**MỞ ĐẦU**

Hiện nay, mạng Internet nói riêng và những thành tựu trong lĩnh vực công nghệ thông tin – truyền thông nói chung đã và đang tiếp tục mở rộng phát triển nhanh chóng, đạt được những dấu mốc quan trọng.Khi nhắc tới ngành công nghệ thông tin – truyền thông và đặc biệt là thế giới ảo trên mạng toàn cầu rộng lớn, chúng ta sẽ phải nhắc ngay đến khái niệm “Dữ liệu đa phương tiện” – một bước đột phá trong lịch sử IT loài người. “Dữ liệu đa phương tiện” là tất cả những dạng dữ liệu được các thiết bị như camera,scanner,micro… tạo ra, xử lí và được truyền thông giữa các thành phần trong mạng thông qua những phương thức đặc thù riêng.Dữ liệu đa phương tiện rất phong phú, nhưng có thể nêu ra một số đối tượng cơ bản, phổ biến nhất như: tệp dữ liệu âm thanh (audio file), tệp dữ liệu hình ảnh/đồ họa số (image/graphic file) hay các tệp phim/hình chuyển động (video file)...

Trong dữ liệu đa phương tiện JPEG là một chuẩn dữ liệu được sử dụng phổ biến hiện nay, cả trong đời sống lẫn công tác nghiên cứu khoa học.Việc ứng dụng chuẩn JPEG trong truyền thông,lưu trữ… đã thực sự mang lại những lợi ích lớn cho người dùng.Thấy được tầm quan trọng của JPEG và để mọi người có được sự hiểu biết đầy đủ về loại chuẩn này chúng em đã quyết định chọn nghiên cứu đề tài bài tập lớn:

***“Kỹ thuật mã hóa nén ảnh theo chuẩn JPEG”***

Việc tìm hiểu, khảo sát về các đối tượng dữ liệu đa phương tiện đòi hỏi công sức và chi phí rất lớn.Trong suốt quá trình thực hiện bài tập này, nhóm đã rất nỗ lực tìm hiểu, khảo sát, đánh giá về những nội dung cần thực hiện. Tuy nhiên không tránh khỏi những thiếu sót, sơ suất rất mong sự góp ý của cô giáo và các bạn.

Để hoàn thành đề tài này chúng em xin chân thành cảm ơn:

* PGS-TS. Nguyễn Thị Hoàng Lan đã nhiệt tình chỉ bảo chúng em trong suốt quá trình thực hiện bài tập lớn.
* Các bạn trong lớp TTM K52 đã đóng góp ý kiến và có những nhận xét quý báu.

**NỘI DUNG**

**1.Giới thiệu chung về chuẩn JPEG.**

JPEG ( Joint Photographic Expert Group ) là tên của một tổ chức nghiên cứu về các chuẩn nén ảnh (trước đây là ISO) được thành lập vào năm 1982. Năm 1986, JPEG chính thức được thiết lập nhờ sự kết hợp giữa nhóm ISO/IEC và ITV. Tiêu chuẩn này có thể được ứng dụng trong nhiều lĩnh vực : lưu trữ ảnh, Fax màu, truyền ảnh báo chí, ảnh cho y học, camera số…

Tiêu chuẩn JPEG (Joint Photographic Experts Group) được định ra cho nén ảnh tĩnh đơn sắc và màu. Công nghệ nén ảnh JPEG là một trong những công nghệ nén ảnh hiệu quả, cho phép làm việc với các ảnh có nhiều màu và kích cỡ lớn. Tỷ lệ nén ảnh đạt mức vài chục lần. Thông thường các ảnh hiện nay dùng 8 bit (1 byte) hay 256 mức cường độ tương ứng của các màu đỏ, xanh lá cây và xanh da trời. Mỗi điểm của ảnh cần 3 byte để lưu mã màu, và lượng byte một ảnh màu này chiếm gấp 24 lần ảnh trắng đen cùng cỡ. Với những ảnh này các phương pháp nén ảnh không mất mát thông tin chỉ đạt tỉ lệ 2:1 hay 3:1.Nó không đáp ứng được yêu cầu với một số ứng dụng. Ưu điểm cao của phương pháp này là ảnh đã nén sau khi giải nén sẽ trùng khớp với ảnh ban đầu. Một số phương pháp nén khác không để mất thông tin như của Lempel - Ziv and Welch (LZW) có thể cho hệ số nén tới 6:1. Nhưng như thế cũng chưa thật đáp ứng yêu cầu đòi hỏi thực tế.Phương pháp nén ảnh theo thuẩn JPEG có thể cho hệ số nén tới 80:1 hay lớn hơn, nhưng ta phải chịu mất thông tin (ảnh sau khi bung nén khác với ảnh ban đầu), lượng thông tin mất mát tăng dần theo hệ số nén. Tuy nhiên sự mất mát thông tin này không bị làm một cách cẩu thả. JPEG tiến hành sửa đổi thông tin ảnh khi nén sao cho ảnh mới gần giống như ảnh cũ, khiến phần đông mọi người không nhận thấy sự khác biệt. Và ta hoàn toàn có thể quản lý sự mất mát này bằng cách hạn chế hệ số nén. Như thế người dùng có thể cân nhắc giữa cái lợi của việc tiết kiệm bộ nhớ và mức độ mất thông tin của ảnh, để chọn phương án thích hợp.

Phương pháp nén ảnh JPEG dựa trên nguyên lý sau: ảnh màu trong không gian của 3 màu RGB (red Green Blue) được biến đổi về hệ YUV (hay YCbCr) (điều này không phải là nhất thiết, nhưng nếu thực hiện thì cho kết quả nén cao hơn). Hệ YUV là kết quả nghiên cứu của các nhà sản xuất vô tuyến truyền hình hệ Pal, Secam và NTSC, nhận thấy tín hiệu video có thể phân ra 3 thành phần Y, U, V (cũng như phân theo màu chuẩn đỏ, xanh lá cây và xanh da trời). Và một điều ­­­ là hệ nhãn thị của con người rất nhạy cảm với thành phần Y và kém nhạy cảm với hai loại U và V. Phương pháp JPEG đã nắm bắt phát hiện này để tách những thông tin thừa của ảnh. Tỉ số nén thành phần Y của ảnh nhỏ hơn so với U, V, bởi người ta ít nhận thấy sự thay đổi của U và V so với Y.JPEG cũng được sử dụng cho nhiều ứng dụng với ảnh động bởi vì nó cho chất lượng ảnh khôi phục khá tốt và ít tính toán hơn so với nén MPEG**. JPEG nén hình ảnh được sử dụng rộng rãi trong việc phát triển trang web so với bitmap (phần mở rộng .bmp). Những ảnh JPEG chiếm không gian ít hơn và do đó có thể được tải về nhanh chóng khi chúng ta truy cập một trang web.** Nén JPEG có thể thực hiện bởi bốn phương mã hóa đó là:

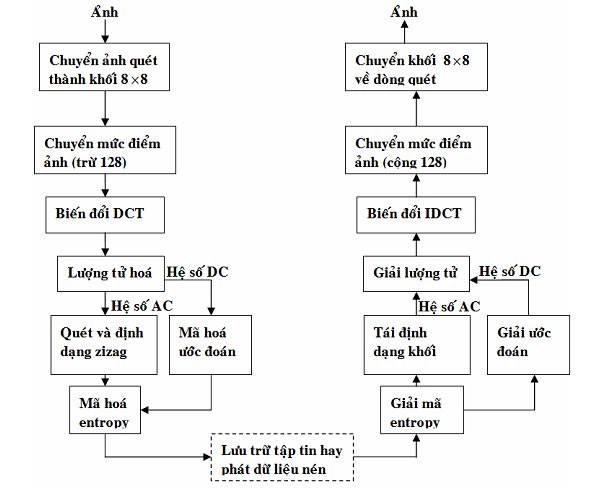
+ Mã tuần tự (sequential DCT-based) : ảnh được mã hóa theo kiểu quét từ trái qua phải, từ trên xuống dưới dựa trên khối DCT.

+ Mã hóa lũy tiến (progressive DCT-based): ảnh được mã hóa bằng kiểu quét phức hợp theo chế độ phân giải không gian cho các ứng dụng trên kiểu băng hẹp và do đó thời gian truyền dẫn có dài.

+ Mã hóa không tổn thất (lossless) : ảnh được đảm bảo khôi phục chính xác cho mỗi giá trị mẫu của nguồn. Thông tin không cần thiết sẽ mới cắt bỏ cho nên hiệu quả nén thấp hơn so với phương pháp có tổn thất.

+ Mã hóa phân cấp (hierarchical) : ảnh được mã hóa ở chế độ phân giải không gian phức hợp, để cho những ảnh có độ phân giải thấp có thểđược truy xuất và hiển thị mà không cần giải nén như những ảnh có độ phân giải trong không gian cao hơn.

Mã hóa không tổn thất không sử dụng cho video động bởi vì nó cung cấp một tỉ lệ nén không đủ cao. Tỉ lệ nén ảnh tĩnh có thể đạt từ 1/10 đến 1/50 mà không làm ảnh hưởng đến chất lượng hiển thị của ảnh. Khai triển DCT được chọn là kỹ thuật then chốt trong JPEG vì nó cho ảnh nén chất lượng tốt nhất tại tốc độ bit thấp và giải thuật chuyển đổi nhanh và dễ dàng thực hiện bằng phần cứng. Hình dưới là sơ đồ mã hóa và giải nén JPEG



Sơ đồ mã hóa và giải nén JPEG

Tất cả các block có cùng kích thước và mỗi block là một ma trận điểm ảnh 8×8 pixel được lấy từ một ảnh màn hình theo chiều từ trái sang phải, từ trên xuống dưới. Kích thước block là 8×8 được chọn bởi hai lý do sau:

+ Thứ nhất, qua việc nghiên cứu cho thấy hàm tương quan suy giảm rất nhanh khi khoảng cách giữa các pixel vượt quá 8.

+ Thứ hai, là sự tiện lợi cho việc tính toán và thiết kế phần cứng. Nói chung, độ phức tạp về tính toán sẽ tăng nếu kích thước block tăng.

Dung lượng lưu trữ ảnh cũng phụ thuộc vào loại ảnh quét. Nếu quét liên tục thì các block bao gồm các mẫu từ các dòng liên tục (lúc này nén ảnh theo-frame). Ngược lại, trong trường hợp quét cách dòng, trong một block chỉ có các mẫu của một nửa ảnh . Tóm lại, việc chia hình ảnh thành các ảnh con (block) sẽ thực sự có ý nghĩa cho bước chuyển vị tiếp theo.

**2.Phương pháp nén ảnh theo chuẩn JPEG.**

Công đoạn đầu tiên của hầu hết các quá trình nén là xác định thông tin dư thừa trong miền không gian của ảnh . Nén không gian được thực hiện bởi phép biến đổi cosin rời rạc DCT (Discrete Cosine Transform). DCT biến đổi dữ liệu trong miền không gian thành dữ liệu trong dạng tần số. Mục đích của quá trình biến đổi là thay đổi dữ liệu biểu diễn thông tin: dữ liệu của ảnh con tập trung vào một phần nhỏ các hệ số hàm truyền. Việc mã hóa và truyền chỉ thực hiện đối với các hệ số năng lượng này, và có thể cho kết quả tốt khi tạo lại tín hiệu video có chất lượng cao. DCT đã trở thành tiêu chuẩn quốc tế cho các hệ thống mã chuyển vị bởi nó có đặc tính gói năng lượng tốt, cho kết quả là số thực và có các thuật toán nhanh để thể hiện chúng

Quá trình mã hóa gồm nhiều công đoạn được mô tả trong hình dưới đây:

|  |
| --- |
| P  Mã hóa  Lượng  Tử hóa  DCT  8X8  H  ảnh nén  Â  8X8  N  K  8X8  H  Ố  Bảng Mã hóa  Bảng lượng tử hóa  8X8  I |

Quá trình giải nén là quá trình ngược lại:

Lượng

Tử hóa

Ngược

DCT

Ngược

Giải

mã

Ảnh ban đầu

ảnh nén

Bảng Mã hóa

Bảng lượng tử hóa

Quá trình mã hóa,giải mã và lượng tử hóa dựa vào các thông tin quan trọng trong phần Header của file ảnh.

2.1.Quá trình phân khối.

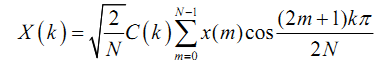
Chuẩn JPEG phân ảnh thành các khối 8x8 để giảm thời gian tính toán cũng như làm tăng độ chính xác khi tính toán.

Do các điểm ảnh lân cận có độ tương quan cao,do đó phép biến đổi DCT cho từng khối nhỏ sẽ tập trung năng lượng vào một số ít các hệ số biến đổi.Việc loại bớt một số hệ số năng lượng thấp trong các khối chỉ tạo ra mất mát thông tin cục bộ giúp nâng cao chất lượng ảnh.

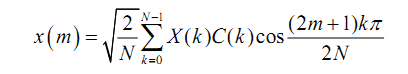
2.2.Biến đổi DCT.

2.2.1DCT một chiều

DCT một chiều biến đổi biên độ tín hiệu tại các điểm rời rạc theo thời gian hoặc không gian thành chuỗi các hệ số rời rạc, mỗi hệ số biểu diễn biên độ của một thành phần tần số nhất định có trong tín hiệu gốc. Hệ số đầu tiên biểu diễn mức DC trung bình của tín hiệu. Từ trái sang phải, các hệ số thể hiện các thành phần tần số không gian cao hơn của tín hiệu và được gọi là các hệ số AC. Thông thường, nhiều hệ số AC có giá trị sẽ gần hoặc bằng 0. Quá trình biến đổi DCT thuận (FDCT) dùng trong tiêu chuẩn JPEG được định nghĩa như sau:



Hàm biến đổi DCT ngược (một chiều):



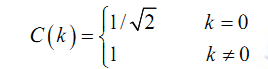
trong đó: X(k) là chuỗi kết quả.

x(m) là giá trị của mẫu m.

k-chỉ số của hệ số khai triển.

m-chỉ số của mẫu.

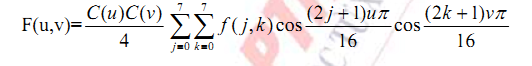
N- số mẫu có trong tín hiệu



2.2.2 DCT hai chiều

Để tách tương quan nội dung ảnh cao hơn, mã hóa DCT hai chiều (2-D) được dùng cho các khối 8×8 giá trị các điểm chói. Quá trình biến đổi DCT tiến FDCT (forward DCT) dùng trong tiêu chuẩn JPEG được định nghĩa như sau:

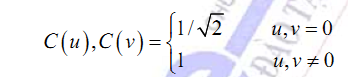
Biến đổi DCT là một trong những công đoạn quan trọng trong JPEG.Nhiệm vụ của nó là tập trung năng lượng vào một số các giá trị để giải tương quan tốt nhất nhằm nâng cao tỉ số nén.

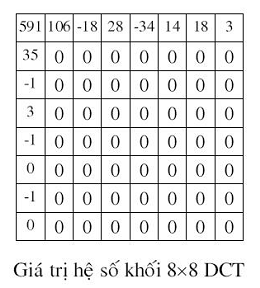
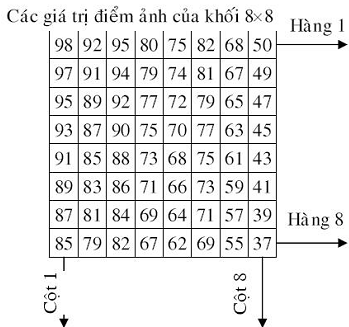


trong đó:

f(j,k)- các mẫu gốc trong khối 8×8 pixel.

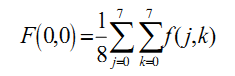
F(u,v)-các hệ số của khối DCT 8×8.





Hình :Biểu diễn ma trận trước và sau khi biến đổi DCT

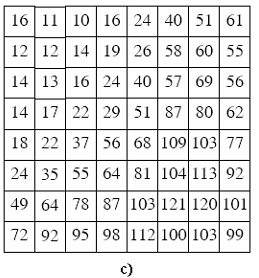
Phương trình trên là một liên kết của hai phương trình DCT một chiều, một cho tần số ngang và một cho tần số đứng. Giá trị trung bình của block 8x8 chính là hệ số thứ nhất (khi u,v= 0)



Phương trình này cộng tất cả các giá trị pixel trong khối 8×8 và chia kết quả cho 8. Kết quả phép tính bằng 8 lần giá trị pixel trung bình trong khối. Do đó hệ số thứ nhất được gọi là hệ số DC. Các hệ số khác, dưới giá trị thành phần một chiều, biểu diễn các tần số cao hơn theo chiều dọc. Các hệ số ở về phía bên phải của thành phần một chiều biểu thị các tần số cao hơn theo chiều ngang. Hệ số trên cùng ở cận phải (0,7) sẽ đặc trưng cho tín hiệu có tần số cao nhất theo phương nằm ngang của ma trận 8×8, và hệ số hàng cuối bên trái (7,0) sẽ đặc trưng cho tín hiệu có tần số cao nhất theo phương thẳng đứng. Còn các hệ số khác ứng với những phối hợp khác nhau của các tần số theo chiều dọc và chiều ngang.

Nếu dùng quá trình DCT cho các tín hiệu số thành phần Y, Cr,Cb thì các tín hiệu Cr,Cb có biên độ cực đại ±128 ( giá trị nhị phân trong hệ thống lấy mẫu 8 bit), còn tín hiệu Y có một khoảng cực đại từ 0 đến 255 giá trị nhị phân. Để đơn giản việc thiết kế bộ mã hóa DCT, tín hiệu Y được dịch mức xuống dưới bằng cách trừ 128 từ từng giá trị pixel trong khối để có khoảng cực đại của tín hiệu giống như đối với các tín hiệu CR và CB. Ở phần giải mã DCT, giá trị này (128) được cộng vào các giá trị pixel chói. Giá trị hệ số DC của khối DCT có một khoảng từ –1024 đến 1016. Đối với hệ số AC ( với u,v=1,2,...,7), C(u) và C(v)=1 và các giá trị cực đại của nó nằm trong khoảng ±1020 theo phương trình FDCT. Khối 8×8 các giá trị của hệ số DCT đưa ra 1 giá trị DC lớn (ví dụ =591), biểu diễn độ sáng trung bình của khối 8×8 và các giá trị rất nhỏ của các thành phần tần số cao theo chiều ngang và chiều đứng.

Để loại bỏ một số giá trị không cần thiết thì chúng ta cần chia với bảng trọng số



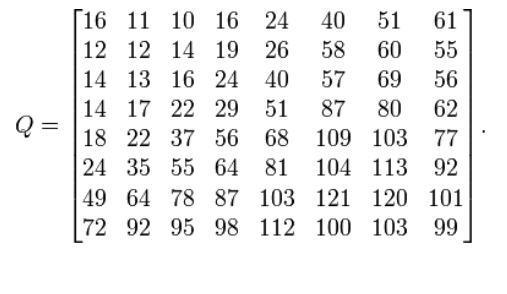
Hình :Bảng trọng số Q(u,v)

Tóm lại, DCT làm giảm độ tương quan không gian của thông tin trong block. Điều đó cho phép biễu diễn thích hợp ở miền DCT do các hệ số DCT có xu hướng có phần dư thừa ít hơn. Điều này có nghĩa là DCT gói một phần lớn năng lượng tín hiệu vào các thành phần biến đổi có tần số tương đối thấp để lưu trữ hoặc truyền dẫn, tạo 0 và các giá trị rất thấp đối với thành phần tần số cao. Nhờ đặc tính của hệ thống nhìn của mắt người, các hệ số DCT có thể được mã hóa phù hợp, chỉ các hệ số DCT quan trọng nhất mới được mã hóa và truyền đi. DCT thuận kết hợp với DCT nghịch sẽ không cho tổn thất nếu độ dài từ mã của hệ số là 13 đến 14 bits cho tín hiệu video đầu vào được số hóa bằng các mẫu dài 8 bit. Nếu hệ số được lượng tử hóa bằng 11 bit (hoặc ngắn hơn), thì nén bằng DCT sẽ có tổn hao.Sau khi thực hiên DCT năng lượng tập trung chủ yếu ở miền tần số thấp.

2.3.Lượng tử hóa.

Bước tiếp theo của quá trình nén trong ảnh là lượng tử hóa các hệ số F(u,v) sao cho làm giảm được số lượng bit cần thiết. Các hệ số tương ứng với tần số thấp có các giá trị lớn hơn, và như vậy nó chứa phần năng lượng chính của tín hiệu, do đó phải lượng tử hóa với độ chính xác cao. Riêng hệ số một chiều đòi hỏi độ chính xác cao nhất, bởi lẽ nó biểu thị giá trị độ chói trung bình của từng khối phần tử ảnh. Hệ thống thị giác của chúng ta quan sát rất tốt với những sự thay đổi nhỏ về độ sáng trong một miền tương đối rộng (Thành phần tần số thấp) nhưng ngược lại đối với các thành phần tần số cao.Chính điều này giúp ta giảm các thông tin đối với các thành phần tần số cao.Điều này được thực hiện bàng cách chia các tần số cho các hằng số cố định và làm tròn tới giá trị số nguyên gần nhất.Sau quá trình này giá trị các thành phần tần số cao được làm tròn tới giá trị 0 trong khi đó các thành phần tần thấp cũng nhỏ đi chính vì thế làm giảm số bit cho mỗi giá trị của các thành phần tần số thấp. Bất kỳ một sai sót nào trong quá trình lượng tử hệ số một chiều đều có khả năng nhận biết dễ dàng bởi nó làm thay đổi mức độ chói trung bình của khối. Ngược lại, với các hệ số tương ứng với tần số cao và có các giá trị nhỏ, thì có thể biểu diễn lại bằng tập giá trị nhỏ hơn hẳn các giá trị cho phép.

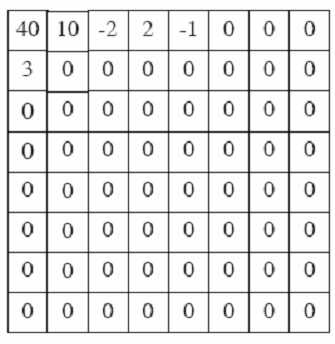
Sau đây là bảng hàng số chuẩn cho quá trình lượng tử hóa:



Tại mỗi vị trí của X người ta chia giá trị tại đó cho các hằng số tại vị trí tương ứng sau đó làm tròn với số nguyên gần nhất.

B(u,v) =

B(0,0)=== 40

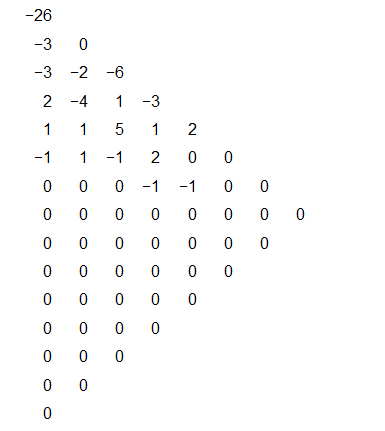
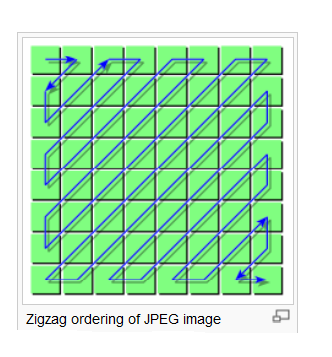


Hình :Kết quả của ma trận sau khi nhân với Q(u,v)

Cần phải xác định là trong quá trình lượng tử hóa có trọng số có xảy ra mất thông tin, gây tổn hao. Đây là bước tổn hao duy nhất trong thuật toán nén. Mức độ tổn hao phụ thuộc vào giá trị các hệ số trên bảng lượng tử. Các thành phần DC và tần số thấp là các thông số nhạy cảm nhất của khối pixel gốc. Hệ số DC sẽ được lượng tử với độ chính xác 12 bit nhằm tránh các nhiễu xuất hiện giữa các khối điểm ảnh. Ngược lại, các hệ số tần số cao có thể lượng tử hóa thô với độ chính xác 2 bit-do khả năng cảm nhận của mắt người giảm ở tần số cao. Theo đó, hệ số chia trong bảng lượng tử hóa là nhỏ đối với các hệ số có tần số thấp và tăng từ từ đối với các hệ số có tần số cao hơn.Việc biến đổi sao cho chất lượng hình ảnh do mắt người cảm nhận tốt, phụ thuộc vào các thành phần tần số và sự biến đổi chi tiết ảnh từng vùng trong miền không gian. Các ảnh càng chi tiết thì hệ số thành phần tần số cao càng lớn.

2.4. Quá trình mã hóa hay nén.

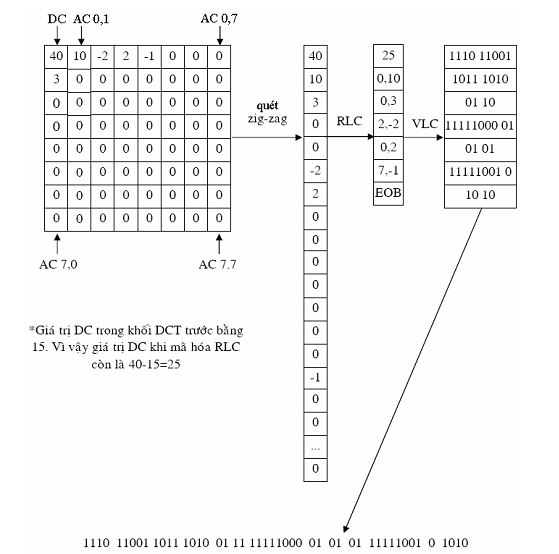
Để mã hóa entropy các hệ số được lượng tử hóa Fq(u,v), trước hết, cần biến đổi mảng hai chiều của các hệ số Fq(u,v) thành chuỗi số một chiều bằng cách quét zig-zag. Để nâng cao hiệu quả nén cho mỗi bộ hệ số người ta thường xếp chúng lại theo thứ tự Zigzac.Tác dụng sắp xếp lại theo thứ tự Zigzac là tạo ra nhiều loạt hệ số giống nhau.Ta biết rằng năng lượng của khối hệ số sẽ giảm dần từ góc trên bên trái xuống góc dưới bên phải vì vậy việc lấy Zigzac sẽ tạo điều kiện cho các hệ số sấp xỉ nhau.

 Việc xử lý 64 hệ số của khối 8x8 pixel bằng cách quét zig-zag làm tăng tối đa chuỗi các giá trị 0 và do vậy làm tăng hiệu quả nén khi dùng RLC.

2.4.1. Mã hóa độ dài chạy (RLC)

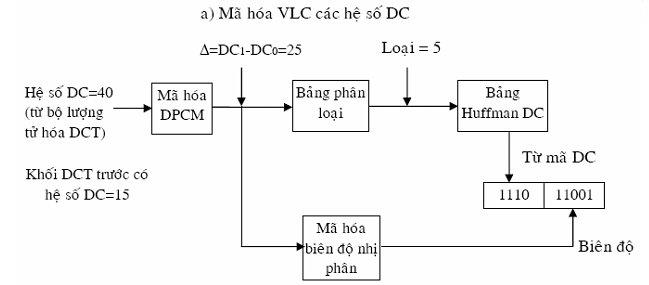
Các giá trị lượng tử hóa có thể chỉ biểu diễn nhờ các từ mã có độ dài cố định hay đồng đều, tức là các giá trị lượng tử hóa biễu diễn bằng cùng một số bit. Tuy nhiên hiệu quả của việc mã hóa không cao. Để cải tiến hiệu quả người ta dùng mã hóa entropy. Mã hóa entropy dùng những đặc tính thống kê của tín hiệu được mã hóa. Một tín hiệu, ở đây là giá trị pixel hoặc các hệ số chuyển vị, có chứa một lượng thông tin (entropy) tùy theo những xác suất của những giá trị hay sự kiện khác nhau xuất hiện. Ví dụ những từ mã nào ít xảy ra hơn sẽ có nhiều thông tin hơn từ mã hay xảy ra. Khi dùng mã hóa entropy có hai vấn đề đặt ra: thứ nhất, mã hóa entropy làm tăng độ phức tạp và yêu cầu bộ nhớ lớn hơn so với mã độ dài cố định. Kỹ thuật RLC được dùng để mã hóa có hiệu quả các hệ số DCT đã lượng tử hóa hơn là dùng trực tiếp cho số liệu ảnh. Sau quá trình quét zig-zag ở trên, RLC sẽ được thực thi. Một hệ số khác 0 sau giá trị DC được mã hóa bằng 1 từ mã bao gồm 2 thông số: số lượng 0 chạy trước 1 hệ số riêng khác 0 và mức của nó sau khi lượng tử hóa. RLC thực chất là việc thay thế các hệ số có giá trị 0 bằng số lượng các chữ số 0 xuất hiện.

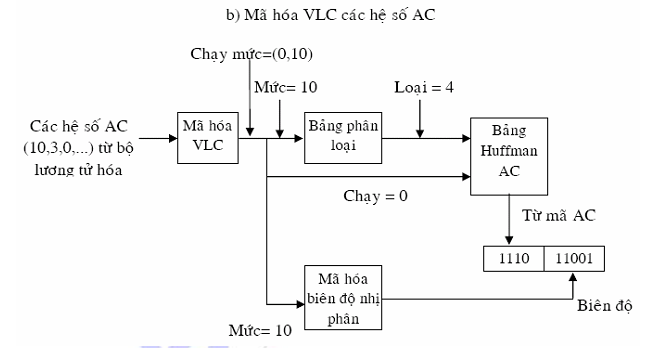
Hình dưới biểu diễn ví dụ về RLC : Ở đây, giá trị 10 không có giá trị 0 nào trước đó được biễu diễn bằng <0,10>; giá trị –2 có hai giá trị 0 đứng trước được biễu diễn bằng <2,-2>v.v... Riêng một dấu đặc biệt là End of Block (EOB) được dùng để cho biết tất cả các hệ số tiếp theo trong khối bằng 0. Trong ví dụ này, ta có một chuỗi 49 từ mã với giá trị 0. Như vậy chỉ xét riêng 49 từ mã giá trị 0 được nén xuống chỉ còn 3 từ mã. Điều này chứng tỏ hiệu suất nén rất cao của mã hóa RLC. Nén bằng mã RLC là quá trình nén không tổn hao.



2.4.2. Mã hóa độ dài thay đổi VLC

Các từ mã RLC tiếp tục được mã hóa bằng cách đặt các từ mã ngắn cho các mức có xác suất xuất hiện cao và các từ mã dài cho các mức có xác suất xuất hiện thấp. Tại đầu ra VLC, tất cả các từ mã của cùng một khối DCT được kết hợp tạo thành một dòng tín hiệu ra. Trong ví dụ trên, số liệu tương ứng với khối DCT ban đầu (8x8x8 bit =512 bit) được giảm thành 48 bits sau khi mã hóa VLC. Hệ số nén trong trường hợp này bằng 512/48=10,6. Hệ số nén cũng thường được tính bằng số bit biễu diễn điểm ảnh. Trong ví dụ trên, 48 bit biểu diễn cho 64 điểm ảnh, theo đó thu được hệ số nén tương ứng là 48/64=0,75 (bit/điểm ảnh). Mã hóa VLC tự nó là một kỹ thuật mã hóa không tổn thất, nó cho phép giảm thêm tốc độ dòng bit (đã được giải tương quan, làm tròn, và giảm qua quá trình lượng tử hóa. Tại đầu ra VLC, tất cả các từ mã của cùng một khối DCT được kết hợp tạo thành một dòng tín hiệu ra. Trong ví dụ trên, số liệu tương ứng với khối DCT ban đầu (8x8x8 bit =512 được giảm thành 48 bits sau khi mã hóa VLC.Rõ ràng chúng ta thấy được rằng tỉ lệ nén là rất cao. Mã hóa VLC tự nó là một kỹ thuật mã hóa không tổn thất, nó cho phép giảm thêm tốc độ dòng bit (đã được giải tương quan, làm tròn, và giảm qua quá trình lượng tử hóa DCT). Quá trình mã hóa VLC cho hệ số DC và các hệ số AC được mổ tả trong sơ đồ khối sau :



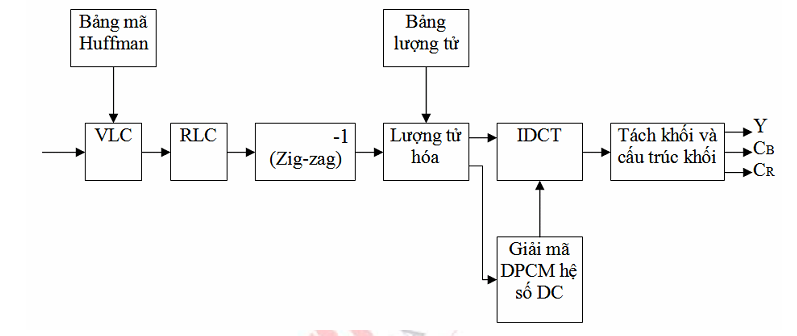


2.5.Quá trình giải nén

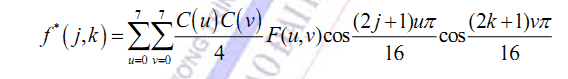
Quá trình giải nén trong ảnh dựa trên cơ sở thực hiện thuật toán ngược với quá trình nén.

Các bảng Huffman và lượng tử hóa giống như các bảng của bộ mã hóa DCT được dùng để tạo lại các giá trị hệ số DCT của một khối 8x8 pixel. Quá trình lượng tử hóa ngược R(u,v) được tiến hành theo biểu thức :

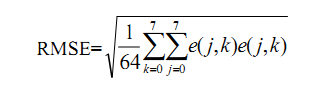
R(u,v)=Fq(u,v)Q(u,v)



Quá trình biến đổi DCT ngược (IDCT) tạo lại khối giá trị các điểm ban đầu theo biểu thức:

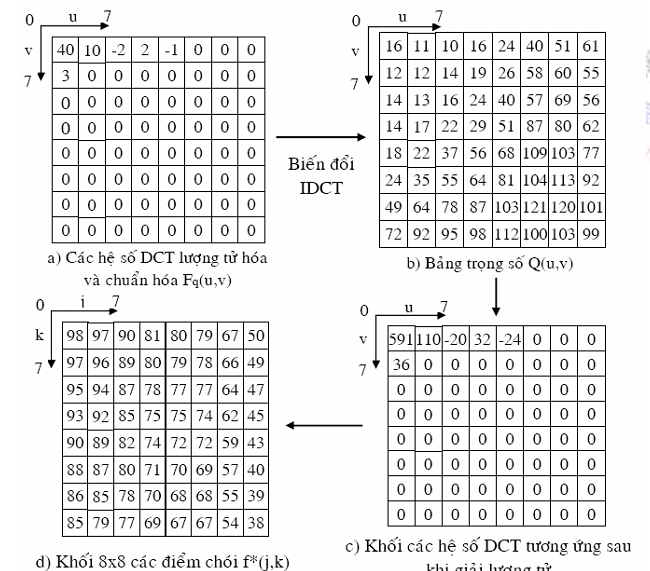


Để đánh giá chất lượng ảnh khôi phục, ta sử dụng các đại lượng đo là giá trị sai số trung bình bình phương (RMSE) và hệ số biên độ đỉnh tín hiệu trên nhiễu (PSNR: peak signal-to-noise ration):

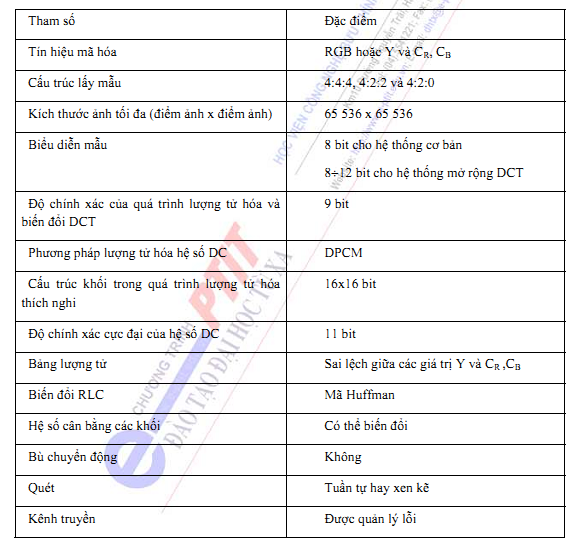




Và đây ví dụ biểu diễn quá trình phục hồi lại cá điểm ảnh



Bảng dưới đây trình bày các tham số tiêu chuẩn của JPEG



3.Vai trò của phương pháp biến đổi DCT

DCT (Discrete Cosine Transform) là phép biến đổi Cosin rời rạc để chuyển tín hiệu từ miền thời gian hay không gian sang miền tần số . Đặc điểm của phép biến đổi này là tín hiệu ảnh trong miền không gian chuyển sang miền tần số thì các thành phần DC và các thành phần AC mang hầu hết các thông tin chứa trong ảnh gốc. Trong đó, DC là thành phần quan trọng nhất mang độ chói trung bình của ảnh, các thành phần AC chứa các thông tin về chi tiết của ảnh. Sau đó, khi qua tầng lượng tử hoá, các hệ số ít quan trọng sẽ bị loại bỏ bớt và chỉ giữ lại một số hệ số hệ số đầu tiên gọi là hệ số DCT.Vai trỏ chủ yếu của phương pháp DCT là giảm độ dư thừa dữ liệu trong pixcel ở miền tần số cao .Bởi vì bất kì một giá trị pixcel nào đó cũng có thể dự đoán từ các giá trị pixcel lân cận của nó nên thông tin từ các pixcel riêng lẻ là tương đối nhỏ.Để giảm độ dư thừa của các pixcel đặc biệt là các pixcel ở miền tần số cao,thì phương pháp DCT là vô cùng thích hợp.Không những vậy sau khi biến đổi DCT thì hàm giải tương quan giảm đi một cách đáng kể.Chính vì vậy mà hiệu suất nén đạt được tỉ số nén cao.Điều quan trọng là khối DCT đóng vai trò quan trọng trong quá trình lượng tử hóa khi thiết kế hệ thống nén video vì nó ảnh hưởng trực tiếp đến việc cho lại chất lượng ảnh khôi phục tốt hay xấu.Có thể nói chính điều đó đã làm lên môt chuẩn JPEG được ứng dựng rộng rãi như vậy.

**ĐỊNH DẠNG ẢNH TIFF**

1. Những kiến thức cơ bản về định dạng TIFF
   1. Khái niệm

TIFF là cụm từ viết tắt của Tagged Image File Format, một chuẩn định dạng file để lưu trữ ảnh. Kể từ khi được ra đời vào khoảng những năm 80 của thế kỉ trước, TIFF là một định dạng ảnh công nghiệp được áp dụng rộng rãi trong các công nghệ in ảnh màu và quét (scan) ảnh.

Định dạng TIFF được xây dựng đầu tiên bởi tập đoàn Aldus (Aldus Corporation) mà sau này sát nhập vào với Adobe Systems vào khoảng đầu những năm 1990 sau khi gặp thất bại trên thị trường công nghệ so với các hãng khác như QuarkXPress. Từ đó cho tới nay, Adobe nắm bản quyền với toàn bộ các vấn đề liên quan tới định dạng ảnh TIFF.

Ban đầu, TIFF là định dạng ảnh chỉ cho phép lưu trữ ảnh nhị phân, tạo ra các tệp ảnh đen trắng, cụ thể, với mỗi một điểm ảnh thì chỉ có một trong hai màu hoặc trắng hoặc đen. Với sự phát triển của công nghệ sau này, TIFF đã hỗ trợ lưu trữ ảnh đa mức xám, ảnh màu và nhiều dạng ảnh màu hiện đại khác nữa.

Các qui tắc xây dựng chuẩn TIFF không hề cứng nhắc để có thể sử dụng trong nhiều mục đích và chức năng khác nhau (tương tự như định dạng AVI). Mỗi một tệp ảnh TIFF sử dụng các thẻ (header tags) để mô tả về các ứng dụng xử lí tệp tin, cung cấp các thông tin về độ lớn, dung lượng của tệp tin, mức độ màu sắc, dữ liệu ảnh được sắp xếp như thế nào, các thành phần nào trong ảnh được nén... TIFF không bị mất mát dữ liệu hay không bị giảm chất lượng ảnh trong quá trình nén tương tự như định dạng JPEG.

TIFF được sử dụng với nhiều mục đích và ứng dụng trong cuộc sống. Các tệp ảnh TIFF thường được sử dụng trong máy fax số hay các máy in ấn (tạp chí, báo họa…). Tuy nhiên, hiện này người ta thường khuyến cáo về các lỗi có thể phát sinh khi sử dụng định dạng ảnh TIFF bởi các yêu cầu lỏng lẻo về tiêu chuẩn đối với định dạng này. Trong khoa học nghiên cứu ảnh, định dạng TIFF thường được dùng trong các môi trường yêu cầu ảnh có độ chính xác và chất lượng cao.

* 1. Lịch sử ra đời và phát triển

Cho đến nay, một vài phiên bản của định dạng TIFF đã ra đời:

* Năm 1986: định dạng TIFF lần đầu tiên xuất hiện phiên bản 3.0
* Năm 1987: nâng cấp lên phiên bản 4.0
* Năm 1988: nâng cấp lên phiên bản 5.0. Kể từ phiên bản này, định dạng TIFF cho phép lưu trữ với dạng ảnh màu và áp dụng thuật toán nén TLZW để giảm thiểu dụng lượng lưu trữ cho tệp tin.
* Năm 1992: nâng cấp lên phiên bản 6.0. Đây cũng là phiên bản nâng cấp chính thức cuối cùng cho tới hiện nay.
  1. Cấu trúc TIFF

Cấu trúc một tệp ảnh TIFF bắt đầu bởi một trường có độ dài 8 bytes (Image File Header) để trỏ tới đường dẫn của tệp (IFD – Image File Directory). Giá trị của trường này lưu thông tin về ảnh như dữ liệu thực tế của ảnh... Các giá trị độ dài, độ lớn được thể hiện ở hệ thập phân. Kí tự .H theo sau chỉ định dưới dạng hệ thập lục phân.

* + 1. Image File Header

Trường dữ liệu này lưu trữ các thông tin sau về tệp tin TIFF:

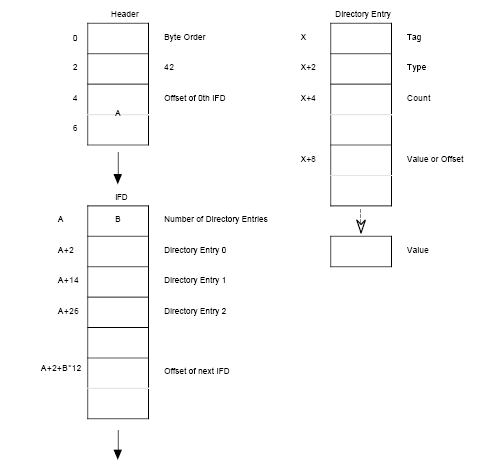
* Bytes 0-1: thứ tự sắp xếp byte trong tệp tin được chỉ ra như sau:

“II” (4949.H)

“MM” (4D4D.H)

Với định dạng “II”, thứ tự sắp xếp các byte được thực hiện theo thứ tự sắp xếp đầu nhỏ (little – endian), bao gồm cả hệ số nguyên 16 và 32 bits. Ngược lại, với định dạng “MM” thì thứ tự sắp xếp các byte được thực hiện theo thứ tự sắp xếp đầu to (big – endian).

* Bytes 2-3: thứ tự sắp xếp các bytes này phụ thuộc vào giá trị bytes 0-1, được sắp xếp tùy ý nhưng việc chọn các giá trị này phải hết sức cẩn thận để xác định một tệp tin TIFF
* Bytes 4-7: phần còn thiếu (phần bù) của giá trị IFD đầu tiên. Đường dẫn địa chỉ có thể nằm ở bất cứ đâu trong tệp tin phía sau header nhưng phải bắt đầu bằng một từ làm mốc (word boundary). Ý nghĩa của bytes offset được sử dụng để xác định nơi lưu trữ dựa vào thông tin phần đầu của tệp TIFF.



* + 1. Image File Directory

Image File Directory (IFD) chứa 2 byte để đếm số lượng đầu vào đường dẫn (như số lượng trường - fields), tiếp nối sau đó là 12 bytes cho các trường (fields) và sau nữa là 4 bytes cho IFD tiếp theo.

Có ít nhất 1 IFD trong 1 tệp TIFF và mỗi IFD lại có ít nhất đầu vào (entry).

* IFD Entry

Mỗi IFD entry có độ dài 12 bytes và gồm các thông tin sau:

Bytes 0-1: Tag định danh cho 1 trường (field)

Bytes 2-3: kiểu của trường (field Type)

Bytes 4-7: số lượng kiểu (Type)

Bytes 8-11: giá trị Offset như đã nói ở trên

* Order

Thứ tự sắp xếp các đầu vào trong IFD được sắp xếp theo thứ tự tăng dần của Tag

* Type

Kiểu của các trường dữ liệu và độ dài tương ứng:

1= BYTE: 8bit số nguyên không dấu

2= ASCII: 8bit trong đó có 7 bit đầu là mã ASCII và bit cuối cùng là giá trị NULL

3= SHORT: 16bit (2-bytes) số nguyên không dấu

4= LONG: 32bit (4-bytes) số nguyên không dấu

5= RATIONAL: gồm 2 kí tự kiểu LONG

* + 1. Multiple Image per TIFF file:

Mỗi tệp TIFF có thể chứa nhiều ảnh (nhiều directory). Mỗi IFD được coi như là một tệp con.

* 1. Phân loại ảnh TIFF
     1. Ảnh nhị phân (Bilevel Images)

1. Màu sắc

Ảnh nhị phân chỉ có 2 màu là đen và trắng.

Tag = 262 (106.H)

Type = SHORT

Values:

0 = White zero

1 = Black zero

1. Nén

Tag = 259 (103.H)

Type = SHORT

Values:

1 = không nén nhưng ép dữ liệu thành dạng bytes nhỏ nhất có thể, loại bỏ đi các phần dữ liệu không cần thiết. Mỗi một dòng dữ liệu được lưu trữ thành các mảng có kiểu dữ liệu là Byte.

2 = nén theo thuật toán CCITT

32773 = nén dạng PackBits

1. Hàng và Cột

Một tệp ảnh được tổ chức thành một mảng lưu trữ các pixel (được chia thành các hàng và cột). Thông tin liên quan tới hàng và cột:

ImageLength

Tag = 257 (101.H)

Type = SHORT or LONG

Mô tả số lượng hàng (có khi được mô tả là các dòng quét - scanlines) trong bức ảnh

ImageWidth

Tag = 256 (100.H)

Type = SHORT or LONG

Mô tả số lượng cột trong ảnh hay số pixel trên một hàng

1. Số chiều vật lí

Các ứng dụng thường tính toán ra dung lượng của các tệp ảnh thông qua các chỉ số về số lượng cột và dòng của ảnh. Một số thông tin liên quan:

ResolutionUnit

Tag = 296 (128.H)

Type = SHORT

Values:

1 = không có đơn vị đo cụ thể, dùng cho các tệp ảnh không có giá trị cụ thể về tỉ lệ hình chữ nhật

2 = Inch.

3 = Centimeter.

Mặc định = 2 (inch).

XResolution

Tag = 282 (11A.H)

Type = RATIONAL

Cho biết số lượng pixel theo độ rộng (chiều ngang) tương ứng với độ phân giải của ảnh

YResolution

Tag = 283 (11B.H)

Type = RATIONAL

Cho biết số lượng pixel theo độ dài (chiều dọc) tương ứng với độ phân giải của ảnh

1. Định vị dữ liệu

Dữ liệu của mỗi bức ảnh được nén hay không nén đều được lưu trữ toàn bộ trong tệp TIFF. TIFF cũng cho phép chia các ảnh thành các dải để thuận tiện cho việc lưu trữ, xử lí dữ liệu và các thao tác vào ra khác... Các thông tin liên quan tới việc lưu trữ và kích thước của các dải được mô tả như sau:

RowsPerStrip

Tag = 278 (116.H)

Type = SHORT or LONG

Số lượng dòng trong mỗi dải

StripOffsets

Tag = 273 (111.H)

Type = SHORT or LONG

Số byte bù trong mỗi dải.

StripByteCounts

Tag = 279 (117.H)

Type = SHORT or LONG

Số bytes biểu diễn dữ liệu cho mỗi dải sau khi được nén.

* + 1. Ảnh đa mức xám (Grayscale Images)

Ảnh đa mức xám được cải tiến từ ảnh nhị phân. Với ảnh nhị phân, ta chỉ có thể lưu lại ảnh chỉ với 2 màu đen và trắng thì với ảnh đa mức xám, ta có thể lưu ảnh mà tại mỗi điểm ảnh sẽ có nhiều mức độ xám (độ chói) khác nhau. Mặc định, ảnh đa mức xám bao gồm tất cả các thông tin có trong ảnh nhị phân, trừ một số thông tin cải tiến khác như sau:

Ảnh đa mức xám không nén hoặc nén với thuật toán PackBits (32773).

BitsPerSample

Tag = 258 (102.H)

Type = SHORT

Số bits cần để lưu trữ cho mỗi thành phần, cho phép lưu từ 16 đến 256 mức độ xám khác nhau.

* + 1. Ảnh đa màu sắc (Palette-color Images)

Tương tự ảnh đa mức xám, ảnh đa màu sắc cũng được tổ chức theo dạng các pixel (điểm ảnh). Với mỗi pixel, ảnh này sẽ lưu trữ theo các mức màu sắc khác nhau (thay vì các mức độ chói trong ảnh đa mức xám đơn thuần) theo chuẩn màu RGB. Ảnh đa màu sắc bao gồm tất cả các thông tin tương tự với ảnh đa mức xám nhưng có một số thông tin cải tiến khác như:

ColorMap

Tag = 320 (140.H)

Type = SHORT

N = 3 \* (2\*\*BitsPerSample)

Trường này khởi gán các giá trị màu sắc tương ứng trong bảng màu RGB cho mỗi một điểm ảnh. VD: nếu một pixel được lưu giá trị 0 thì tương ứng sẽ có bộ ba Red, Green, Blue số 0 trong bảng màu RGB. Theo đúng thứ tự đó, ta sẽ có các giá trị màu từ đen (0,0,0) cho tới trắng (65535, 65535, 65535).

* + 1. Ảnh màu RGB đầy đủ (RGB Full Color Images)

Với định dạng này thì tại mỗi pixel sẽ gồm 3 thành phần màu: Red, Green, Blue chứ không ánh xạ các giá trị màu thông qua bảng màu RGB ColorMap như trước. Ảnh này có đầy đủ các tính chất như ảnh đa màu sắc nhưng có một số đặc điểm khác như sau:

SamplesPerPixel

Tag = 277 (115.H)

Type = SHORT

Số lượng thành phần màu cho mỗi pixel, ở đây là 3 thành phần màu Red, Green, Blue.

* + 1. Một số phân loại ảnh TIFF cải tiến

1. Tiled Images

Với định dạng này, tệp TIFF sẽ chia nhỏ ảnh ra thành các dải nhỏ nhằm thuận tiện cho việc tổ chức dữ liệu và nén ảnh. Kích thước của các dải này được thể hiện qua chỉ số TileWidth và TileLength

1. CMYK Images

Với định dạng này, mỗi pixel trong ảnh sẽ gồm 4 thành phần, mỗi một thành phần sẽ là giá trị của từng màu cyan, magenta, yellow và black. Đó cũng là bốn thành phần màu cơ bản được sử dụng trong việc in ấn.

Mối quan hệ giữa hệ màu CMYK với hệ màu RGB như sau:

cyan = 255-red

magenta = 255-green

yellow = 255-blue

black = black

1. YCbCr Images

Định dạng này ra đời nhằm phục vụ cho việc hiển thị ảnh, tín hiệu video trên truyền hình hay các thiết bị số. Mối quan hệ giữa hệ YCbCr với hệ RGB được biểu diễn như sau:

Y = ( LumaRed \* R + LumaGreen \* G + LumaBlue \* B )

Cb = ( B - Y ) / ( 2 - 2 \* LumaBlue )

Cr = ( R - Y ) / ( 2 - 2 \* LumaRed )

Trong đó, các tham số được xác định:

LumaRed = 0.299

LumaGreen = 0.587

LumaBlue = 0.114

1. CIEL\*a\*b Images

CIEL\*a\*b là không gian màu được xây dựng dựa trên sự phân chia của độ sáng và các kênh màu, theo sự cảm nhận của con người. Sự tương quan giữa hệ màu CIEL\*a\*b và hệ màu RGB là:

0.60700.17400.2000 R X

0.29900.58700.1140 \* G = Y

0.00000.06601.1110 B Z

1. Các kĩ thuật nén ảnh TIFF
   1. PackBits
2. Ý tưởng thuật toán

Ý tưởng cơ bản của thuật giải thuật packbits là thay thế chuỗi các ký tự giống nhau liên tiếp bằng một hoặc một vài ký tự đại diện cho chuỗi đó.

Đối với các tệp ảnh TIFF, ta có thể nén và giải nén ảnh đa mức xám, ảnh màu hay ảnh nhị phân sử dụng thuật toán Packbits – một phương pháp không làm mất thông tin. Thuật toán Packbits chạy nhanh, được áp dụng rộng rã và rất thích hợp để nén với các bức ảnh rời rạc như các văn bản scan từ ngoài vào. Các chương trình ứng dụng có thể lưu trữ các bức ảnh đã được nén trong định dạng TIFF phiên bản 5.0. Packbits là thuật toán nén logic có thể sử dụng uyển chuyển trên nhiều hệ thống khác nhau.

Đây là phương pháp mã hóa theo từng byte tương ứng với type = 32773 của định dạng TIFF, thường được sử dụng như một phương pháp nén phổ biến trong dòng máy Macintosh của Apple.

Ví dụ:

* Chuỗi đầu vào: AA AA AA 80 00 2A AA AA AA AA 80 00 2A 22 AA AA AA AA AA AA AA AA AA AA
* Chuỗi sau khi nén sẽ gồm các ký tự với ý nghĩa như sau: (các ký tự in đậm là các số nguyên có dấu)

**FE** AA: (-(-2)+1) = 3 bytes cho ký tự: AA

**02** 80 00 2A: (2) + 1 = 3 bytes cho các ký tự rời rạc

**FD** AA: (-(-3)+1) = 4 bytes cho ký tự: AA

**03** 80 00 2A 22: (3) + 1 = 4 bytes các ký tự rời rạc.

**F7** AA: (-(-9)+1) = 10 bytes cho ký tự: AA

* Chuỗi ký tự sau khi được mã hóa là: FE AA 02 80 00 2A FD AA 03 80 00 2A 22 F7 AA

Giải nén dữ liệu:

Với thuật toán nén dữ liệu Packbits thì không có cách nào để xác định điểm kết thức của dòng dữ liệu xử lí. Chính vì thế, phải luốn xác định trước kích thước của dữ liệu trước khi được nén (không nén) để giúp cho việc mã hóa Packbits xác định được điểm kết thúc của dòng dữ liệu nén.

Sơ đồ giả ngôn ngữ của thuật toán giải nén Packbits:

*Vòng lặp thực hiện cho đến khi duyệt đến cuối chuỗi dữ liệu :*

*Đọc các byte kế tiếp đến n*

*Nếu n nằm trong khoảng 0 tới 127, ta sẽ copy n+1 bytes tiếp theo*

*Nếu n nằm trong khoảng -127 tới -1, ta sẽ copy –n+1 byte kế tiếp*

*Ngược lại, nếu n = -128, noop*

*Kết thúc vòng lặp*

1. Đặc điểm của thuật toán

Các đặc điểm nổi trội

* Đơn giản về mặt thuật toán mã hóa và giải mã
* Tốc độ tính toán nhanh.
* Tỉ số nén phụ thuộc vào độ dài chuỗi các ký tự liên tiếp giống nhau.

Một số luật ràng buộc trong quá trình nén:

* Nén riêng rẽ từng dòng; không nén vượt quá đường biên của dòng
* Số lượng bytes không được nén ở mỗi dòng được xác định bởi (ImageWidth + 7)/8. Nếu ảnh không được nén yêu cầu một số chẵn bytes một dòng thì sẽ giải nén theo bộ đệm từ (word-aligned buffers)
* Nếu một thao tác chạy lớn hơn 128 bytes thì quá trình mã hóa những phần còn lại của thao tác đó sẽ được thực hiện tương tự.
  1. Modified Huffman

Đây là phương pháp nén nhị phân nằm trong giải thuật CCITT Group 3 1D.

1. Ý tưởng thuật toán:

Giống như Packbits, phương pháp nén Modified Huffman sẽ thay thế chuỗi các ký tự giống nhau liên tiếp bằng một hoặc một vài ký tự đại diện cho chuỗi đó.

Chuỗi các ký tự giống nhau liên tiếp có độ dài khác nhau sẽ được thay thế bởi một chuỗi được ánh xạ trong bảng mã hóa theo độ dài. Các chuỗi này được chia ra làm hai loại:

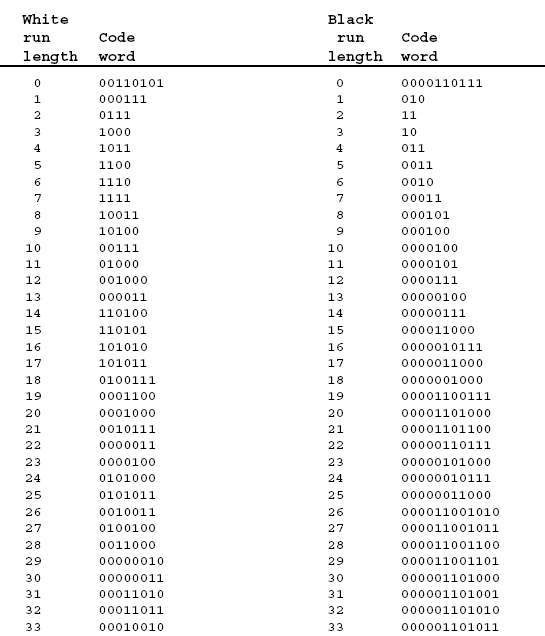
* chuỗi các bit 1: black bit
* chuỗi các bit 0: white bit

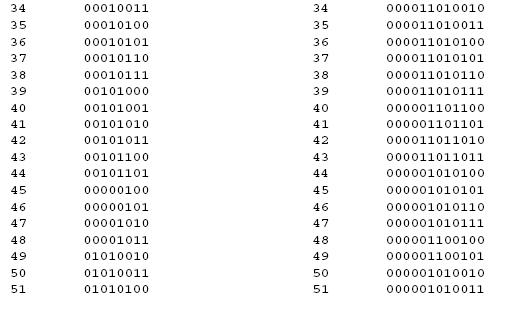
Nếu độ dài chuỗi từ 0-63 bit (pixel) được mã hóa theo bảng mã hóa **Terminating.**

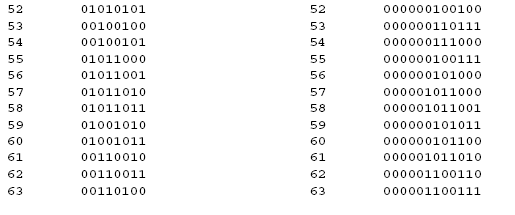
Nếu độ dài dải từ 64-2623 (2560+63) bit được mã hóa theo bảng mã hóa **Make-up.**

Nếu độ dài dải lơn hơn hoặc bằng 2624 bit thì được mã hóa trước tiên theo bảng mã hóa Make-up 2560, còn lại, nếu các phần nằm ngoài dải đó thì sẽ được mã hóa theo bảng mã hóa **additional Make-up.**

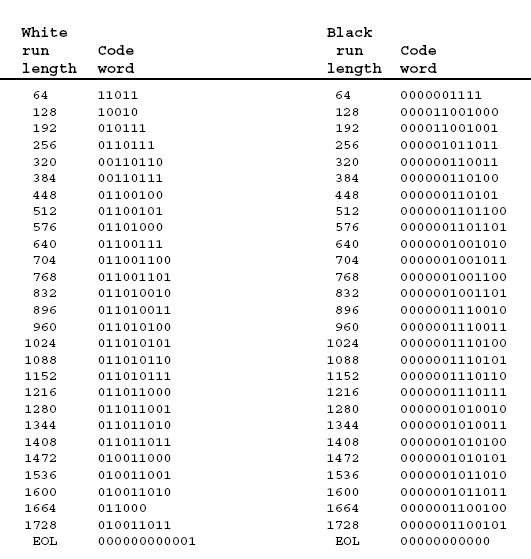
*Bảng từ mã Terminating code*



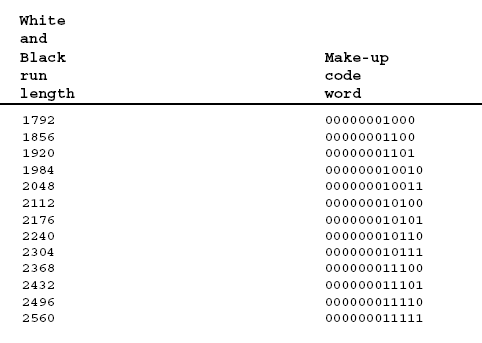




*Bảng từ mã Make-up code*



*Bảng từ mã Additional Make-up code*



1. Đặc điểm thuật toán:

* Phương pháp này có thể nén được ở tỉ số cao (khoảng 10 lần so với tệp gốc).
* Phương pháp này cho phép tùy chỉnh các tham số truyền vào cho quá trình cải tiến mã hóa.
* Thuật toán này có hiệu quả với các ảnh đa mức xám và ảnh màu tương tự như với ảnh nhị phân.
* Các tệp sau khi được mã hóa với thuật toán này rất linh động, có thể sử dụng hiệu quả ở nhiều môi trường khác nhau.
  1. Modified READ Coding (CCITT Group 3 2D)

1. Ý tưởng thuật toán

Thuật toán này còn có tên khác là CCITT Bilevel. Đây là cải tiến của phương pháp Modified Huffman (CCITT Group 3 1D)

Thay vì việc mã hóa từng dòng không phụ thuộc lẫn nhau như Modified Huffman, phương pháp Modified READ coding sẽ mã hóa sự sai khác trong một dòng nhưng dựa vào dòng đã mã hóa trước đó. (dòng tham chiếu – reference line).

Phương pháp này tốt hơn Modified Huffman với các ảnh có sự tương quan giữa hai chiều lớn ( như chữ viết tay).

Đối với định dạng TIFF ta có thể áp dụng thuật toán nén và giải nén CCITT Group 3 2D và CCITT Group 4 2D đối với ảnh nhị phân. FAX là tên phổ thông được dùng cho chuẩn nén CCITT Group 3 và Group 4, áp dụng cho các tệp văn bản được truyền qua các thiết bị thoại như FAX; tuy nhiên, hầu hết các máy FAX chỉ hỗ trợ chuẩn định dạng Group 3. FAX là phương thức nén không mất mát dữ liệu.

Các ứng dụng có thể cho phép nén ảnh nhị phân thành dạng file FAX, lưu trữ các ảnh FAX dạng tệp TIFF 5.0 và giải nén ảnh để hiển thị các ảnh FAX ngay trên màn hình.

CCITT cung cấp phương pháp nén hiệu quả cho các tệp tin được scan trực tiếp từ văn bản bên ngoài.

Ý tưởng thực hiện của thuật toán như sau:

+ Bước 1: nén dòng đầu tiên bằng phương pháp mã hóa Modified Huffman và coi đây là dòng tham chiếu (reference line).

+ Bước 2: Chọn dòng kết tiếp để mã hóa (coding line). Chọn các cặp cùng màu giữa hai dòng. Mã hóa sự lệch nhau giữa các cặp này theo một trong các chứ độ “vertical mode”, “pass mode” hoặc “horizontal mode”

* Vertical Mode: Nếu khoảng cách của cặp nhỏ hơn hoặc bằng 3 bit (pixel).
  + Nếu cách nhau 0 pixel: 1
  + Nếu cách nhau 1 pixel: 011
  + Nếu cách nhau 2 pixel: 00 0011
  + Nếu cách nhau 3 pixel: 000 0011
* Pass Mode: Nếu khoảng cách của cặp lớn hơn 3 bit và dòng tham chiếu thay đổi trước. Được mã hóa bằng chuỗi 0001.
* Horizontal Mode: Nếu khoảng cách của cặp lớn hơn 3 bit và dòng đang mã hóa thay đổi trước.

+ Bước 3: Chọn dòng vừa mã hóa là dòng tham chiếu (reference line)

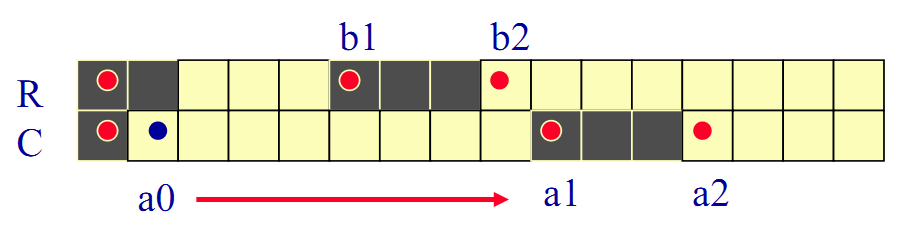
+ Bước 4: lặp lại các bước 2 và 3 cho tới khi K dòng được mã hóa

+ Bước 5: sau khi đã mã hóa được K dòng (tính cả dòng mã hóa theo phương pháp Modified Huffman), dòng K+1 sẽ được mã hóa bằng phương pháp Modified Huffman và chọn dòng K+1 làm dòng tham chiếu. Việc làm này sẽ hạn chế việc các dòng bị nhieeunx làm ảnh hưởng tới việc mã hóa các dòng tiếp theo dẫn tới sự sai lệch thông tin của toàn bộ bức ảnh.

+ Bước 6: Tiếp tục lặp lại các bước 2,3,4,5 cho tới khi hoàn thành toàn bộ file ảnh.

1. Ví dụ

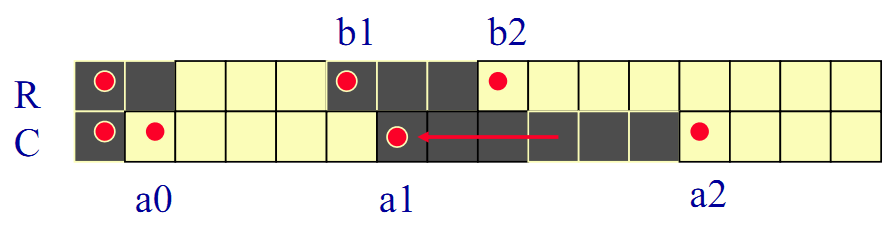
* Mã hóa theo “Pass mode”



* + Dòng R là dòng tham chiếu đã được mã hóa bằng Modified Huffman (reference line)
  + Dòng C là dòng đang mã hóa. (coding line)
  + Sự thay đổi ở dòng đang mã hóa đầu tiên được đánh dấu bởi a0.
  + Các cặp cùng màu là: (b1,a1), (b2,a2).
  + Vì (b1,a1) cách nhau 4 bit và sự thay đổi bắt đầu ở dòng tham chiếu nên a1 sẽ được mã hóa theo kiểu "pass mode".

Tương tự như vậy với (b2,a2) – cũng sẽ được mã hóa theo kiểu "pass mode".

* Mã hóa theo “Vertical mode”



* + Tương tự như trên các cặp cùng màu là: (b1, a1), (b2,a2)
  + Cặp (b1,a1) cách nhau 1 bit, vì vậy a1 sẽ được mã hóa theo kiểu "vertical mode".
  + Cặp (b2, a2) vẫn sẽ được mã hóa theo kiểu "pass mode" giống như ví dụ trên.

1. Đặc điểm của thuật toán

* Phương pháp cho tỉ số nén khá tốt với ảnh nhị phân, thích hợp cho việc truyền thông (gửi FAX).
* Tốc độ tính toán nhanh và dung lượng ảnh đã được mã hóa nhỏ hơn so với phương pháp Modified Huffman.
  1. LZW

1. Ý tưởng thuật toán:

Thuật toán LZW (được phát minh bởi Lempel - Zip and Welch) được áp dụng để nén ảnh trong tệp TIFF phiên bản 5.0 không làm mất mát dữ liệu, với tỉ số nén trung bình 2:1 và thường được áp dụng trong nén các ảnh đồ họa tượng trưng.

Thuật toán LZW sử dụng phép thay thế các ký tự trong chuỗi đầu vào dựa trên bảng giải mã hóa (giải mã) có sẵn.

Bảng giải mã hóa (giải mã) này không cần được lưu kèm với dữ liệu trong quá trình nén. Người mã hóa và người giải mã sẽ tự xây dựng bảng giải mã.

Bảng mã hóa sẽ được xây dựng giữa vào việc tách chuỗi cần mã hóa hóa thành các chuỗi con theo thứ tự từ điển.

* Quá trình nén dữ liệu

+ Bước 1: Xây dựng từ điển từ chuỗi đầu vào, đánh số cho các ký tự trong từ điển: chia nhỏ chuỗi cần mã hóa thành nhiều chuỗi con theo thứ tự từ điển của chuỗi con.

+ Bước 2: Đánh số thứ tự i cho các chuỗi con.

+ Bước 3: Mã hóa từng chuỗi con theo phép thay thế. Mỗi chuỗi con sẽ được thay thế bởi: Phần con trỏ, trỏ đến số thứ tự của chuỗi tiền tố: số thứ tự ở dạng nhị phân của chuỗi tiền tố. Phần bit sai khác với chuỗi tiền tố.

+ Bước 4: Ghép các phần đã mã hóa của chuỗi con theo thứ tự ban đầu

Giả ngôn ngữ mô phỏng cho quá trình nén với giải thuật LZW:

*InitializeStringTable();*

*WriteCode(ClearCode);*

*W = the empty string;*

*for each character in the strip {*

*K = GetNextCharacter();*

*if W+K is in the string table {*

*W = W+K; /\* string concatenation \*/*

*}*

*else {*

*WriteCode (CodeFromString(W));*

*AddTableEntry(W+K);*

*W = K;*

*}*

*} /\* end of for loop \*/*

*WriteCode (CodeFromString(W));*

*WriteCode (EndOfInformation);*

* Quá trình giải nén dữ liệu

+ Bước 1: Dựa vào chuỗi đã được mã hóa, xây dựng lại từ điển.

+ Bước 2: Giải mã chuỗi đã mã hóa dựa vào từ điển tự xây dựng lại.

Mã thực thi mô phỏng cho quá trình giải nén với giải thuật LZW:

*while ((Code = GetNextCode()) != EoiCode) {*

*if (Code == ClearCode) {*

*InitializeTable();*

*Code = GetNextCode();*

*if (Code == EoiCode) break;*

*WriteString(StringFromCode(Code));*

*OldCode = Code;*

*} /\* end of ClearCode case \*/*

*else {*

*if (IsInTable(Code)) {*

*WriteString(StringFromCode(Code));*

*AddStringToTable(StringFromCode(OldCode*

*)+FirstChar(StringFromCode(Code)));*

*OldCode = Code;*

*} else {*

*OutString = StringFromCode(OldCode) +*

*FirstChar(StringFromCode(OldCode));*

*WriteString(OutString);*

*AddStringToTable(OutString);*

*OldCode = Code;*

*}*

*} /\* end of not-ClearCode case \*/*

*} /\* end of while loop \*/*

1. Đặc điểm thuật toán

Thuật toán này được phát minh bởi Lempel-Ziv và Welch với một số đặc trưng sau:

* LZW làm việc với các bức ảnh có giá trị bit biểu diễn khác nhau, đa dạng
* LZW có chi phí tính toán tồi nhất ở mức chấp nhận được
* LZW quản lí tốt đối với các thành phần lặp lại ở mức độ rộng
* LZW thực hiện nhanh cả với quá trình nén và giải nén
* LZW là phương thức mã hóa không mất mát dữ liệu. Tuy nhiên, nếu ta loại bộ nhiễu hoặc các thông tin dư thừa trong ảnh thì LZW nén ảnh xuống được với kích thước nhỏ hơn.
* Không cần phải lưu trữ từ điển, bên giải mã có thể tự xây dựng lại trong quá trình giải mã
* Số lượng từ trong từ điển tăng lên không giới hạn
* Các chuỗi con có độ dài lớn sẽ xuất hiện trễ quá trình xây dựng từ điển

LZW hoạt động tốt cả với các ảnh nhị phân. Giải thuật này khi đem ra so sánh thì vượt trội hơn giải thuật Packbits và khá ngang bằng với giải thuật CCITT 1D.

1. Ví dụ mô phỏng

* Chuỗi cần mã hóa: 1 0110 1010 0010
* Mã hóa:

+ Xây dựng các chuỗi con theo thứ tự từ điển. Các chuỗi con lần lượt là: 1, 0, 11, 01, 010, 00, 10

+ Xây dựng bảng mã hóa

* Trong thuật toán LZ78, chuỗi con thứ n, ta sẽ cần (log2n) bits cho phần con trỏ
* Phần tử đầu tiên của từ điển có số thự tự 0, và là phần tử rỗng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Số thứ tự | Số thự tự ở dạng nhị phân | Số bit cần dùng cho con trỏ | Chuỗi con | Chuỗi mã hóa  (con trỏ, bit sai khác) |
| 0 | 000 | 0 | Ø |  |
| 1 | 001 | 0 | 1 | Ø,1 |
| 2 | 010 | 1 | 0 | 0,0 |
| 3 | 011 | 2 | 11 | 01,1 |
| 4 | 100 | 2 | 01 | 10,1 |
| 5 | 101 | 3 | 010 | 100,0 |
| 6 | 110 | 3 | 00 | 010,0 |
| 7 | 111 | 3 | 10 | 001,0 |

* Chuỗi mã hóa là: 1 0001 1101 1000 0100 0010
* Giải mã :

Xây dựng lại từ điển : tách chuỗi mã hóa thành các chuỗi con gồm 2 thành phần :

+ Phần con trỏ đến số thứ tự chuỗi tiền tố: (log2n) bits

+ Phần bit khác với chuỗi tiền tố

|  |  |  |  |
| --- | --- | --- | --- |
| Số thứ tự | Số bit cần dùng cho con trỏ | Các chuỗi con  (con trỏ, bit sai khác) | Chuỗi con |
| 1 | 0 | ,1 | 1 |
| 2 | 1 | 0,0 | 0 |
| 3 | 1 | 01,1 | 11 |
| 4 | 2 | 10,1 | 01 |
| 5 | 3 | 100,0 | 010 |
| 6 | 3 | 010,0 | 00 |
| 7 | 3 | 001,0 | 10 |

* Chuỗi ban đầu là : 1 0110 1010 0010

1. So sánh TIFF và JPEG
   1. Xác định tiêu chí so sánh

Xét 2 bức ảnh định dạng TIFF và JPEG có cùng độ phân giải 512x512, ta đánh giá sự khác biệt về:

* Tỉ số nén
* Dung lượng các tệp ảnh (KB) sau khi nén
* Chất lượng hiển thị (trực quan)
  1. Lựa chọn môi trường, công cụ
* Sử dụng phần mềm Advanced JPEG Compressor
  1. Phân tích tình huống
* Ta sử dụng 1 bức ảnh ban đầu (raw image) được chụp từ máy ảnh lưu dưới dạng ảnh .bmp (Bitmap)
* Nén ảnh theo định dạng TIFF, tiếp đó xác định dung lượng tệp ảnh.
* Nén ảnh theo định dạng JPEG, tiếp đó xác định dung lượng tệp ảnh.
* So sánh kết quả về dung lượng của ảnh sau khi được nén theo hai định dạng trên.
  1. Quá trình thực hiện
* Ảnh đầu vào là một bức ảnh kích thước 512x512 ở hệ màu RGB với dung lượng 769 KB:



Ảnh ban đầu (Bitmap)

* Nén bức ảnh trên theo định dạng .tiff và định dạng .jpeg ở các tỉ số nén khác nhau sử dụng hệ màu YCrCb:

|  |  |
| --- | --- |
| Ảnh TIFF  Tỉ số nén 1:1.1  Dung lượng: 721KB | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color.tiff |
| Ảnh JPEG  Tỉ số nén kém nhất có thể 1:3.7  Dung lượng: 209KB | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color4.jpg |
| Ảnh JPEG  Tỉ số nén 1:8.5  Dung lượng: 91KB | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color8.jpg |
| Ảnh JPEG  Tỉ số nén 1:32  Dung lượng: 25KB | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color32.jpg |
| Ảnh JPEG  Tỉ số nén 1:101.5  Dung lượng: 8KB | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color100.jpg |
| Ảnh JPEG  Tỉ số nén tốt nhất có thể 1:277.7 | B:\Tai Lieu\Hoctap\20102\GoogleCode\Xu li du lieu da phuong tien\lena512color277.jpg |

* 1. Đánh giá kết quả

Ở những bức ảnh JPEG có tỉ số nén thấp khoảng 1:4 tới 1:8.5 ta vẫn thu được mức độ hiển thị gần giống với ảnh ban đầu. Với các bức ảnh có tỉ số nén cao hơn, chất lượng ảnh đã bị giảm xuống đáng kể. Các bức ảnh được phóng to để thấy được sự khác biệt theo các tỉ số nén lần lượt là: ảnh gốc, 1:3.7, 1:8.5, 1:32, 1:100







Đối với ảnh TIFF, chất lượng ảnh giống hệt ảnh ban đầu, và khi giải nén ta thu lại đc ảnh gốc với dung lượng file không thay đổi.



Rõ ràng, với cùng một chất lượng ảnh thu được, ảnh JPEG có dung lượng nhỏ hơn hẳn ảnh gốc cũng như ảnh TIFF.

TÀI LIỆU THAM KHẢO

[1] Bài giảng môn Xử lí dữ liệu đa phương tiện – PGS.TS. Nguyễn Thị Hoàng Lan

[2] Nhập môn Xử lí ảnh số - Lương Mạnh Bá, Nguyễn Thanh Thủy

[3] Raw, JPEG and TIFF – Bob Atkins

[4] TIFF 6, Revision 6.0 – Adobe Systems Incorporated – June 3, 1992