**CHƯƠNG 12: New Video Coding Standards: H.264 and H.265**

**12.1 H.264**

Nhóm Video chung (JVT) của ISO/IEC MPEG và ITU-T VCEG (Nhóm chuyên gia mã hóa video) đã phát triển tiêu chuẩn nén video H.264. Trước đây, tiêu chuẩn này được biết đến với tên gọi làm việc là “H.26L”. Bản thảo cuối cùng của phiên bản đầu tiên của H.264 đã được hoàn thành vào tháng 5 năm 2003 [1].

H.264 còn được gọi là MPEG-4 Phần 10, AVC (Mã hóa video nâng cao) [2–4]. Nó thường được gọi là tiêu chuẩn mã hóa video H.264/AVC (hoặc H.264/MPEG-4 AVC).

H.264/AVC cung cấp hiệu suất mã hóa video cao hơn, khả năng nén tốt hơn tới 50% so với MPEG-2 và tốt hơn tới 30% so với H.263+ và MPEG-4 Advanced Simple- Profile, trong khi vẫn duy trì cùng chất lượng của video được nén. Nó bao gồm một loạt các ứng dụng, từ tốc độ bit cao đến tốc độ bit rất thấp. Các tính năng cốt lõi của H.264 được cải thiện đáng kể, cùng với các công cụ mã hóa mới mang lại sự cải thiện đáng kể về tỷ lệ nén, khả năng phục hồi lỗi và chất lượng chủ quan so với các tiêu chuẩn ITU-T và MPEG hiện có. Kể từ đó, nó đã trở thành tiêu chuẩn mặc định cho nhiều ứng dụng khác nhau, ví dụ như đĩa Blu-ray, chương trình phát sóng HDTV, video phát trực tuyến trên Internet, phần mềm web như Flash và Silverlight, và các ứng dụng trên thiết bị di động và cầm tay.

Tương tự như các tiêu chuẩn nén video trước đây, H.264 chỉ định một lược đồ mã hóa lai dựa trên khối hỗ trợ bù chuyển động và mã hóa biến đổi. Một lần nữa, mỗi hình ảnh có thể được tách thành các khối vĩ mô (khối 16×16) và các lát cắt có kích thước tùy ý có thể nhóm nhiều khối vĩ mô thành các đơn vị độc lập. Bộ mã hóa H.264/AVC cơ bản được thể hiện trong Hình 12.1. Các tính năng chính của H.264/AVC là:

* Biến đổi số nguyên trong các khối 4×4. Độ phức tạp thấp, không bị trôi.
* Bù chuyển động có kích thước khối thay đổi, từ 16×16 đến 4×4 trong hình ảnh độ sáng.
* Độ chính xác một phần tư pixel trong các vectơ chuyển động, đạt được bằng cách nội suy.
* Bù chuyển động của nhiều hình ảnh tham chiếu. Không chỉ là các khung P hoặc B để ước tính chuyển động.
* Dự đoán không gian theo hướng cho các khung bên trong.

A diagram of a machine

Description automatically generated

* Lọc giải khối vòng lặp.
* Mã hóa chiều dài biến thích ứng theo ngữ cảnh (CAVLC) và Mã hóa số học nhị phân thích ứng theo ngữ cảnh (CABAC).
* Chống lỗi dữ liệu và mất dữ liệu tốt hơn, linh hoạt hơn trong việc đồng bộ hóa và chuyển đổi luồng video do các bộ giải mã khác nhau tạo ra. Bộ giải mã có năm khối chính sau:
* Giải mã entropy
* Lượng tử hóa ngược và biến đổi các pixel còn lại
* Bù chuyển động hoặc dự đoán nội bộ
* Tái tạo
* Lọc giải khối vòng lặp trên các pixel được tái tạo.

**12.1.1 Bù chuyển động**

Tương tự như MPEG-2 và H.263, H.264 sử dụng công nghệ mã hóa lai, tức là kết hợp giữa dự đoán chuyển động giữa các hình ảnh và dự đoán không gian trong hình ảnh, và mã hóa chuyển đổi trên các lỗi còn lại.

**Bù chuyển động theo kích thước khối thay đổi**

Như trước đây, ước tính chuyển động giữa các khung trong H.264 cũng dựa trên khối. Theo mặc định, các khối vĩ mô có kích thước 16 × 16. Một khối vĩ mô cũng có thể được chia thành bốn phân vùng 8 × 8. Trong khi tiến hành ước tính chuyển động, mỗi khối vĩ mô hoặc mỗi các phân vùng có thể được phân đoạn thành các phân vùng nhỏ hơn như thể hiện trong Hình 12.2. Bốn tùy chọn trên cùng là từ khối macro 16 × 16 (còn gọi là Kiểu M) và bốn tùy chọn dưới cùng là từ mỗi phân vùng 8 × 8 (còn gọi là Kiểu 8 × 8).

A group of squares with numbers

Description automatically generated

**Hình 12.2** Phân đoạn khối macro để ước tính chuyển động trong H.264. Phân đoạn t*rên* cùng của khối macro. Phân đoạn *dưới* cùng của phân vùng 8 × 8

**Độ chính xác của một phần tư điểm ảnh**

Độ chính xác của bù chuyển động là độ chính xác một phần tư pixel trong hình ảnh độ sáng. Hình 12.3 minh họa cách các giá trị pixel ở vị trí một nửa pixel và một phần tư pixel có thể được suy ra bằng cách nội suy. Để suy ra các giá trị pixel ở vị trí một nửa pixel được dán nhãn b và h, các giá trị trung gian b1 và h1 trước tiên được suy ra bằng cách áp dụng các bộ lọc sáu lần nhấn như bên dưới.

= *E* − 5*F* + 20*G* + 20*H* − 5*I* + *J*

= *A* − 5*C* + 20*G* + 20*M* − 5*R* + *T*.

Giá trị của b và h sau đó được lấy theo công thức sau, được cắt theo phạm vi từ 0–255.

*b* = (+ 16) ⇒ 5

*h* = (+ 16) ⇒ 5.

Ký hiệu đặc biệt “⇒” trong các công thức trên biểu thị một phép dịch chuyển sang phải. Ví dụ, b = ( + 16) ⇒ 5 tương đương với b = round(/) = round(/32). Tuy nhiên, phép dịch chuyển hiệu quả hơn nhiều so với lệnh gọi hàm round.

Điểm ảnh giữa 'j' được lấy bằng

*j*1 = −+20 +20 −5 +,

A screenshot of a crossword puzzle

Description automatically generated

**Hình 12.3** Nội suy cho các mẫu phân số trong H.264. *Các chữ cái viết ho*a cho biết các pixel trên lưới hình ảnh. *Các chữ cái viết thường* cho biết pixel ở vị trí nửa pixel và một phần tư pixel

trong đó các giá trị trung gian , , , và được suy ra theo cách tương tự như . Sau đó, giá trị của j thu được bằng cách sau, được cắt vào phạm vi từ 0 đến 255.

*j* = ( *j*1 + 512) ⇒ 10.

Các giá trị pixel tại các vị trí pixel một phần tư được gắn nhãn *a*, *c*, *d*, *n*, *f*, *i*, *k* và *q* thu được bằng cách lấy trung bình các giá trị của hai pixel gần nhất tại vị trí số nguyên và nửa pixel. Chẳng hạn,

*a* = (*G* + *b* + 1) ⇒ 1.

Cuối cùng, các giá trị pixel tại các vị trí pixel một phần tư được gắn nhãn *e*, *g*, *p* và *r* thu được bằng cách lấy trung bình giá trị của hai pixel gần nhất ở vị trí nửa pixel theo hướng chéo. Chẳng hạn,

*e* = (*b* + *h* + 1) ⇒ 1.

**Additional Options in Group of Pictures**

Như thể hiện trong Hình 11.3, trong các tiêu chuẩn MPEG trước đây, *Nhóm ảnh (GOP)* bắt đầu và kết thúc bằng khung chữ I. Ở giữa nó có khung P và khung B. Khung chữ I hoặc khung chữ P có thể được sử dụng làm *khung tham chiếu*. Các khối vĩ mô trong khung P được dự đoán bằng dự đoán chuyển tiếp và các khối vĩ mô trong khung B được dự đoán bằng sự kết hợp giữa dự đoán tiến và dự đoán lùi. H.264 sẽ tiếp tục hỗ trợ cấu trúc GOP "cổ điển" này. Ngoài ra, nó sẽ hỗ trợ các cấu trúc GOP sau đây.

*Không có khung B*

Dự đoán các macroblocks trong khung B phải chịu nhiều độ trễ hơn và yêu cầu lưu trữ nhiều hơn cho các khung I và P cần thiết vì dự đoán hai chiều. Trong tùy chọn này, chỉ cho phép khung I và P. Mặc dù hiệu quả nén tương đối thấp, nhưng nó phù hợp hơn cho một số ứng dụng nhất định, ví dụ: hội nghị truyền hình trong đó độ trễ tối thiểu là mong muốn hơn. Điều này tương thích với các mục tiêu của Hồ sơ cơ sở hoặc Hồ sơ cơ sở bị ràng buộc của H.264.

*Nhiều khung tham chiếu*

Để tìm ra kết quả phù hợp nhất cho mỗi macroblock trong khung chữ P, H.264 cho phép tối đa *N* khung tham chiếu. Hình 12.4 minh họa một ví dụ trong đó *N* = 4. Khung tham chiếu cho *P1 là I0, các khung tham chiếu cho là , và ,,. . . Đối với P4, các khung tham chiếu là , , và . Mặc dù điều này cải thiện hiệu quả nén, nhưng nó đòi hỏi nhiều tính toán hơn trong ước tính chuyển động tại bộ mã hóa. Hơn nữa, nó đòi hỏi một bộ đệm lớn hơn để lưu trữ tối đa N* khung hình tại bộ mã hóa và bộ giải mã.

*Cấu trúc dự đoán phân cấp*

Cấu trúc dự đoán phân cấp cũng được cho phép theo các tùy chọn dự đoán linh hoạt của H.264. Ví dụ: chúng ta có thể có GOP bắt đầu và kết thúc bằng  *và . Ở giữa có 11 khung B liên tiếp, đến . Đầu tiên, được dự đoán sử dụng và làm tham chiếu. Tiếp theo, được dự đoán bằng cách sử dụng và B6, và B9 được dự đoán bằng cách sử dụng B6 và I12 làm tài liệu tham khảo. Cuối cùng, B1 và B2 được dự đoán bằng I0 và B3, B4 và B5 được dự đoán bằng B3 và B6, v.v. Cấu trúc phân cấp này có thể được xem là có các lớp. Trong ví dụ này, Lớp 0: I0, I12; Lớp 1: B6; Lớp 2: , B9; Lớp 3: B1, B2, B4, B5, B7, B8, B10, B11. Thông thường, các tham số lượng tử hóa ngày càng lớn hơn sẽ được liên kết với các lớp cao hơn để kiểm soát hiệu quả nén. Điều này được chứng minh là hiệu quả hơn IBBP. . . Cấu trúc được sử dụng trong các tiêu chuẩn mã hóa video trước đây để dự đoán thời gian.*

**12.1.2 IntegerTransform**

Như trong các tiêu chuẩn mã hóa video trước đây, H.264 sử dụng mã hóa Transform sau khi thu được các macroblocks dif-ference. Một trong những tính năng quan trọng nhất trong H.264/AVC là sử dụng *Biến đổi số nguyên*.

Biến đổi Cosin rời rạc (DCT) trong các tiêu chuẩn mã hóa video trước đây được biết là gây ra *sự thay đổi dự đoán* do tính toán dấu phẩy động và làm tròn

A diagram of a diagram

Description automatically generated

**Fig.12.4** Anillustrationofmulti-referenceframes

Lỗi trong biến đổi và biến đổi nghịch đảo. Nó cũng chậm do nhiều phép nhân dấu phẩy động. H.264 cho phép 4 × 4 khối và các dự đoán khác nhau; Ngay cả mã hóa nội bộ cũng dựa vào dự đoán không gian theo sau là mã hóa biến đổi. Do đó, nó rất nhạy cảm với sự thay đổi dự đoán. Ví dụ: khối 4 × 4 có thể được dự đoán từ một khối nội bộ lân cận và khối lân cận có thể được dự đoán từ một khối lân cận khác, v.v. Kết quả là, sự thay đổi dự đoán có thể được tích lũy, gây ra một lỗi lớn.

Với các sơ đồ dự đoán P và I mạnh mẽ và chính xác trong H.264, người ta nhận ra rằng mối tương quan không gian trong các pixel dư thường rất thấp. Do đó, một độ chính xác nguyên đơn giản 4 × 4 DCT là đủ để nén năng lượng. Số học số nguyên cho phép biến đổi nghịch đảo chính xác trên tất cả các bộ xử lý và loại bỏ các vấn đề không khớp bộ mã hóa / giải mã trong các codec dựa trên biến đổi trước đó. H.264 cũng cung cấp một sơ đồ lượng tử hóa với *các kích thước bước phi tuyến để thu được tỷ lệ chính xác ở cả hai đầu cao và thấp của thang lượng tử hóa.*

4 × 4 biến đổi trong H.264 xấp xỉ DCT và IDCT. Chúng chỉ liên quan đến các phép toán số học số nguyên, 16 bit. Chúng có thể được thực hiện rất hiệu quả. Như đã thảo luận trong Chap.8, DCT 2D có thể tách rời: nó có thể được thực hiện bằng hai biến đổi 1D liên tiếp, tức là theo hướng thẳng đứng trước và sau đó là hướng ngang. Điều này có thể được thực hiện bằng hai phép nhân ma trận: **F = T × f × TT** , trong đó **f** là dữ liệu đầu vào, **F** là dữ liệu được chuyển đổi và **T** là cái gọi là ma trận DCT. Ma trận DCT là trực giao, tức là tất cả các hàng đều trực giao và chúng

Tất cả đều có định mức 1.Ma trận DCT 4 × 4 *T4 có thể được viết là:*

A black and white image of a mathematical equation

Description automatically generated

where a = ½, b= cos , and c = cos .

Để rút ra một biến đổi số nguyên 4 × 4 được chia tỷ lệ xấp xỉ **T4**, chúng ta có thể chỉ cần chia tỷ lệ các mục nhập của **T4** lên và làm tròn chúng thành các số nguyên gần nhất [5]:

**H** = round(α · **T4**).

Khi α = 26, ta có

A number with black numbers

Description automatically generated with medium confidence

Similar to *T*4 , this matrix has some nice properties: all its rows are orthogonal; they also have the same norm, because 4×132 = 2×(172 +72). However, this matrix has a dynamic range gain of 52 (i.e., 4×13). Since it is used twice in **F** = **H**×**f** ×in order to transform the columns and then the rows of **f**, the total gain is 522 = 2704. Because log2 2704 ≡ 11.4, it would require 12 more bits for coefficients in **F** than the number of bits required for data in the original **f**. This would make the 16-bit arithmetic insufficient, and would require 32-bit arithmetic.

Hence, it is proposed in [5] that α = 2.5 in Eq. 12.1. This yields

A number and a mathematical equation

Description automatically generated with medium confidence

Ma trận **H** mới này vẫn trực giao, mặc dù các hàng của nó không còn giống nhau nữa. chuẩn mực. Để khôi phục tính chất trực chuẩn, chúng ta có thể rút ra một cách đơn giản như sau ma trận bằng cách chia tất cả các mục trong hàng trong H cho , trong đó H\_ij , là mục nhập thứ j của hàng thứ i trong H. Tuy nhiên, đây sẽ không còn là một phép biến đổi số nguyên nữa.

A number of mathematical equations

Description automatically generated with medium confidence

Trong việc thực hiện H.264, vấn đề chuẩn hóa này bị hoãn lại. Nó được hợp nhất vào quá trình lượng tử hóa trong bước tiếp theo, vì chúng ta có thể chỉ cần điều chỉnh các giá trị trong ma trận lượng tử hóa để đạt được cả mục tiêu lượng tử hóa và chuẩn hóa.

Bởi vì **H** là trực giao, chúng ta có thể đã sử dụng làm biến đổi nghịch đảo , miễn là vấn đề chuẩn hóa được quan tâm. Một lần nữa, vì chúng ta cũng có thể giải quyết vấn đề này sau này ở bước khử lượng tử hóa, chúng ta chỉ cần giới thiệu một biến đổi nghịch đảo ad hoc Hinv để sử dụng:

A number and numbers on a white background

Description automatically generated with medium confidence

**Hinv** về cơ bản là **HT** , nhưng với cột thứ hai và thứ tư được thu nhỏ lại 1/2. Điều này là do dải động của dữ liệu đầu vào vào **Hinv** lớn hơn dải động của **H**. Do đó, việc giảm tỷ lệ hơn nữa được áp dụng cho các cột nếu không sẽ có mức tăng dải động cao hơn.

H.264 cũng hỗ trợ the 8×8 số nguyên biến đổi . Nó như trong Eq.12.4.Chúng tôi sẽ sử dụng **H**, phiên bản 4 × 4, trong các cuộc thảo luận của chúng tôi trừ khi có ghi chú khác.

A number with numbers on it

Description automatically generated with medium confidence

**12.1.3 QuantizationandScaling**

Như trong các tiêu chuẩn nén video trước đây, *lượng tử hóa* được sử dụng sau khi chuyển đổi. Thay vì thiết kế ma trận lượng tử hóa đơn giản, H.264 có thiết kế tinh vi hơn[5] hoàn thành cả hai nhiệm vụ lượng tử hóa và chia tỷ lệ (chuẩn hóa) của H.

**Biến đổi số nguyên và lượng tử hóa**

Cho f là ma trận đầu vào 4 × 4 và Fˆ là đầu ra được biến đổi và sau đó lượng tử hóa. Việc biến đổi, chia tỷ lệ và lượng tử hóa số nguyên chuyển tiếp được thực hiện theo:

A black text on a white background

Description automatically generated

Ở đây, '×' biểu thị phép nhân ma trận, trong khi '·' biểu thị phép nhân từng phần tử. **H** giống như trong Eq. 12.2. **Mf** là ma trận lượng tử hóa 4 × 4 có nguồn gốc từ **m** là ma trận 6 × 3 (xem Bảng 12.1). *Q P* là *tham số quantizaton*.

Đối với 0 *QP* <6, chúng tôi có

A table of maths with black letters

Description automatically generated with medium confidence

Đối với *Q P* ⊕ 6, mỗi phần tử **m**(*Q P*, *k*) được thay thế bằng **m**(*Q P*%6, *k*)/

A table of numbers and a number

Description automatically generated with medium confidence

Việc lượng tử hóa được theo sau bởi một bước mở rộng quy mô khác, có thể được thực hiện bằng cách dịch chuyển phải "⇒15".

**Biến đổi số nguyên nghịch đảo và khử lượng tử hóa**

Hãy để **f** ̃ là kết quả phi lượng tử hóa và sau đó biến đổi nghịch đảo. Việc chia tỷ lệ, khử lượng tử hóa và biến đổi số nguyên nghịch đảo được thực hiện theo:



**Hinv** giống như trong Eq. 12.3. **Vi** là ma trận khử lượng tử hóa 4 × 4 có nguồn gốc từ **v** là ma trận 6 × 3 (xem Bảng 12.2). Đối với 0 *Q P* < 6, chúng ta có

A group of black letters

Description automatically generated

Đối với *Q P* ⊕ 6, mỗi phần tử **v**(*Q P*, *k*) được thay thế bằng **v**(*Q P*%6, *k*) · 2≤Q P/6∗.

Việc phi lượng tử hóa cũng được theo sau bởi một bước mở rộng quy mô khác, có thể được ngụ ý bằng sự dịch chuyển phải "⇒6".

**12.1.4 Ví dụ về biến đổi và lượng tử hóa số nguyên H.264**

Phần này cho thấy một số ví dụ về biến đổi và lượng tử hóa số nguyên H.264 và nghịch đảo của chúng bằng cách sử dụng các tham số lượng tử hóa khác nhau *QP. Dữ liệu đầu vào là ma trận 4 × 4* **f** với các giá trị tùy ý, các hệ số được chuyển đổi và sau đó lượng tử hóa được tính bằng **Fˆ . Mf** và **Vi** là ma trận lượng tử hóa và phi lượng tử hóa, và *f* ̃ là đầu ra phi lượng tử hóa và sau đó biến đổi nghịch đảo. Để so sánh, chúng tôi cũng sẽ hiển thị tổn thất nén ε = **f** − **f** ̃.

Để cải thiện hiệu suất biến dạng tốc độ, H.264 áp dụng lượng tử hóa vùng chết (còn được gọi là *midtread* như đã thảo luận trong Chap. 8). Nó có thể được mô tả như một hàm biến một số thực *x* thành một số nguyên *Z* , như sau:

A black and white symbol

Description automatically generated

trong đó *x* là giá trị được chia tỷ lệ như đã thảo luận trong phần trước. Theo mặc định, *b* = 0,5, hàm trên sau đó tương đương với hàm *tròn* như được chỉ định trong Eq. (12,5) hoặc (12,7). Để giảm thiểu sai số lượng tử hóa, H.264 thực sự áp dụng lượng tử hóa thích ứng, trong đó chiều rộng của vùng chết có thể được kiểm soát bởi *b*. Ví dụ: *b* = 1/3 cho mã hóa nội bộ và *b* = 1/6 cho mã hóa liên kết. Để đơn giản, trong các ví dụ sau, chúng ta chỉ sử dụng *b* = 0,5.

Hình 12.5a cho thấy kết quả khi *QP* = 0. Đây là tùy chọn cung cấp tổn thất nén ít nhất có thể. Các giá trị của và được xác định theo Eqs. (12.6) và (12.8). Vì không có sự thu nhỏ các giá trị F trong bước lượng tử hóa, f ̃ được tái tạo hoàn toàn giống với **f**, tức là **f** ̃ = **f**.

Hình 12,5b cho thấy kết quả khi *Q P* = 6. Như mong đợi, so với các mục ma trận tương quan cho *Q P* = 0, các giá trị tính bằng giảm một nửa và các giá trị trong **Vi** là khoảng hai lần. Hệ số lượng tử hóa hiện nay xấp xỉ 1,25 (tức là tương đương với *qstep* ≡ 1,25). Kết quả là, **f** ̃ √ = **f**. Sự mất mát nhẹ có thể được quan sát thấy trong ε.

Tương tự, khi *Q P* = 18, kết quả nằm trong Hình 12,5c. Các giá trị tính bằng và các giá trị trong được giảm thêm hoặc tăng tương ứng, theo hệ số 4 so với các giá trị của *Q P* = 6. Hệ số lượng tử hóa xấp xỉ 5 (tức là tương đương với *qstep* ≡ 5). Sự mất mát thể hiện trong ε bây giờ là đáng kể hơn.

Có lẽ một kết quả thú vị hơn được thể hiện trong Hình 12.5d khi *QP* = 30. Hệ số lượng tử hóa hiện nay xấp xỉ 20 (tức là tương đương với *qstep* ≡ 20). Ngoại trừ giá trị DC lượng tử hóa lớn hơn tính bằng *Fˆ vẫn bằng không, tất cả các hệ số AC đều trở thành không. Kết quả là, f ̃ được xây dựng lại* có 80 trong tất cả các mục. Tổn thất nén trong ε không còn được chấp nhận.

**12.1.5 Mã hóa nội bộ**

H.264 khai thác dự *đoán không gian nhiều* hơn so với các tiêu chuẩn video trước đây như MPEG-2 và H.263 +. Các macroblock được mã hóa nội bộ được dự đoán bằng cách sử dụng một số pixel được tái tạo lân cận (sử dụng các pixel được tái tạo nội bộ hoặc liên mã hóa).

A table of numbers and equations

Description automatically generated

**Fig. 12.5** Examples of H.264 integer transform and quantization with various QPs: **a** QP=0; **b** QP=6; **c** QP=18; **d** QP=30

A diagram of a sunburst

Description automatically generated

**Hình 12.6** Dự đoán khung hình nội bộ H.264. **dự** đoán Intra\_4 × 4 sử dụng các mẫu lân cận từ A đến M. **b** Tám hướng cho intra\_4 × 4 dự đoán

A diagram of a mode

Description automatically generated with medium confidence

**Hình 12.7** Sáu trong số chín chế độ dự đoán đầu tiên intra\_ 4×4 trong dự đoán nội khung H.264

Tương tự như kích thước khối thay đổi trong bù chuyển động giữa các hình ảnh, các kích thước khối dự đoán nội bộ khác nhau (4 × 4 hoặc 16 × 16) có thể được chọn cho mỗi macroblock được mã hóa nội bộ. Như thể hiện trong Hình 12.6, có chín chế độ dự đoán cho 4 × 4 khối trong Intra\_4 × 4. Hình 12.7 minh họa thêm sáu trong số chúng.

Các chế độ dự đoán 0, 1, 3 và 4 khá đơn giản. Ví dụ: trong Chế độ 0, giá trị của pixel A sẽ được sử dụng làm giá trị dự đoán cho tất cả các pixel trong cột đầu tiên (tức là cột bên dưới A), giá trị của pixel B sẽ được sử dụng làm giá trị dự đoán cho tất cả các pixel trong cột thứ hai, v.v. Chế độ 2 (DC) là một chế độ đặc biệt trong đó giá trị trung bình của tám hàng xóm được mã hóa trước đó (A đến D và I đến L) được sử dụng làm giá trị dự đoán cho tất cả các pixel trong khối 4 × 4.

Chế độ 5 (Dọc-Phải), Chế độ 6 (Ngang-Xuống), Chế độ 7 (Dọc-Trái) và Chế độ 8 (Ngang-Lên) cũng tương tự nhau. Như thể hiện trong Hình 12.7, hướng dự đoán trong Chế độ 5 là xuống và sang phải theo tỷ lệ 2: 1 (tức là 2 pixel xuống và 1 pixel sang phải hoặc khoảng 26,6 độ sang phải). Điều này hoạt động tốt cho các pixel ở hàng thứ hai và thứ tư. Ví dụ: nếu các chỉ số hàng và cột của khối 4 × 4 nằm trong phạm vi 0..3, thì giá trị dự đoán cho các pixel tại [1, 1] và [3, 2] sẽ là giá trị của A. Tuy nhiên, các pixel ở hàng thứ nhất và thứ ba sẽ không thể sử dụng một giá trị duy nhất từ bất kỳ hàng xóm nào được mã hóa trước đó. Thay vào đó, nó phải được ngoại suy từ hai trong số họ. Ví dụ: giá trị dự đoán cho các pixel tại [0, 0] và [2, 1] sẽ là sự kết hợp tỷ lệ thuận của các giá trị của M và A.

Đối với mỗi chế độ dự đoán, giá trị dự đoán và giá trị thực tế sẽ được so sánh để tạo ra lỗi dự đoán. Chế độ tạo ra lỗi dự đoán ít nhất sẽ được chọn làm chế độ dự đoán cho khối. Các lỗi dự đoán (dư) sau đó được gửi đến mã hóa chuyển đổi nơi biến đổi số nguyên 4 × 4 được sử dụng. Mỗi khối 4 × 4 trong một macroblock có thể có một chế độ dự đoán khác nhau. Dự đoán nội bộ tinh vi rất mạnh mẽ vì nó làm giảm đáng kể lượng dữ liệu được truyền khi dự đoán thời gian không thành công.

Chỉ có 4 chế độ dự đoán cho 16×16 khối trong Intra\_16×16. Chế độ 0 (Dọc), Chế độ 1 (Ngang) và Chế độ 2 (DC) rất giống với Intra\_4 × 4 ở trên ngoại trừ kích thước khối lớn hơn. Đối với Chế độ 3 (Mặt phẳng) duy nhất cho 16 × 16 khối, chức năng mặt phẳng (tuyến tính) được lắp vào các mẫu trên và trái trong khối 16 × 16 như dự đoán.

Tóm lại, bốn chế độ sau đây được chỉ định cho mã hóa nội bộ:

* Intra\_4 × 4 cho macroblocks luma
* Intra\_16 × 16 cho macroblocks luma
* Mã hóa intra cho các macroblock sắc độ — nó sử dụng bốn chế độ dự đoán giống như
* trong Intra\_16 × 16 Luma. Kích thước khối dự đoán là 8 × 8 cho 4: 2: 0, 8 × 16 cho
* 4: 2: 2 và 16 × 16 để lấy mẫu sắc độ 4: 4: 4.
* I\_PCM (Điều chế mã xung) — bỏ qua dự đoán không gian và biến đổi

mã hóa và gửi trực tiếp các giá trị luma và chroma pixel được mã hóa PCM (độ dài cố định). Nó được gọi trong những trường hợp hiếm hoi khi các chế độ dự đoán khác không tạo ra bất kỳ nén / giảm dữ liệu nào.

**12.1.6 Trong vòng lặp chặn lọc**

Một trong những thiếu sót nổi bật của các phương pháp mã hóa dựa trên khối là sự phân loại của các cấu trúc khối có thể nhìn thấy không mong muốn. Các pixel ở ranh giới khối thường được tái tạo kém chính xác hơn: chúng có xu hướng trông giống như các pixel bên trong trong cùng một khối, do đó sự xuất hiện nhân tạo của các khối.

H.264 chỉ định bộ lọc bỏ chặn thích ứng tín hiệu, trong đó một bộ lọc được áp dụng trên 4 × 4 cạnh khối. Độ dài, cường độ và loại bộ lọc (bỏ chặn / làm mịn) khác nhau, tùy thuộc vào các tham số mã hóa macroblock (mã hóa nội bộ hoặc liên mã, chênh lệch khung tham chiếu, hệ số được mã hóa, v.v.) và kích hoạt không gian (phát hiện cạnh), để chặn các tạo tác được loại bỏ mà không làm biến dạng các đặc điểm trực quan. Bộ lọc bỏ chặn H.264 rất quan trọng trong việc tăng chất lượng chủ quan của video.

A graph of a graph

Description automatically generated with medium confidence

Ranh giới khối

**Hình 12.8** De chặn cạnh a1-D trên ranh giới khối

Như thể hiện trong Hình 12.1, trong H.264, quá trình lọc bỏ chặn diễn ra trong vòng lặp, sau khi biến đổi nghịch đảo trong bộ mã hóa, trước khi dữ liệu khối được giải mã được đưa đến Ước tính chuyển động.

Hình 12.8 mô tả cạnh 1-D được đơn giản hóa, trong đó chiều cao của các pixel *p0, q0, v.v., cho biết giá trị của chúng. Chức năng của bộ lọc bỏ chặn* về cơ bản là làm mịn các cạnh khối. Ví dụ: "lọc bốn lần nhấn" sẽ lấy một số giá trị trung bình có trọng số của các giá trị *p1, p0, q0 và q1 để tạo p0 hoặc q0 mới.*

Rõ ràng, các cạnh thực trên ranh giới khối sẽ cần được bảo vệ khỏi bộ lọc bỏ chặn. Việc lọc bỏ chặn trên *p0 và q0 sẽ chỉ được áp dụng nếu đáp ứng tất cả các tiêu chí sau:*

|p0 −q0|<α(*QP*),

|p0 − *p1| < β(QP*),

|q0 −q1|<β(*QP*),

trong đó α và β là các ngưỡng, và chúng là các hàm của tham số lượng tử hóa *Q P* như được định nghĩa trong tiêu chuẩn. Chúng thấp hơn khi *Q P* nhỏ hơn. Điều này là do khi *Q P* nhỏ, một sự khác biệt tương đối đáng kể, ví dụ: | *P0 − Q0 | có khả năng được gây ra bởi một cạnh thực.*

Ngoài *p0 và q0, bộ lọc bỏ chặn trên p1 hoặc q1 sẽ được áp dụng nếu |p0−p2|<β(QP*) hoặc |q0−q2|<β(*QP*)

A table of numbers and lines

Description automatically generated with medium confidence

**12.1.7 Mã hóa Entropy**

H.264 đã phát triển một tập hợp các phương pháp mã hóa entropy tinh vi. Khi entropy\_ coding\_mode = 0, một mã Exponential-Golomb (Exp-Golomb) đơn giản được sử dụng cho dữ liệu tiêu đề, vectơ chuyển động và dữ liệu không còn lại khác, trong khi *Mã hóa độ dài biến thích ứng theo ngữ cảnh (CAVLC)*  phức tạp hơn được sử dụng cho các hệ số dư lượng tử hóa. Khi entropy\_coding\_mode = 1, *Mã hóa số học nhị phân thích ứng theo ngữ cảnh (CABAC)* được sử dụng (xem Phần 12.1.9).

**Mã Exp-Golomb đơn giản**

Mã Exponential-Golomb (Exp-Golomb) đơn giản được sử dụng cho dữ liệu tiêu đề, v.v., được gọi là mã Exp-Golomb bậc 0 (*EG0). Nó là một mã nhị phân, và nó bao gồm ba phần:*

[Tiền tố] [1] [Hậu tố]

Tiền tố là một chuỗi *l* số không. Cho một số không dấu (dương) *N* là

được mã hóa, *l* = . Hậu tố *S* là số nhị phân *N* + 1 − được biểu diễn bằng *l* bit.

Như thể hiện trong Bảng 12.3, nếu N = 4 không dấu , thì l = = 2, Tiền tố00; hậu tố S là số nhị phân S = 4 + 1− = 1 được biểu diễn trong 2 bit, tức là 01. Do đó, mã Exp-Golomb cho *N* = 4 là 00101.

Để giải mã từ mã Exp-Golomb *EG0 cho N không dấu*, bạn có thể làm theo các bước sau:

1. Đọc theo trình tự các số không liên tiếp, *l* = number\_of\_zeros.

2. Bỏ qua '1' tiếp theo.

3. Đọc trong *l*  bit tiếp theo và gán cho *S.*

*4. N = S−1 + .*

Các số không dấu được sử dụng để chỉ ra, ví dụ: loại macroblock, chỉ mục khung tham chiếu, v.v. Đối với các số đã ký, ví dụ: chênh lệch vectơ chuyển động, chúng sẽ đơn giản là

A table of numbers and letters

Description automatically generated

được ép vào để tạo ra một tập hợp các mục nhập bảng mới như được liệt kê trong cột thứ hai (Bảng 12.3).

***kth Order Exp-Golomb Code***

Nói chung, mã Exp-Golomb có thể có thứ tự cao hơn, tức là thứ *tự thứ k EGk*. Tương tự, nó là một mã nhị phân và bao gồm ba phần: [Tiền tố] [1] [Hậu tố]. Tiền tố là một chuỗi *l* số không. Cho một số không dấu (dương) *N* được mã hóa, *l* = . Hậu tố *S* là số nhị phân *N* + được biểu diễn bằng *l* + *k* bit.

Ví dụ: *mã EG1 cho N* = 4 là 0110. Đó là bởi vì *l* = . = 1, tiền tố là 0; hậu tố là biểu diễn nhị phân của 4 + = 2, trong *l* + *k* = 1 + 1 = 2 bit là 10. Bảng 12.4 cung cấp một số ví dụ về mã Exp-Golomb bậc nhất và thứ hai cho các số không âm.

Để giải mã *từ mã Exp-Golomb bậc k EGk* cho N không dấu, có thể làm theo các bước sau:

1. Đọc theo trình tự các số không liên tiếp, *l* = number\_of\_zeros.

2. Bỏ qua '1' tiếp theo.

3. Đọc tiếp theo l + k bit và gán cho S.

4. *N = S-*.

A grid of numbers and symbols

Description automatically generated

**Hình 12.9** Ví dụ: Khối dữ liệu A4×4 cho bộ mã hóa CAVLC

**12.1.8 Context-AdaptiveVariableLengthCoding(CAVLC)**

Các tiêu chuẩn mã hóa video trước đây như MPEG-2 và H.263 sử dụng VLC cố định. Trong CAVLC [6,7], nhiều bảng VLC được xác định trước cho từng loại dữ liệu (zero-runs, levels, v.v.) và các quy tắc được xác định trước dự đoán bảng VLC tối ưu dựa trên ngữ cảnh, ví dụ: các khối lân cận đã được giải mã trước đó.

Được biết, các ma trận (theo mặc định 4×4) chứa các hệ số tần số lượng tử hóa của dữ liệu còn lại thường thưa thớt, tức là chứa nhiều số không. Ngay cả khi chúng không bằng không, các hệ số lượng tử hóa cho các tần số cao hơn thường là +1 hoặc −1 (cái gọi là "trailing\_1s"). CAVLC khai thác các đặc điểm này bằng cách trích xuất cẩn thận các tham số sau từ dữ liệu khối hiện tại:

* Tổng số hệ số khác 0 (TotalCoeff) và số ±1 (Trailing\_1s).
* Dấu hiệu của Trailing\_1s.
* Mức (dấu và độ lớn) của các hệ số khác không (không Trailing\_1s).
* Tổng số số không trước hệ số khác không cuối cùng.
* Chạy các số không (zeros\_left và run\_before) trước mỗi hệ số khác không.

Hình 12.9 cho thấy một ví dụ về khối 4 × 4 của dữ liệu còn lại sau khi biến đổi và lượng tử hóa. Sau khi quét ngoằn ngoèo, chuỗi 1-D là: 0 3 0 −2 1 0 −1 1 0 0 0 0 0 0 0 0. Bảng 12.5 cung cấp chi tiết về mã CAVLC được tạo. Các hệ số khác 0 được xử lý theo thứ tự ngược lại, tức là hệ số cuối cùng được lập chỉ mục bởi '4' (Trailing\_1 [4]) được kiểm tra trước, v.v.

Chúng tôi sẽ giải thích ngắn gọn quá trình tạo mã cho ví dụ trên.

• Acode0000100isgeneratedforTotalCoeffs=5,Trailing\_1s=3.Thesearetra cứu từ "Bảng 9–5—ánh xạ coeff\_token tới TotalCoeff và TrailingOnes" trong tiêu chuẩn H.264 (2003). Người ta quan sát thấy rằng, nói chung, số lượng các rạn san hô khác không trong các khối lân cận là tương tự nhau, do đó mã cho coeff\_token thích ứng với ngữ cảnh. Đối với mỗi cặp TotalCoeffs và Trailing\_1s, mã của nó có thể được gán một trong bốn giá trị có thể tùy thuộc vào số lượng hệ số khác 0 trong các khối ở trên và bên trái của khối hiện tại. Nếu các khối nhàm chán neigh- có một số lượng nhỏ các số không, thì một phép gán mã ủng hộ

A table of numbers and symbols

Description automatically generated with medium confidence

TotalCoeffs nhỏ trong khối hiện tại sẽ được sử dụng (tức là TotalCoeffs nhỏ được gán mã rất ngắn và TotalCoeffs lớn được gán mã đặc biệt dài) và ngược lại. Trong ví dụ này, người ta giả định rằng số hệ số khác không trong hai khối lân cận nhỏ hơn 2.

* Dấu '+' được gán mã 0 và '-' mã 1.
* Việc lựa chọn mã VLC cho Cấp độ một lần nữa thích ứng với ngữ cảnh, nó phụ thuộc vào
* độ lớn của các Cấp độ được mã hóa gần đây. Theo thứ tự ngược lại, hệ số khác 0 đầu tiên là −2. Ban đầu, SuffixLength = 0, vì vậy mã cho −2 là 0001 (tiền tố). Sau đó, SuffixLength được tăng thêm 1, do đó nonzero 3 tiếp theo nhận được mã 001 (tiền tố) 0 (hậu tố). Độ lớn của các mức có xu hướng tăng (khi được kiểm tra theo thứ tự ngược lại), do đó Độ dài hậu tố được tăng lên một cách thích ứng để phù hợp với cường độ lớn hơn. Để biết thêm chi tiết, độc giả được tham khảo [4,6,7].
* Tổng số không là 3. Nó có mã 111.
* Năm hàng cuối cùng trong Bảng 12.5 ghi lại thông tin cho các lần chạy số không trong

khối hiện tại. Ví dụ: đối với hệ số khác 0 cuối cùng '1', 3 số không ở phía trước nó và không có số 0 nào ngay trước nó. Mã 11 được tra cứu từ Bảng 9-10 trong tiêu chuẩn H.264. Để minh họa, chúng tôi trích xuất một phần của nó dưới đây là Bảng 12.6. Điều này cũng sẽ giải thích các mã trong ba hàng tiếp theo, 10, 1 và 01. Đi xuống hàng cuối cùng, chỉ còn lại một số không cho hệ số khác không (cuối cùng) duy nhất, bộ mã hóa và bộ giải mã có thể xác định rõ ràng, do đó không cần mã cho run\_before.

Trong ví dụ này, chuỗi kết quả của mã là 0000100 0 1 0 0001 001 0 111 11 10 1 01. Dựa trên nó, bộ giải mã có thể tái tạo dữ liệu khối.

A table of numbers and digits

Description automatically generated with medium confidence

**12.1.9 Context-AdaptiveBinarySố học Mã hóa (CABAC)**

Các phương pháp mã hóa entropy dựa trên VLC (bao gồm CAVLC) không hiệu quả trong giao dịch với các ký hiệu có xác suất lớn hơn 0,5, vì thường là tối thiểu

của 1 bit phải được gán cho mỗi biểu tượng, có thể lớn hơn nhiều so với *chính nó*

*Thông tin* được đo bằng *, trong đó* là xác suất của ký hiệu. Để tốt hơn *i*

hiệu quả mã hóa trong H.264 Cấu hình chính và cao, Mã hóa số học nhị phân thích ứng theo ngữ cảnh (CABAC) [8] được sử dụng cho một số dữ liệu và hệ số dư lượng tử hóa khi entropy\_coding\_mode = 1.As thể hiện trong Hình 12.10, CABAC có ba thành phần chính:

• **Song ngữ.** Tất cả dữ liệu phi nhị phân được chuyển đổi thành chuỗi bit nhị phân (*bin*) đầu tiên vì CABAC sử dụng Mã hóa số học nhị phân. Năm sơ đồ có thể được sử dụng để nhị phân hóa: (a) Unary (U) — đối với *N* ⊕ 0, nó là N 1s theo sau là 0 kết thúc. Ví dụ: nó là 111110 cho 5. (b) Unary cắt ngắn (TU)—tương tự như U, nhưng không có kết thúc 0. (c) *mã Exp-Golomb bậc k. (d) Ghép nối của (a) và (c). (e) Sơ đồ nhị phân có độ dài cố định.*

• **Mô hình hóa bối cảnh.** Bước này liên quan đến việc lựa chọn và truy cập mô hình ngữ cảnh. *Chế độ mã hóa thông thường* dành cho hầu hết các ký hiệu, ví dụ: loại macroblock, mvd, thông tin về chế độ dự đoán, thông tin về kiểm soát lát cắt và macroblock và dữ liệu còn lại. Các "mô hình ngữ cảnh" khác nhau được xây dựng để lưu trữ các khả năng xác suất có điều kiện cho các thùng của biểu tượng nhị phân là 1 hoặc 0. Các mô hình xác suất được lấy từ các số liệu thống kê của bối cảnh, tức là các ký hiệu và thùng được mã hóa gần đây. *Chế độ mã hóa Bypass* không sử dụng mô hình ngữ cảnh; nó được sử dụng để tăng tốc quá trình mã hóa.

• **Mã hóa số học nhị phân.** Để đạt hiệu quả, một phương pháp mã hóa số học nhị phân được phát triển [8]. Dưới đây là mô tả ngắn gọn về *Mã hóa số học nhị phân* trong H.264. Như thể hiện trong Chap. 7, mã hóa số học liên quan đến các phân chia đệ quy của phạm vi cur-rent. Nhiều phép nhân liên quan đến đó là nhược điểm chính của nó về chi phí tính toán khi so sánh với các phương pháp mã hóa entropy khác. Mã hóa số học nhị phân chỉ liên quan đến hai ký hiệu, vì vậy số lượng phép nhân giảm đi rất nhiều.

Trong trường hợp nhị phân, chúng tôi đặt tên cho hai ký hiệu *là LPS (Biểu tượng ít có khả năng xảy ra nhất)* và *MPS (Biểu tượng có thể xảy ra nhất),* phạm vi là *R*, cận dưới của *R* là *L*. (Lưu ý:

A diagram of a software model

Description automatically generated

**Hình 12.10** Sơ đồ khối của CABAC trong H.264

LPS là khoảng trên và MPS là khoảng dưới trong *R*.) Nếu xác suất của LPS là , thì = 1− và quy trình sau đây có thể được sử dụng để tạo phạm vi tiếp theo với ký hiệu mới *S*:

**THỦ TỤC Tính toán phạm vi trong mã hóa số học nhị phân**

BEGIN

Nếu *S* là MPS.

*R* = *R* × (1 − );

Thì // *S* là LPS.

*L* = *L* +

*R* × (1 − ); *R* = *R* × ;

Kết thúc

Tuy nhiên, phép nhân trong *R* × sẽ tốn kém tính toán. Các sơ đồ mã hóa số học nhị phân "không nhân" khác nhau đã được phát triển, ví dụ, bộ mã hóa Q cho hình ảnh nhị phân, bộ mã hóa QM được cải tiến và bộ mã hóa MQ được áp dụng vào năm JPEG2000. Mã hóa số học nhị phân H.264 phương pháp được phát triển bởi Marpe et al. [8] là cái gọi là *M-coder (Modulo Coder).* Ở đây, phép nhân trong *R* × *PL P S* được thay thế bằng tra cứu bảng. Trong chế độ Mã hóa thùng thông thường, bảng có 4 × 64 giá trị sản phẩm được tính toán trước để cho phép bốn giá trị khác nhau cho *R* và 64 giá trị cho *pLPS*. (Chế độ Mã hóa Bypass Bin giả định một mô hình xác suất thống nhất, tức là, ≡ để đơn giản hóa và tăng tốc quá trình.)

Rõ ràng, do kích thước hạn chế của bảng tra cứu, độ chính xác của các giá trị sản phẩm bị hạn chế. Do đó, các phương pháp không có phép nhân này được gọi là *Mã hóa số học giảm độ chính xác thực hiện*. Nghiên cứu trước đây cho thấy tác động của độ chính xác giảm đối với độ dài mã là tối thiểu.

Việc thực hiện CABAC có rất nhiều chi tiết. Độc giả được giới thiệu đến [8] để thảo luận chi tiết.

**12.1.10 H.264 Profiles**

Như trước đây, một số cấu hình được cung cấp để phù hợp với nhu cầu của các ứng dụng khác nhau, từ thiết bị di động đến phát sóng HDTV. Hình 12.11 cung cấp tổng quan về các cấu hình H.264 [4,9].

**Hồ sơ cơ sở**

Cấu hình cơ sở của H.264 dành cho các ứng dụng đàm thoại thời gian thực, chẳng hạn như hội nghị truyền hình. Nó chứa tất cả các công cụ mã hóa cốt lõi của H.264 được thảo luận ở trên và các công cụ phục hồi lỗi bổ sung sau đây, để cho phép các nhà cung cấp dịch vụ dễ bị lỗi như IP và mạng không dây:

• **Thứ tự lát tùy ý (ASO).** Thứ tự giải mã của các lát cắt trong một bức tranh có thể không tuân theo thứ tự tăng đơn điệu. Điều này cho phép giải mã các gói không theo thứ tự trong mạng chuyển mạch gói, do đó giảm độ trễ.

• **Thứ tự macroblock linh hoạt (FMO).** Macroblocks có thể được giải mã theo bất kỳ thứ tự nào, chẳng hạn như các mẫu bàn cờ, không chỉ là thứ tự quét raster. Điều này rất hữu ích trên các mạng dễ bị lỗi, do đó việc mất một lát dẫn đến mất các macroblocks rải rác trong hình, có thể dễ dàng che giấu khỏi mắt người. Tính năng này cũng có thể giúp giảm jitter và độ trễ, vì bộ giải mã có thể quyết định không chờ đợi các lát cắt muộn và vẫn có thể tạo ra hình ảnh chấp nhận được.

• **Lát thừa.** Các bản sao dự phòng của các lát cắt có thể được giải mã, để cải thiện hơn nữa khả năng phục hồi lỗi.

**Hồ sơ chính**

Cấu hình chính được xác định bởi H.264 đại diện cho các ứng dụng không có độ trễ thấp như độ nét tiêu chuẩn (SD), truyền hình phát sóng kỹ thuật số và phương tiện lưu trữ. Cấu hình chính chứa tất cả các tính năng cấu hình Đường cơ sở (ngoại trừ ASO, FMO và các lát dự phòng) cộng với các tính năng không có độ trễ thấp và độ phức tạp cao hơn sau đây, để có hiệu quả ép tối đa:

* **B lát.** Chế độ dự đoán kép trong H.264 đã được thực hiện linh hoạt hơn so với các tiêu chuẩn hiện có. Hình ảnh dự đoán kép cũng có thể được sử dụng làm khung tham chiếu. Hai khung tham chiếu cho mỗi macroblock có thể theo bất kỳ hướng thời gian nào, miễn là chúng có sẵn trong bộ đệm khung tham chiếu. Do đó, ngoài dự đoán kép tiến + lùi thông thường, việc dự đoán lùi + lùi hoặc tiến + tiến cũng là hợp pháp.
* **Mã hóa số học nhị phân thích ứng theo ngữ cảnh (CABAC).** Chế độ mã hóa này thay thế mã hóa entropy dựa trên VLC bằng mã hóa số học nhị phân sử dụng mô hình thống kê thích ứng khác cho các loại dữ liệu và ngữ cảnh khác nhau.
* **Dự đoán có trọng số.** Trọng lượng toàn cầu (hệ số nhân và độ bù) để sửa đổi các mẫu dự đoán bù chuyển động có thể được chỉ định cho mỗi lát, để dự đoán sự thay đổi ánh sáng và các hiệu ứng toàn cầu khác, chẳng hạn như mờ dần.

**Hồ sơ mở rộng**

Cấu hình eXtended (hoặc hồ sơ X) được thiết kế cho các ứng dụng phát trực tuyến video mới. Cấu hình này cho phép các tính năng không có độ trễ thấp, tính năng chuyển đổi bitstream và nhiều công cụ phục hồi lỗi hơn. Nó bao gồm tất cả các tính năng hồ sơ Baseline cộng với những điều sau đây:

* B lát.
* Dự đoán có trọng số.
* **Phân vùng dữ liệu lát cắt.** Phân vùng này cắt dữ liệu với tầm quan trọng khác nhau thành
* Các chuỗi riêng biệt (thông tin tiêu đề, thông tin còn lại) để áp đặt nhiều hơn-
* Dữ liệu TANT có thể được truyền trên các kênh đáng tin cậy hơn.
* **Các loại lát cắt SP (Chuyển mạch P) và SI (Chuyển đổi I).** Đây là những lát cắt con-

Chế độ dự đoán thời gian đặc biệt, để cho phép chuyển đổi hiệu quả các luồng bit được tạo bởi các bộ giải mã khác nhau. Chúng cũng tạo điều kiện thuận lợi cho việc tiến / lùi nhanh và truy cập ngẫu nhiên.

**Cấu hình cao**

H.264 / AVC cũng có bốn cấu hình cao cho các ứng dụng đòi hỏi chất lượng video cao hơn, tức là Độ nét cao (HD).

* **Cấu hình cao** Cấu hình này được thông qua bởi định dạng Blu-ray Disc và DVB HDTV
* phát sóng. Nó hỗ trợ biến đổi số nguyên 8 × 8 cho các phần của hình ảnh không có nhiều chi tiết và biến đổi số nguyên 4 × 4 cho các phần có chi tiết. Nó cũng cho phép dự đoán 8 × 8 Intra để có hiệu quả mã hóa tốt hơn, đặc biệt là đối với các video có độ phân giải cao hơn. Nó cung cấp ma trận tỷ lệ lượng tử có thể điều chỉnh và các tham số lượng tử riêng biệt cho Cb và Cr. Nó cũng hỗ trợ video đơn sắc (4: 0: 0).
* **Cấu hình High 10** Hỗ trợ 9 hoặc 10 bit cho mỗi mẫu.
* **Cấu hình 4: 2: 2 cao** Hỗ trợ lấy mẫu phụ sắc độ 4: 2: 2.
* **Cấu hình dự đoán cao 4: 4: 4** Nó hỗ trợ lấy mẫu sắc độ lên đến 4: 4: 4, lên đến 14

bit trên mỗi mẫu, mã hóa các mặt phẳng màu riêng biệt và mã hóa dự đoán không mất dữ liệu hiệu quả.

**12.1.11 Mã hóa video có thể mở rộng H.264**

Phần mở rộng *Mã hóa video có thể mở rộng (SVC)* của tiêu chuẩn H.264 / AVC đã được phê duyệt vào năm 2007 [10]. Nó cung cấp khả năng mở rộng bitstream đặc biệt quan trọng để truyền dữ liệu đa phương tiện thông qua các mạng khác nhau có thể có băng thông rất khác nhau.

Tương tự như MPEG-2 và MPEG-4, H.264 / AVC SVC cung cấp *khả năng mở rộng tạm thời*, *khả năng mở rộng không gian*, *khả năng mở rộng chất lượng và các* kết hợp có thể có của chúng. So với các tiêu chuẩn trước đây, hiệu quả mã hóa được cải thiện rất nhiều. Các chức năng khác như tốc độ bit và thích ứng năng lượng, và sự xuống cấp duyên dáng trong truyền mạng mất dữ liệu cũng được cung cấp.

Chúng tôi đã đề cập đến các vấn đề về khả năng mở rộng thời gian, khả năng mở rộng không gian, khả năng mở rộng chất lượng (SNR) và các kết hợp có thể có của chúng một cách chi tiết đầy đủ theo MPEG-2 trong Chap. 11. Vì các khái niệm và cách tiếp cận cơ bản rất giống nhau, chúng tôi sẽ không thảo luận chi tiết về chủ đề này trong chương này. Để biết thêm thông tin về H264 / AVC SVC, độc giả được tham khảo [10] và Phụ lục G mở rộng của tiêu chuẩn H.264 / AVC.

**12.1.12 Mã hóa video đa chế độ xem H.264**

*Mã hóa video MultiView (MVC)* là một vấn đề mới nổi. Nó có các ứng dụng tiềm năng trong một số lĩnh vực mới như *Free Viewpoint Video (FVV),* nơi người dùng có thể chỉ định chế độ xem ưa thích của họ. Merkle et al.[11] đã mô tả một số cấu trúc dự đoán MVC có thể có. Hình 12.12 cho thấy một ví dụ nhỏ trong đó chỉ có bốn chế độ xem. Hai tính năng quan trọng nhất là: • **Dự đoán phỏng vấn** Vì có sự dư thừa rõ ràng giữa nhiều

xem, cấu trúc IPPP, ví dụ, có thể được sử dụng cho cái gọi là Khóa

A diagram of a diagram

Description automatically generated

**Hình 12.12** Cấu trúc dự đoán H.264MVC

Hình ảnh (hình ảnh đầu tiên và thứ chín trong mỗi chế độ xem trong hình). Cấu trúc dự đoán phỏng vấn này tất nhiên có thể được mở rộng sang các cấu trúc khác, ví dụ: IBBP. Điều này thậm chí còn có lợi hơn khi có nhiều quan điểm hơn.

• **Hình ảnh B phân cấp** Để dự đoán thời gian trong mỗi chế độ xem, có thể sử dụng một hệ thống phân cấp của hình ảnh B, ví dụ: *B1, B2, B3, tương tự như các phân cấp được thảo luận trong Phần 12.1.1. Nó được thực hiện khả thi vì H.264 / AVC linh hoạt hơn trong việc hỗ trợ các sơ đồ dự đoán khác nhau ở cấp độ hình ảnh / trình tự. Như đã thảo luận trước đó, các tham số lượng tử hóa ngày càng lớn hơn thường có thể được áp dụng xuống hệ thống phân cấp để kiểm soát hiệu quả nén.*

**12.2 H.265**

HEVC (Mã hóa video hiệu quả cao) [12,13] là tiêu chuẩn mới nhất được phát triển bởi Nhóm hợp tác chung về mã hóa video (JCT-VC) từ các nhóm ITU-T VCEG (Nhóm chuyên gia mã hóa video) và ISO / IEC MPEG. Dự thảo cuối cùng của tiêu chuẩn được đưa ra vào tháng 1 năm 2013. Trong ISO / IEC, HEVC trở thành MPEG-H

Phần 2 (ISO/IEC 23008-2). Nó còn được gọi là Khuyến nghị ITU-T H.265 [14], đó là thuật ngữ chúng tôi sẽ sử dụng trong cuốn sách này.

Sự phát triển của tiêu chuẩn mới này phần lớn được thúc đẩy bởi hai yếu tố: (a) Sự cần thiết phải cải thiện hơn nữa hiệu quả mã hóa do sự phục hồi video ngày càng tăng (ví dụ: lên đến 8k × 4k trong UHDTV). (b) Sự cần thiết phải tăng tốc các phương pháp mã hóa / giải mã phức tạp hơn bằng cách khai thác các thiết bị và thuật toán xử lý song song ngày càng có sẵn. Mục tiêu ban đầu là giảm thêm 50% kích thước của video nén (với cùng chất lượng hình ảnh) từ H.264 và có báo cáo rằng mục tiêu này đã vượt quá. Với hiệu suất nén vượt trội so với MPEG-2, H.264 và H.265 hiện là những ứng cử viên hàng đầu để mang toàn bộ nội dung video trên nhiều ứng dụng tiềm năng.

Tại thời điểm này, định dạng mặc định cho video màu trong H.265 là YCbCr. Trong các cấu hình chính, lấy mẫu phụ sắc độ là 4: 2: 0.

Các tính năng chính của H.265 là:

* Bù chuyển động kích thước khối thay đổi, từ 4 × 4 lên đến 64 × 64 trong hình ảnh luma.
* Cấu trúc macroblock được thay thế bằng cấu trúc quadtree của các khối mã hóa tại
* Các cấp độ và kích cỡ khác nhau.
* Thăm dò xử lý song song.
* Số nguyênbiến đổi invarioussizes,from4×4,8×8,16×16to32×32.
* Cải thiện các phương pháp nội suy cho độ chính xác một phần tư pixel trong vectơ chuyển động.
* Dự đoán không gian định hướng mở rộng (33 hướng góc) để mã hóa nội bộ.
* Việc sử dụng tiềm năng của DST (Biến đổi hình sin rời rạc) trong mã hóa luma intra.
* Bộ lọc trong vòng lặp bao gồm bỏ chặn-lọc và SAO (Bù thích ứng mẫu).
* Chỉ CABAC (Mã hóa số học nhị phân thích ứng theo ngữ cảnh) sẽ được sử dụng, tức là không
* thêm CAVLC.

**12.2.1 Bồi thường chuyển động**

Như trong các tiêu chuẩn mã hóa video trước đây, H.265 vẫn sử dụng công nghệ *mã hóa lai*, tức là sự kết hợp giữa các dự đoán liên / nội bộ và mã hóa biến đổi 2D trên các lỗi còn lại.

Kích thước khối thay đổi được sử dụng trong các dự đoán liên / nội bộ như trong H.264. Tuy nhiên, nhiều phân vùng của các khối dự đoán và biến đổi được khuyến khích để giảm các lỗi dự đoán. Không giống như các tiêu chuẩn mã hóa video trước đây, H.265 không sử dụng cấu trúc đơn giản và cố định của **macroblocks**. Thay vào đó, một hệ thống phân cấp quadtree của các khối khác nhau được giới thiệu như dưới đây cho hiệu quả của nó.

* + **CTB** và **CTU** (Coding Tree Block and Coding Tree Unit): CTB là khối lớn nhất, gốc trong hệ thống phân cấp quadtree. Kích thước của luma CTB là *N* × *N*, trong đó *N* có thể là 16, 32 hoặc 64. CTB sắc độ có kích thước bằng một nửa, tức là *N* / 2 × *N* /2. Một CTU bao gồm 1 luma CTB và 2 CTB sắc độ.
  + **CB** và **CU** (Coding Block and Coding Unit): CTB bao gồm các CB được tổ chức theo cấu trúc quadtree. CB là một khối vuông có thể nhỏ tới 8 × 8 trong luma và 4 × 4 trong hình ảnh sắc độ. Các CB trong CTB được đi qua và mã hóa theo thứ tự Z. Một luma CB và hai CB sắc độ tạo thành CU.

A diagram of a tree

Description automatically generated

* **PB** và **PU** (Khối dự đoán và Đơn vị dự đoán): Một CB có thể được chia thành các PB cho mục đích dự đoán. Chế độ dự đoán cho CU có thể là hình ảnh nội bộ (không gian) hoặc liên hình ảnh (thời gian). Đối với dự đoán nội bộ, kích thước của CB và PB thường giống nhau; ngoại trừ khi CB là 8 × 8, cho phép chia thành bốn PB để mỗi PB có thể có một chế độ dự đoán khác nhau. Đối với dự đoán inter, luma hoặc chroma CB có thể được chia thành một, hai hoặc bốn PB, tức là, PB có thể không phải là hình vuông mặc dù luôn có hình chữ nhật. PU chứa các PB luma và chroma và cú pháp dự đoán của chúng.
* **TB** và **TU** (Transform Block and Transform Unit): Một CB có thể được chia thành TB với mục đích chuyển đổi mã hóa các lỗi còn lại. Điều này được thể hiện trong cùng một cấu trúc quadtree, do đó nó rất hiệu quả. Phạm vi của kích thước TB là 32 × 32 xuống còn 4 × 4. Trong H.265, các TB được phép trải dài qua các ranh giới PB trong các CU được dự đoán giữa các để đạt được hiệu quả mã hóa cao hơn. TU bao gồm TB từ hình ảnh luma và sắc độ.

Hình 12.13 minh họa một ví dụ trong đó CTB được phân chia thành CB và sau đó tiếp tục thành TB trong cấu trúc quadtree. Trong ví dụ này, CTB ban đầu là 64 × 64 và TB nhỏ nhất là 4 × 4.

**Lát và gạch**

Như trong H.264, H.265 hỗ trợ các lát có độ dài bất kỳ bao gồm một chuỗi các CTU (Hình 12.14a). Chúng có thể là I-slices, P-slices, hoặc B-slices.

Ngoài các lát, khái niệm *Tile* được giới thiệu để tạo điều kiện xử lý song song giữa nhiều Tile. Gạch là một cấu trúc hình chữ nhật bao gồm CTU (Hình 12.14b); Nó cũng có thể chứa nhiều lát.

Một tính năng bổ sung là bao gồm công nghệ *Xử lý song song mặt sóng (WPP).*  Về cơ bản, các hàng CTU có thể được xử lý song song, trong nhiều luồng theo cách mặt sóng được hiển thị trong (Hình 12.14c).

Hiện tại, tiêu chuẩn không cho phép sử dụng hỗn hợp cả Gạch và Mặt sóng.

**Độ chính xác của một phần tư điểm ảnh trong hình ảnh Luma**

Trong dự đoán giữa các hình ảnh cho hình ảnh luma, độ chính xác cho các vectơ chuyển động một lần nữa ở mức một phần tư pixel như trong H.264. Các giá trị tại các vị trí subpixel được suy ra thông qua nội suy. Như thể hiện trong Bảng 12.7, bộ lọc tám lần nhấn được sử dụng cho các vị trí nửa pixel (ví dụ: vị trí *b* trong Hình 12.15) và bộ lọc qfilter bảy lần nhấn được sử dụng cho các vị trí pixel một phần tư (ví dụ: vị trí *a* và *c* trong Hình 12.15).

Như được hiển thị bên dưới, tất cả các giá trị tại các vị trí subpixel được suy ra thông qua các bước lọc có thể tách rời theo chiều dọc và chiều ngang. Điều này khác với H.264 sử dụng bộ lọc sáu lần nhấn để lấy các giá trị ở vị trí nửa pixel và sau đó tính trung bình để có được các giá trị ở vị trí pixel một phần tư.

Các giá trị tại các vị trí a, b và c có thể được suy ra bằng cách sử dụng như sau:

A black and white text

Description automatically generated with medium confidence

A white paper with black text and black squares

Description automatically generated with medium confidence

A table with numbers and symbols

Description automatically generated

A math equations with numbers

Description automatically generated

Việc triển khai thực tế liên quan đến sự dịch chuyển phải bởi (*B* - 8) bit sau các tính toán trên, trong đó *B* ⊕ 8 là số bit trên mỗi mẫu hình ảnh.

Rõ ràng là bộ lọc tám lần nhấn là đối xứng, do đó nó hoạt động tốt đối với các vị trí nửa pixel nằm ở giữa các pixel trên lưới hình ảnh. Bộ lọc qfilter bảy lần nhấn không đối xứng, rất phù hợp với các vị trí pixel một phần tư không ở giữa. Cách xử lý khác nhau một cách tinh tế của *a* và *c* trong Eqs. (12.9) và (12.11) phản ánh bản chất của hoạt động bất đối xứng này. Về cơ bản, qfilter[1 − *t*] là một phiên bản lật của qfilter[*t*]. Ví dụ, *ai*, *j* gần nhất với *Ai*. *j* , và nó sẽ rút ra nhiều nhất từ *Ai*, *j* với trọng số 58; trong khi *ai*, *j* sẽ rút ra nhiều nhất từ *Ai*, *j* +1 với trọng số 58.

Tương tự, các giá trị tại các vị trí *d*, *e* và *f* có thể được suy ra bằng cách sử dụng như sau:

A math equations with numbers

Description automatically generated with medium confidenceA group of math equations

Description automatically generated

**12.2.2 Biến đổi số nguyên**

Như trong H.264, mã hóa biến đổi được áp dụng cho phần dư lỗi dự đoán. Biến đổi 2-D được thực hiện bằng cách áp dụng biến đổi 1-D theo hướng dọc và sau đó là ngang. Điều này được thực hiện bởi hai phép nhân ma trận: **F** = **H** × **f** × **HT,** trong đó **f** là dữ liệu dư đầu vào và **F** là dữ liệu được chuyển đổi. **H** là ma trận Biến đổi số nguyên xấp xỉ ma trận DCT.

Kích thước khối chuyển đổi của 4 × 4, 8 × 8, 16 × 16 và 32 × 32 được hỗ trợ. Chỉ có một Ma trận biến đổi số nguyên, tức  **là H32×32** được chỉ định trong H.265. Các ma trận khác cho các bệnh lao nhỏ hơn là các phiên bản được lấy mẫu phụ của **H32×32**. Ví dụ: **H16×16 được** hiển thị bên dưới dành cho 16 × 16 TB.

A table of numbers with numbers on it

Description automatically generated

**H8×8** có thể thu được bằng cách sử dụng 8 mục đầu tiên của Hàng 0, 2, 4, 6, ... của **H16×16**. Đối với **H4×4**, sử dụng 4 mục đầu tiên của Hàng 0, 4, 8 và 12.

A number with numbers on it

Description automatically generated with medium confidence

So với Eq. (12.2), các mục trong **H4×4** rõ ràng có cường độ lớn hơn nhiều. Để sử dụng số học 16 bit và bộ nhớ 16 bit, dải động của kết quả trung gian từ phép nhân ma trận đầu tiên phải được giảm bằng cách giới thiệu phép dịch chuyển phải 7 bit và thao tác cắt 16 bit.

**12.2.3 Lượng tử hóa và mở rộng quy mô**

Không giống như ma trận H trong H.264 (Eq. 12.2), các số trong ma trận biến đổi số nguyên H.265, ví dụ, Eq. (12.15), tương ứng rất gần với giá trị thực tế của các hàm cơ sở DCT. Do đó, các yếu tố mở rộng quy mô đặc biệt như được xây dựng trong Bảng 12.1 và 12.2 không còn cần thiết nữa.

Để lượng tử hóa, ma trận lượng tử hóa và tham số QP tương tự như trong H.264 được sử dụng. Phạm vi của QP là [0, 51]. Tương tự, kích thước bước lượng tử hóa tăng gấp đôi khi giá trị QP tăng thêm 6.

**12.2.4 Mã hóa nội bộ**

Như trong H.264, dự đoán không gian được sử dụng trong mã hóa nội bộ trong H.265. Các mẫu ranh giới lân cận từ các khối ở trên cùng và / hoặc bên trái của khối hiện tại được sử dụng cho các dự đoán. Các lỗi dự đoán sau đó được gửi để mã hóa chuyển đổi. Kích thước khối biến đổi (TB) nằm trong khoảng từ 4 × 4 đến 32 × 32 trong mã hóa Intra trong H.265. Do (a) kích thước bệnh lao có khả năng lớn hơn nhiều và (b) nỗ lực giảm lỗi dự đoán, số lượng chế độ dự đoán có thể tăng từ 9 trong H.264 lên 35 trong H.265. Như thể hiện trong Hình 12.16a, Chế độ 2 đến Chế độ 34 là các chế độ dự đoán Intra\_angular. Lưu ý, chênh lệch góc giữa các chế độ được cố tình làm không đều, ví dụ: để làm cho nó dày đặc hơn gần hướng ngang hoặc dọc. Hầu hết các mẫu cần thiết cho dự đoán góc sẽ ở vị trí subpixel. Nội suy tuyến tính của hai pixel gần nhất tại các vị trí số nguyên được sử dụng và độ chính xác lên tới 1/32 pixel.

Hai chế độ dự đoán đặc biệt là Chế độ 0: Intra\_Planar và Chế độ 1: Intra\_DC. Chúng tương tự như trong H.264. Trong Intra\_DC, trung bình của các mẫu tham chiếu được sử dụng làm dự đoán. Về Intra\_planar, khác với H.264, tất cả bốn góc đều được sử dụng để dự đoán phẳng, tức là, hai dự đoán mặt phẳng sẽ được thực hiện và giá trị trung bình của chúng sẽ được áp dụng.

**12.2.5 Rời rạcSineTransform**

Năm Intra\_4 × 4, đối với các khối dư luma, HEVC đã giới thiệu một biến đổi thay thế dựa trên một trong các biến thể của *Biến đổi hình sin rời rạc (DST)* (được gọi là DST-VII) [15]. Đó là bởi vì các dự đoán nội bộ dựa trên các mẫu ranh giới lân cận ở trên cùng hoặc bên trái của khối. Lỗi dự đoán có xu hướng tăng đối với các nút trong khối ở xa đỉnh hoặc bên trái lân cận. Nói chung, DST được tìm thấy để đối phó với tình huống này tốt hơn DCT ở bước mã hóa chuyển đổi.

Ma trận số nguyên cho DST có thể được mô tả bởi:

A black text on a white background

Description automatically generated

A diagram of a graph

Description automatically generated

**where** *i* = 1,.., *N* và *j* = 1,.., *N* là các chỉ số hàng và cột, và kích thước khối là *N* × *N*.

Khi *N* = 4, thu được HDST sau:

A number with numbers and lines

Description automatically generated with medium confidence

Saxena và Fernandes [16,17] tiếp tục nghiên cứu lợi ích của việc kết hợp DCT và DST, tức là, cho phép DCT hoặc DST ở một trong hai biến đổi 1-D, bởi vì DST và DCT được hiển thị để giành chiến thắng theo (các) hướng dọc và / hoặc ngang cho các chế độ dự đoán nhất định. Mặc dù có hơn 30 hướng dự đoán nội bộ khác nhau trong H.265, chúng phân loại các chế độ dự đoán thành:

* Loại 1 — các mẫu để dự đoán đều từ các hàng xóm bên trái của khối hiện tại (Hình 12.17a) hoặc tất cả từ các hàng xóm hàng đầu của khối hiện tại (Hình 12.17b).
* Loại 2 — các mẫu để dự đoán là từ cả hàng xóm trên cùng và bên trái của khối hiện tại (Hình 12.17c, d).
* DC — một chế độ dự đoán đặc biệt trong đó trung bình của một tập hợp các mẫu lân cận cố định được sử dụng.

Bảng 12.8 cho thấy một số khuyến nghị của họ.

A table with a number of lines

Description automatically generated with medium confidence

**12.2.6 In-LoopFiltering**

Tương tự như H.264, các quy trình lọc trong vòng lặp được áp dụng để loại bỏ các tạo tác khối. Ngoài việc Lọc bỏ chặn, H.265 cũng giới thiệu *quy trình* Bù đắp thích ứng mẫu (SAO).

**Bỏ chặn lọc**

Thay vì áp dụng bộ lọc bỏ chặn cho 4 × 4 khối như trong H.264, nó chỉ được áp dụng cho các cạnh nằm trên lưới hình ảnh 8 × 8. Điều này làm giảm độ phức tạp tính toán và đặc biệt tốt cho xử lý song song vì cơ hội thay đổi tầng tại các mẫu gần đó giảm đáng kể. Chất lượng hình ảnh vẫn tốt, một phần là do quy trình SAO được mô tả dưới đây.

Bộ lọc khử chặn được áp dụng đầu tiên cho các cạnh dọc, sau đó đến các cạnh ngang trong ảnh, do đó cho phép xử lý song song. Ngoài ra, nó có thể được áp dụng CTB bởi CTB.

**Độ lệch thích ứng mẫu (SAO)**

Quá trình SAO có thể được gọi tùy chọn sau khi lọc bỏ chặn. Về cơ bản, một giá trị bù đắp được thêm vào mỗi mẫu dựa trên các điều kiện nhất định được mô tả bên dưới.

**A screenshot of a table

Description automatically generated**

Hai chế độ được xác định để áp dụng SAO: *Chế độ bù băng tần* và *chế độ bù cạnh*.

Trong chế độ bù băng tần, phạm vi biên độ mẫu được chia thành 32 dải. Độ lệch băng tần có thể được thêm vào các giá trị mẫu trong bốn dải liên tiếp đồng thời. Điều này giúp giảm "hiện vật dải" ở những khu vực nhẵn.

Trong chế độ bù cạnh, thông tin gradient (cạnh) được phân tích đầu tiên. Hình 12.18 mô tả bốn hướng gradient (cạnh) có thể: (a) ngang, (b) dọc và (c, d) đường chéo. Độ lệch dương hoặc âm, hoặc độ lệch bằng không có thể được thêm vào p mẫu dựa trên những điều sau:

• Tích cực: *pisalocalminimum (p < n0 & p < n1), orpisanedgepixel*

(*p* < *n0 & p* = *n1 hoặc p* = *n0 & p* < *n1).*

*• Tiêu cực: pisalocalmaximum (p > n0 & p >n1), orpisanedgepixel*

(*p* > *n0 & p* = *n1 hoặc p* = *n0 & p* > *n1).*

*• Số không: Không có điều nào ở trên.*

**12.2.7 Mã hóa Entropy**

H.265 chỉ sử dụng CABAC trong mã hóa entropy, tức là CAVLC không còn được sử dụng nữa. Do cây mã hóa mới được giới thiệu và cấu trúc cây biến đổi, độ sâu cây giờ đây trở thành một phần quan trọng của *mô hình ngữ cảnh* bên cạnh bối cảnh lân cận không gian trong H.264 / AVC. Kết quả là, số lượng bối cảnh được giảm và hiệu quả mã hóa entropy được cải thiện hơn nữa.

Không giống như các tiêu chuẩn video trước đây, ba phương pháp quét đơn giản được xác định để đọc trong các hệ số biến đổi, tức là *Đường chéo lên bên phải, Ngang* và *Dọc*. Mục tiêu vẫn là tối đa hóa độ dài của zero-runs. Quá trình quét luôn diễn ra trong 4 × 4 khối con bất kể kích thước TB. Quét chéo lên bên phải được sử dụng cho tất cả các khối liên dự đoán và cho các khối nội dự đoán là 16 × 16 hoặc 32 × 32. Đối với các khối nội dự đoán là 4 × 4 hoặc 8 × 8, các hướng sau được sử dụng: Ngang— cho các hướng dự đoán gần với dọc, Dọc—cho các hướng dự đoán gần ngang, Đường chéo lên bên phải—cho các hướng dự đoán khác.

Có nhiều cải tiến như làm thế nào để mã hóa các hệ số biến đổi khác không một cách hiệu quả [12,14]. Ngoài ra, một trong những mục tiêu của việc triển khai CABAC mới trong H.265 là đơn giản hóa các biểu diễn ngữ cảnh của nó để thông lượng của nó có thể được tăng lên. Để biết chi tiết, độc giả được tham khảo [18] cung cấp một tài liệu tham khảo tuyệt vời.

A table with numbers and text

Description automatically generated

**12.2.8 Chế độ mã hóa đặc biệt**

Ba chế độ mã hóa đặc biệt được định nghĩa trong H.265. Chúng có thể được áp dụng ở cấp độ CU hoặc TU.

* **I\_PCM** Như trong H.264, dự đoán, mã hóa biến đổi, lượng tử hóa và entropy
* Các bước mã hóa được bỏ qua. Các mẫu được mã hóa PCM (chiều dài cố định) được gửi trực tiếp. Nó được gọi khi các chế độ dự đoán khác không tạo ra bất kỳ sự giảm dữ liệu nào.
* **Các** lỗi còn lại từ các dự đoán liên hoặc nội bộ được gửi trực tiếp đến mã hóa entropy, do đó để tránh bất kỳ bước mất mát nào, đặc biệt là lượng tử hóa sau khi mã hóa biến đổi.
* **Biến đổi bỏ qua** Chỉ có bước biến đổi được bỏ qua. Điều này hoạt động đối với một số dữ liệu nhất định (ví dụ: hình ảnh hoặc đồ họa do máy tính tạo). Nó chỉ có thể được áp dụng cho 4 × 4 TB.

**12.2.9 Hồ sơ H.265**

Tại thời điểm này, chỉ có ba cấu hình được xác định: *Cấu hình chính*, *Cấu hình chính 10* và *Cấu hình ảnh tĩnh chính*, mặc dù ngày càng có nhiều cấu hình cao hơn được mong đợi trong tương lai.

Định dạng mặc định cho màu là YCbCr. Trong tất cả các cấu hình chính, lấy mẫu phụ sắc độ là 4: 2: 0. Mỗi mẫu có 8 bit, ngoại trừ trong Main 10, có 10 bit.

Ví dụ: một số định dạng video được hỗ trợ ở các cấp độ khác nhau trong Cấu hình chính được liệt kê trong Bảng 12.9. Như được hiển thị, tổng số cấp độ được đề xuất là 13. Nó bao gồm các video có độ phân giải rất thấp, ví dụ: QCIF (176 × 144) ở Cấp độ 1, cũng như các video có độ phân giải rất cao, ví dụ: UHDTV (8.192 × 4.320) ở Cấp độ 6, 6.1 và 6.2.

Khi tính toán Kích thước hình ảnh Max Luma, chiều rộng và chiều cao được làm tròn lên bội số gần nhất của 64 (như một yêu cầu thực hiện). Ví dụ: 176 × 144 trở thành 192 × 192 = 36.864.

A table of information

Description automatically generated

Như được hiển thị, HDTV hiện tại ở Cấp độ 4 và 4.1 cho Tốc độ khung hình 30 và 60 khung hình / giây. Tốc độ bit tối đa của video nén ở cái gọi là Bậc chính lần lượt là 12 Mbit / giây và 20 Mbit / giây. Ở High Tier, chúng có thể cao hơn tới 2,5 lần. Các video UHDTV ở Cấp độ 5 trở lên sẽ yêu cầu tốc độ bit cao hơn nhiều, đây vẫn là một thách thức đối với tất cả các khía cạnh của đa phương tiện bao gồm lưu trữ, truyền dữ liệu và thiết bị hiển thị.

**12.3 So sánh hiệu quả mã hóa video**

Khi so sánh hiệu quả mã hóa của các phương pháp nén video khác nhau, một thực tế phổ biến là so sánh tốc độ bit của các luồng bit video được mã hóa ở cùng chất *lượng*. Các phương pháp đánh giá chất lượng video có thể là *khách quan* hoặc *chủ quan*: cái trước được thực hiện tự động bởi máy tính và cái sau đòi hỏi sự phán đoán của con người.

**12.3.1 Đánh giá khách quan**

Tiêu chí phổ biến nhất được sử dụng để đánh giá khách quan là tỷ lệ tín hiệu trên nhiễu đỉnh (PSNR). Như được định nghĩa trong Phần 8.3. Đối với hình ảnh, đó là:

A black text on a white background

Description automatically generated

trong đó *Imax* là giá trị cường độ tối đa, ví dụ: 255 cho hình ảnh 8 bit và *M S E* là *Lỗi bình phương trung bình* giữa ảnh gốc *I* và hình ảnh nén *I* ̃. Đối với video, PSNR thường là trung bình của PSNR của hình ảnh trong chuỗi video.

Ohm et al. [19] đã báo cáo nhiều kết quả thí nghiệm của họ. Ví dụ: Table12.10 liệt kê mức giảm tốc độ bit trung bình khi các phương pháp nén video khác nhau được so sánh ở cùng một PSNR, trong trường hợp này trong phạm vi 32–42 dB. Dữ liệu thử nghiệm là các video giải trí thường có chất lượng cao hơn và có độ phân giải cao hơn. Các ký hiệu là: MP — Main Profile, HP — High Profile, ASP — Advanced Simple Profile. Ví dụ: khi H.265 MP được so sánh với H264 / MPEG-4 AVC HP, tiết kiệm 35,4% được thực hiện.

**12.3.2 Đánh giá chủ quan**

Ưu điểm chính của PSNR là dễ tính toán. Tuy nhiên, nó không nhất thiết phản ánh chất lượng như cảm nhận của con người, tức là chất lượng thị giác. Một ví dụ rõ ràng sẽ là thêm (hoặc trừ) một lượng nhỏ và cố định vào các giá trị cường độ của tất cả các pixel trong ảnh. Trực quan (chủ quan), chúng tôi có thể không nhận thấy bất kỳ thay đổi chất lượng nào. Mặt khác, PSNR chắc chắn sẽ bị ảnh hưởng.

Phương pháp đánh giá chủ quan hình ảnh truyền hình được quy định trong Khuyến nghị ITU-R BT.500. Phiên bản mới nhất của nó là BT.500-13 [20] sửa đổi vào năm 2012. Trong thí nghiệm của Ohm et al. [19], các video clip gốc và nén được hiển thị liên tiếp cho các đối tượng con người (cái gọi là phương pháp *kích thích kép*). Các đối tượng được yêu cầu chấm điểm video theo chất lượng của chúng, 0–thấp nhất, 10–cao nhất. Điểm ý kiến trung bình (MOS), là trung bình cộng của điểm số của họ, được sử dụng như

Thước đo chất lượng chủ quan của các phương pháp nén video khác nhau. Được biết, khi so sánh với H.264 / MPEG-4 AVC HP, ở cùng chất lượng chủ quan, đối với chín video thử nghiệm cho các ứng dụng giải trí, tốc độ bit trung bình giảm H.265 MP dao động từ 29,8 đến 66,6%, với mức trung bình của 49,3%. Điều này rất gần với mục tiêu ban đầu là giảm 50%. Đánh giá chất lượng video (VQA) là một lĩnh vực nghiên cứu tích cực. Những nỗ lực chính là tìm ra các số liệu tốt hơn so với các biện pháp đơn giản như PSNR, vì vậy các đánh giá có thể được tiến hành một cách khách quan (bằng máy tính) và kết quả của chúng sẽ tương đương với các đối tượng của con người. Wang et al. [21] đã trình bày  *chỉ số Tương tự về cấu trúc (SSIM)* ghi lại một số thông tin cấu trúc hình ảnh đơn giản (ví dụ: độ chói và độ tương phản). Nó đã trở nên rất phổ biến trong đánh giá chất lượng hình ảnh và video. Peng et al. [22] đã trình bày một cuộc khảo sát ngắn gọn về VQA và một số liệu tốt dựa trên một cuốn tiểu thuyết biểu diễn kết cấu không thời gian.

**12.4 Exercises**

1. Biến đổi số nguyên được sử dụng trong H.264 và H.265. (a) Mối quan hệ giữa DCT và Biến đổi số nguyên là gì? (b) Những ưu điểm chính của việc sử dụng Biến đổi số nguyên thay vì DCT là gì?
2. H.264 và H.265 sử dụng độ chính xác một phần tư điểm ảnh trong bù chuyển động.
   1. Lý do chính mà độ chính xác subpixel (trong trường hợp này là một phần tư pixel) được ủng hộ là gì?
   2. H.264 và H.265 khác nhau như thế nào khi có được ở vị trí một phần tư pixel?
3. Từ Eq. 12.15, suy ra **H8×8** cho Biến đổi số nguyên trong H.265.
4. H.264 và H.265 hỗ trợ *lọc chặn trong vòng lặp*.
5. Tại sao bỏ chặn là một ý tưởng hay? Nhược điểm của nó là gì?

(b) Sự khác biệt chính trong việc triển khai H.264 và H.265 là gì?

(c) Bên cạnh việc lọc bỏ chặn, H.265 làm gì để cải thiện chất lượng hình ảnh?

1. Kể tên ít nhất ba tính năng trong H.265 tạo điều kiện xử lý song song.
2. Đưa ra ít nhất ba lý do để lập luận rằng PSNR không nhất thiết phải là thước đo tốt để đánh giá chất lượng video.
3. Mã hóa khung P trong H.264 sử dụng *Biến đổi số nguyên*. Đối với bài tập này, giả sử:

A black text on a white background

Description automatically generated

1. (a) Hai ưu điểm của việc sử dụng Biến đổi số nguyên là gì?
2. (b) Giả sử Khung mục tiêu dưới đây là khung chữ P. Để đơn giản, giả sử kích thước của macroblock là 4 × 4. Đối với macroblock được hiển thị trong Khung Đích:
3. Vector chuyển động nên là gì?
4. Các giá trị của *f* (*i*, *j*) trong trường hợp này là gì?
5. Hiển thị tất cả các giá trị của *F*(*u*, v).

A table of numbers with a few words

Description automatically generated with medium confidence

1. Viết một chương trình cho *bộ mã hóa và giải mã Exp-Golomb thứ k. (a) Từ mã EG0 cho N* = 110 không dấu là gì? (b) Cho một *000000011010011 mã EG0, N không dấu được giải mã là gì*? (c) *Từ mã EG3 cho N* = 110 không dấu là gì?
2. Viết một chương trình để thực hiện nén video với bù chuyển động, mã hóa chuyển đổi và lượng tử hóa cho bộ mã hóa và giải mã H.26 \* được đơn giản hóa.
   * Sử dụng 4: 2: 0 để lấy mẫu phụ sắc độ.
   * Chọn chuỗi khung hình video (I-, P-, B-frames) tương tự như MPEG-1, 2. Không xen kẽ.
   * Đối với I-frame, thực hiện mã hóa dự đoán H.264 Intra\_4 × 4.
   * Đối với khung hình P và B, chỉ sử dụng 8 × 8 để ước tính chuyển động. Sử dụng tìm kiếm logarit cho vectơ chuyển động. Sau đó, sử dụng Biến đổi số nguyên 4 × 4 như trong H.264.

* • Use the quantization and scaling matrices as specified in Eqs. 12.5 and 12.7. Control and show the effect of various levels of compression and quantization losses.
* • Do not implement the entropy coding part. Optionally, you may include any publicly available code for this.

10. Write a program to verify the results in Table 12.8. For example, to show that DST will produce shorter code than DCT for Category 2 directional predictions.