11,25 dòng và đối với thành phần sắc độ nó được dẫn ra là 6,25 dòng (xem Fig.8). Việc đáp ứng các yêu cầu bộ đệm dòng được tăng lên trên các chip phần cứng được biết đến là mối quan tâm khi bộ đệm dòng cần được lưu trữ trên bộ nhớ “trên chip”, nhờ vậy làm tăng vùng chip phần cứng. Do đó sáng chế giải quyết vấn đề làm giảm các yêu cầu bộ đệm dòng đối với VTM-3.0 ở giai đoạn lọc vòng lặp. Sáng chế này chủ yếu giải quyết việc giảm bộ đệm dòng của phương pháp “phân loại khối ALF” và “lọc ALF”.

Để làm giảm các yêu cầu bộ đệm dòng của VTM-3.0,

Việc phân loại khối ALF được thích nghi để đảm bảo rằng việc phân loại khối ALF không cần các điểm ảnh bất kỳ ở trên hoặc ở dưới ranh giới ảo (virtual boundary - VB). Các ranh giới ảo (VB) là các ranh giới CTU theo chiều ngang được dịch chuyển lên trên bởi các điểm ảnh “N”. Đối với mỗi CTU, SAO và ALF có thể xử lý các mẫu ở trên VB trước khi CTU thấp hơn đến nhưng không thể xử lý các mẫu ở dưới VB cho đến khi CTU thấp hơn đến, mà chủ yếu được gây ra bởi vì độ trễ trong việc lọc giải khối theo chiều thẳng đứng.

Trong VTM-3.0, N có thể lấy trị số nguyên dương mà lớn hơn hoặc bằng 4. Cụ thể là sáng chế cung cấp thêm chi tiết khi N=4 và khi N=6,

Để làm giảm các yêu cầu bộ đệm dòng, sáng chế chủ yếu thích nghi việc phân loại khối ALF sao cho cách mà tập hợp con của các mẫu ban đầu được sử dụng trong việc phân loại khối còn được sử dụng.

Sáng chế còn sử dụng cách tiếp cận “dịch chuyển lưới” đối với các trường hợp khi kích thước của sổ phân loại khối ALF (P x P) không là bội số nguyên của N. Trong VTM-3.0 hiện tại việc phân loại khối ALF được thực hiện đối với các khối 4

x 4. Do đó P được thiết đặt tới 4 cho VTM-3.0. Cách tiếp cận dịch chuyển lưới chủ yếu dịch chuyển cửa sổ phân loại khối ALF bởi số nguyên của các mẫu, sao cho cửa sổ phân loại khối P x P đã cho sẽ không bao giờ đi qua ranh giới ảo. Do đó đối với các mẫu ban đầu trong hình ảnh mà tương đương với “P- N”, việc phân loại khối ALF được thực hiện chỉ nhờ sử dụng cửa sổ khối 2 x 2 hoặc các dòng “P-N” thứ nhất sử dụng lại quyết định phân loại khối của khối 4 x4 bắt đầu ngay ở (“P- N” + 1)

Việc lọc ALF được thích nghi để sử dụng các phiên bản bị biến dạng của bộ lọc cho cả thành phần độ chói và sắc độ hoặc bằng cách sử dụng việc đệm hoặc bằng cách bổ sung các hệ số bộ lọc không được sử dụng vào hệ số trung tâm. Theo cách như vậy các bộ đệm dòng bổ sung không được yêu cầu cho việc lọc ALF

Cốt lõi của sáng chế là sử dụng phiên bản được cải biến của việc phân loại khối ALF ban đầu sao cho việc phân loại khối sẽ không yêu cầu các bộ đệm dòng bổ sung bất kỳ. Ngoài ra việc lọc ALF được cải biến theo cách mà nó không cần các bộ đệm bổ sung bất kỳ. Phiên bản được cải biến của việc phân loại khối chỉ sử dụng tập hợp con của các mẫu mà ban đầu được sử dụng trong việc phân loại khối ban đầu.

Như được thể hiện trên Fig.8, các yêu cầu bộ đệm dòng độ chói cho VVC là 11,25 dòng độ chói, khi ranh giới ảo được xem xét tới 4 dòng ở trên ranh giới CTU.

Phần giải thích của yêu cầu bộ đệm dòng như sau: Việc giải khối của mép theo chiều ngang chồng lấp với mép CTU không thể được thực hiện khi các quyết định và việc lọc yêu cầu các dòng K,L,M,M từ CTU thứ nhất và các dòng O,P.. từ CTU dưới cùng. Do đó việc giải khối của các mép theo chiều ngang chồng lấp với ranh giới CTU được trì hoãn cho đến khi CTU thấp hơn đến. Do đó các dòng K, L, M, N mà là các mẫu độ chói được tái cấu trúc phải được lưu trữ trên bộ đệm dòng (4 dòng). Sau đó việc lọc SAO có thể được thực hiện cho các dòng A đến J. Dòng J có thể là SAO được lọc khi việc giải khối không thay đổi các mẫu ở dòng K. Đối với việc lọc SAO của dòng K, quyết định phân loại độ dịch mép chỉ được lưu trữ trong bộ đệm dòng (mà là 0,25 dòng độ chói). Việc lọc ALF chỉ có thể được thực hiện đối với các dòng A-F. Như được thể hiện trên Fig.1, việc phân loại ALF được thực hiện đối với

mỗi khối 4×4 . Mỗi việc phân loại khối 4×4 cần cửa sổ hoạt động có kích thước 8×8 mà lần lượt cần cửa sổ 9×9 để tính toán 1d Laplacian để xác định độ dốc.

Do đó đối với việc phân loại khối của các dòng G, H, I, J cần các mẫu được lọc SAO dưới ranh giới ảo. Ngoài ra các dòng D, E, F các mẫu được lọc SAO được yêu cầu cho việc phân loại mẫu. Ngoài ra việc lọc ALF của dòng G cần ba dòng được lọc SAO D, E, F từ trên. Do đó tổng yêu cầu bộ đệm dòng như sau:

- Các dòng K-N (các điểm ảnh DF theo chiều ngang): 4
- Các dòng D-J (các điểm ảnh SAO được lọc): 7
- Các trị số phân loại độ dịch mép SAO giữa dòng J và dòng K: 0,25
- Do đó tổng số lượng của các dòng độ chói là $7 + 4 + 0,25 = 11,25$

Phân loại khối được cải biến khi ranh giới ảo là 4 dòng ở trên ranh giới CTU

Fig.9a và Fig.9b mô tả cách thức việc phân loại khối ALF được thực hiện đối với các dòng ở trên và ở dưới ranh giới ảo.

Giải pháp 1 (Phân loại khối ALF được cắt ngắn không đối xứng)

Như được thể hiện trên Fig.9a và Fig.18a, đối với khối 4×4 ngay ở trên ranh giới ảo, chỉ các mẫu mà ở trên ranh giới khối được sử dụng cho việc phân loại khối.

Như được thể hiện trên Fig.9b, đối với việc phân loại khối của khối 4×4 ngay ở dưới ranh giới ảo, chỉ các mẫu mà ở dưới ranh giới ảo được sử dụng.

Trên Fig.9a, Fig.22a (trường hợp khi khối 4×4 ở trên VB), việc phân loại khối đối với khối 4×4 bắt đầu ở dòng G, sử dụng các dòng D, E, F, F, H, J, J. Do việc tính toán độ dốc của các mẫu thuộc về dòng J dòng cần nhiều hơn một dòng ở dưới, do đó dòng J được đệm tới dòng K.

Hơn nữa việc tính toán hoạt động mà được sử dụng được cải biến như sau: $\text{avgVar}[x][y] = \text{varTab}[\text{Clip3}(0, 15, (\text{sumOfHV}[x \gg 2][y \gg 2] * \text{Ac}) \gg (3 + \text{BitDepth}_Y))]$ trong đó trị số Ac là 96 đối với các khối 4×4 gần ranh giới ảo và 64 đối với các khối còn lại.

Theo cách khác, đối với việc phân loại khối của khối bắt đầu ở dòng G, chỉ các dòng D, E, F, G được sử dụng. Dòng I còn có thể được sử dụng cho việc phân

loại nhưng cách thực hiện hiện tại của việc phân loại khối ALF trong VTM 3.0 tính toán các độ dốc cho mọi tập hợp của 2 dòng mà khiến nó công kênh để bao gồm dòng I trong việc phân loại.

Giải pháp thứ hai là sử dụng việc đệm đơn giản. Dòng J được đệm tới dòng K, L, M và sau đó các dòng bắt đầu từ D đến M có thể được sử dụng cho việc phân loại khối.

Trên Fig.9b, Fig.22b và Fig.18b, (trường hợp khi khối 4 x 4 ở dưới VB), việc phân loại khối đối với khối bắt đầu ở dòng K sử dụng các dòng K, L, M, N, O, P, Q. Do việc tính toán độ dốc của các mẫu thuộc về dòng K dòng cần nhiều hơn một dòng ở trên, do đó dòng K được đệm tới dòng J. Hơn nữa việc tính toán hoạt động mà được sử dụng được cải biến như sau: $\text{avgVar}[x][y] = \text{varTab}[\text{Clip3}(0, 15, (\text{sumOfHV}[x \gg 2][y \gg 2] * \text{Ac}) \gg (3 + \text{BitDepth}_Y))]$ trong đó trị số Ac là 96 đối với các khối gần ranh giới ảo và 64 đối với các khối còn lại.

Theo cách khác, đối với việc phân loại khối của khối bắt đầu ở dòng K, chỉ các dòng M, N, O, P, Q được sử dụng. Dòng L còn có thể được sử dụng cho việc phân loại tuy nhiên cách thực hiện hiện tại của việc phân loại khối ALF trong VTM 3.0 tính toán các độ dốc cho mọi tập hợp của 2 dòng mà khiến nó công kênh để bao gồm dòng L trong việc phân loại.

Giải pháp thứ hai là sử dụng việc đệm đơn giản. Dòng K được đệm tới dòng J, I, H và sau đó các dòng bắt đầu từ H đến Q có thể được sử dụng cho việc phân loại khối

Giải pháp 2 (Phân loại khối ALF được cắt ngắn đối xứng)

Như được thể hiện trên Fig.10 việc phân loại khối ALF được cắt ngắn đối xứng chỉ sử dụng 4 mẫu bên trong (a,b,c,d) để phân loại khối đã cho. Trên Fig.10a, đối với khối bắt đầu ở dòng G, chỉ các mẫu thuộc về các dòng G và I (các mẫu a, b, c, d) được sử dụng trong các phép tính độ dốc. Không có việc đệm bổ sung nào được yêu cầu trong trường hợp này. Tương tự trên Fig.10b, đối với khối bắt đầu ở dòng K, chỉ các mẫu thuộc về các dòng L và M (các mẫu a, b, c, d) được sử dụng trong các

phép tính độ dốc. Không có việc đệm bổ sung nào cũng được yêu cầu trong trường hợp này. Giải pháp này có ưu điểm là độ dốc được tính toán nhờ sử dụng hình dạng khối đối xứng mà có thể có lợi đối với chất lượng hình ảnh chủ quan cụ thể là khi nội dung hình ảnh thường thay đổi bên trong cửa sổ cục bộ đã cho. Ngoài ra giải pháp ít phức tạp hơn về mặt tính toán do nó tính toán độ dốc nhờ sử dụng chỉ 4 mẫu (a, b, c, d).

Fig.11 mô tả trường hợp đặc biệt khi việc phân loại khối ALF có thể sử dụng thêm hai dòng của bộ đệm dòng mà có thể lưu trữ các mẫu được lọc SAO. Trong trường hợp này, việc phân loại khối ALF có thể sử dụng hai dòng bổ sung cả ở trên và ở dưới ranh giới ảo đối với các khối lần lượt bắt đầu ở dòng G và dòng K.

Việc lọc ALF được cải biến khi ranh giới ảo là 4 dòng

Fig.12 mô tả việc lọc ALF được cải biến để tránh khỏi các bộ đệm dòng độ chói ALF. Đối với việc lọc của các dòng H, J, I (ở trên ranh giới ảo) và các dòng K, L, M (ở dưới ranh giới ảo), các phiên bản bị biến dạng của bộ lọc độ chói 7×7 được sử dụng.

Hai giải pháp được đề xuất. Bộ lọc loại bỏ các hệ số mà ở trên và ở dưới ranh giới ảo theo kiểu đối xứng. Ví dụ, đối với việc lọc dòng M, hệ số c_0 nằm ở trên ranh giới ảo và ngược lại đối với việc lọc dòng H. Do đó trong trường hợp này đối với giải pháp 1, hệ số được loại bỏ c_0 được bổ sung vào hệ số lân cận mà ngay theo chiều thẳng đứng. Do đó $2' = c_2 + c_0$. Theo giải pháp 2, hệ số được loại bỏ được bổ sung vào hệ số trung tâm, dẫn đến $c_{12}' = c_{12} + 2 * c_0$

Tương tự đối với việc lọc dòng L và dòng I các trị số hệ số được cải biến đối với giải pháp 1:

$$c_5' = c_5 + c_1$$

$$c_6' = c_6 + c_2 + c_0$$

$$c_7' = c_7 + c_3$$

Đối với giải pháp 2: $c_{12}' = c_{12} + 2 * c_0 + 2 * c_1 + 2 * c_2 + 2 * c_3$

Tương tự đối với việc lọc dòng J và dòng K các trị số hệ số được cải biến đối

với giải pháp 1:

$$c10' = c10 + c4 + c8$$

$$c11' = c11 + c5 + c7 + c1 + c3$$

$$c12' = c12 + 2 * c6 + 2 * c2 + 2 * c0$$

Giải pháp này cũng có thể được coi là việc đệm đối xứng như được thể hiện trên Fig.21. Như được thể hiện trên Fig 21a, đối với việc lọc của dòng M, dòng K được đệm tới dòng J, và dòng J được đệm tới dòng K.

Như được thể hiện trên Fig.21b, đối với việc lọc của dòng L, dòng K được đệm tới dòng J và dòng I. Tương tự đối với việc lọc của dòng I, dòng J được đệm tới dòng K và dòng L.

Như được thể hiện trên Fig.21c, đối với việc lọc của dòng K, dòng K được đệm tới các dòng J, I và H. Tương tự đối với việc lọc của dòng J, dòng J được đệm tới các dòng K, L, và M.

Quy trình xử lý tương tự như được nêu trên cũng được thực hiện đối với việc lọc ALF sắc độ.

Các sự thay đổi đặc tả cú pháp đối với giải pháp 1 được đưa ra dưới đây:

2.1 Khái quát

Các đầu vào của quy trình xử lý này là các mảng mẫu hình ảnh được tái cấu trúc trước bộ lọc vòng lặp thích nghi `recPictureL`, `recPictureCb` và `recPictureCr`.

Các đầu ra của quy trình xử lý này là các mảng mẫu hình ảnh được tái cấu trúc được cải biến sau bộ lọc vòng lặp thích nghi `alfPictureL`, `alfPictureCb` và `alfPictureCr`.

Các trị số mẫu trong các mảng mẫu hình ảnh được tái cấu trúc được cải biến sau bộ lọc vòng lặp thích nghi `alfPictureL`, `alfPictureCb` và `alfPictureCr`, ban đầu được thiết đặt lần lượt bằng các trị số mẫu trong các mảng mẫu hình ảnh được tái cấu trúc trước bộ lọc vòng lặp thích nghi `recPictureL`, `recPictureCb` và `recPictureCr`.

Khi trị số của `tile_group_alf_enabled_flag` bằng 1, đối với mọi đơn vị cây mã hóa với vị trí khối cây mã hóa độ chói (`rx`, `ry`), trong đó `rx = 0..PicWidthInCtbs - 1` và

$ry = 0..PicHeightInCtbs - 1$, phân sau đây áp dụng:

Khi trị số của $alf_ctb_flag[0][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu độ chói như được quy định trong khoản 2.2 được truy xuất với $recPictureL$, $alfPictureL$, và vị trí khối cây mã hóa độ chói $(xCtb, yCtb)$ được thiết đặt bằng $(rx \ll CtbLog2SizeY, ry \ll CtbLog2SizeY)$ là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureL$.

Khi trị số của $alf_ctb_flag[1][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu sắc độ như được quy định trong khoản 2.4 được truy xuất với $recPicture$ được thiết đặt bằng $recPictureCb$, $alfPicture$ được thiết đặt bằng $alfPictureCb$, và vị trí khối cây mã hóa sắc độ $(xCtbC, yCtbC)$ được thiết đặt bằng $(rx \ll (CtbLog2SizeY - 1), ry \ll (CtbLog2SizeY - 1))$ là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureCb$.

Khi trị số của $alf_ctb_flag[2][rx][ry]$ bằng 1, quy trình lọc khối cây mã hóa đối với các mẫu sắc độ như được quy định trong khoản 2.4 được truy xuất với $recPicture$ được thiết đặt bằng $recPictureCr$, $alfPicture$ được thiết đặt bằng $alfPictureCr$, và vị trí khối cây mã hóa sắc độ $(xCtbC, yCtbC)$ được thiết đặt bằng $(rx \ll (CtbLog2SizeY - 1), ry \ll (CtbLog2SizeY - 1))$ là các đầu vào, và đầu ra là hình ảnh được lọc được cải biến $alfPictureCr$.

2.2 Quy trình lọc khối cây mã hóa đối với các mẫu độ chói

Các đầu vào của quy trình xử lý này là:

mảng mẫu hình ảnh độ chói được tái cấu trúc $recPictureL$ trước quy trình lọc vòng lặp thích nghi,

mảng mẫu hình ảnh độ chói được tái cấu trúc được lọc $alfPictureL$,

vị trí độ chói $(xCtb, yCtb)$ xác định mẫu trên cùng bên trái của khối cây mã hóa độ chói hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu hình ảnh độ chói được tái cấu trúc được lọc được cải biến $alfPictureL$.

Quy trình dẫn ra đối với chỉ số bộ lọc khoản 2.3 được truy xuất với vị trí $(xCtb, yCtb)$

và mảng mẫu hình ảnh độ chói được tái cấu trúc `recPictureL` là các đầu vào, và `filtIdx[x][y]` và `transposeIdx[x][y]` với $x, y = 0..CtbSizeY - 1$ là các đầu ra.

Đối với việc dẫn ra của các mẫu độ chói được tái cấu trúc được lọc `alfPictureL[x][y]`, mỗi mẫu độ chói được tái cấu trúc bên trong khối cây mã hóa độ chói hiện tại `recPictureL[x][y]` được lọc như sau với $x, y = 0..CtbSizeY - 1$:

Mảng của các hệ số bộ lọc độ chói $f[j]$ tương ứng với bộ lọc được quy định bởi `filtIdx[x][y]` được dẫn ra như sau với $j = 0..12$:

$$f[j] = AlfCoeffL[filtIdx[x][y]][j]$$

Các hệ số bộ lọc độ chói `filterCoeff` được dẫn ra tùy thuộc vào `transposeIdx[x][y]` như sau:

Nếu `transposeIndex[x][y] == 1`,

$$filterCoeff[] = \{ f[9], f[4], f[10], f[8], f[1], f[5], f[11], f[7], f[3], f[0], f[2], f[6], f[12] \}$$

Theo cách khác, nếu `transposeIndex[x][y] == 2`,

$$filterCoeff[] = \{ f[0], f[3], f[2], f[1], f[8], f[7], f[6], f[5], f[4], f[9], f[10], f[11], f[12] \}$$

Theo cách khác, nếu `transposeIndex[x][y] == 3`,

$$filterCoeff[] = \{ f[9], f[8], f[10], f[4], f[3], f[7], f[11], f[5], f[1], f[0], f[2], f[6], f[12] \}$$

Theo cách khác,

$$filterCoeff[] = \{ f[0], f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9], f[10], f[11], f[12] \}$$

Các vị trí (hx, vy) đối với mỗi trong số các mẫu độ chói tương ứng (x, y) bên trong mảng đã cho `recPicture` của các mẫu độ chói được dẫn ra như sau:

$$hx = Clip3(0, pic_width_in_luma_samples - 1, xCtb + x)$$

$$vy = y$$

Độ dịch vị trí mẫu đầu vào bộ lọc $r = 0 \dots 6$ được dẫn ra theo vị trí mẫu độ chói y như được thể hiện trong bảng 2-2a dưới đây

Bảng 2-2a – Bản mô tả về độ dịch vị trí mẫu đầu vào bộ lọc $r=0..6$ theo vị trí mẫu độ chói y

| y | $r1$ | $r2$ | $r3$ | $r4$ | $r5$ | $r6$ |
|---|------|------|------|------|------|------|
| $y == \text{CtbSizeY} - 5$ (or) $\text{CtbSizeY} - 4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $y == \text{CtbSizeY} - 6$ (or) $\text{CtbSizeY} - 3$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $y == \text{CtbSizeY} - 7$ (or) $\text{CtbSizeY} - 2$ | 1 | 1 | 2 | 2 | 2 | 2 |
| $y_{\text{Ctb}} + y == \text{pic_height_in_luma_samples} - 1$ | 0 | 1 | 0 | 2 | 0 | 3 |
| $y_{\text{Ctb}} + y == \text{pic_height_in_luma_samples} - 2$ | 1 | 1 | 1 | 2 | 1 | 3 |
| $y_{\text{Ctb}} + y == \text{pic_height_in_luma_samples} - 3$ | 1 | 1 | 2 | 2 | 2 | 3 |
| $y_{\text{Ctb}} + y == 0$ | 1 | 0 | 2 | 0 | 3 | 0 |
| $y_{\text{Ctb}} + y == 1$ | 1 | 1 | 2 | 1 | 3 | 1 |
| $y_{\text{Ctb}} + y == 2$ | 1 | 1 | 2 | 2 | 3 | 2 |
| Các trường hợp khác | 1 | 1 | 2 | 2 | 3 | 3 |

Tổng biến được dẫn ra như sau:

$$\begin{aligned} \text{tổng} = & \text{filterCoeff}[0] * (\text{recPictureL}[hx, vy + r5] + \text{recPictureL}[hx, vy - r6]) + \\ & \text{filterCoeff}[1] * (\text{recPictureL}[hx + 1, vy + r3] + \text{recPictureL}[hx - 1, vy - \\ & r4]) + \\ & \text{filterCoeff}[2] * (\text{recPictureL}[hx, vy + r3] + \text{recPictureL}[hx, vy - r4]) + \\ & \text{filterCoeff}[3] * (\text{recPictureL}[hx - 1, vy + r3] + \text{recPictureL}[hx + 1, vy - \\ & r4]) + \\ & \text{filterCoeff}[4] * (\text{recPictureL}[hx + 2, vy + r1] + \text{recPictureL}[hx - 2, vy - \\ & r2]) + \end{aligned}$$

$$\begin{aligned}
& \text{filterCoeff}[5] * (\text{recPictureL}[\text{hx} + 1, \text{vy} + \text{r1}] + \text{recPictureL}[\text{hx} - 1, \text{vy} - \\
& \text{r2}]) + \\
& \text{filterCoeff}[6] * (\text{recPictureL}[\text{hx}, \text{vy} + \text{r1}] + \text{recPictureL}[\text{hx}, \text{vy} - \text{r2}]) + \\
& (8-1155) \\
& \text{filterCoeff}[7] * (\text{recPictureL}[\text{hx} - 1, \text{vy} + \text{r1}] + \text{recPictureL}[\text{hx} + 1, \text{vy} - \\
& \text{r2}]) + \\
& \text{filterCoeff}[8] * (\text{recPictureL}[\text{hx} - 2, \text{vy} + \text{r1}] + \text{recPictureL}[\text{hx} + 2, \text{vy} - \\
& \text{r2}]) + \\
& \text{filterCoeff}[9] * (\text{recPictureL}[\text{hx} + 3, \text{vy}] + \text{recPictureL}[\text{hx} - 3, \text{vy}]) + \\
& \text{filterCoeff}[10] * (\text{recPictureL}[\text{hx} + 2, \text{vy}] + \text{recPictureL}[\text{hx} - 2, \text{vy}]) + \\
& \text{filterCoeff}[11] * (\text{recPictureL}[\text{hx} + 1, \text{vy}] + \text{recPictureL}[\text{hx} - 1, \text{vy}]) + \\
& \text{filterCoeff}[12] * \text{recPictureL}[\text{hx}, \text{vy}]
\end{aligned}$$

tổng = (tổng + 64) >> 7

Mẫu hình ảnh độ chói được tái cấu trúc được lọc được cải biến $\text{alfPictureL}[\text{xCtb} + \text{x}][\text{yCtb} + \text{y}]$ được dẫn ra như sau:

$$\text{alfPictureL}[\text{xCtb} + \text{x}][\text{yCtb} + \text{y}] = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, \text{sum}).$$

2.3 Quy trình dẫn ra đối với chỉ số bộ lọc và chuyển vị ALF cho các mẫu độ chói

Các đầu vào của quy trình xử lý này là:

vị trí độ chói (xCtb, yCtb) xác định mẫu trên cùng bên trái của khối cây mã hóa độ chói hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại,

mảng mẫu hình ảnh độ chói được tái cấu trúc recPictureL trước quy trình lọc vòng lặp thích nghi.

Các đầu ra của quy trình xử lý này là

mảng chỉ số bộ lọc phân loại $\text{filtIdx}[\text{x}][\text{y}]$ với $\text{x}, \text{y} = 0.. \text{CtbSizeY} - 1$,

mảng chỉ số chuyển vị $\text{transposeIdx}[\text{x}][\text{y}]$ với $\text{x}, \text{y} = 0.. \text{CtbSizeY} - 1$.

Các vị trí (hx, vy) đối với mỗi trong số các mẫu độ chói tương ứng (x, y) bên trong mảng đã cho recPicture của các mẫu độ chói được dẫn ra như sau:

$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x)$

nếu $yCtb + CtbSizeY \geq \text{pic_height_in_luma_samples}$

$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y)$

Theo cách khác, nếu $y < (CtbSizeY - 4)$,

$vy = \text{Clip3}(0, yCtb + CtbSizeY - 5, yCtb + y)$

Theo cách khác,

$vy = \text{Clip3}(yCtb + CtbSizeY - 4, \text{pic_height_in_luma_samples} - 1, yCtb + y)$

Mảng chỉ số bộ lọc phân loại filtIdx và mảng chỉ số chuyển vị transposeIdx được dẫn ra bởi các bước theo thứ tự sau đây:

Các biến $\text{filtH}[x][y]$, $\text{filtV}[x][y]$, $\text{filtD0}[x][y]$ và $\text{filtD1}[x][y]$ với $x, y = -2..CtbSizeY + 1$ được dẫn ra như sau:

Nếu cả x và y là các số chẵn hoặc cả x và y là các số không chẵn, thì phần sau đây áp dụng:

$\text{filtH}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y] - \text{recPicture}[hxCtb+x+1, vyCtb+y])$

$\text{filtV}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x, vyCtb+y-1] - \text{recPicture}[hxCtb+x, vyCtb+y+1])$

$\text{filtD0}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y-1] - \text{recPicture}[hxCtb+x+1, vyCtb+y+1])$

$\text{filtD1}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x+1, vyCtb+y-1] - \text{recPicture}[hxCtb+x-1, vyCtb+y+1])$

Theo cách khác, $\text{filtH}[x][y]$, $\text{filtV}[x][y]$, $\text{filtD0}[x][y]$ và $\text{filtD1}[x][y]$ được thiết đặt bằng 0.

Các biến $\text{varTempH1}[x][y]$, $\text{varTempV1}[x][y]$, $\text{varTempD01}[x][y]$, $\text{varTempD11}[x][y]$ và $\text{varTemp}[x][y]$ với $x, y = 0..(CtbSizeY - 1) \gg 2$ được dẫn ra như sau:

Nếu $(y \ll 2) == \text{CtbSizeY} - 8$ và $y\text{Ctb} + \text{CtbSizeY} < \text{pic_height_in_luma_samples} - 1$ sau đó

$$\text{sumH}[x][y] = \sum_i \sum_j \text{filtH}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = -2..3$$

$$\text{sumV}[x][y] = \sum_i \sum_j \text{filtV}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = -2..3$$

$$\text{sumD0}[x][y] = \sum_i \sum_j \text{filtD0}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = -2..3$$

$$\text{sumD1}[x][y] = \sum_i \sum_j \text{filtD1}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = -2..3$$

biến Ac được thiết đặt là 96

Theo cách khác, nếu $(y \ll 2) == \text{CtbSizeY} - 4$ và $y\text{Ctb} + \text{CtbSizeY} < \text{pic_height_in_luma_samples} - 1$ sau đó

$$\text{sumH}[x][y] = \sum_i \sum_j \text{filtH}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = 0..5$$

$$\text{sumV}[x][y] = \sum_i \sum_j \text{filtV}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = 0..5$$

$$\text{sumD0}[x][y] = \sum_i \sum_j \text{filtD0}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = 0..5$$

$$\text{sumD1}[x][y] = \sum_i \sum_j \text{filtD1}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i = -2..5 \text{ và } j = 0..5$$

biến Ac được thiết đặt là 96

theo cách khác

$$\text{sumH}[x][y] = \sum_i \sum_j \text{filtH}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumV}[x][y] = \sum_i \sum_j \text{filtV}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD0}[x][y] = \sum_i \sum_j \text{filtD0}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD1}[x][y] = \sum_i \sum_j \text{filtD1}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumOfHV}[x][y] = \text{sumH}[x][y] + \text{sumV}[x][y] \quad (8-1168)$$

biến Ac được thiết đặt là 64

Các biến $\text{dir1}[x][y]$, $\text{dir2}[x][y]$ và $\text{dirS}[x][y]$ với $x, y = 0..CtbSizeY - 1$ được dẫn ra như sau:

Các biến hv1 , hv0 và dirHV được dẫn ra như sau:

Nếu $\text{sumV}[x \gg 2][y \gg 2]$ lớn hơn $\text{sumH}[x \gg 2][y \gg 2]$, phần sau đây áp dụng:

$$\text{hv1} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 1;$$

Theo cách khác, phần sau đây áp dụng:

$$\text{hv1} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 3.$$

Các biến d1 , d0 và dirD được dẫn ra như sau:

Nếu $\text{sumD0}[x \gg 2][y \gg 2]$ lớn hơn $\text{sumD1}[x \gg 2][y \gg 2]$, phần sau đây áp dụng:

$$\text{d1} = \text{sumD0}[x \gg 2][y \gg 2]$$

$$\text{d0} = \text{sumD1}[x \gg 2][y \gg 2]$$

$$\text{dirD} = 0 \quad ;$$

Theo cách khác, phần sau đây áp dụng:

$$\text{d1} = \text{sumD1}[x \gg 2][y \gg 2]$$

$$\text{d0} = \text{sumD0}[x \gg 2][y \gg 2]$$

$$\text{dirD} = 2 \quad .$$

Các biến $hvd1$, $hvd0$, được dẫn ra như sau:

$$hvd1 = (d1 * hv0 > hv1 * d0) ? d1 : hv1$$

$$hvd0 = (d1 * hv0 > hv1 * d0) ? d0 : hv0.$$

Các biến $dirS[x][y]$, $dir1[x][y]$ và $dir2[x][y]$ được dẫn ra như sau:

$$dir1[x][y] = (d1 * hv0 > hv1 * d0) ? dirD : dirHV$$

$$dir2[x][y] = (d1 * hv0 > hv1 * d0) ? dirHV : dirD$$

$$dirS[x][y] = (hvd1 > 2 * hvd0) ? 1 : ((hvd1 * 2 > 9 * hvd0) ? 2 : 0).$$

Biến $avgVar[x][y]$ với $x, y = 0..CtbSizeY - 1$ được dẫn ra như sau:

$$varTab[] = \{ 0, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4 \}$$

$$avgVar[x][y] = varTab[Clip3(0, 15, (sumOfHV[x >> 2][y >> 2] * Ac) >> (3 + BitDepthY))].$$

Mảng chỉ số bộ lọc phân loại $filtIdx[x][y]$ và mảng chỉ số chuyển vị $transposeIdx[x][y]$ với $x = y = 0..CtbSizeY - 1$ được dẫn ra như sau:

$$transposeTable[] = \{ 0, 1, 0, 2, 2, 3, 1, 3 \}$$

$$transposeIdx[x][y] = transposeTable[dir1[x][y] * 2 + (dir2[x][y] >> 1)]$$

$$filtIdx[x][y] = avgVar[x][y]$$

Khi $dirS[x][y]$ không bằng 0, $filtIdx[x][y]$ được cải biến như sau:

$$filtIdx[x][y] += (((dir1[x][y] \& 0x1) \ll 1) + dirS[x][y]) * 5.$$

2.4 Quy trình lọc khối cây mã hóa đối với các mẫu sắc độ

Các đầu vào của quy trình xử lý này là:

mảng mẫu hình ảnh sắc độ được tái cấu trúc $recPicture$ trước quy trình lọc vòng lặp thích nghi,

mảng mẫu hình ảnh sắc độ được tái cấu trúc được lọc $alfPicture$,

vị trí sắc độ $(xCtbC, yCtbC)$ xác định mẫu trên cùng bên trái của khối cây mã hóa sắc độ hiện tại liên quan tới mẫu trên cùng bên trái của hình ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu hình ảnh sắc độ được tái cấu trúc được lọc được cải biến alfPicture.

Kích thước của khối cây mã hóa sắc độ hiện tại ctbSizeC được dẫn ra như sau:

$$ctbSizeC = CtbSizeY / SubWidthC$$

Đối với việc dẫn ra của các mẫu sắc độ được tái cấu trúc được lọc alfPicture[x][y], mỗi mẫu sắc độ được tái cấu trúc bên trong khối cây mã hóa sắc độ hiện tại recPicture[x][y] được lọc như sau với $x, y = 0..ctbSizeC - 1$:

Các vị trí (hx, vy) đối với mỗi trong số các mẫu sắc độ tương ứng (x, y) bên trong mảng đã cho recPicture của các mẫu sắc độ được dẫn ra như sau:

$$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, xCtbC + x)$$

$$vy = y$$

Độ dịch vị trí mẫu đầu vào bộ lọc $r = 0..4$ được dẫn ra theo vị trí mẫu sắc độ y như được thể hiện trong bảng 2.4 dưới đây

Bảng 2.4b – Bản mô tả về độ dịch vị trí mẫu đầu vào bộ lọc $r=0..4$ theo vị trí mẫu sắc độ y

| y | r1 | r2 | r3 | r4 |
|---|----|----|----|----|
| $y == CtbSizeC - 2$ (or) $CtbSizeC - 3$ | 0 | 0 | 0 | 0 |
| $y == CtbSizeC - 1$ (or) $CtbSizeC - 4$ | 1 | 1 | 1 | 1 |
| $y == CtbSizeY - 7$ (or) $CtbSizeY - 2$ | 1 | 1 | 2 | 2 |
| $yCtb + y == (\text{pic_height_in_luma_samples} / \text{SubHeightC} - 1) - 1$ | 0 | 1 | 0 | 2 |
| $yCtb + y == (\text{pic_height_in_luma_samples} / \text{SubHeightC} - 1) - 2$ | 1 | 1 | 1 | 2 |
| $yCtb + y == 0$ | 1 | 0 | 2 | 0 |
| $yCtb + y == 1$ | 1 | 1 | 2 | 1 |

| | | | | |
|---------------------|---|---|---|---|
| Các trường hợp khác | 1 | 1 | 2 | 2 |
|---------------------|---|---|---|---|

Tổng biến được dẫn ra như sau:

$$\begin{aligned}
\text{tổng} &= \text{AlfCoeffC}[0] * (\text{recPicture}[hx, vy + r3] + \text{recPicture}[hx, vy - r4]) + \\
&\quad \text{AlfCoeffC}[1] * (\text{recPicture}[hx + 1, vy + r1] + \text{recPicture}[hx - 1, vy - r2]) \\
+ & \\
&\quad \text{AlfCoeffC}[2] * (\text{recPicture}[hx, vy + r1] + \text{recPicture}[hx, vy - r2]) + \\
&\quad \text{AlfCoeffC}[3] * (\text{recPicture}[hx - 1, vy + r1] + \text{recPicture}[hx + 1, vy - r2]) \\
+ & \\
&\quad \text{AlfCoeffC}[4] * (\text{recPicture}[hx + 2, vy] + \text{recPicture}[hx - 2, vy]) + \\
&\quad \text{AlfCoeffC}[5] * (\text{recPicture}[hx + 1, vy] + \text{recPicture}[hx - 1, vy]) + \\
&\quad \text{AlfCoeffC}[6] * \text{recPicture}[hx, vy]
\end{aligned}$$

$$\text{tổng} = (\text{tổng} + 64) \gg 7$$

Mẫu hình ảnh sắc độ được tái cấu trúc được lọc được cải biến $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y]$ được dẫn ra như sau:

$$\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum}).$$

Đối với giải pháp 2:

$$\begin{aligned}
c_{12}' &= c_{12} + 2 * c_0 + 2 * c_1 + 2 * c_2 + 2 * c_3 + 2 * c_4 + 2 * c_5 + 2 * c_6 \\
&\quad + 2 * c_7 + 2 * c_8
\end{aligned}$$

Giải pháp bổ sung (giải pháp 3) mà được mô tả trên Fig.12c cũng có thể được sử dụng để thực hiện việc lọc của các dòng H, J, I (ở trên ranh giới ảo) và các dòng K, L, M (ở dưới ranh giới ảo). Dựa vào hướng mép ưu thế đối với khối đã cho, bộ lọc ALF được cải biến.

Phải lưu ý rằng thuật toán bất kỳ mà có thể dẫn ra hướng mép ưu thế đối với khối đã cho có thể được sử dụng và không giới hạn ở thuật toán được giải thích dưới đây. Hướng mép ưu thế đối với khối đã cho có thể được tính toán nhờ sử dụng các trị số độ dốc được tính toán cho khối đã cho nghĩa là, nhờ sử dụng các trị số độ dốc g_v, g_h, g_{d1} và g_{d2} và trị số định hướng D được tính toán trong suốt thời gian phân loại khối ALF. Nếu trị số định hướng D là 0, 1, 2 sau đó như được mô tả trên

Fig.12c, hình ảnh phụ c, giải pháp 1 (bổ sung các hệ số theo chiều thẳng đứng) được lựa chọn. Nếu trị số định hướng D là 3 hoặc 4 (trường hợp khi định hướng của khối có mép “theo đường chéo” mạnh), sau đó điều kiện thêm sau đây g_{d1} lớn hơn hoặc bằng g_{d2} được kiểm tra. Nếu đúng, sau đó độ dốc 135 độ được lựa chọn là hướng mép ưu thế và do đó như được mô tả trên Fig.12c, hình ảnh phụ b, việc lọc được cải biến được thực hiện như sau:

$$\text{Đối với các dòng M và H } c3' = c3 + c0$$

$$\text{Đối với các dòng L và I } c6' = c6 + c1$$

$$c7' = c7 + c2$$

$$c8' = c3 + c8 + c0$$

Đối với các dòng J và K:

$$c9' = c9 + c8 + c3 + c0$$

$$c10' = c10 + c7 + c2$$

$$c11' = c1 + c6 + c11 + c4$$

$$c12' = c12 + 2 * c5$$

Mặt khác nếu g_{d2} lớn hơn g_{d1} sau đó như được mô tả trên Fig.12c, hình ảnh phụ a, việc lọc được cải biến được thực hiện như sau:

$$\text{Đối với các dòng M và H } c1' = c1 + c0$$

$$\text{Đối với các dòng L và I } c4' = c1 + c4 + c0$$

$$c5' = c5 + c2$$

$$c6' = c6 + c3$$

Đối với các dòng J và K:

$$c9' = c9 + c4 + c1 + c0$$

$$c10' = c10 + c5 + c2$$

$$c11' = c8 + c7 + c6 + c3 + c11$$

$$c12' = c12 + 2 * c7$$

Phải lưu ý rằng nếu bộ lọc đã cho được biến đổi như được đưa ra trong bảng 1, sau đó tất cả các giải pháp nêu trên vẫn có thể được sử dụng. Các phương trình theo các giải pháp nêu trên được đưa ra đối với trường hợp khi không có phép biến đổi nào được áp dụng. Đối với bộ lọc được biến đổi, các hệ số theo các phương trình có thể được thay thế lần lượt dựa vào các phương trình 17, 18, 19.

Giải pháp bổ sung (giải pháp 4) mà được mô tả trên Fig.19a và Fig.19b, ngắt việc lọc khi tổng tuyệt đối của các hệ số được đệm lớn hơn ngưỡng (T) lần tổng tuyệt đối của các hệ số không được đệm. Ví dụ, đối với

Các dòng M và H, $if (2 * abs (c0) > (T * (2 * (abs (c1) + abs (c2) + abs(c3) + abs(c4) + abs(c5) + abs(c6) + abs(c7) + abs(c8) + abs(c9) + abs(c10) + abs(c11)) + abs(c12))))$, sau đó việc lọc ALF được ngắt đối với các dòng M và H.

Tương tự đối với các dòng L và I, $if (2 * (abs (c0) + abs (c1) + abs (c2) + abs (c3)) > (T * (2 * (abs(c4) + abs(c5) + abs(c6) + abs(c7) + abs(c8) + abs(c9) + abs(c10) + abs(c11)) + abs(c12))))$, sau đó việc lọc ALF được ngắt đối với các dòng L và I.

Tương tự đối với các dòng J và K, $if (2 * (abs (c0) + abs (c1) + abs (c2) + abs (c3) + abs(c4) + abs(c5) + abs(c6) + abs(c7) + abs(c8)) > (T * (2 * (abs(c9) + abs(c10) + abs(c11)) + abs(c12))))$, sau đó việc lọc ALF được ngắt đối với các dòng J và K.

Trong đó T thường là ngưỡng mà có thể lấy các trị số phân số trong khoảng giữa 0 và 1. Ví dụ, trị số T có thể là 0,5, 0,6 hoặc 0,7

Tương tự đối với sắc độ, như được mô tả trên Fig.20, đối với các dòng N và K, $if (2 * abs (c0) > (T * (2 * (abs (c1) + abs (c2) + abs(c3) + abs(c4) + abs(c5)) + abs(c6))))$, sau đó việc lọc ALF được ngắt đối với các dòng N và K.

đổi với các dòng M và L, $if ((2 * (abs(c0) + abs(c1) + abs(c2) + abs(c3)) > (T * (2 * (abs(c4) + abs(c5)) + abs(c6))))$), sau đó việc lọc ALF được ngắt đổi với các dòng M và L.

hàm $abs(x)$ được định rõ như sau:

$$Abs(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

Theo giải pháp thay thế, thao tác lọc được ngắt nếu dựa vào phép biến đổi hình học được xác định và tiệm cận theo chiều ngang của mẫu điểm ảnh được lọc tới ranh giới ảo. Nói cách khác giả sử rằng mẫu điểm ảnh được lọc là N mẫu ở trên hoặc ở dưới ranh giới ảo (khoảng cách theo chiều ngang là N tới ranh giới theo chiều ngang) và phép biến đổi bộ lọc hình học được xác định là phép biến đổi_X (transformation_X). Thao tác lọc được ngắt cho mẫu điểm ảnh nếu;

- Nếu N nhỏ hơn ngưỡng định trước, cụ thể là thr,
- Và nếu phép biến đổi_X trong số tập hợp của các phép biến đổi mà được định trước hoặc được chỉ báo ở dòng bit.

Theo một ví dụ cụ thể thr bằng 1 (chỉ báo hàng của các mẫu mà là một dòng ở trên và một dòng ở dưới ranh giới ảo). Theo ví dụ khác tập hợp của các phép biến đổi bao gồm ít nhất một trong số ‘Không có phép biến đổi’, ‘Đường chéo’, ‘Lật theo chiều thẳng đứng’, ‘Quay’. Theo ví dụ khác thr bằng 1 và tập hợp của các phép biến đổi bao gồm ít nhất một trong số ‘Không có phép biến đổi’, ‘Đường chéo’, ‘Lật theo chiều thẳng đứng’, ‘Quay’. Theo ví dụ khác tập hợp của các phép biến đổi bao gồm đường chéo và quay.

Lưu ý rằng tập hợp của các phép biến đổi là tập hợp con của tất cả các phép biến đổi bộ lọc khả thi. Theo tài liệu JVET1001_v7 tài liệu “mã hóa video đa năng (Bản dự thảo 3)” (có thể truy cập công khai tại <http://phenix.it-sudparis.eu/jvet/>), có 4 phép biến đổi bộ lọc khả thi. Trong tài liệu bộ lọc phép biến đổi được xác định theo phần phụ “8.5.4.3 Quy trình dẫn ra đối với chỉ số bộ lọc và chuyển vị ALF cho các mẫu độ chói”, và phép biến đổi được xác định được chỉ báo bởi biến `transposeIdx`. Các trị số mà có thể được giả sử bởi `transposeIdx` là 0, 1, 2 và 3.

Theo một ví dụ thao tác lọc được ngắt đối với mẫu điểm ảnh nếu transposeIdx được xác định bằng trị số định trước. Lưu ý rằng tập hợp của các trị số định trước bao gồm ít nhất một trị số giữa 0 và 3. Lưu ý thêm rằng tập hợp của các trị số định trước không bao gồm tất cả các trị số khả thi (tất cả 0, 1, 2 và 3).

Theo ví dụ khác thao tác lọc được ngắt nếu điểm ảnh được lọc là một mẫu ở trên hoặc ở dưới ranh giới ảo và phép biến đổi bộ lọc được xác định trong số tập hợp của các trị số định trước.

Các yêu cầu bộ đệm dòng đối với các mẫu sắc độ:

Fig.13 mô tả các yêu cầu bộ đệm dòng đối với các mẫu sắc độ. Khi việc giải khối chỉ sử dụng 2 mẫu và cải biến nhiều nhất 1 mẫu, các yêu cầu bộ đệm dòng ALF sắc độ xuống tới 6,25 dòng. Do không có sự phân loại khối rõ ràng được thực hiện đối với các mẫu sắc độ, nên chỉ việc lọc phải được cải biến ở các ranh giới ảo (VB). Fig.14 thể hiện phiên bản bị biến dạng của bộ lọc được tạo ra hoặc bằng cách sử dụng việc đệm (giải pháp 1) hoặc bằng cách bổ sung hệ số được loại bỏ vào hệ số trung tâm (giải pháp 2)

Phân loại khối được cải biến khi ranh giới ảo là 6 dòng

Fig.15 thể hiện các yêu cầu bộ đệm dòng ALF khi bộ lọc giải khối sử dụng 6 dòng của bộ đệm dòng. Do đó ranh giới ảo là 6 dòng ở trên ranh giới CTU. Đối với trường hợp này, tổng yêu cầu bộ đệm dòng bộ lọc vòng lặp tăng lên tới 15,25 dòng.

Để làm giảm các yêu cầu bộ đệm dòng, Fig.16 mô tả một giải pháp mà được giải thích dưới đây:

Việc phân loại khối của khối 4 x 4 bắt đầu ở dòng C đạt được bằng cách đệm dòng H tới dòng I để đảm bảo việc phân loại khối không yêu cầu dòng bất kỳ ở dưới ranh giới ảo.

Các dòng G, H sử dụng lại quyết định phân loại khối từ khối C, D, E, F. Các dòng I, J sử dụng lại quyết định phân loại khối từ các dòng K, L, M, N (hoặc) các dòng I, J, K, L có thể được coi là một phân đoạn khối 4x 4 và việc phân loại được thực hiện đối với 4 dòng này, và sau đó các dòng M, N sử dụng lại quyết định phân

loại khối từ khối K, L, M, N.

Fig.17 sử dụng cách tiếp cận dịch chuyển lưới trong đó hai dòng thứ nhất ở ranh giới hình ảnh trên cùng và hai dòng cuối cùng ở ranh giới hình ảnh dưới cùng được phân loại và được lọc tách biệt và do đó lưới lọc và phân loại ALF được di chuyển thêm 2 mẫu xuống ở trục Y. Giải pháp này có ưu điểm là khối 4×4 đã cho không bao giờ chồng lấp hoặc đi qua ranh giới ảo. Nói chung khi kích thước ranh giới ảo không là bội số của 4 (kích thước phân loại khối ALF), sau đó lưới lọc ALF có thể được dịch chuyển nhờ sử dụng công thức sau đây:

$((\text{Kích thước ranh giới ảo}) / 4) - 4$. Đối với việc lọc thực tế của nó, phiên bản bị biến dạng của bộ lọc được giải thích trên Fig.12a, Fig.12b và Fig.12c có thể được sử dụng.

Các phương án khác của sáng chế được đề xuất trong phần sau đây. Cần lưu ý rằng việc đánh số được sử dụng trong phần sau đây không cần thiết cần phải tuân theo việc đánh số được sử dụng trong các phần trước đó.

Phương án 1. Phương pháp lọc trị số mẫu của hình ảnh, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối, trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n ,

trong đó phương pháp này bao gồm các bước, với t nguyên nhỏ hơn k và với j nguyên trong khoảng từ 1 đến m :

tạo ra, với i nguyên trong khoảng từ 1 đến $k-t-1$, trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-t-1$, và/hoặc

tạo ra, với i nguyên trong khoảng từ $k-t$ đến k , trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-t$.

Phương án 2. Phương pháp lọc các trị số mẫu của hình ảnh, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối,

trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n ,

trong đó phương pháp này bao gồm các bước, với t nguyên nhỏ hơn k và đối với mỗi i từ 1 đến k , và mỗi j từ 1 đến m :

tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó nếu i trong khoảng từ 1 đến $k-t-1$, tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-t-1$, và trong đó nếu i trong khoảng từ $k-t$ đến k , tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-t$.

Phương án 3. Phương pháp theo phương án 1 hoặc 2, trong đó khối thứ nhất là đơn vị cây mã hóa (CTU) thứ nhất, và khối thứ hai là CTU thứ hai.

Phương án 4. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 3, trong đó t là 2, 3, 4, 5 hoặc 6.

Phương án 5. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 4,

trong đó đối với mỗi i trong khoảng từ 1 đến $k-t-1$ và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc bao gồm nhiều mẫu $p(i', j')$ với i' trong khoảng từ 1 đến $k-t-1$; và

trong đó đối với mỗi i trong khoảng từ $k-t$ đến k và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc bao gồm nhiều mẫu $p(i', j')$ với i' bằng hoặc lớn hơn $k-t$.

Phương án 6. Phương pháp theo phương án 5, trong đó đối với mỗi i trong khoảng từ $k-t$ đến k và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc còn bao gồm một hoặc nhiều mẫu $p(i', j')$ với i' lớn hơn k .

Phương án 7. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 6, trong đó bộ lọc là bộ lọc trong vòng lặp.

Phương án 8. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 7, trong đó bộ lọc bao gồm bộ lọc giải khối.

Phương án 9. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 8, trong đó bộ lọc bao gồm bộ lọc giải khối, bộ lọc độ dịch thích nghi mẫu (SAO), và bộ lọc vòng lặp thích nghi (ALF),

trong đó SAO được áp dụng sau bộ lọc giải khối, và ALF được áp dụng sau SAO.

Phương án 10. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 9, trong đó phương pháp bao gồm các bước:

xác định các trị số mẫu của khối thứ nhất;

áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất ở các hàng từ 1 đến $k-t-1$;

lưu trữ các trị số mẫu của khối thứ nhất từ các hàng $k-t$ đến k trong bộ đệm;

xác định các trị số mẫu của khối thứ hai; và

áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất ở các hàng $k-t$ mà đã được lưu trữ bộ đệm.

Phương án 11. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 10, trong đó bộ lọc là bộ lọc vòng lặp thích nghi, và bộ lọc thích nghi được xác định nhờ sử dụng bước phân loại mà truy cập các mẫu của khối hiện tại và cả cá mẫu xung quanh khối.

Phương án 12. Phương pháp theo phương án 11, trong đó khi các mẫu xung quanh khối không khả dụng, bộ lọc thích nghi được xác định nhờ sử dụng trị số chuẩn hóa, trong đó trị số chuẩn hóa là trị số nguyên dương và phụ thuộc vào số lượng các mẫu có thể truy cập được xung quanh khối đã cho.

Phương án 13. Phương pháp theo phương án 11 hoặc 12, trong đó trị số chuẩn hóa có mối tương quan ngược với số lượng các mẫu có thể truy cập được.

Phương án 14. Phương pháp theo phương án bất kỳ trong số các phương án từ 10 đến 13, trong đó trị số chuẩn hóa là 96, khi không phải tất cả các mẫu xung quanh khối khả dụng.

Phương án 15. Phương pháp theo phương án bất kỳ trong số các phương án từ 10 đến 14, trong đó trị số chuẩn hóa là 64, khi tất cả các mẫu xung quanh khối khả dụng.

Phương án 16. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 10, trong đó bộ lọc là bộ lọc vòng lặp thích nghi.

Phương án 17. Phương pháp theo một phương án bất kỳ trong số các phương án từ 1 đến 10, trong đó thao tác đệm đối xứng được áp dụng khi các mẫu được yêu cầu không thể truy cập được để lọc mẫu đã cho.

Phương án 18. Phương pháp theo phương án 17, trong đó thao tác đệm đối xứng được định rõ khi thay thế các mẫu không khả dụng bởi các mẫu lân cận trung gian mà khả dụng đối với thao tác lọc.

Phương án 19. Phương pháp theo phương án bất kỳ trong số phương án 17 hoặc 18, trong đó thao tác đệm đối xứng được định rõ là phương pháp mà bộ lọc được làm cho đối xứng bằng cách áp dụng thao tác đệm ở cả hai phía của bộ lọc mặc dù ở một phía, các mẫu bộ lọc vẫn có thể truy cập được.

Phương án 20. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 19, trong đó hình ảnh được bao gồm trong khung của chuỗi video. Phương án 21. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 20, được thực hiện để mã hóa hoặc giải mã hình ảnh, hoặc được thực hiện để mã hóa hoặc giải mã chuỗi video mà bao gồm hình ảnh.

Mặc dù các phương án của sáng chế chủ yếu đã được mô tả dựa vào việc mã video, nhưng cần lưu ý rằng các phương án của hệ thống mã 10, bộ mã hóa 20 và bộ giải mã 30 (và một cách tương ứng hệ thống 10) và các phương án còn lại được mô tả ở đây cũng có thể được tạo cấu hình cho việc xử lý hoặc mã hóa hình ảnh tĩnh, nghĩa là, việc xử lý hoặc mã hóa của hình ảnh riêng độc lập với hình ảnh trước đó hoặc liên tiếp bất kỳ như trong việc mã hóa video. Nói chung chỉ các bộ phận dự báo liên kết 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp việc mã hóa xử lý hình ảnh giới hạn ở hình ảnh đơn 17. Tất cả các tính năng khác (cũng được gọi là các công cụ hoặc các công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng như nhau cho việc xử lý hình ảnh tĩnh, ví dụ, tính toán phần dư 204/304, biến đổi 206, lượng tử hóa 208, lượng tử hóa ngược 210/310, biến đổi (ngược) 212/312, phân chia 262/362, dự báo bên trong 254/354, và/hoặc lọc vòng lặp 220, 320, và mã hóa entropi 270 và giải mã entropi 304.

Các phương án, ví dụ, về bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, ví dụ, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được thực hiện ở phần cứng, phần mềm, phần sụn, hoặc sự kết hợp bất kỳ của chúng. Nếu được thực hiện ở phần mềm, các chức năng có thể được lưu trữ trên phương tiện đọc được bởi máy tính hoặc được truyền qua các phương tiện truyền thông như một hoặc nhiều lệnh hoặc mã và được thực thi bởi bộ phận xử lý dựa vào phần cứng. Các phương tiện đọc được bởi máy tính có thể bao gồm các phương tiện lưu trữ đọc được bởi máy tính, mà tương ứng với phương tiện hữu hình chẳng hạn như các phương tiện lưu trữ dữ liệu, hoặc các phương tiện truyền thông bao gồm phương tiện bất kỳ mà tạo điều kiện thuận lợi cho việc chuyển đổi của chương trình máy tính từ một địa điểm tới địa điểm khác, ví dụ, theo giao thức truyền thông. Theo cách thức này, các phương tiện đọc được bởi máy tính thường có thể tương ứng với (1) các phương tiện

lưu trữ đọc được bởi máy tính hữu hình mà không nhất thời hoặc (2) phương tiện truyền thông chẳng hạn như tín hiệu hoặc sóng mang. Các phương tiện lưu trữ dữ liệu có thể là các phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất các lệnh, mã và/hoặc các cấu trúc dữ liệu cho việc thực hiện của các kỹ thuật được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện đọc được bởi máy tính.

Bộ mã hóa 20 và/hoặc bộ giải mã 30 có thể bao gồm hệ mạch xử lý 46 được tạo cấu hình để thực hiện các thao tác và các phương pháp khác nhau được mô tả ở đây. Hệ mạch xử lý 46 có thể bao gồm phần cứng và phần mềm. Phần cứng có thể bao gồm hệ mạch tương tự hoặc hệ mạch kỹ thuật số, hoặc cả hệ mạch tương tự và kỹ thuật số. Hệ mạch kỹ thuật số có thể bao gồm các thành phần chẳng hạn như các mạch tích hợp chuyên dụng (application-specific integrated circuit - ASIC), các mảng công lập trình được dạng trường (field-programmable array - FPGA), các bộ xử lý tín hiệu kỹ thuật số (digital signal processor - DSP), hoặc các bộ xử lý đa năng. Cụ thể là, các bộ lọc được đề cập ở đây có thể được thực hiện ở phần cứng hoặc ở phần mềm hoặc sự kết hợp của phần cứng và phần mềm. Theo một phương án, hệ mạch xử lý bao gồm một hoặc nhiều bộ xử lý và bộ nhớ không nhất thời được kết nối với một hoặc nhiều bộ xử lý. Bộ nhớ không nhất thời có thể mang mã chương trình có thể thực thi được mà, khi được thực thi bởi một hoặc nhiều bộ xử lý, khiến bộ mã hóa 20 và/hoặc bộ giải mã 30 thực hiện các thao tác hoặc các phương pháp được mô tả ở đây.

Bằng ví dụ, và không giới hạn, các phương tiện lưu trữ đọc được bởi máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc thiết bị lưu trữ đĩa quang, thiết bị lưu trữ đĩa từ khác, hoặc các thiết bị lưu trữ từ khác, bộ nhớ chớp, hoặc phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn dưới dạng của các lệnh hoặc các cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Ngoài ra, sự kết hợp bất kỳ được gọi thích hợp là phương tiện đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ trang mạng, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường dây thuê bao kỹ thuật số (digital subscriber line - DSL), hoặc các công nghệ không

dây chẳng hạn như hồng ngoại, radio, và vi sóng, sau đó cáp đồng trục, cáp sợi quang, cáp xoắn đôi, DSL, hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng được bao gồm theo định nghĩa về phương tiện. Tuy nhiên, cần hiểu rằng các phương tiện lưu trữ đọc được bởi máy tính và các phương tiện lưu trữ dữ liệu không bao gồm các sự kết nối, các sóng mang, các tín hiệu, hoặc các phương tiện nhất thời khác, nhưng thay vào đó được hướng đến các phương tiện lưu trữ hữu hình, không nhất thời. Đĩa (disk) và đĩa (disc), như được sử dụng ở đây, bao gồm đĩa compac (compact disc - CD), đĩa laze, đĩa quang, đĩa đa năng kỹ thuật số (digital versatile disc - DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa thường được tái tạo dữ liệu từ, trong khi các đĩa tái tạo dữ liệu quang với các laze. Các sự kết hợp nêu trên cũng cần được bao gồm nằm trong phạm vi của các phương tiện đọc được bởi máy tính.

Các lệnh có thể được thực thi bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu kỹ thuật số (DSP), bộ vi xử lý đa năng, các mạch tích hợp chuyên dụng (ASIC), các mảng cổng lập trình được dạng trường (FPGA), hoặc hệ mạch logic rời rạc hoặc được tích hợp tương đương khác. Theo đó, thuật ngữ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến bất kỳ trong số cấu trúc nêu trên hoặc cấu trúc khác bất kỳ thích hợp cho việc thực hiện của các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, tính năng được mô tả ở đây có thể được đưa ra nằm trong các môđun phần cứng và/hoặc phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp trong bộ mã hóa/giải mã được kết hợp. Ngoài ra, các kỹ thuật có thể được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc các thành phần logic.

Các kỹ thuật của sáng chế có thể được thực hiện trong nhiều thiết bị hoặc nhiều máy, bao gồm thiết bị cầm tay không dây, mạch tích hợp (integrated circuit - IC) hoặc tập hợp của các IC (ví dụ, bộ chip). Các thành phần, các môđun, hoặc các bộ phận khác nhau được mô tả theo sáng chế nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không cần thiết yêu cầu thực hiện bởi các bộ phận phần cứng khác nhau. Hơn nữa, như được mô tả ở trên, các bộ phận khác nhau có thể được kết hợp trong bộ phận phần

cứng bộ mã hóa/giải mã hoặc được cung cấp bởi tập hợp của các bộ phận phần cứng tương tác, bao gồm một hoặc nhiều bộ xử lý như được nêu trên, kết hợp với phần mềm và/hoặc phần sụn thích hợp.

Nhằm tham chiếu, các toán tử logic sau đây được định rõ như sau:

$x \ \&\& \ y$ Logic Boolean "và" của x và y

$x \ || \ y$ Logic Boolean "hoặc" của x và y

$! \ y$ Logic Boolean "không"

$x \ ? \ y \ : \ z$ Nếu x là đúng hoặc không bằng 0, thì đánh giá tới trị số của y ; theo cách khác, đánh giá tới trị số của z .

Nhằm tham chiếu, các toán tử quan hệ sau đây được định rõ như sau:

$>$ Lớn hơn

$>=$ Lớn hơn hoặc bằng

$<$ Nhỏ hơn

$<=$ Nhỏ hơn hoặc bằng

$==$ Bằng

$!=$ Không bằng

Khi toán tử quan hệ được áp dụng tới thành phần cú pháp hoặc biến mà đã được gán trị số "na" (không áp dụng được), trị số "na" được coi là trị số phân biệt cho thành phần cú pháp hoặc biến. Trị số "na" không được coi là không bằng trị số khác bất kỳ.

Nhằm tham chiếu, các toán tử thao tác bit sau đây được định rõ như sau:

$\&$ Thao tác bit "và (and)". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép toán trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

| Thao tác bit "hoặc (or)". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép toán trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

^Thao tác bit "phép toán loại trừ hoặc (exclusive or)". Khi thực hiện phép toán trên các đối số nguyên, thực hiện phép toán trên sự biểu diễn bù hai của trị số nguyên. Khi thực hiện phép toán trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

$x \gg y$ Sự dịch chuyển phải số học của sự biểu diễn số nguyên bù hai của các chữ số nhị phân x với y . Hàm này được định rõ chỉ cho các trị số nguyên không âm của y . Các bit được dịch chuyển thành các bit quan trọng nhất (most significant bit - MSB) do kết quả của sự dịch chuyển phải có trị số bằng MSB của x trước phép toán dịch chuyển.

$x \ll y$ Sự dịch chuyển trái số học của sự biểu diễn số nguyên bù hai của các chữ số nhị phân x với y . Hàm này được định rõ chỉ cho các trị số nguyên không âm của y . Các bit được dịch chuyển thành các bit ít quan trọng nhất (least significant bit - LSB) là kết quả của sự dịch chuyển trái có trị số bằng 0.

Tóm lại, sáng chế đề xuất phương pháp lọc trị số mẫu của hình ảnh, hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối, trong đó hình ảnh bao gồm ít nhất $n \times m$ (n lần m) mẫu $p(i,j)$ được sắp xếp theo n hàng và m cột, trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , trong đó các mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và các mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , trong đó phương pháp bao gồm các bước, với q nguyên nhỏ hơn k và với j nguyên trong khoảng từ 1 đến m : tạo ra, với i nguyên trong khoảng từ 1 đến $k-q-1$, trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp

của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' lớn hơn $k-q-1$, và/hoặc tạo ra, với i nguyên trong khoảng từ $k-q$ đến k , trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm các mẫu $p(i',j')$ bất kỳ với chỉ số hàng i' nhỏ hơn $k-q$.

Sáng chế đề cập đến các đơn sáng chế Hoa Kỳ US 62/784,535 được nộp vào ngày 23 tháng 12 năm 2018, US 62/787,735 được nộp vào ngày 02 tháng 01 năm 2019, và US 62/822,979 được nộp vào ngày 24 tháng 03 năm 2019. Các nội dung của các đơn này được kết hợp ở đây nhằm viện dẫn.

YÊU CẦU BẢO HỘ

1. Thiết bị lọc trị số mẫu của hình ảnh mà có khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối theo chiều ngang, trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu được sắp xếp theo n hàng và m cột, trong đó mẫu $p(i,j)$ với i từ 1 đến k với i từ 1 đến k thuộc về khối thứ nhất và mẫu $p(i,j)$ với i từ $k+1$ đến n thuộc về khối thứ hai, và trong đó k là số nguyên nhỏ hơn n , và trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , thiết bị này bao gồm:

bộ nhớ chứa các lệnh; và

bộ xử lý truyền thông với bộ nhớ và khi thực thi các lệnh, được tạo cấu hình để:

khi số nguyên i trong khoảng từ 1 đến $k-t-1$, số nguyên j trong khoảng từ 1 đến m và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' lớn hơn $k-t-1$, và

khi số nguyên i trong khoảng từ $k-t$ đến k , số nguyên j trong khoảng từ 1 đến m và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' nhỏ hơn $k-t$,

trong đó i' là chỉ số hàng và j' là chỉ số cột.

2. Thiết bị theo điểm 1, trong đó mẫu hiện tại $p(i,j)$ với chỉ số hàng i trong khoảng từ 1 đến $k-t-1$ nằm phía trên ranh giới ảo, và mẫu hiện tại $p(i,j)$ với chỉ số hàng i trong khoảng từ $k-t$ đến k nằm phía dưới ranh giới ảo.
3. Thiết bị theo điểm 1, trong đó khối thứ nhất là đơn vị cây mã hóa (coding tree unit - CTU) thứ nhất, khối thứ hai là CTU thứ hai, và ranh giới khối theo chiều ngang là ranh giới CTU theo chiều ngang.
4. Thiết bị theo điểm 1, trong đó t là 1, 2, 3, 4, 5 hoặc 6.
5. Thiết bị theo điểm 3, trong đó đối với các mẫu là các mẫu độ chói, ranh giới ảo là 4 dòng độ chói phía trên ranh giới CTU theo chiều ngang, hoặc ranh giới ảo là 4 dòng độ chói phía trên ranh giới dưới cùng của CTU thứ nhất.
6. Thiết bị theo điểm 3, trong đó đối với các mẫu là các mẫu sắc độ, ranh giới ảo là 2 dòng sắc độ phía trên ranh giới CTU theo chiều ngang, hoặc ranh giới ảo là 2 dòng sắc độ phía trên ranh giới dưới cùng của CTU thứ nhất.
7. Thiết bị theo điểm 1,

trong đó đối với mỗi i trong khoảng từ 1 đến $k-t-1$ và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc bao gồm nhiều mẫu $p(i',j')$ với i' trong khoảng từ 1 đến $k-t-1$, và

trong đó đối với mỗi i trong khoảng từ $k-t$ đến k và đối với mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc bao gồm nhiều mẫu $p(i',j')$ với i' bằng hoặc lớn hơn $k-t$.
8. Thiết bị theo điểm 7, trong đó đối với mỗi i trong khoảng từ $k-t$ đến k và đối với

mỗi j từ 1 đến m , tập hợp của các mẫu hỗ trợ bộ lọc còn bao gồm một hoặc nhiều mẫu $p(i',j')$ với i' lớn hơn k .

9. Thiết bị theo điểm 1, trong đó

khi $i' > k-t-1$, mẫu thứ nhất được bố trí theo chiều thẳng đứng phía trên ranh giới ảo được sử dụng là trị số được đệm mà thay thế mẫu thứ hai được bố trí theo chiều thẳng đứng phía dưới ranh giới ảo, và các mẫu hỗ trợ bộ lọc bao gồm mẫu thứ nhất và mẫu thứ hai; và/hoặc

khi $i' < k-t$, mẫu thứ ba được bố trí theo chiều thẳng đứng phía dưới ranh giới ảo được sử dụng như trị số được đệm mà thay thế mẫu thứ tư được bố trí theo chiều thẳng đứng phía trên ranh giới ảo, và các mẫu hỗ trợ bộ lọc bao gồm mẫu thứ ba và mẫu thứ tư,

trong đó số lượng các lần của mẫu thứ nhất hoặc hoặc thứ ba được sử dụng là trị số được đệm được xác định dựa vào khoảng cách của mẫu hiện tại $p(i,j)$ tới ranh giới ảo.

10. Thiết bị theo điểm 5, trong đó tập hợp của các độ dịch vị trí mẫu theo chiều thẳng đứng được kết hợp với tập hợp của các mẫu hỗ trợ bộ lọc được xác định theo vị trí mẫu theo chiều thẳng đứng độ chói của mẫu hiện tại $p(i,j)$ khi ranh giới ảo là 4 dòng độ chói phía trên ranh giới CTU theo chiều ngang.

11. Thiết bị theo điểm 6, trong đó tập hợp của các độ dịch vị trí mẫu theo chiều thẳng đứng được kết hợp với tập hợp của các mẫu hỗ trợ bộ lọc được xác định theo vị trí mẫu theo chiều thẳng đứng sắc độ của mẫu hiện tại $p(i,j)$ khi ranh giới ảo là 2 dòng sắc độ phía trên ranh giới CTU theo chiều ngang.

12. Thiết bị theo điểm 1, trong đó bộ lọc là bộ lọc vòng lặp thích nghi (ALF - adaptive loop filter).

13. Thiết bị theo điểm 1, trong đó bộ xử lý còn được tạo cấu hình để

xác định các trị số mẫu của khối thứ nhất;

áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất trong các hàng 1 đến $k-t-1$;

lưu trữ các trị số mẫu của khối thứ nhất từ các hàng $k-t$ đến k trong bộ đệm;

xác định các trị số mẫu của khối thứ hai; và

áp dụng bộ lọc tới các trị số mẫu của khối thứ nhất trong các hàng $k-t$ đến k mà đã được lưu trữ trong bộ đệm.

14. Thiết bị theo điểm 1, trong đó bộ lọc là bộ lọc vòng lặp thích nghi mà được xác định nhờ sử dụng bước phân loại mà truy cập các mẫu của khối đã cho và các mẫu xung quanh khối đã cho.

15. Thiết bị theo điểm 14, trong đó khi các mẫu xung quanh khối đã cho không khả dụng, bộ lọc thích nghi được xác định nhờ sử dụng trị số chuẩn hóa (A_c) được sử dụng ở bước phân loại, trong đó trị số chuẩn hóa (A_c) là trị số nguyên dương và phụ thuộc vào số lượng các mẫu có thể truy cập được xung quanh khối đã cho; hoặc

trong đó bộ lọc thích nghi được xác định nhờ sử dụng trị số chuẩn hóa (A_c) được sử dụng ở bước phân loại, trong đó trị số chuẩn hóa (A_c) là trị số nguyên dương và phụ thuộc vào khoảng cách của khối đã cho tới ranh giới ảo, hoặc trị số chuẩn hóa (A_c) phụ thuộc vào việc khối đã cho có gần ranh giới ảo hay không; hoặc

trong đó bộ lọc thích nghi được xác định nhờ sử dụng trị số chuẩn hóa (A_c) được

sử dụng ở bước phân loại, trong đó trị số chuẩn hóa (A_c) là 96 để phản hồi lại việc xác định của ranh giới ảo là N dòng phía trên ranh giới CTU theo chiều ngang, $N=2$ hoặc 4.

16. Thiết bị theo điểm 14, trong đó trị số chuẩn hóa (A_c) có mối tương quan ngược với số lượng các mẫu có thể truy cập được.

17. Thiết bị theo điểm 14, trong đó trị số chuẩn hóa là 96, khi không phải tất cả các mẫu xung quanh khối đã cho là khả dụng.

18. Thiết bị theo điểm 14, trong đó trị số chuẩn hóa là 64, khi tất cả các mẫu xung quanh khối đã cho là khả dụng.

19. Thiết bị theo điểm 14, , trong đó khối đã cho là khối 4×4 .

20. Thiết bị theo điểm 19, trong đó đối với khối 4×4 liền kề với ranh giới ảo, trị số chuẩn hóa (A_c) được sử dụng ở bước phân loại là 96; hoặc đối với khối 4×4 mà không liền kề ranh giới ảo, trị số chuẩn hóa là 64.

21. Thiết bị theo điểm 1, trong đó đó thao tác đệm đối xứng được áp dụng tại hoặc gần ranh giới ảo khi các mẫu được yêu cầu không thể truy cập được để lọc mẫu hiện tại $p(i,j)$.

22. Thiết bị theo điểm 21, trong đó thao tác đệm đối xứng được định rõ khi thay thế các mẫu không khả dụng bằng các mẫu lân cận trung gian mà khả dụng đối với thao tác lọc.

23. Thiết bị theo điểm 1, trong đó bộ xử lý được tạo cấu hình để

khi mẫu hiện tại $p(i,j)$ được bố trí theo chiều thẳng đứng phía trên ranh giới ảo, tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$,

trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' lớn hơn $k-t-1$, và trong đó số nguyên i trong khoảng từ 1 đến $k-t-1$, và

khi mẫu hiện tại $p(i,j)$ được bố trí theo chiều thẳng đứng phía dưới ranh giới ảo, tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc, trong đó tập hợp của các mẫu hỗ trợ bộ lọc bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' nhỏ hơn $k-t$, và trong đó số nguyên i trong khoảng từ $k-t$ đến k .

24. Thiết bị theo điểm 23, trong đó đối với mẫu hiện tại $p(i,j)$, chỉ số hàng i tương ứng với vị trí mẫu theo chiều thẳng đứng (y) của mẫu hiện tại $p(i,j)$, và/hoặc đối với các mẫu $p(i',j')$, chỉ số hàng i' tương ứng với vị trí mẫu theo chiều thẳng đứng (y) của các mẫu $p(i',j')$ của các mẫu hỗ trợ bộ lọc.

25. Phương pháp lọc trị số mẫu của hình ảnh mà có khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối theo chiều ngang, trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu được sắp xếp theo n hàng và m cột, trong đó mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và mẫu $p(i,j)$ với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , và trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m , bao gồm:

khi số nguyên i trong khoảng từ 1 đến $k-t-1$, số nguyên j trong khoảng từ 1 đến

m và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' lớn hơn $k-t-1$, và

khi số nguyên i trong khoảng từ $k-t$ đến k , số nguyên j trong khoảng từ 1 đến m và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' nhỏ hơn $k-t$,

trong đó i' là chỉ số hàng và j' là chỉ số cột.

26. Phương tiện bất biến đọc được bởi máy tính lưu trữ các lệnh máy tính để lọc trị số mẫu của hình ảnh, mà khi được thực thi bởi một hoặc nhiều bộ xử lý, làm cho một hoặc nhiều bộ xử lý thực hiện các thao tác sau :

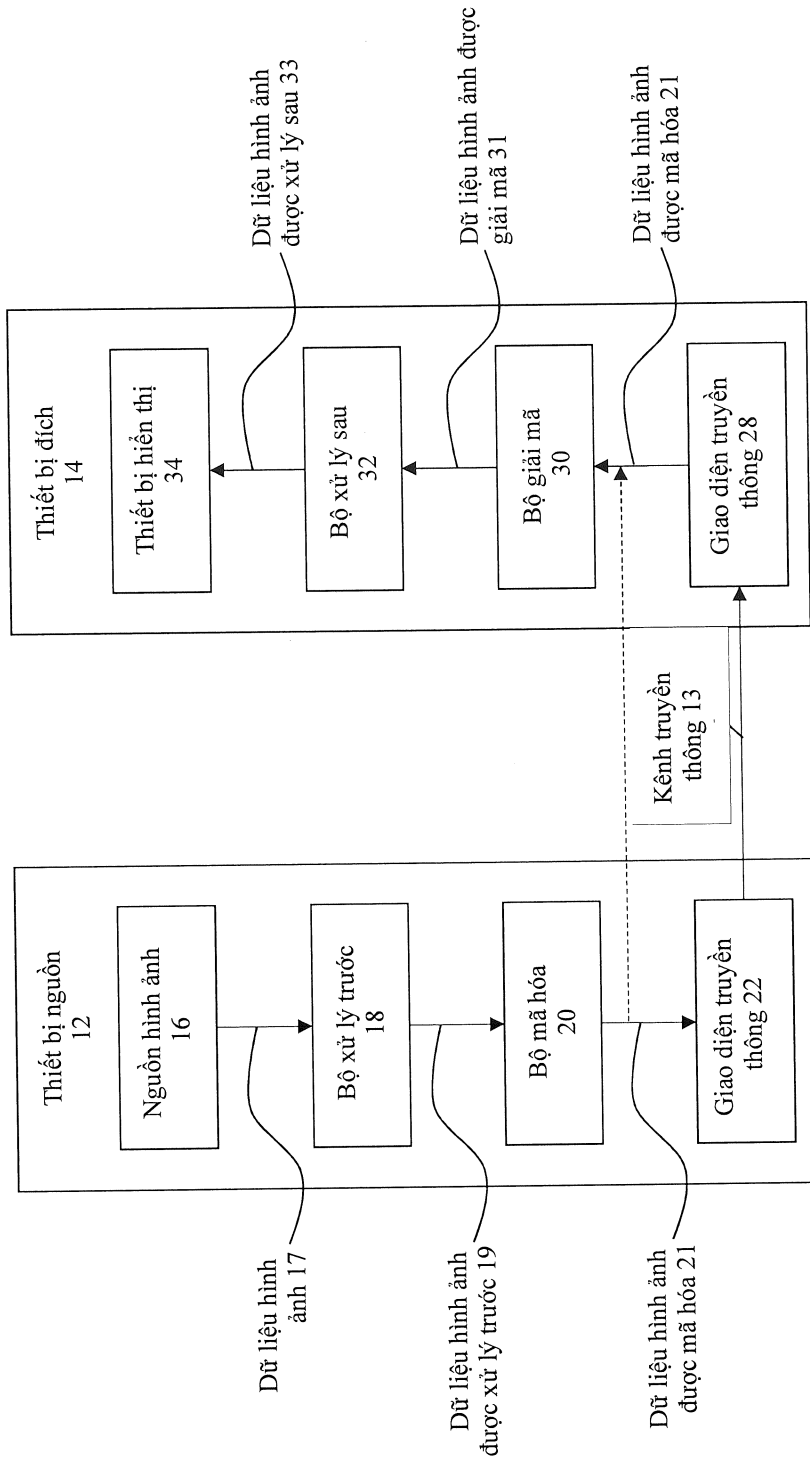
khi số nguyên i trong khoảng từ 1 đến $k-t-1$, số nguyên j trong khoảng từ 1 đến m và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' lớn hơn $k-t-1$, và/hoặc

khi số nguyên i trong khoảng từ $k-t$ đến k , số nguyên j trong khoảng từ 1 đến m

và số nguyên t nhỏ hơn k , tạo ra trị số mẫu được lọc bằng cách áp dụng bộ lọc tới trị số của mẫu hiện tại $p(i,j)$, trong đó bộ lọc được hỗ trợ bởi tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$, trong đó tập hợp của các mẫu hỗ trợ bộ lọc $p(i',j')$ bao gồm mẫu hiện tại $p(i,j)$ và một hoặc nhiều mẫu khác, trong đó tập hợp của các mẫu hỗ trợ bộ lọc không bao gồm bất kỳ các mẫu $p(i',j')$ với chỉ số hàng i' nhỏ hơn $k-t$, và trong đó i' là chỉ số hàng và j' là chỉ số cột;

trong đó hình ảnh bao gồm khối thứ nhất và khối thứ hai, khối thứ nhất và khối thứ hai liền kề nhau trong ranh giới khối theo chiều ngang, trong đó hình ảnh bao gồm ít nhất $n \times m$ mẫu được sắp xếp theo n hàng và m cột, trong đó mẫu $p(i,j)$ với i từ 1 đến k thuộc về khối thứ nhất và mẫu với i từ $k+1$ đến n thuộc về khối thứ hai, trong đó k là số nguyên nhỏ hơn n , và trong đó i là chỉ số hàng mà chạy từ 1 đến n và j là chỉ số cột mà chạy từ 1 đến m .

1/28



10

Fig. 1A

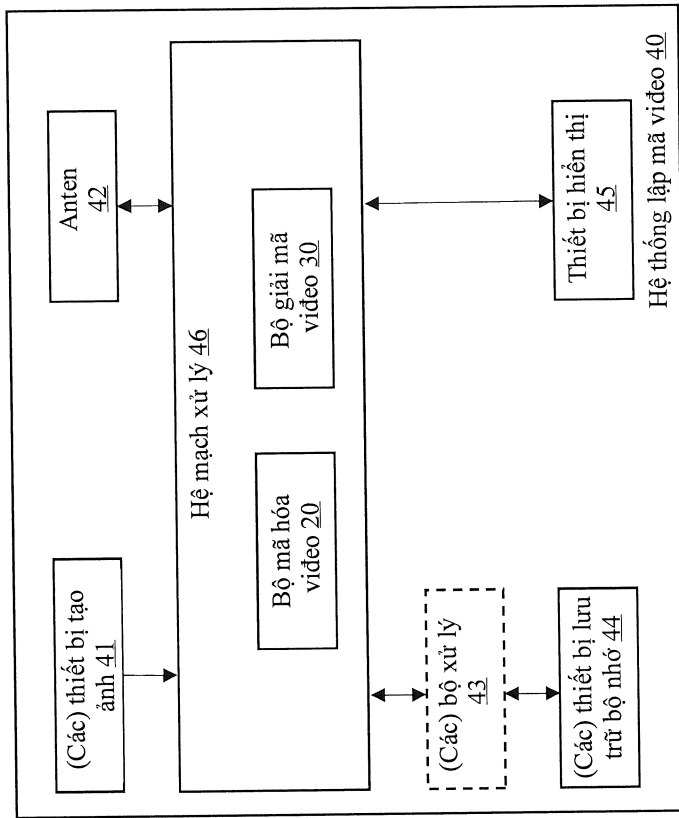


Fig. 1B

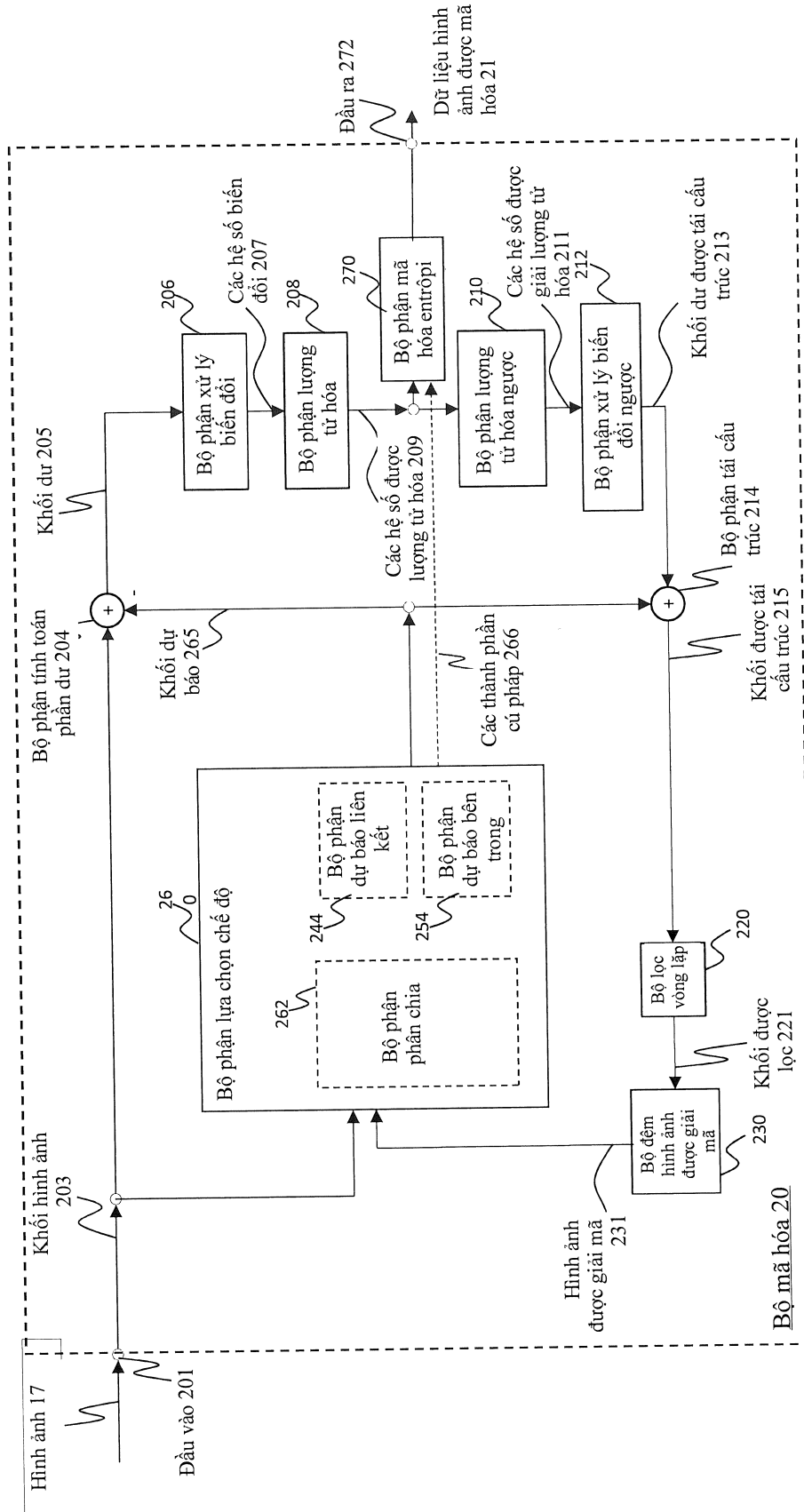


Fig. 2

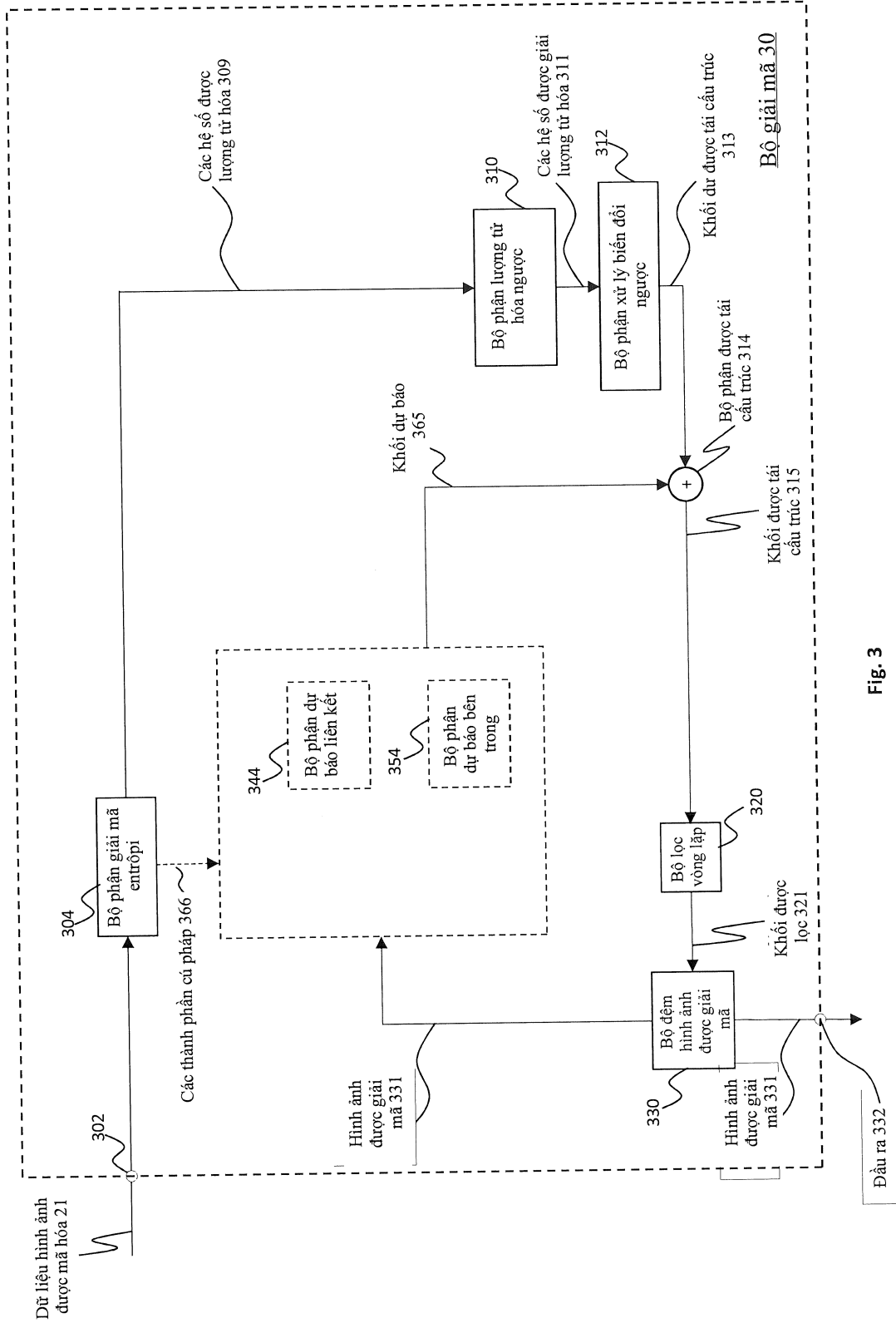
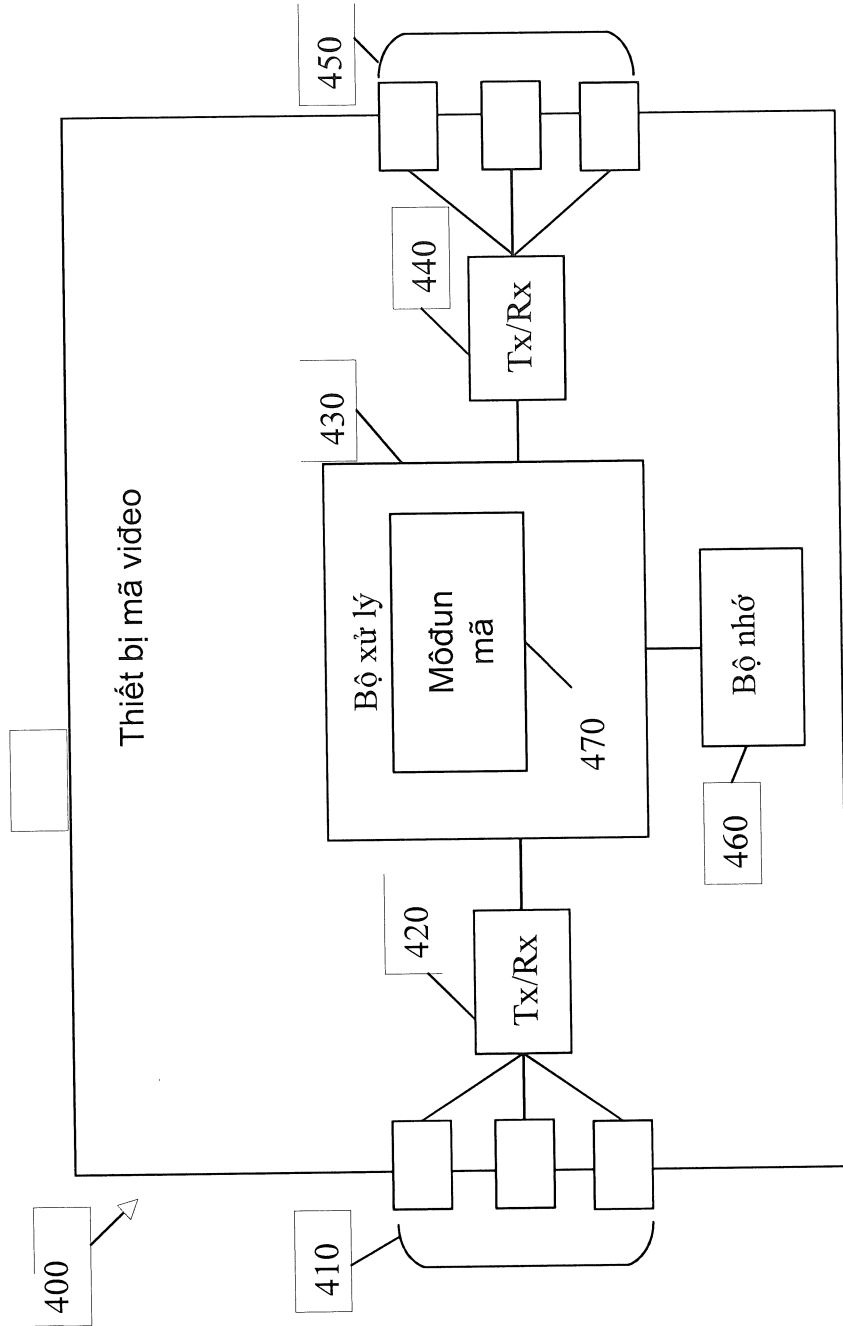


Fig. 3



Các cổng
dòng lên

Fig. 4

Các cổng
dòng xuống

6/28

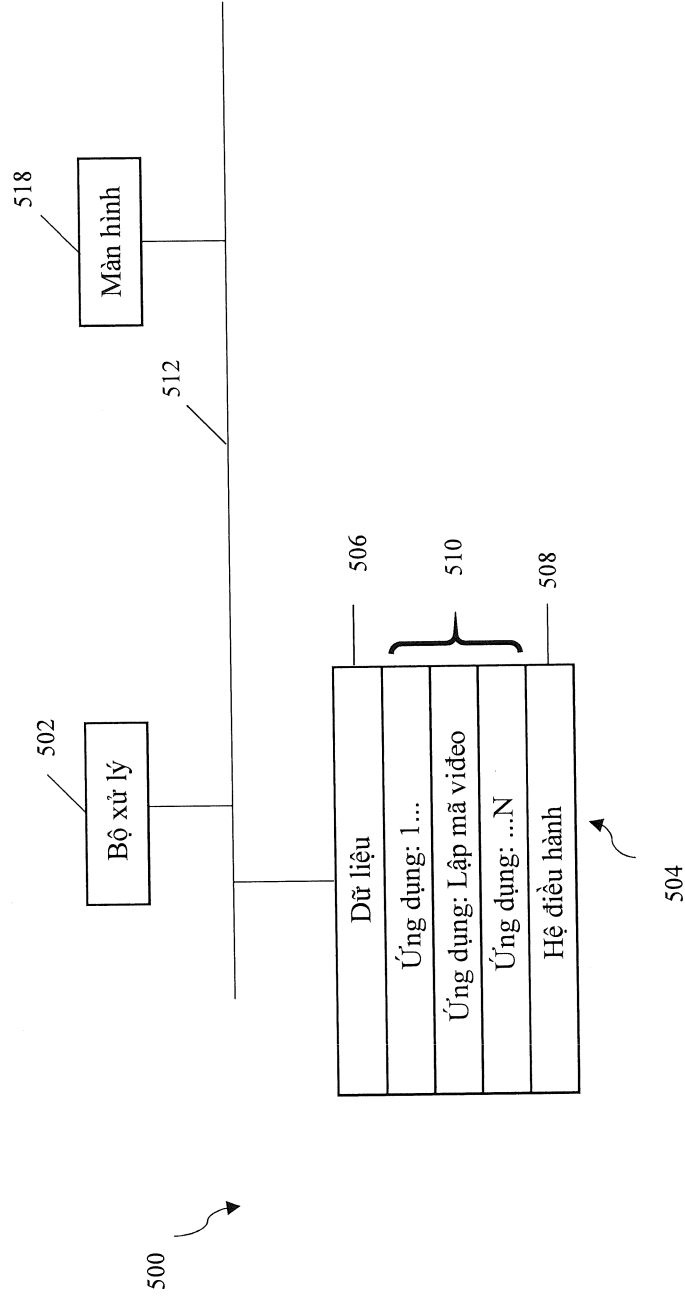


Fig. 5

7/28

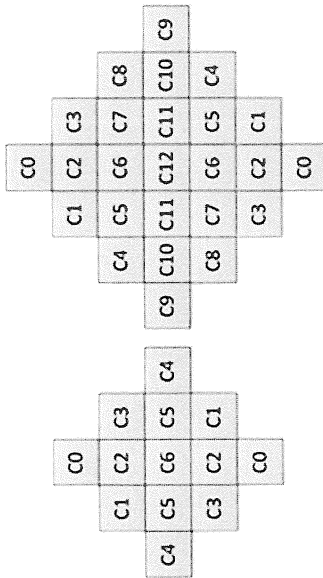


Fig.6. Các hình dạng bộ lọc ALF (kim cương 5 x 5 sắc độ, kim cương 7 x 7 độ chói)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |
| V | V | V | V | V | V | V | V | V | V |

(a) Các vị trí được lấy mẫu con cho độ dốc theo chiều thẳng đứng

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |
| D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 | D1 |

(c) Các vị trí được lấy mẫu con cho độ dốc theo đường chéo

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |
| H | H | H | H | H | H | H | H | H | H |

(b) Các vị trí được lấy mẫu con cho độ dốc theo chiều ngang

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |
| D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 | D2 |

(d) Các vị trí được lấy mẫu con cho độ dốc theo đường chéo

Fig.7. Phân loại khối ALF được lấy mẫu con

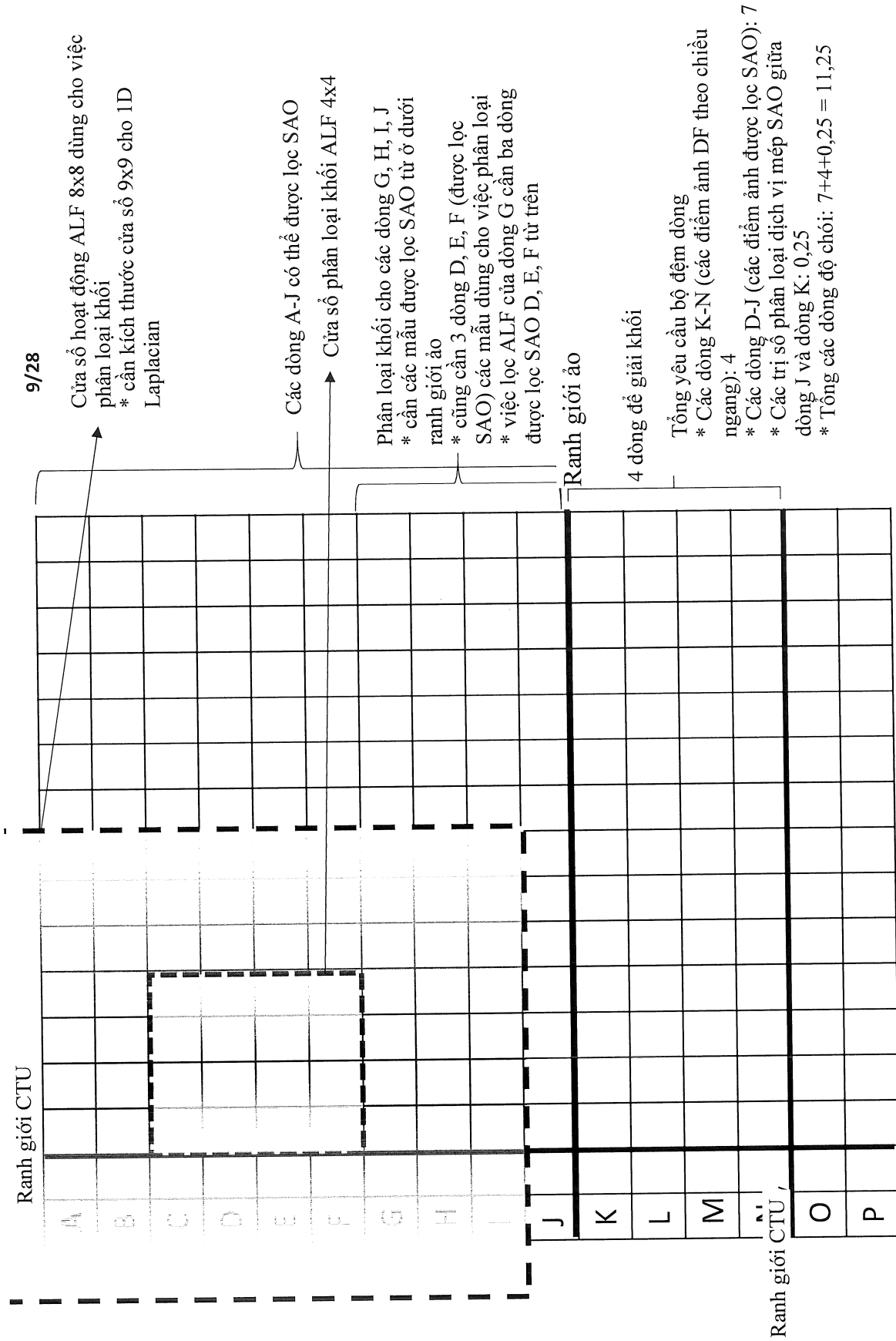


Fig.8. Các yêu cầu bộ lọc dòng độ chói bộ lọc vòng lặp VTM-3.0

10/28

Fig.9. Phân loại ALF được cải biến để tránh khối để tránh khối các bộ đệm dòng độ chói ALF (Biến thể không đối xứng)

* Chúng tôi cần việc phân loại khối ALF được cải biến cho các khối 4x4 ở ranh giới ảo

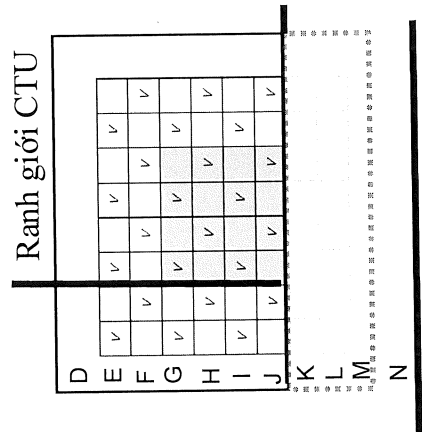


Fig. 9a

Giải pháp 1:

Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB

(4a):

* Chỉ sử dụng các mẫu ở trên VB trong việc phân loại (các mẫu ở các dòng E tới I)

Đối với việc phân loại khối của khối 4x4 trung gian ở dưới VB

(4b):

* Chỉ sử dụng các mẫu ở dưới VB trong việc phân loại (các mẫu ở các dòng P tới L)

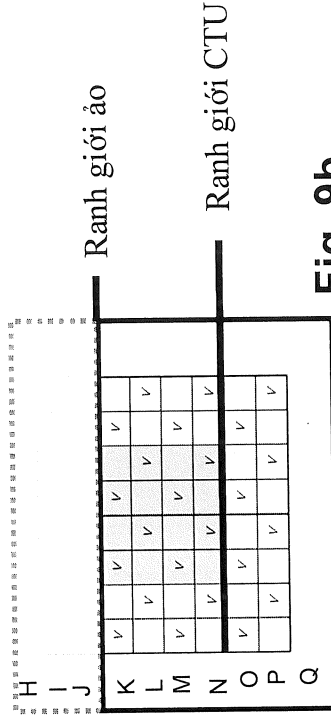


Fig. 9b

Giải pháp 2:

Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB (4a):

* Đệm dòng J tới dòng K, L, M

Đối với việc phân loại khối của khối 4x4 trung gian ở dưới VB (4b):

* Đệm dòng K tới dòng J, I, H

Fig.10. Phân loại ALF được cải biến để tránh khối để tránh khối các bộ đệm dòng độ chói ALF (Biến thể đối xứng)

* Chúng tôi cân việc phân loại khối ALF được cải biến cho các khối 4x4 ở ranh giới ảo

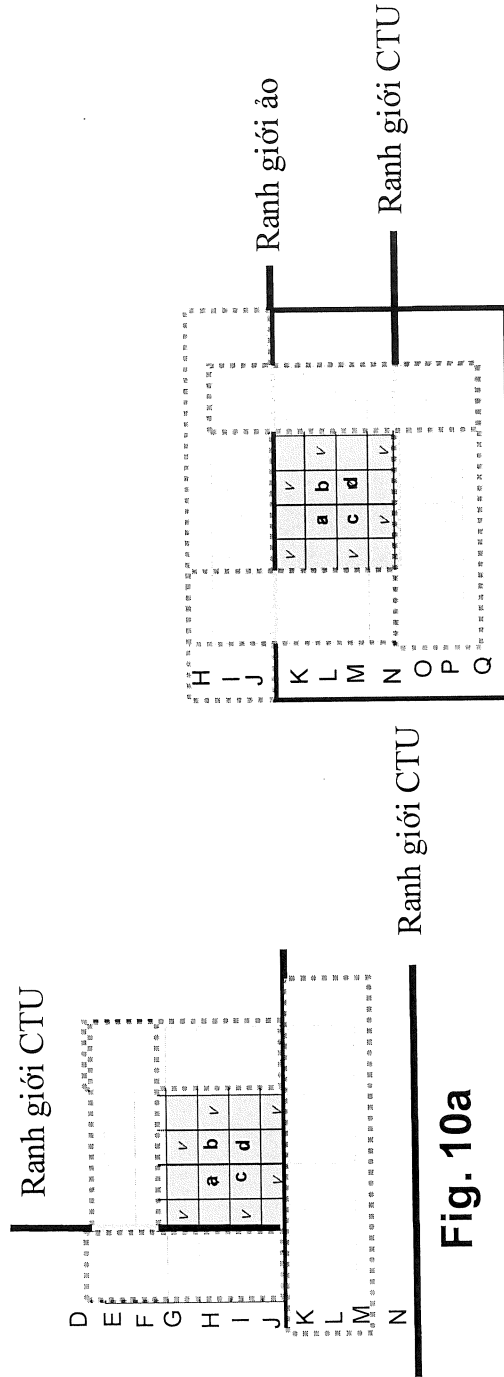


Fig. 10a

Fig. 10b

Giải pháp 1:

Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB (hình vẽ bên trái):

* Chỉ sử dụng các mẫu ở trên VB trong việc phân loại. **Chỉ yếu các mẫu a, b, c, d được sử dụng**

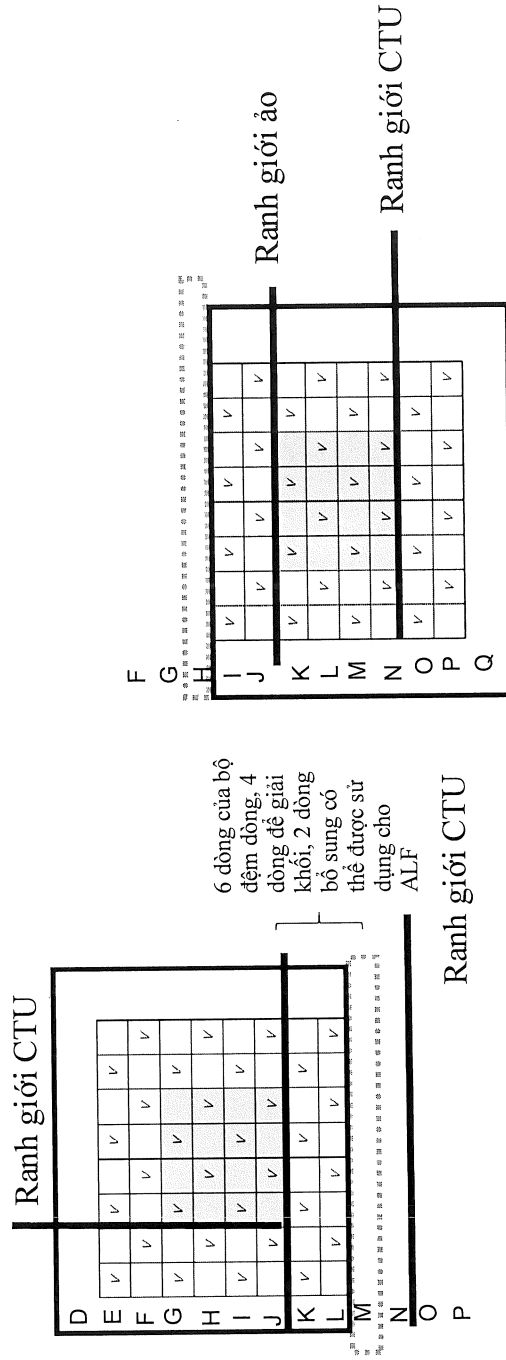
Đối với việc phân loại khối của khối 4x4 trung gian ở dưới VB (hình vẽ bên phải):

* Chỉ sử dụng các mẫu ở dưới VB trong việc phân loại. **Chỉ yếu các mẫu a, b, c, d được sử dụng**

12/28

Fig.11. Phân loại ALF được cải biến để tránh khối các bộ đệm dòng đối chọi ALF với 6 dòng (2 dòng bổ sung có thể được sử dụng cho việc phân loại khối ALF)

* Chúng tôi cần việc phân loại khối ALF được cải biến cho các khối 4x4 ở ranh giới ảo



Giải pháp 1:

Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB (hình ảnh bên trái):

* Các dòng được lọc SAO K, L ở dưới VB còn có thể được sử dụng

Đối với việc phân loại khối của khối 4x4 trung ở dưới VB (hình ảnh bên phải):

* Các dòng được lọc SAO I, J ở trên VB còn có thể được sử dụng

Giải pháp 2:

Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB (hình ảnh bên trái):

* Đệm dòng L tới dòng M

Đối với việc phân loại khối của khối 4x4 trung gian ở dưới VB (hình ảnh bên phải):

* Đệm dòng I tới dòng H

Fig.12a. Phân loại ALF được cải biến để tránh khỏi các bộ đệm dòng độ chói ALF

* Đối với việc lọc ALF của các dòng H, I, J (ở trên ranh giới ảo) và các dòng K, L, M (ở dưới ranh giới ảo) các phiên bản bị biến dạng sau đây của bộ lọc được sử dụng:

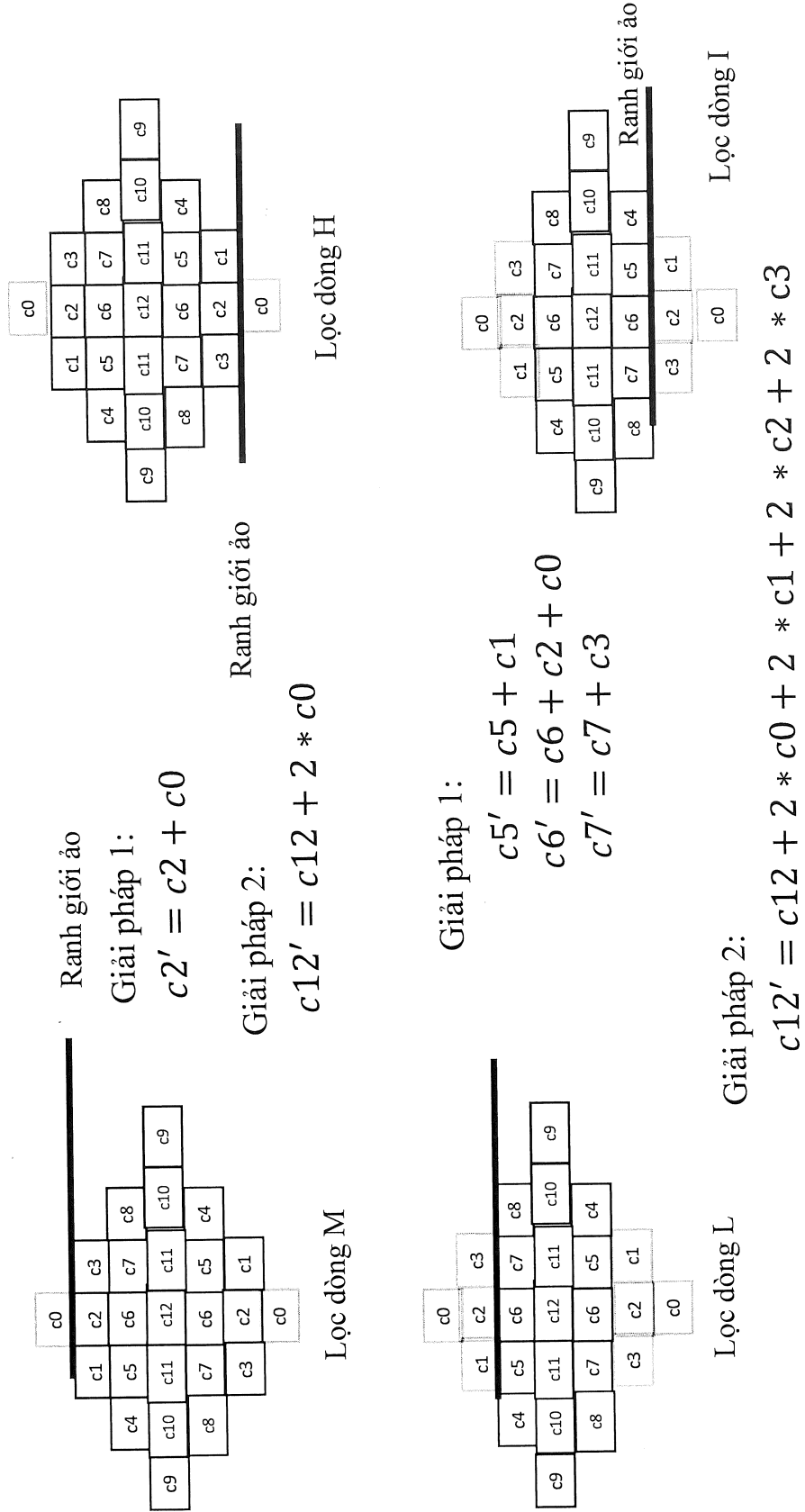


Fig.12b. Phân loại ALF được cải biến để tránh khối các bộ đệm dòng độ chói ALF



Lọc dòng J

Lọc dòng K

Giải pháp 1:

$$\begin{aligned}
 c10' &= c10 + c4 + c8 \\
 c11' &= c11 + c5 + c7 + c1 + c3 \\
 c12' &= c12 + 2 * c6 + 2 * c2 + 2 * c0
 \end{aligned}$$

Giải pháp 2:

$$c12' = c12 + 2 * c0 + 2 * c1 + 2 * c2 + 2 * c3 + 2 * c4 + 2 * c5 + 2 * c6 + 2 * c7 + 2 * c8$$

Fig.12c. Phân loại ALF được cải biến để tránh khởi các bộ đệm dòng độ chói ALF (cải biến bộ lọc dựa vào độ dốc)

| | | | | | | |
|----|-----|-----|-----|-----|-----|----|
| | | c0 | | | | |
| | c1 | c2 | c3 | | | |
| | c4 | c5 | c6 | c7 | c8 | |
| c9 | c10 | c11 | c12 | c11 | c10 | c9 |
| | c8 | c7 | c6 | c5 | c4 | |
| | | c3 | c2 | c1 | | |
| | | | | | | c0 |

a) Độ dốc 45 độ:

Một dòng (dòng M và H):

$$c1' = c1 + c0$$

Hai dòng (dòng L và I):

$$c4' = c1 + c4 + c0$$

$$c5' = c5 + c2$$

$$c6' = c6 + c3$$

Ba dòng (dòng J và K):

$$c9' = c9 + c4 + c1 + c0$$

$$c10' = c10 + c5 + c2$$

$$c11' = c8 + c7 + c6 + c3 + c11$$

$$c12' = c12 + 2 * c7$$

| | | | | | | |
|----|-----|-----|-----|-----|-----|----|
| | | | | | | |
| | c1 | c2 | c3 | | | |
| | c4 | c5 | c6 | c7 | c8 | |
| c9 | c10 | c11 | c12 | c11 | c10 | c9 |
| | c8 | c7 | c6 | c5 | c4 | |
| | | c3 | c2 | c1 | | |
| | | | | | | c0 |

b) Độ dốc 135 độ:

Một dòng (dòng M và H):

$$c3' = c3 + c0$$

Hai dòng (dòng L và I):

$$c6' = c6 + c1$$

$$c7' = c7 + c2$$

$$c8' = c3 + c8 + c0$$

Ba dòng (dòng J và K):

$$c9' = c9 + c8 + c3 + c0$$

$$c10' = c10 + c7 + c2$$

$$c11' = c1 + c6 + c11 + c4$$

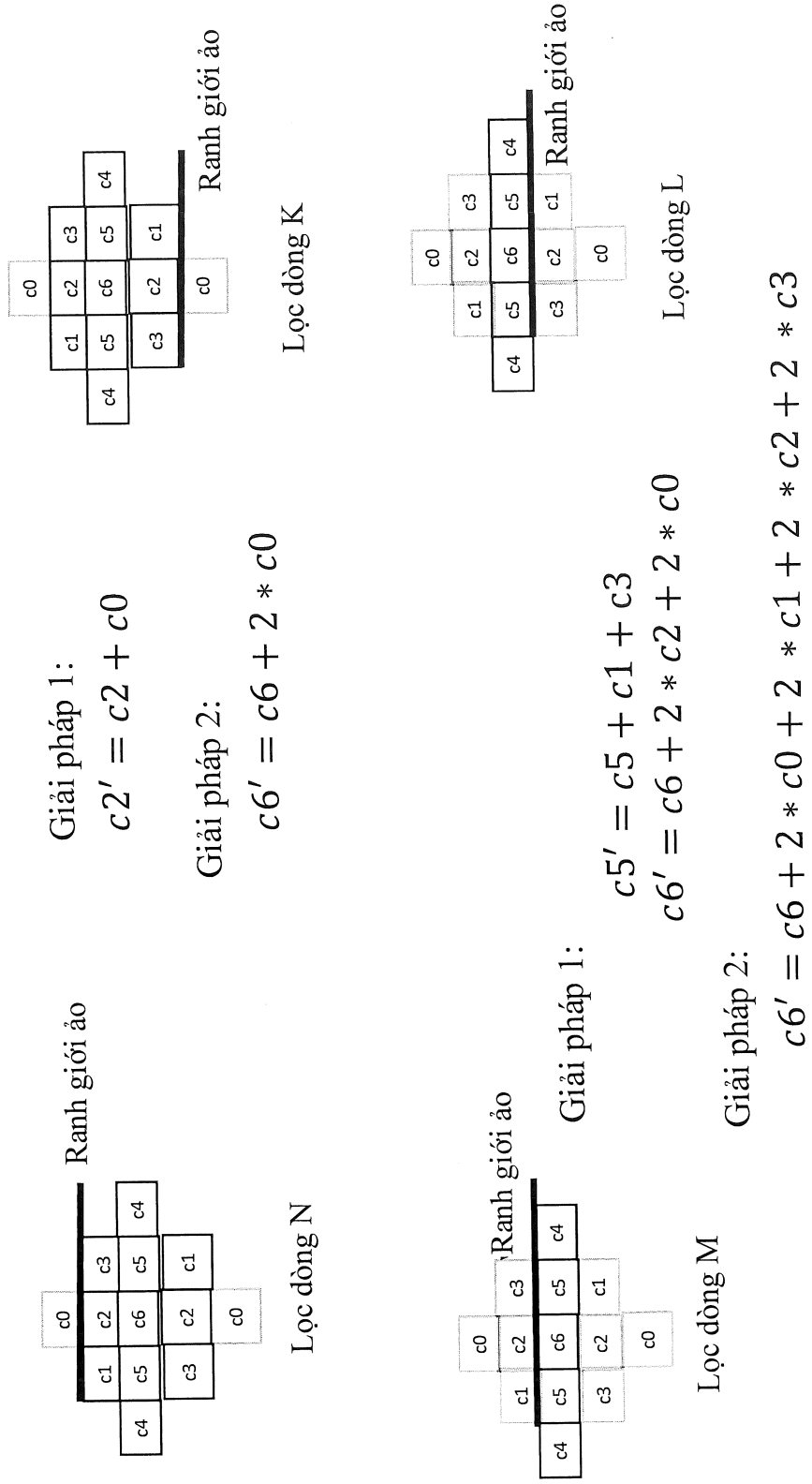
$$c12' = c12 + 2 * c5$$

| | | | | | | |
|----|-----|-----|-----|-----|-----|----|
| | | | | | | |
| | | c1 | | | | |
| | | c2 | c3 | | | |
| | c4 | c5 | c6 | c7 | c8 | |
| c9 | c10 | c11 | c12 | c11 | c10 | c9 |
| | c8 | c7 | c6 | c5 | c4 | |
| | | c3 | c2 | c1 | | |
| | | | | | | c0 |

c) Độ dốc theo chiều thẳng đứng:

Đối với độ dốc dọc theo chiều ngang và theo chiều thẳng đứng, giải pháp 1 từ Fig.12 được sử dụng

Fig.14. Bộ lọc ALF được cải biến cho sắc độ



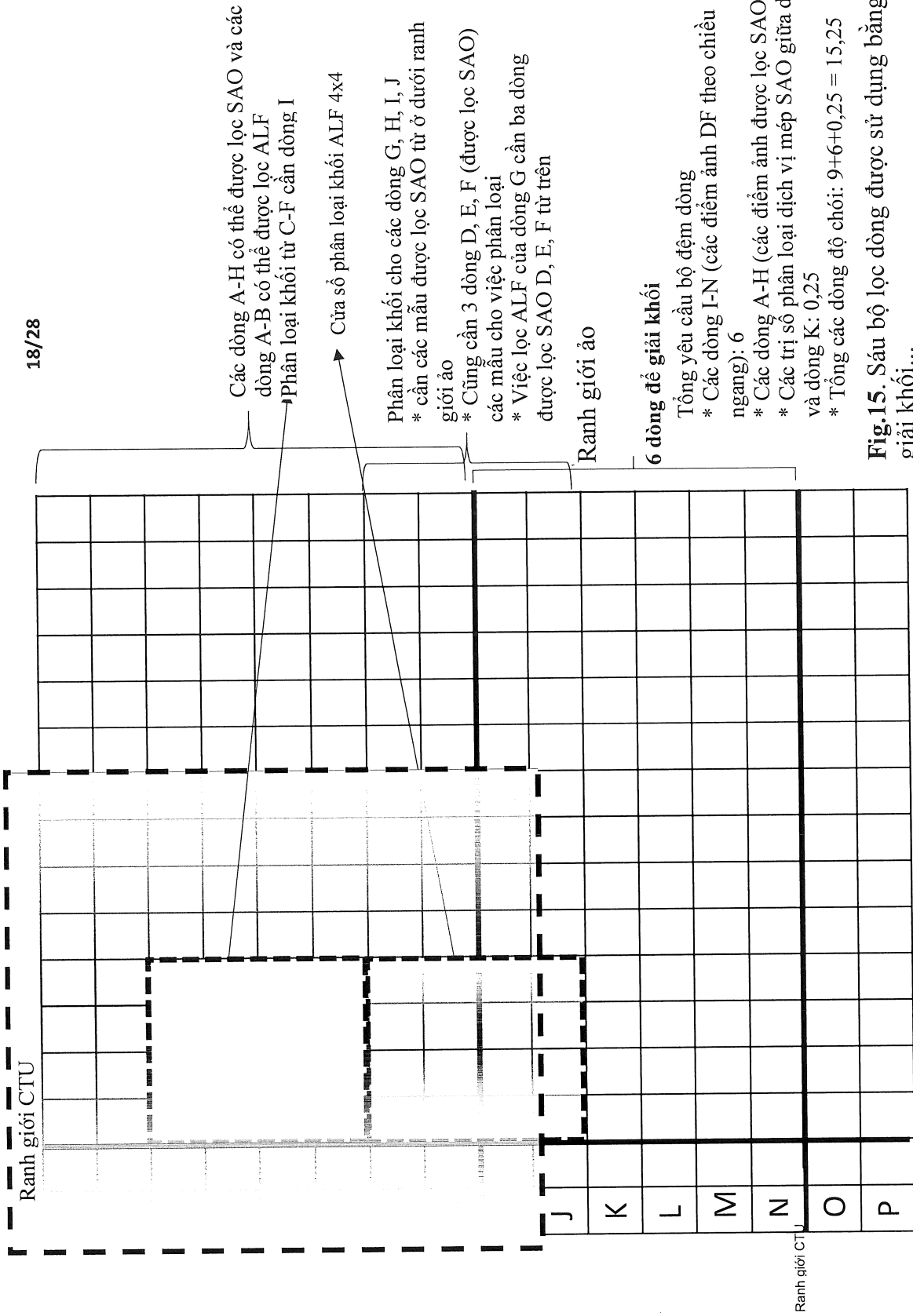


Fig.15. Sáu bộ lọc dòng được sử dụng bằng cách giải khối...

19/28

- * Dòng H được đem tới dòng I để tạo điều kiện thuận lợi cho quyết định phân loại khối của khối 4x4 bắt đầu ở dòng C
- * Các dòng G, H tái sử dụng quyết định phân loại khối từ khối C, D, E, F
- * Các dòng I, J sử dụng quyết định phân loại khối từ khối K, L, M, N
- * Các phiên bản được cắt ngắn của các bộ lọc (các lát 8 và 9) được sử dụng để lọc các dòng F, G, H, I, J, K)

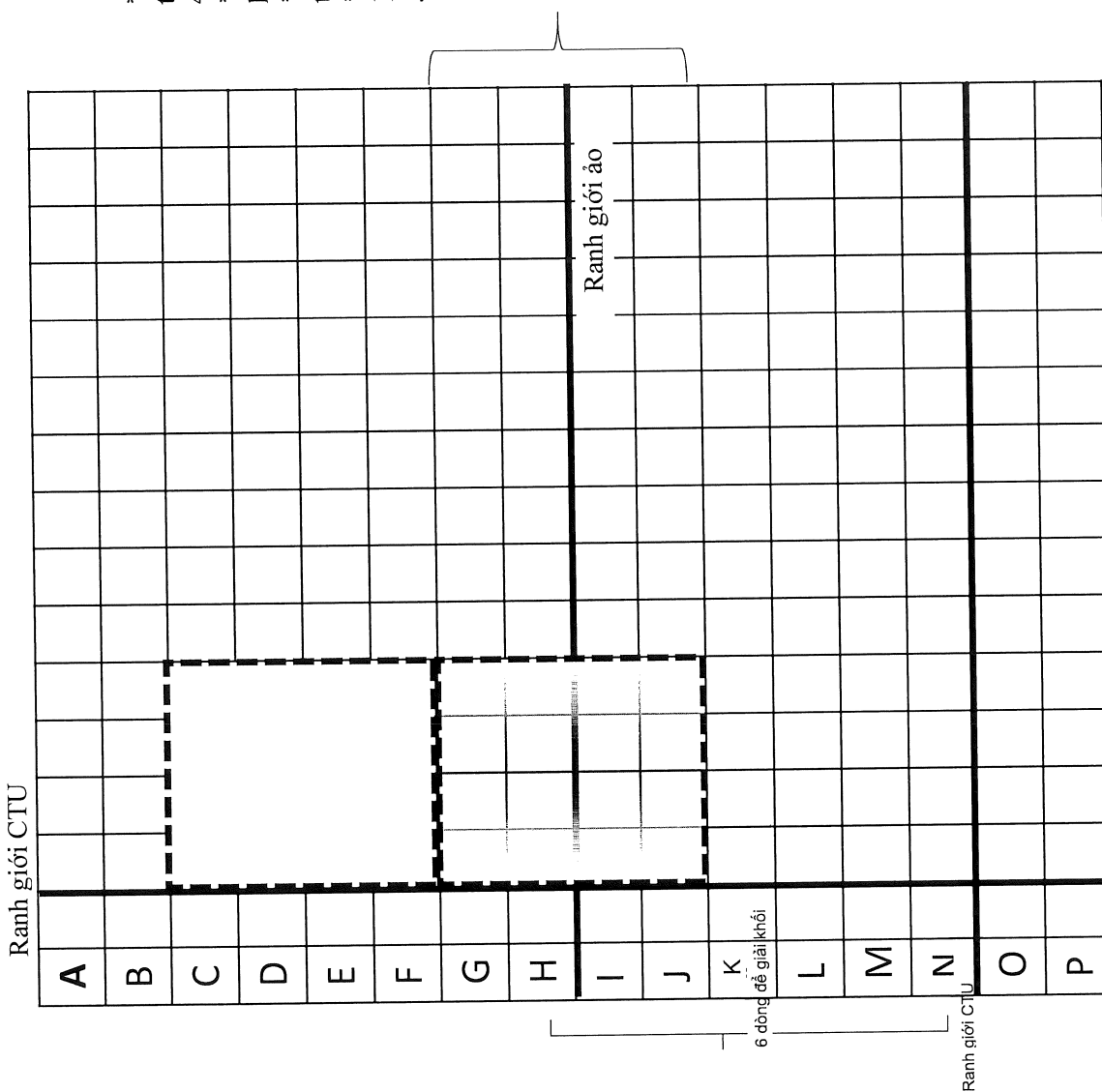
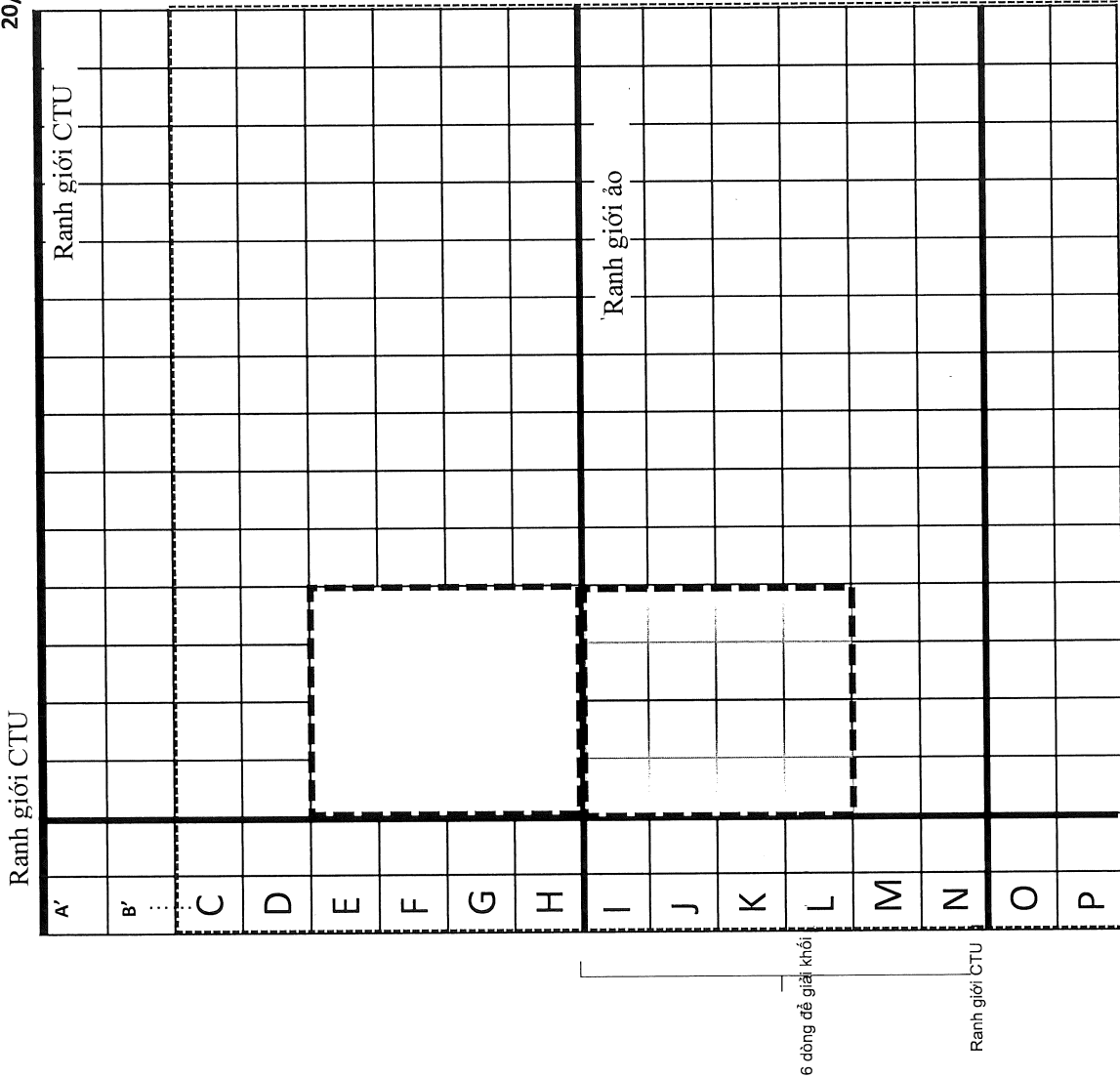


Fig.16. Giải pháp 1. sử dụng lại quyết định phân loại khối ở ranh giới ảo và sử dụng phiên bản được cắt ngắn để lọc

20/28

- Khi ranh giới ảo là 6 dòng ở trên ranh giới CTU, giải pháp sử dụng cách tiếp cận " dịch chuyển lưới".
 - Lưới bộ lọc ALF được dịch chuyển thêm hai dòng ở trục Y để đảm bảo rằng cửa sổ phân loại khối 4x4 đã cho không đi qua ranh giới ảo đã cho
- Nói chung khi kích thước ranh giới ảo không là bội số của 4 (kích thước phân loại khối ALF), lưới có thể được dịch chuyển bởi trị số sau đây trên trục Y:
 - $((\text{Kích thước ranh giới ảo})/4) - 4$
- Ở các ranh giới khung,
 - Ở ranh giới khung trên cùng các dòng A', B' sử dụng lại quyết định phân loại khối từ khối 4x4 bắt đầu ở dòng thứ ba.
 - Tương tự đối với 2 dòng cuối cùng ở ranh giới khung dưới cùng việc phân loại khối sử dụng lại quyết định từ cửa sổ phân loại khối 4x4 ngay ở trên

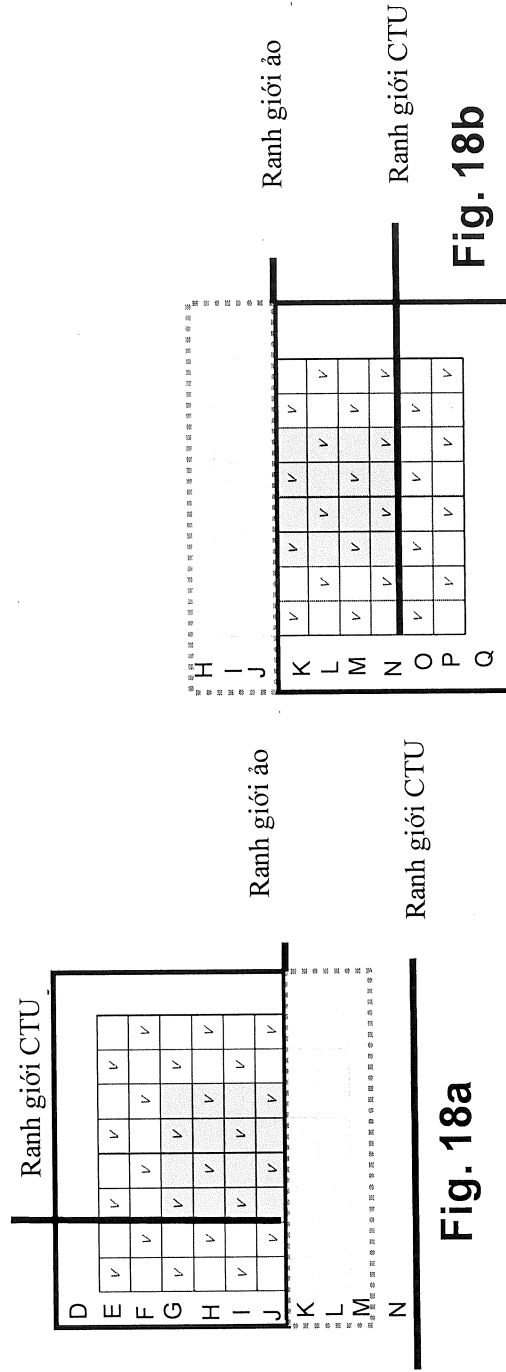


Lưới bộ lọc được dịch chuyển thêm 2 mẫu trên trục y

Fig.17. Giải pháp 2: Cách tiếp cận dịch chuyển lưới cho trường hợp 6 dòng

Fig.18. Phân loại ALF được cải biến để tránh khối độ đệm dòng độ chói ALF

- Phân loại khối ALF được cải biến cho các khối 4x4 ở ranh giới ảo

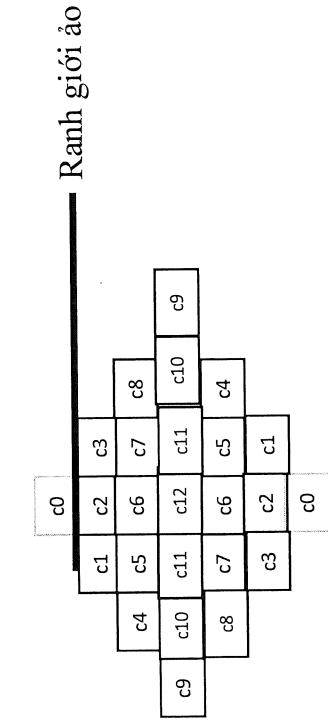


Đối với việc phân loại khối của khối 4x4 trung gian ở trên VB (hình vẽ bên trái):

- * Chỉ sử dụng các mẫu ở trên VB trong việc phân loại (các mẫu ở các dòng E tới H)
 - * Dòng I và J được xem xét cùng nhau trong VTM-3.0, do đó chỉ tới dòng H được sử dụng.
 - * Đệm dòng J tới dòng K cũng để sử dụng các dòng I và J trong việc phân loại
- Đối với việc phân loại khối của khối 4x4 trung gian ở dưới VB (hình vẽ bên phải):
- * Chỉ sử dụng các mẫu ở dưới VB trong việc phân loại (các mẫu ở các dòng P tới M)
 - * Dòng K và L được xem xét cùng nhau trong VTM-3.0, do đó chỉ tới dòng M được sử dụng.
 - * Đệm dòng K tới dòng J cũng để sử dụng các dòng I và J trong việc phân loại

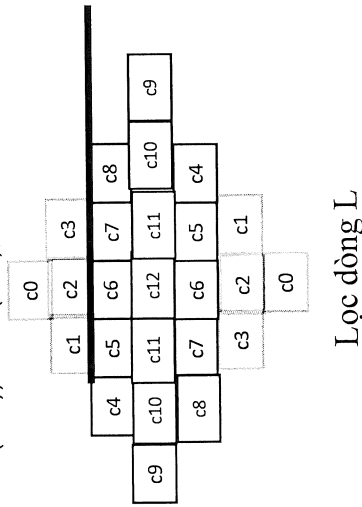
Fig.19a. Bộ lọc ALF được cải biến để tránh khỏi các bộ đệm dòng độ chói ALF

- Đối với việc lọc ALF của các dòng H, I, J (ranh giới ảo ở trên) và các dòng K, L, M (ở dưới ranh giới ảo) việc lọc được ngắt trong các điều kiện nhất định:



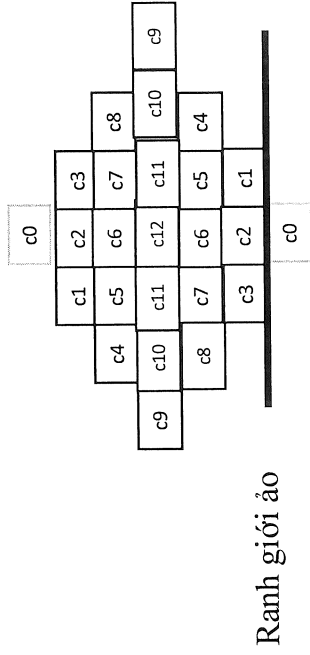
Lọc dòng M

Đối với các dòng M, H, nếu $(2 * \text{abs}(c0) > T * (2 * (\text{abs}(c1) + \text{abs}(c2) + \text{abs}(c3) + \text{abs}(c4) + \text{abs}(c5) + \text{abs}(c6) + \text{abs}(c7) + \text{abs}(c8) + \text{abs}(c9) + \text{abs}(c10) + \text{abs}(c11) + \text{abs}(c12))$, sau đó ngắt việc lọc cho các dòng M và dòng H, theo cách khác bộ lọc như giải pháp 1 được đề cập trên Fig.12a



Lọc dòng L

Đối với các dòng L, I, nếu $(2 * (\text{abs}(c0) + \text{abs}(c1) + \text{abs}(c2) + \text{abs}(c3)) > T * (2 * (\text{abs}(c4) + \text{abs}(c5) + \text{abs}(c6) + \text{abs}(c7) + \text{abs}(c8) + \text{abs}(c9) + \text{abs}(c10) + \text{abs}(c11) + \text{abs}(c12))$, sau đó ngắt việc lọc cho các dòng L và dòng I, theo cách khác bộ lọc như giải pháp 1 được đề cập trên Fig.12a



Lọc dòng H

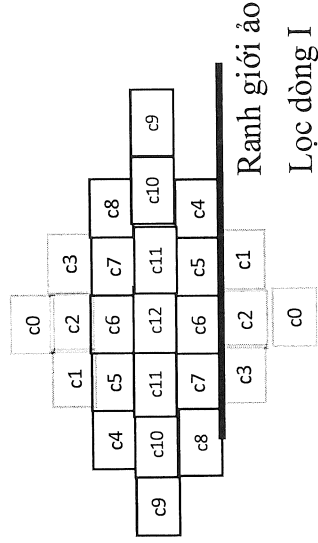
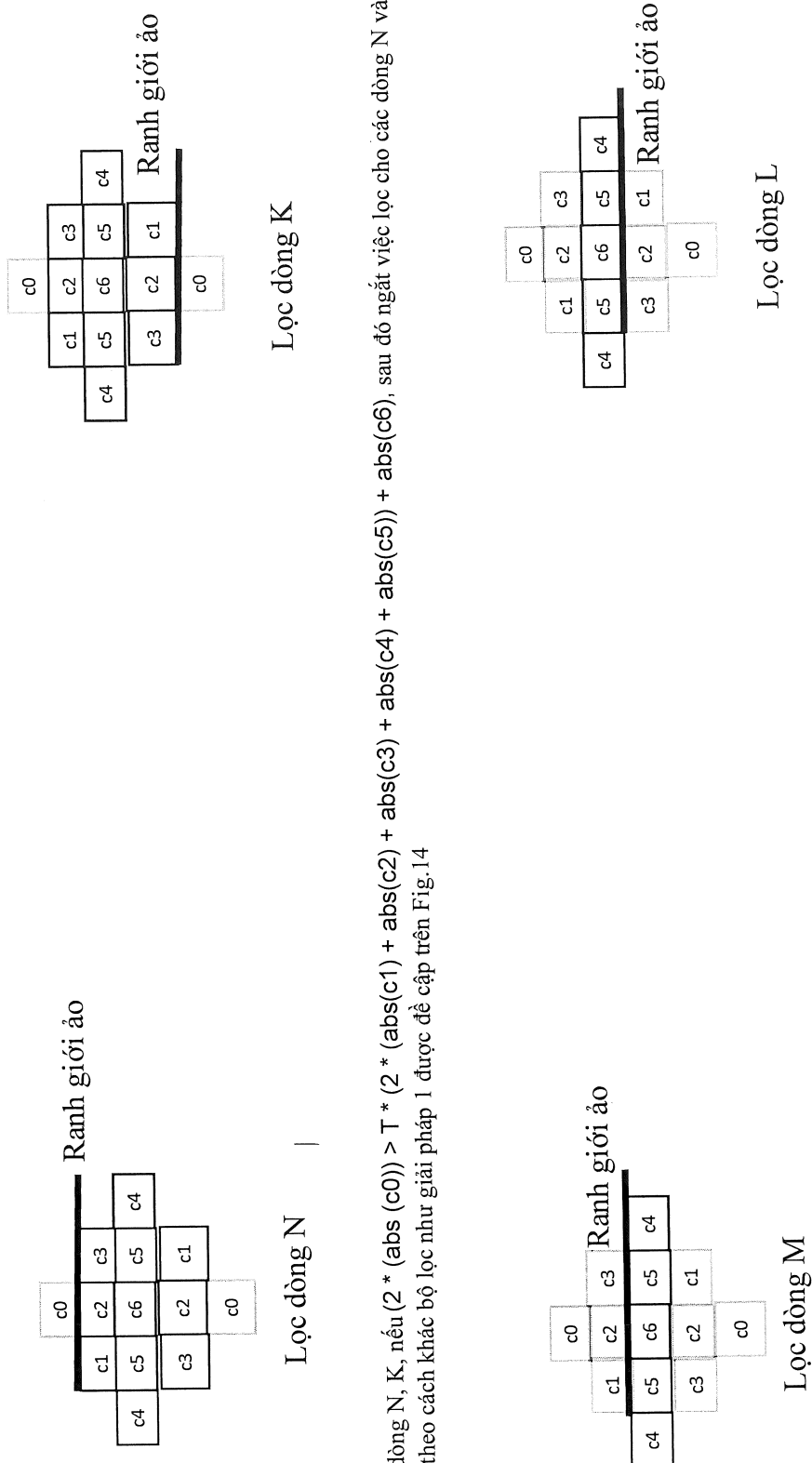


Fig.19b. Bộ lọc ALF được cải biến để tránh khởi các bộ đếm dòng độ chói ALF



Đối với các dòng J, K, nếu $(2 * (abs(c0) + abs(c1) + abs(c2) + abs(c3) + abs(c4) + abs(c5) + abs(c6) + abs(c7) + abs(c8)) > T * (2 * (abs(c9) + abs(c10) + abs(c11)) + abs(c12))$, sau đó ngắt việc lọc cho các dòng J và dòng K, theo cách khác bộ lọc như giải pháp 1 được đề cập trên Fig.12a

Fig.20. Bộ lọc ALF được cài biến cho sắc độ

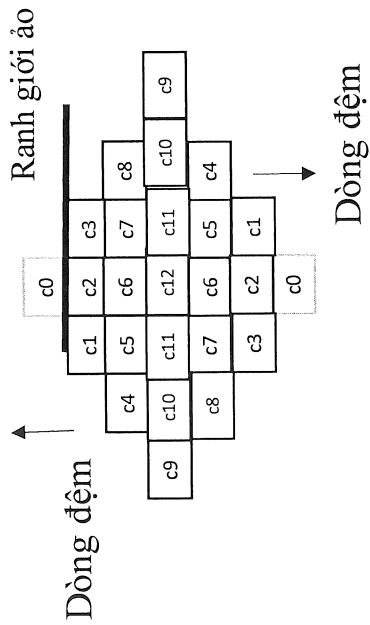


Đối với các dòng N, K, nếu $(2 * (abs(c0)) > T * (2 * (abs(c1) + abs(c2) + abs(c3) + abs(c4) + abs(c5)) + abs(c6))$, sau đó ngắt việc lọc cho các dòng N và dòng K, theo cách khác bộ lọc như giải pháp 1 được đề cập trên Fig.14

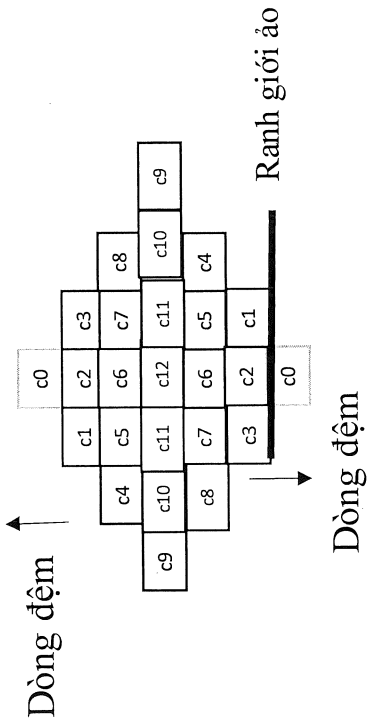
Đối với các dòng M, L, nếu $(2 * (abs(c0) + abs(c1) + abs(c2) + abs(c3)) > T * (2 * (abs(c4) + abs(c5)) + abs(c6))$, sau đó ngắt việc lọc cho các dòng M và dòng L, theo cách khác bộ lọc như giải pháp 1 được đề cập trên Fig.14

Fig.21a. Bộ lọc ALF được cải biến

- Đệm đối xứng ở ranh giới ảo



Lọc dòng M



Lọc dòng H

Fig.21b. Bộ lọc ALF được cải biến

- Đệm đối xứng ở ranh giới ảo

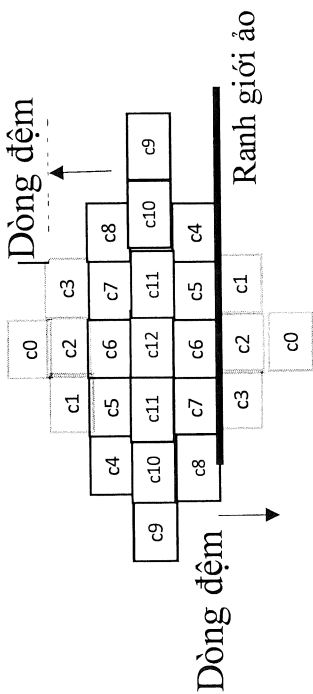
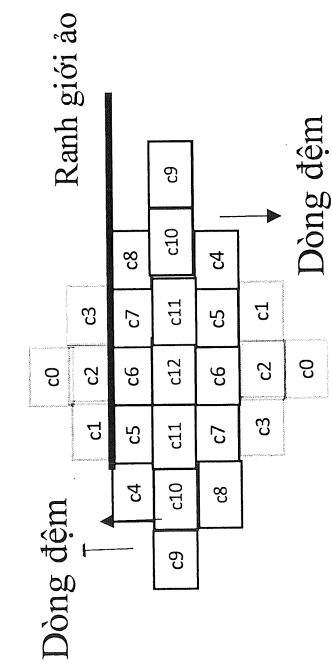


Fig.21c. Bộ lọc ALF được cải biến

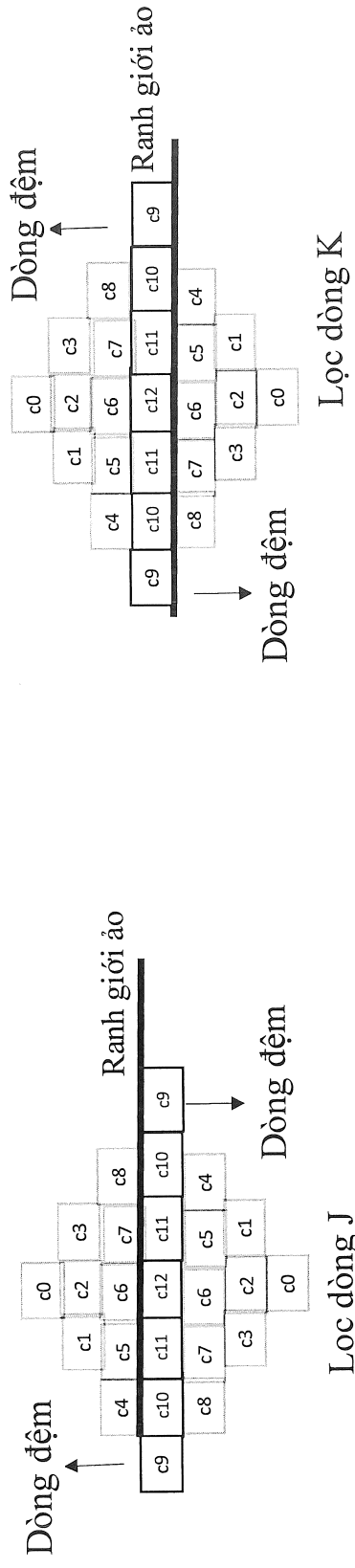
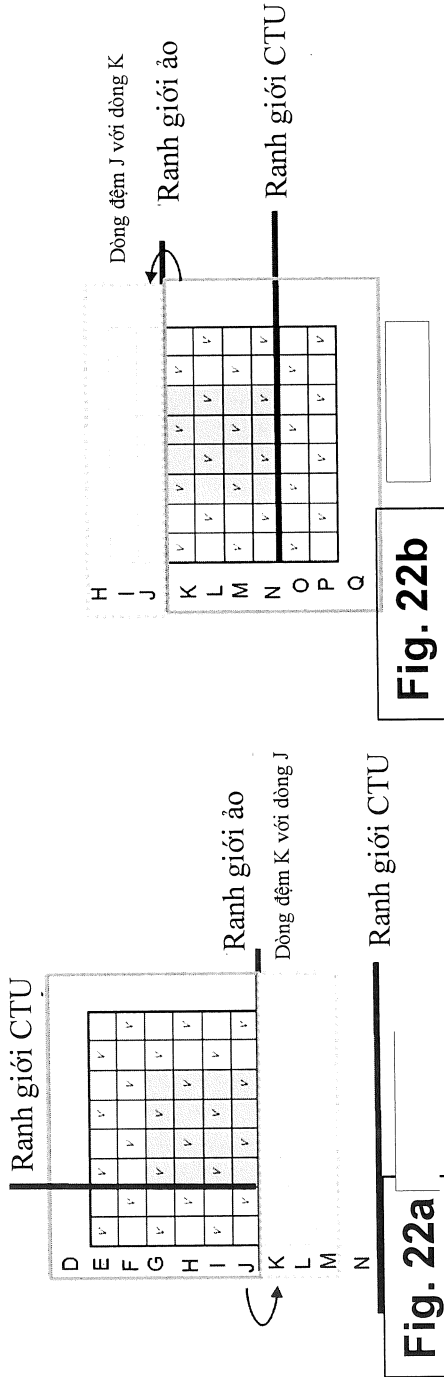


Fig.22. Phân loại khối ALF được cải biến



- Đối với khối 4x4 ở trên VB:
 - Chỉ các mẫu ở trên VB góp phần vào việc phân loại khối ALF và ngược lại
 - Đối với việc tính toán độ dốc của các mẫu ở dòng J, dòng K được đệm với dòng J
- Đối với khối 4x4 ở dưới VB:
 - Chỉ các mẫu ở dưới VB góp phần vào việc phân loại khối ALF và ngược lại
 - Đối với việc tính toán độ dốc của các mẫu ở dòng J, dòng K được đệm với dòng J
- Việc tính toán hoạt động sử dụng chuẩn hóa khác khi ít mẫu hơn được sử dụng cho việc phân loại khối.
 - Do đó phương trình sau đây được sử dụng trong việc tính toán hoạt động:
 - $avgVar[x][y] = varTab[Clip3(0, 15, (sumOfHV[x >> 2][y >> 2] * Ac) >> (3 + BitDepth_y))]$, trong đó trị số Ac là 96 đối với các khối gần ranh giới ảo và 64 đối với các khối còn lại