**Tìm hiểu về H.264**

H.264 giảm yêu cầu băng thông trong khi chất lượng ảnh tương đương MPEG-2 và MPEG-4. Công nghệ này giúp cho tăng cường khả năng nén không gian và nén thời gian, cho hình ảnh truyền rất nhanh trên mạng LAN, internet. Với những ưu việt của mình, H.264 đang đựơc ứng dụng rất nhiều vào ngành an ninh.

**1. Giới thiệu chung**

Kể từ khi mới xuất hiện vào đầu những năm 90, chuẩn nén video MPEG-2 đã hoàn toàn thống lĩnh thế giới truyền thông. Cũng trong thập kỷ này, chuẩn nén MPEG-2 đã được cải tiến về nhiều mặt. Giờ đây nó có tốc độ bit thấp hơn và việc ứng dụng nó được mở rộng hơn nhờ có các kỹ thuật như đoán chuyển động, tiền xử lý, xử lý đối ngẫu và phân bổ tốc độ bit tùy theo tình huống thông qua ghép kênh thống kê.

Tuy nhiên, chuẩn nén MPEG-2 cũng không thể được phát triển một cách vô hạn định. Thực tế hiện nay cho thấy chuẩn nén này đã đạt đến hết giới hạn ứng dụng của mình trong lĩnh vực truyền truyền hình từ sản xuất tiền kỳ đến hậu kỳ và lưu trữ Video số. Bên cạnh đó, nhu cầu nén Video lại đang ngày một tăng cao kèm theo sự phát triển mạnh mẽ của mạng IP mà tiêu biểu là mạng Internet. Khối lượng nội dung mà các công ty truyền thông cũng như các nhà cung cấp dịch vụ thông tin có thể mang lại ngày càng lớn, ngoài ra họ còn có thể cung cấp nhiều dịch vụ theo yêu cầu thông qua hệ thống cáp, vệ tinh và các hạ tầng viễn thông đặt biệt là mạng Internet.

Các tiêu chuẩn mã hoá Video ra đời và phát triển với mục tiêu cung cấp các phương tiện cần thiết để tạo ra sự thống nhất giữa các hệ thống được thiết kế bởi những nhà sản xuất khác nhau đối với mọi loại ứng dụng Video; Nhờ vậy thị trường Video có điều kiện tăng trưởng mạnh. Chính vì lý do này nên những người sử dụng bộ giải mã cần có một chuẩn nén mới để đi tiếp chặng đường mà MPEG-2 đã bỏ dở.

Hiệp hội viễn thông quốc tế (ITU) và tổ chức tiêu chuẩn quốc tế/ Uỷ ban kỹ thuật điện tử quốc tế (ISO/IEC) là hai tổ chức phát triển các tiêu chuẩn mã hoá Video. Theo ITU-T, các tiêu chuẩn mã hoá Video được coi là các khuyến nghị gọi tắt là chuẩn H.26x (H.261, H.262, H.263 và H.264). Với tiêu chuẩn ISO/IEC, chúng được gọi là MPEG-x (như MPEG-1, MPEG-2 và MPEG-4).

Những khuyến nghị của ITU được thiết kế dành cho các ứng dụng truyền thông Video thời gian thực như Video Conferencing hay điện thoại truyền hình. Mặt khác, những tiêu chuẩn MPEG được thiết kế hướng tới mục tiêu lưu trữ Video chẳng hạn như trên đĩa quang DVD, quảng bá Video số trên mạng cáp, đường truyền số DSL, truyền hình vệ tinh hay những ứng dụng truyền dòng Video trên mạng Internet hoặc thông qua mạng không dây (wireless).

Với đối tượng để truyền dẫn Video là mạng Internet thì ứng cử viên hàng đầu là chuẩn nén MPEG-4 AVC, còn được gọi là H.264, MPEG-4 part 10, H.26L hoặc JVT.

**2. Tính kế thừa của chuẩn nén H.264**

Mục tiêu chính của chuẩn nén H.264 đang phát triển nhằm cung cấp Video có chất lượng tốt hơn nhiều so với những chuẩn nén Video trước đây. Điều này có thể đạt được nhờ sự kế thừa các lợi điểm của các chuẩn nén Video trước đây. Không chỉ thế, chuẩn nén H.264 còn kế thừa phần lớn lợi điểm của các tiêu chuẩn trước đó là H.263 và MPEG-4 bao gồm 4 đặc điểm chính như sau:

* Phân chia mỗi hình ảnh thành các Block (bao gồm nhiều điểm ảnh), do vậy quá trình xử lý từng ảnh có thể được tiếp cận tới mức Block.
* Khai thác triệt để sự dư thừa về mặt không gian tồn tại giữa các hình ảnh liên tiếp bởi một vài mã của những Block gốc thông qua dự đoán về không gian, phép biến đổi, quá trình lượng tử và mã hoá Entropy (hay mã có độ dài thay đổi VLC).
* Khai thác sự phụ thuộc tạm thời của các Block của các hình ảnh liên tiếp bởi vậy chỉ cần mã hoá những chi tiết thay đổi giữa các ảnh liên tiếp. Việc này được thực hiện thông qua dự đoán và bù chuyển động. Với bất kỳ Block nào cũng có thể được thực hiện từ một hoặc vài ảnh mã hoá trước đó hay ảnh được mã hoá sau đó để quyết định véc tơ chuyển động, các véc tơ này được sử dụng trong bộ mã hoá và giải mã để dự đoán các loại Block.
* Khai thác tất cả sự dư thừa về không gian còn lại trong ảnh bằng việc mã các block dư thừa. Ví dụ như sự khác biệt giữa block gốc và Block dự đoán sẽ được mã hoá thông qua quá trình biến đổi, lượng tử hoá và mã hoá Entropy.

**3. Cơ chế nén ảnh của H.264 (MPEG-4 AVC)**

Với chuẩn nén H264, mỗi hình ảnh được phân chia thành nhiều Block, mỗi block tương ứng với một số lượng nhất định các MacroBlock. Ví dụ một hình ảnh có độ phân giải QCIF (tương đương với số lượng điểm ảnh 176x144) sẽ được chia thành 99 MacroBlock với kích cỡ 16x16. Một sự phân đoạn các MacroBlock tương tự được sử dụng các kích cỡ ảnh khác. Thành phần chói của ảnh được lấy mẫu tương ứng với độ phân giải của ảnh đó, trong khi đó thành phần màu CR và CB được lấy mẫu với tần số thấp hơn theo 2 chiều ngang và dọc. Thêm vào đó mỗi hình ảnh có thể được phân thành số nguyên lần các lát mỏng (slice), việc này rất có giá trị cho việc tái đồng bộ trong trường hợp lỗi dữ liệu.

Mỗi hình ảnh thu được được xem như một ảnh I. Ảnh I là ảnh được mã hoá bởi việc áp dụng trực tiếp các phép biến đổi lên các MacroBlock khác nhau trong ảnh. Các ảnh I được mã hoá sẽ có kích cỡ lớn bởi nó được xây dựng từ một khối lượng lớn thông tin của bản thân ảnh hiện tại mà không sử dụng bất cứ thông tin nào từ miền thời gian trong quá trình xử lý mã hoá để tăng hiệu quả xử lý mã hoá bên trong trong H.264.

**3.1. Giảm bớt độ dư thừa**

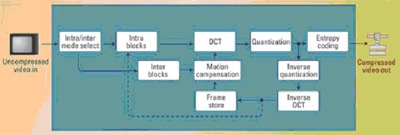
Cũng giống như các bộ lập giải mã khác, H.264 nén video bằng cách giảm bớt độ dư thừa cả về không gian và thời gian trong hình ảnh. Những dư thừa về mặt thời gian là những hình ảnh giống nhau lặp đi lặp lại từ khung (frame) này sang khung khác, ví dụ như phần phông nền không chuyển động của một chương trình đối thoại trên truyền hình. Dư thừa về không gian là những chi tiết giống nhau xuất hiện trong cùng một khung, ví dụ như nhiều điểm ảnh giống nhau tạo thành một bầu trời xanh. Hình 1 biểu diễn một cách sơ lược các bước mà bộ lập giải mã MPEG-4 phải tiến hành để nén không gian và thời gian.

**3.2. Chọn chế độ, phân chia và chế ngự**

Bộ lập giải mã bắt đầu bằng việc quyết định loại khung cần nén tại một thời điểm nhất định và chọn chế độ mã hoá phù hợp. Chế độ "trong khối" tạo ra ảnh "I", trong khi chế độ "giữa khối" tạo ra khung "P" hoặc "B". Sau đó, bộ mã hoá sẽ chia ảnh thành hàng trăm hàng và cột các điểm ảnh của ảnh video số chưa nén thành các khối nhỏ hơn, mỗi khối có chứa một vài hàng và cột điểm ảnh.

**3.3. Nén theo miền thời gian**

Khi bộ mã hoá đang hoạt động ở chế độ "giữa khối" (inter), khối này sẽ phải qua công đoạn hiệu chỉnh chuyển động. Quá trình này sẽ phát hiện ra bất kỳ chuyển động nào diễn ra giữa khối đó và một khối tương ứng ở một hoặc hơn một ảnh tham chiếu đã được lưu trữ từ trước, sau đó tạo ra một khối "chênh lệch" hoặc "lỗi". Thao tác này sẽ giảm bớt dữ liệu trong mỗi block một cách hiệu quả do chỉ phải trình bày chuyển động của nó mà thôi. Tiếp đến là công đoạn biến đổi côsin rời rạc (DCT) để bắt đầu nén theo miền không gian. Khi bộ mã hoá hoạt động ở chế độ "trong khối" (intra), khối này sẽ bỏ qua công đoạn hiệu chỉnh chuyển động và tới thẳng công đoạn DCT.

****

*Hình 1. Sơ đồ khối mã hoá MPEG, đường đứt nét đặc trưng cho phần bổ sung của MPEG-4 AVC trong việc nén theo miền không gian.*

**3.4. Nén theo miền không gian**

Các khối thường có chứa các điểm ảnh tương tự hoặc thậm chí giống hệt nhau. Trong nhiều trường hợp, các điểm ảnh thường không thay đổi mấy (nếu có). Như vậy có nghĩa là tần số thay đổi giá trị điểm ảnh trong khối này là rất thấp. Những khối như thế được gọi là khối có tần số không gian thấp. Bộ lập mã lợi dụng đặc điểm này bằng cách chuyển đổi các giá trị điểm ảnh của khối thành các thông tin tần số trong công đoạn biến đổi côsin rời rạc.

*\* Biến đổi cosin rời rạc:*

Công đoạn DCT biến đổi các giá trị điểm ảnh của khối thành một ma trận gồm các hệ số tần số ngang, dọc đặt trong không gian tần số. Khi khối ban đầu có tần số không gian thấp, DCT sẽ tập hợp phần lớn năng lượng tần số vào góc tần số thấp của mạng. Nhờ vậy, những hệ số tần số thấp ở góc đó sẽ có giá trị cao hơn.

Một số lượng lớn các hệ số khác còn lại trên ma trận đều là các hệ số có tần số cao, năng lượng thấp và có giá trị thấp. Hệ số DC và một vài hệ số tần số thấp sẽ hàm chứa phần lớn thông tin được mô tả trong khối ban đầu. Điều này có nghĩa là bộ lập mã có thể loại bỏ phần lớn hệ số tần số cao còn lại mà không làm giảm đáng kể chất lượng hình ảnh của khối.

Bộ lập mã chuẩn bị các hệ số cho công đoạn này bằng cách quét chéo mạng lưới theo đường zig-zag, bắt đầu từ hệ số DC và qua vị trí của các hệ số ngang dọc tăng dần. Do vậy nó tạo ra được một chuỗi hệ số được sắp xếp theo tần số.

*\* Lượng tử hoá và mã hoá entropy:*

Tại đây thao tác nén không gian mới thực sự diễn ra. Dựa trên một hệ số tỷ lệ (có thể điều chỉnh bởi bộ mã hoá), bộ lượng tử hoá sẽ cân đối tất cả các giá trị hệ số. Do phần lớn hệ số đi ra từ DCT đều mang năng lượng cao nhưng giá trị thấp nên bộ lượng tử hoá sẽ làm tròn chúng thành 0. Kết quả là một chuỗi các giá trị hệ số đã được lượng tử hoá bắt đầu bằng một số giá trị cao ở đầu chuỗi, theo sau là một hàng dài các hệ số đã được lượng tử hoá về 0. Bộ lập mã entropy có thể theo dõi số lượng các giá trị 0 liên tiếp trong một chuỗi mà không cần mã hoá chúng, nhờ vậy giảm bớt được khối lượng dữ liệu trong mỗi chuỗi.

**4. Các ưu điểm nổi bật của chuẩn nén H.264**

**4.1. Ưu điểm của nén không gian**

Chuẩn nén MPEG-4 AVC có hai cải tiến mới trong lĩnh vực nén không gian. Trước hết, bộ lập mã này có thể tiến hành nén không gian tại các macroblock 16x16 điểm ảnh thay vì các block 8x8 như trước đây. Điều này giúp tăng cường đáng kể khả năng nén không gian đối với các hình ảnh có chứa nhiều khoảng lớn các điểm ảnh giống nhau.

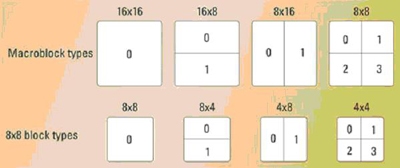
Thứ hai là thao tác nén được tiến hành trong miền không gian trước khi công đoạn DCT diễn ra. Chuẩn nén MPEG-4 AVC so sánh macroblock hiện thời với các macroblock kế bên trong cùng một khung, tính toán độ chênh lệch, và sau đó sẽ chỉ gửi đoạn chênh lệch tới DCT. Hoặc là nó có thể chia nhỏ macroblock 16x16 điểm ảnh thành các khối 4x4 nhỏ hơn và so sánh từng khối này với các khối kế bên trong cùng một macroblock. Điều này giúp cải thiện khả năng nén ảnh chi tiết.

**4.2. Ưu điểm của nén thời gian**

Điểm cải tiến lớn nhất ở MPEG-4 AVC là chế độ mã hoá giữa. Những phương pháp tiên tiến ở chế độ này khiến cho nén thời gian đạt đến một cấp độ cao hơn nhiều, cùng với chất lượng chuyển động tốt hơn so với các chuẩn MPEG trước đây.

**4.3. Kích cỡ khối**

Ở chế độ giữa khối, MPEG-2 chỉ hỗ trợ các macroblock 16x16 điểm ảnh, không đủ độ phân giải để mã hoá chính xác các chuyển động phức tạp hoặc phi tuyến tính, ví dụ như phóng to thu nhỏ. Ngược lại, MPEG-4 AVC lại tăng cường hiệu chỉnh chuyển động bằng cách cho phép bộ lập mã biến đổi kích cỡ thành phần chói của mỗi macroblock. (Bộ lập mã sử dụng thành phần chói như vậy là do mắt người nhạy cảm với chuyển động chói hơn nhiều so với chuyển động màu.) Như có thể thấy trong Hình 2, MPEG-4 AVC có thể chia thành phần chói của từng macroblock thành 4 cỡ: 16x16, 16x8, 8x16 hoặc 8x8. Khi sử dụng khối 8x8, nó còn có thể chia tiếp 4 khối 8x8 này thành 4 cỡ nữa là 8x8, 8x4, 4x8 hoặc 4x4.



|  |
| --- |
|  |

*Hình 2. MPEG-4 AVC có thể phân chia thành phần chói của từng MacroBlock theo nhiều cách để tối ưu hoá việc bù chuyển động.*

Việc phân chia các macroblock cho phép bộ lập mã xử lý được một vài loại chuyển động tuỳ theo độ phức tạp của chuyển động đó cũng như nguồn lực về tốc độ bit. Nhìn chung, kích cỡ phân chia lớn phù hợp với việc xử lý chuyển động tại các khu vực giống nhau trong ảnh, trong khi đó kích cỡ phân chia nhỏ lại rất có ích khi xử lý chuyển động tại các chỗ có nhiều chi tiết hơn. Kết quả là chất lượng hình ảnh cao hơn, ít bị vỡ khối hơn.

Các cuộc thử nghiệm đã chỉ ra rằng việc sắp xếp hợp lý các khung có thể tăng tỷ lệ nén thêm 15%. MPEG-4 AVC lấy phần chói của ảnh gốc và sử dụng các macroblock đã được chia nhỏ tại các khu vực có nhiều chi tiết nhằm tăng cường khả năng hiệu chỉnh chuyển động.

*\* Độ chính xác trong hiệu chỉnh chuyển động:*

Trong đa số trường hợp, chuyển động tại rìa mỗi macroblock hay khối thường diễn ra với độ phân giải nhỏ hơn một điểm ảnh. Do vậy, chuẩn nén MPEG-4 AVC có thể đảm bảo độ chính xác trong hiệu chỉnh chuyển động lên tới 1/4 hoặc 1/8 điểm ảnh, trong khi các chuẩn MPEG trước đây chỉ dừng lại ở mức 1/2 điểm ảnh. Khả năng đạt mức chính xác 1/8 ảnh điểm của MPEG-4 AVC giúp tăng hiệu suất mã hoá tại tốc độ bit cao và độ phân giải video cao. Các thử nghiệm cho thấy độ chính xác đến 1/4 điểm ảnh có thể làm giảm tốc độ bit xuống hơn 15% so với độ chính xác 1 điểm ảnh.

*\* Chọn nhiều hình tham chiếu:*

Chuẩn nén MPEG-2 chỉ dựa trên 2 khung tham chiếu để dự đoán các chuyển động mang tính chu kỳ, giống như trong trò kéo quân. Tuy nhiên, khi camera thay đổi góc quay hay chuyển qua chuyển lại giữa các cảnh, việc chỉ sử dụng 2 khung tham chiếu không còn phù hợp để dự đoán chính xác chuyển động. Tương tự như vậy, để đoán trước các chuyển động phức tạp như sóng biển hay một vụ nổ, ta cần phải có nhiều hơn 2 khung tham chiếu. Vì thế, chuẩn MPEG-4 AVC cho phép có tới 5 khung tham chiếu phục vụ cho việc mã hoá giữa khung. Kết quả là chất lượng video tốt hơn và hiệu suất nén cao hơn.

*\* Giải khối tích hợp:*

Video số sau khi nén thường tạo ra một hiệu ứng gọi là "kết khối", có thể thấy rõ tại điểm giao nhau giữa các khối, đặc biệt là khi có tốc độ bit thấp. Hiệu ứng này là do công đoạn xử lý sử dụng nhiều loại chuyển động và bộ lượng tử khác nhau. Đối với MPEG-2, cách duy nhất để ngăn chặn hiệu ứng này là sử dụng các cơ chế hậu xử lý phù hợp, tuy nhiên các cơ chế này lại không tương thích được với tất cả các máy thu. Chuẩn nén MPEG-4 AVC đưa vào sử dụng một bộ lọc giải khối hoạt động ở hai cấp độ: macroblock 16x16 và khối 4x4. Việc giải khối thường tạo ra một tỉ số tín hiệu trên nhiễu (PSNR) cực điểm thấp hơn, tuy nhiên nhìn một cách chủ quan thì nó tạo ra hình ảnh chất lượng tốt hơn.

**4.4. Ưu điểm về lượng tử hoá và biến đổi.**

Chấm di động 8x8 DCT cùng với dung sai của lỗi làm tròn chính là phần cốt lõi của các chuẩn MPEG trước đây. MPEG-4 AVC độc đáo hơn ở chỗ nó sử dụng biến đổi không gian nguyên (gần giống như DCT) đối với các khối 4x4 điểm ảnh. Kích cỡ nhỏ giúp giảm bớt hiện tượng "kết khối", trong khi thông số nguyên tuyệt đối giúp loại bỏ nguy cơ không thích ứng giữa bộ lập mã và giải mã trong phép biến đổi ngược. Thêm vào đó, dãy hệ số xích lượng tử lớn hơn khiến cho cơ chế kiểm soát tốc độ dữ liệu ở bộ lập mã hoạt động một cách linh hoạt hơn dựa trên một tỉ lệ phức hợp vào khoảng 12,5% thay cho một mức tăng lượng gia không đổi.

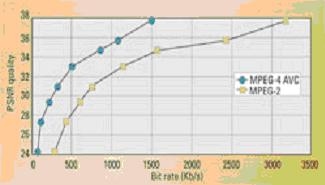
**4.5. Ưu điểm đối với mã hoá entropy.**

Sau khi tiến hành hiệu chỉnh, biến đổi và lượng tử hoá chuyển động, các bộ lập mã MPEG trước đây sẽ vạch ra các symbol biểu diễn véctơ chuyển động và hệ số đã lượng tử hoá thành các bit thực sự. Ví dụ như chuẩn nén MPEG-2 sử dụng phương pháp mã có chiều dài biến thiên tĩnh (VLC) không thể tối ưu hoá trong môi trường video thời gian thực (trong đó nội dung và các cảnh biến đổi theo thời gian).

MPEG-4 AVC sử dụng mã hoá thuật toán nhị phân theo tình huống CABAC (Context-Adaptive Binary Arithmetic Coding). Hiệu suất mã hoá của CABAC cao hơn hẳn nhờ khả năng thích nghi với các thay đổi có thể xảy ra trong phân bổ symbol. Ví dụ, nó có thể khai thác sự tương quan giữa các symbol và từ đó sử dụng sự tương quan bit và thuật toán mã hoá. Cơ chế này có thể giúp tiết kiệm thêm một lượng bit vào khoảng hơn 5%.

**4.6. Một ví dụ về ưu thế của MPEG-4 so với MPEG-2.**

Hình 3 là biểu đồ so sánh chất lượng giữa MPEG-4 AVC với MPEG-2. Nó so sánh hoạt động của các bộ lập mã tối tân khi mã hoá một đoạn video 30 khung/s có độ phân giải CIF ghi hình một trận thi đấu tennis.

**

*Hình 3. So sánh chất lượng và tốc độ giữa MPEG-4 và MPEG-2.*

|  |  |  |  |
| --- | --- | --- | --- |
| Codec | MPEG-2 part 2 (ASP) | H.263 | MPEG-2 |
| MPEG-4 AVC | 39% | 49& | 64% |

*Bảng 1 biểu diễn mức tiết kiệm tốc độ bit của MPEG-4 AVC so với một số chuẩn hiện có khác.*

**5. Kết luận.**

MPEG-4 AVC đánh dấu một bước ngoặt trong lĩnh vực nén video, áp dụng các kỹ thuật tiên tiến nhằm mục đích sử dụng băng thông hiệu quả hơn và đem lại chất lượng ảnh cao hơn. Với các kỹ thuật này, MPEG-4 AVC có thể giảm tốc độ bit xuống hơn 50% so với chuẩn MPEG-2. Tuy nhiên, MPEG-4 AVC đòi hỏi một cấp độ phức tạp cao hơn trong cả quá trình lập mã lẫn giải mã. Mặc dù vậy, thử thách này hoàn toàn có thể chinh phục được nhờ có những tiến bộ mới liên tiếp trong khả năng xử lý phần mềm cũng như phần cứng. Điều này có nghĩa là MPEG-4 AVC là một ứng cử viên nặng ký có khả năng thay thế MPEG-2 trong những năm sắp tới.

**6. Phụ lục.**

**6.1. Giới thiệu setting x264 cho MEGUI**

Bài này mình sẽ giới thiệu về ý nghĩa các settings x264. Trong MeGui khi ấn Config, ta thấy có 1 loạt tab như Frame-Type, Rate Control, Analysis xuất hiện cùng với 1 mớ thông số khó hiểu. Những thông số đó chính là setting của x264. Tùy vào cách chỉnh những thông số này mà chất lượng bản encode sẽ khác nhau.

**6.1.1. Cơ bản về Frame**

Trong tab Frame-Type, ta thấy có 1 vài thông số ghi B-Frames, I-Frames hay P-Frames. Để hiểu các loại Frames này, ta cần tìm hiểu sơ lược về Frames.

Frame là gì? Nói dễ hiểu thì Frame là khung hình. Chẳng hạn mọi người vẫn hay nghe 24 hình / giây, mỗi hình đó chính là 1 Frame.

Video được nén với công nghệ mới hiện nay thường bao gồm 3 loại Frames: I, P và B. Mỗi loại Frames được tạo ra bởi các giải thuật khác nhau và có các đặc điểm lợi hại khác nhau.

* **Intra (I) Frames (hay còn gọi là Key Frames)** : Lưu trữ toàn bộ hình ảnh, do đó độ nén là thấp nhất ~ dung lượng lớn nhất. Do lưu trữ toàn bộ hình ảnh nên I Frames không đòi hỏi cần phải có Frames khác để Decode. Khi xem phim, ta muốn tua đến 1 đoạn nào đó của phim đòi hỏi ở đó phải có I Frames. Đôi khi tua phim ta thường gặp trường hợp bị lag, phim đứng 1 lúc rồi mới chạy, đó là do điểm ta tua tới không có I Frames, nó sẽ tìm cho đến khi nào gặp được I Frames mới chạy được.
* **Predictive (P) Frames:** Chỉ lưu trữ những điểm khác biệt giữa nó với khung hình trước nó (ta gọi là tham khảo đến khung hình trước) ==> Dung lượng nhỏ hơn I Frames. Khung hình trước đó có thể là 1 I Frame hoặc 1 P Frame khác.   
    
  Để dễ hiểu, hãy xem 1 ví dụ thực tế như thế này. Người A giới thiệu về bản thân, 1 loạt thông tin như họ tên, nghề nghiệp, quê quán, giới tính, cha mẹ .... Để lưu trữ thông tin về người A, ta cần viết đầy 1 trang giấy A4, không thể thiếu gì. Người B là anh em sinh đôi với người A, có tất cả thông tin giống hệt người A, chỉ khác tên. Do đó, để lưu trữ thông tin người B, ta chỉ cần viết lại 2 dòng, dòng thứ 1 viết tên, dòng thứ 2 viết "Các thông tin khác giống hệt người A". Trong ví dụ này, người A chính là I Frames, còn người B là P Frames.  
    
  Trong encode phim, giả sử có 1 đoạn phim là 1 người đi ngang qua màn hình. P Frame chỉ cần lưu giữ lại thông tin chuyển động của người đó mà không cần lưu thông tin phong cảnh xung quanh vì phong cảnh hoàn toàn đứng yên không thay đổi.

Như đã nói ở trên, bản thân P Frames cũng có thể được tham khảo bởi 1 Frame khác. Để làm được điều này thì P Frames đó phải chứa đủ lượng thông tin cần thiết (đương nhiên là không cần nhiều như I Frames).

* **Bidirectional (B) Frames**: tương tự P Frames, B Frames cũng chỉ lưu thông tin khác nhau giữa nó và Frames khác. Nếu P Frames chỉ sử dụng các Frames trước nó để tham khảo thì B Frames tham khảo cả các Frames trước và sau nó, và B Frames không cho phép các Frames khác tham khảo (Ngoại trừ "B-frame pyramid"). Điều này giúp B Frames đạt độ nén tốt nhất ~ dung lượng nhỏ nhất trong các loại Frames và nếu được đặt đúng chỗ có thể cho ra chất lượng không hề giảm sút. x264 còn hỗ trợ "B-frame pyramids", loại B Frames này cho phép các B-Frames khác tham khảo tới nó.

Ưu điểm của các loại Frames tham khảo các Frames khác quá rõ ràng là làm giảm dung lượng video được nén. Nhược điểm của việc này là nếu Frame Y tham khảo Frame X mà Frame X lại bị lỗi thì lỗi đó sẽ truyền sang Frame Y và cứ tiếp tục như vậy cho các Frame nào tham khảo Y.

Tóm lại, 1 file video sẽ là 1 tổ hợp các Frames I, P và B, 1 vài Frames tham khảo những Frames khác. Sự thành công của việc tổ hợp và tham khảo này sẽ quyết định chất lượng video.  
  
**6.1.2. Ý nghĩa các thông số liên quan đến Frame-Type**

**bframes:** Lựa chọn số lượng B-Frames liên tục tối đa mà x264 có thể sử dụng. Nếu không sử dụng B-Frames, 1 chuỗi Frames có thể có dạng như sau: IPPPPP...PI. Nếu sử dụng B-Frames = 2, tối đa 2 P-Frames có thể thay thế bằng B-Frames: IBPBBPBPPPB...PI.

**Encode với Command Line**: --bframes n (n là số lượng B-Frames)

**Encode với MeGui:** Number of B-frames.

**Adaptive - b-adapt:** Cho phép x264 quyết định mức độ sử dụng B-Frames nhiều hay ít.

* 0. Luôn sử dụng B-Frames.
* 1. Giải thuật nhanh ("Fast"). Thời gian encode nhanh hơn chút ít khi giá trị --b-frames càng cao. Nếu sử dụng giá trị này, không bao giờ nên để bframes thấp hơn 16.
* 2. Giải thuật tối ưu ("Optimal"). Thời gian encode chậm đáng kể khi giá trị --b-frames càng cao.

**Command Line:** --no-b-adapt hoặc --b-adapt n

**MeGui:**Adaptive B-frames.

**Bias - b-bias:** Giá trị càng cao thì x264 sẽ sử dụng càng nhiều B-Frames để thay thế cho P-Frames. Nếu giá trị này được set bằng 100 tương đương việc b-adapt = 0.

**Command Line:** --b-bias n

**MeGui:**B-frame bias.

**Pyramid - b-pyramid:** Cho phép B Frames được tham khảo bởi các Frames khác. Nếu không có thông số này, chỉ I hoặc P Frames được tham khảo bởi các Frames khác. Để sử dụng thông số này, cần phải dùng ít nhất 2 B-frames. Chỉ khi encode Bluray mới sử dụng none hoặc strict, còn lại nên để default là normal.

**Command Line:** --b-pyramid n

**MeGui:**B-pyramid.  
  
**Weighted Prediction - weightb:** Kích hoạt chức năng đánh giá trọng số cho B-Frames, từ đó cho ra kết quả chính xác và hiệu quả hơn trong quá trình encode.

**Command Line:** --weightb n

**MeGui:**Weighted Prediction for B-frames.

**Reference Frames - ref:** Số lượng Frames mà 1 Frame có thể tham khảo tới. Số này càng lớn chất lượng sẽ tăng tuy nhiên thời gian encode cũng sẽ tăng theo đáng kể. Tuy nhiên tham số này vẫn là tham số xứng đáng trong việc đổi thời gian lấy chất lượng. Thông thường giá trị này nên để trong khoảng từ 3-5. Với các nội dung có sự lặp lại, tương tự về mặt hình ảnh, giá trị này có thể để lên 8-10.

Lưu ý là khi sử dụng Level 4.1(đã cấu hình ở bài trước), số lượng Reference Frames tối đa cho video 720p là 9 và 1080p là 4. Để quá cao sẽ dẫn đến dư thừa lãng phí, và yêu cầu phần cứng cao hơn để có thể decode. Tổng quát, ta có thể tính số lượng Reference Frames tối đa cho Level 4.1 theo công thức sau:

Code:

**--ref = Truncate(8388608/(width \* height))**

Truncate là làm tròn đến số nguyên nhỏ hơn gần nó nhất. Chẳng hạn phép chia cho ra kết quả 4.165 -> 4 hay 9.856 -> 9

**Command Line:** --ref n

**MeGui:**Number of Reference Frames

**GOP Size (Min/Max) - min-keyint, keyint :** Số lượng Frames tối thiểu và tối đa trước đi 1 Key Frame được chèn vào bởi x264.

Ví dụ giá trị min-keyint hay Min GOP Size thường được chọn bằng với giá trị fps. Fps là số lượng Frame có trong 1 giây. Thông thường Fps=24 ==> Ít nhất sau 24 Frames mới có 1 Key Frame được chèn vào. Việc chọn giá trị này sẽ đảm bảo không bao giờ có 2 Key Frames cùng xuất hiện trong vòng 1 giây (việc đó sẽ chỉ làm cho dung lượng bản encode tăng cao mà chất lượng hầu như không khác biệt).

Tương tự, Max GOP Size = X sẽ đảm bảo phải có 1 Key Frame xuất hiện cứ sau X Frame. 1 Video có quá ít Key Frame sẽ dẫn đến việc chất lượng giảm sút, và dễ gây hiện tượng lag khi nhảy đến 1 thời điểm nhất định của video (do điểm đó không có Key Frame, video sẽ phải tìm kiếm Key Frame tiếp theo xuất hiện).

Như đã nói, giá trị Min GOP Size(min-keyint) thường được chọn bằng với giá trị fps. Còn Max GOP Size(keyint) được khuyến cáo bằng 10 lần giá trị fps (đảm bảo rằng nếu hiện tượng lag xuất hiện cũng sẽ chỉ kéo dài trong tối đa 10 giây). Giá trị lớn nhất mà min-keyint có thể nhận là **keyint/2+1**

**Command Line:** --keyint m / --min-keyint n (m, n là số lượng max và min)

**MeGui:**Maximum & Minimum GOP Size.

**Scene Cut - scenecut:** Cho x264 quyết định mức độ nhạy khi chuyển cảnh. x264 sẽ tính toán giá trị cho mỗi frame để xem độ khác biệt giữa nó với các frame trước. Từ giá trị này, nó sẽ quyết định khi nào video xuất hiện việc chuyển cảnh và do đó sẽ cần phải chèn vào Key Frame. Giá trị này càng lớn, x264 sẽ càng nhạy hơn với các phần chuyển cảnh. Ví dụ 1 video với các camera chuyển động ít sẽ đòi hỏi giá trị này cao hơn.

**Command Line:** --scenecut n hoặc --no-scenecut

**MeGui:**Bỏ chọn Adative I-Frames Decision tuơng đuơng --no-scenecut. Number of Extra I-Frames tuơng đương giá trị n trong --scenecut n.

**CABAC:** Tăng hiện quả trong việc nén video. Khuyến cáo luôn chọn trừ khi video được encode để xem trên các phuơng tiện bị giới hạn khả năng decode (như các phương tiện di động iPod, PSP...).

**Command Line:** --no-cabac (không sử dụng CABAC).

**MeGui:**Chọn hoặc bỏ chọn CABAC.

**Deblock aka Loop filter :** Đây là 1 thông số **rất quan trọng trong việc encode** để khử những khối vuông - blocks. Blocks sẽ gây ra 1 hiệu ứng gọi là artifact (từ điển gọi là giả tạo, mình cũng chả biết dịch thế nào) - cần phải loại bỏ.

x264 có 2 tham số liên quan đến việc deblock gọi là Deblocking strength(Alpha Deblocking) và Deblocking threshold(Beta Deblocking). Deblock setting thường được viết dưới dạng X:Y, số đầu là Alpha Deblocking và số sau là Beta Deblocking.

Alpha Deblocking quyết định độ mạnh yếu của việc deblock trên toàn bộ hình ảnh, giá trị càng cao tức là việc deblock sẽ càng hiệu quả, nhưng đồng thời cũng phá hủy đi nhiều chi tiết hình ảnh và khiến hình ảnh có cảm giác dịu hay nhòe đi.

Beta Deblocking quyết định độ mạnh yếu của việc phát hiện ra block của x264. Nếu giá trị này càng cao, x264 sẽ xem video càng có nhiều block hơn so với giá trị thấp.

Ví dụ, nếu giá trị Beta Deblocking và Alpha Deblocking đều được đặt quá cao, x264 sẽ phát hiện ra quá nhiều block (mặc dù đôi khi nó không gây ra artifacts) và áp dụng quá nhiều filter vào để khử chúng đi, khiến hình ảnh mất chi tiết.

Cả 2 thông số này đều có thể nhận giá trị từ -6 đến 6. Lời khuyên là đừng bao giờ xuống quá -3 và lên quá 3.

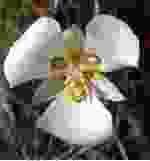
**Command Line:** --no-deblock hoặc --deblock X:Y

**MeGui:**Deblocking strength, Deblocking threshold

**6.2. Nhiễu khối vuông và nhiễu vòng trong video nén**

***Nhiễu khối vuông và nhiễu vòng là gì?***

Nhiễu khối vuông hay còn gọi hiệu ứng ô vuông trong các ảnh hay video đã được nén như trong **hình 1;** và**hình 2** là nhiễu vòng, nó hiển thị mờ ảo xung quanh các đối tượng ảnh.

[](http://2.bp.blogspot.com/-ClsJKTF_HNo/Tafo6p7kaSI/AAAAAAAAAAc/z7OG3_r4tXY/s1600/Picture1.png) [](http://2.bp.blogspot.com/-6sHCPHWtBRI/TafpLEWmvEI/AAAAAAAAAAg/1YjmOvCVAhE/s1600/Picture2.png)

Hình 1: Nhiễu khối vuông (blocking artifact) Hình 2: Nhiễu vòng (Ringing artifact)

***Vì sao có các loại nhiễu này?***

Như đã biết, một tập tin video bao gồm nhiều các khung hình riêng lẻ (gọi là frame). Các frame liên tiếp nhau có sự tương quan rất lớn, giữa các frame ở phía bộ mã hóa (encoder) chứa nhiều thông tin trùng hợp (hay dư thừa) không nhất thiết phải chuyển toàn bộ sang phía bộ giải mã (decoder) để đọc. Sự khác nhau giữa chúng đôi khi chỉ là các chuyển động của đối tượng so với background. Chính vì vậy, hầu hết các giải thuật nén ảnh và video đều có quá trình lượng tử hóa (quantization). Lượng tử hóa được xem là một trong những bước then chốt nhằm mục đích giảm thiểu tối đa thông tin dư thừa khi lưu trữ và truyền thông tin ảnh hay video.

Hơn nữa, trong mỗi frame, để có được tính tương quan cao nhất việc lượng tử hóa thường được thực hiện trong các vùng 8x8 hay 16x16. Các vùng này được xử lý một cách độc lập bằng biến đổi sang miền tần số chẳng hạn Cosin rời rạc (DCT) ở encoder và sau đó được biến đổi ngược lại bằng IDCT (inverse DCT) ở decoder. Những xử lý này đã dẫn đến các ảnh hưởng làm mất các thành phần tần số thấp gây ra hiệu ứng khối vuông (blocking artifact) – **hình 1**, hay hiệu ứng vòng (ringing artifact) do mất các thành phần năng lượng tần số cao – **hình 2**.

Việc nén này làm giảm đáng kể dữ liệu dư thừa cho việc lưu trữ cũng như truyền thông tin hình ảnh và video. Đối với các chuẩn nén cho audio (MP3) cũng có áp dụng các phép biến đổi cơ bản trong quá trình lượng tử hóa, tuy nhiên chúng ta vẫn khó hoặc không cảm nhận được các loại nhiễu này bằng thính giác. Còn đối với các chuẩn nén cho ảnh, phổ biến như JPEG, JPEG2000 hay MPEG thì mắt thường có thể thấy rõ được những loại nhiễu này.

Các bộ lọc xử lý nhiễu này (thường được gọi là De-blocking filter và De-ringing filter) gồm có 4 loại cơ bản sau:

* ***Lọc tuyến tính hay phi tuyến*** (Linear/non-linear filtering): Phương pháp này có ưu điểm về thời gian tính toán nhưng các kết quả thường bị mờ và làm mất các chi tiết đối tượng ở biên và cạnh.
* ***Lặp phục hồi quy tắc*** (Iterative regularization restoration): Phương pháp này sử dụng tham số regularization có ưu điểm duy trì các chi tiết của đối tượng ảnh nhưng với việc sử dụng kĩ thuật lặp nên thời gian tính toán cao. Dó đó, dẫn đến vấn đề “thắt cổ chai” trong các ứng dụng video thời gian thực.
* ***Lọc lặp*** (Loop filter): hiệu quả cho giảm hiệu ứng khối vuông tuy nhiên độ phức tạp tính toán cao và không giảm được hiệu ứng vòng hiển thị xung quanh các đối tượng ảnh.
* ***Hậu xử lý*** (Post-processing): phương pháp này có ưu điểm là không cần tác động tới các thủ tục mã hóa đã tồn tại trong khi đó xử lý được cả 2 loại nhiễu blocking và ringing. Và một điều đặc biệt là nó không làm tăng tốc độ bit (bit-rate) – một yếu tố trọng yếu để đảm bảo đáp ứng tính chất thời gian thực của video. Dưới đây là một vài kết quả của phương pháp Post-processing đã được thực hiện cho các chuỗi video kiểu định dạng tập tin YUV.

[](http://1.bp.blogspot.com/-D5vEOZiYqfY/TafrcnQCsyI/AAAAAAAAAAk/IfaokXektFs/s1600/Picture3.png)

Hình 3: ảnh thực nghiệm nhiễu khối vuông - Drama (đã được phóng to X2)

[](http://1.bp.blogspot.com/--cfY23BbEIQ/Tafr1FoqPVI/AAAAAAAAAAo/C8Yf0eK8vmU/s1600/Picture4.png)

 Hình 4: ảnh thực nghiệm được lọc - Drama (đã được phóng to X2)

[](http://4.bp.blogspot.com/-Dd1lnXUuIiI/Tafx_D9ZEaI/AAAAAAAAAA0/DhY9F9nrIIk/s1600/1.JPG)

Hình 5: Ảnh thực nghiệm nhiễu - Swimming (đã phóng to X2)

[](http://2.bp.blogspot.com/-trUGygiuhms/TafyHGCp9NI/AAAAAAAAAA4/j7oIcrWSqEw/s1600/2.JPG)

 Hình 6: Ảnh thực nghiệm đã được lọc - Swimming (đã phóng to X2)

***Nhận xét:*** cũng như các bài toán về phục hồi khác, kết quả đạt được không phải là duy nhất và không bao giờ là tối ưu nhất, tức là giống hoàn toàn ảnh gốc khi chưa bị nén bởi vì JPEG, JPEG2000, MP3, hay MPEG đều là các giải thuật nén tổn thất (bởi một phần do xử lý lượng tử hóa). Việc đạt được kết quả tốt cần phải tính đến 3 yếu tố: vừa đảm báo tối thiểu tốc độ tính toán thời gian thực, vừa bảo toàn chi tiết đối tượng trong frame, và vừa lọc được càng nhiều càng tốt các hiệu ứng của lượng tử hóa.