Xây dựng phương pháp mã hóa thông tin hình ảnh đơn giản dựa trên Chuẩn mã hóa H.265

Bùi Duy Nam
Khoa Điện tử Viễn thông
Trường Đại học Công Nghệ - ĐHQGHN
Hà Nội, Việt Nam
18020936@vnu.edu.vn

***Tóm tắt nội dung*—Trong bài này, em trình bày một phương pháp mã hóa hình ảnh đơn giản dựa trên các khối cơ bản của chuẩn mã hõa H.265/HEVC với các khối bao gồm biến đổi, lượng tử hóa, mã hóa,... Kết quả của bộ mã hóa được thể hiện định tính thông qua hình ảnh trực quan cùng với định lượng thông qua chỉ số PSNR, tỉ số nén, thời gian mã hóa, ...**

***Index Terms*—Mã hóa hình ảnh, Biến đổi DCT, Lượng tử hóa, Mã hóa Số học**

I. Giới thiệu

Trong những năm gần đây, các thiết bị số hiện đại ra đời và ngày càng nhiều, lượng hình ảnh và video cần lưu trữ và truyền tải ngày càng gia tăng. Việc lưu trữ các dữ liệu thô là vô cùng tốn kém. Các chuẩn mã hóa liên tục được ra đời với hiệu năng ngày một cao. Trong đó điển hình là các chuẩn mã hóa H.265/HEVC, H.264/AVC, MPEG, AVI,...

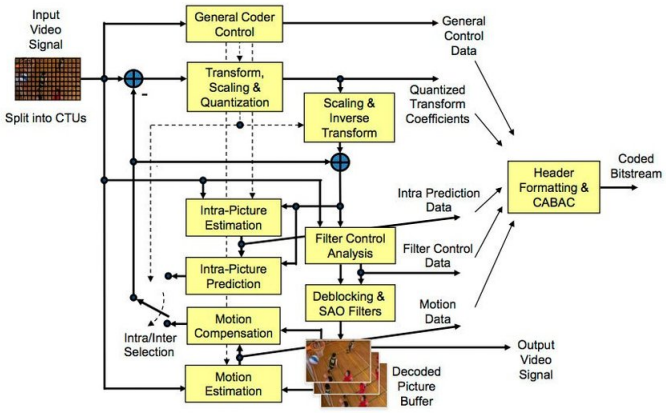
Việc nắm vững các khối trong chuẩn mã hóa hình ảnh, video có vai trò rất quan trọng trong việc phát triển, cải tiến các chuẩn mã hóa sau này. Trong phần tiếp theo của bài này, em trình bày tổng quan về mô hình mã hóa chuẩn H.265/HEVC (phần II). Tiếp đó, phần III sẽ trình bày về mô hình mã hóa ình ảnh đơn giản của em với chi tiết các khối trong bộ mã hóa. Phần IV đưa ra kết quả và thảo luận về hiệu năng nén của bộ mã hóa. Cuối cùng là phần kết luận được trình bày tại phần V.

II. Tổng quan về Chuẩn mã hóa H.265 (HEVC)

Mã hóa video hiệu quả cao (HEVC), còn được gọi là H.265 và MPEG-H Phần 2, là một tiêu chuẩn nén video được thiết kế như một phần của dự án MPEG-H như một sự kế thừa cho Mã hóa video nâng cao được sử dụng rộng rãi (AVC, H. 264, hoặc MPEG-4 Phần 10). So với AVC, HEVC cung cấp khả năng nén dữ liệu tốt hơn từ 25% đến 50% ở cùng mức chất lượng video hoặc chất lượng video được cải thiện đáng kể ở cùng tốc độ bit. Nó hỗ trợ độ phân giải lên đến 8192x4320, bao gồm 8K UHD và không giống như AVC 8-bit chủ yếu, cấu hình Main10 có độ trung thực cao hơn của HEVC đã được tích hợp vào gần như tất cả các phần cứng hỗ trợ.

Trong khi AVC sử dụng biến đổi cosin rời rạc số nguyên (DCT) với kích thước khối 4x4 và 8x8, HEVC sử dụng biến đổi DCT và DST số nguyên với các kích thước khối khác nhau giữa 4x4 và 32x32. Định dạng Hình ảnh Hiệu quả Cao (HEIF) dựa trên HEVC. Tính đến năm 2019, HEVC được 43% nhà phát triển video sử dụng và là định dạng mã hóa video được sử dụng rộng rãi thứ hai sau AVC.

HEVC là phần mở rộng của các khái niệm trong H.264 / MPEG-4 AVC. Cả hai đều hoạt động bằng cách so sánh các phần khác nhau của khung video để tìm các khu vực bị thừa, cả trong một khung hình đơn lẻ và giữa các khung hình liên tiếp. Những vùng dư thừa này sau đó được thay thế bằng một mô tả ngắn thay vì các pixel ban đầu. Những thay đổi chính đối với HEVC bao gồm việc mở rộng khu vực so sánh mẫu và mã hóa sự khác biệt từ 16x16 pixel thành kích thước lên đến 64x64, cải thiện phân đoạn kích thước khối có thể thay đổi, cải thiện dự đoán "nội bộ" trong cùng một hình ảnh, cải thiện chuyển động dự đoán vectơ và hợp nhất vùng chuyển động, cải tiến lọc bù chuyển động và một bước lọc bổ sung được gọi là lọc bù thích ứng mẫu. Việc sử dụng hiệu quả những cải tiến này đòi hỏi khả năng xử lý tín hiệu cao hơn nhiều để nén video, nhưng ít ảnh hưởng hơn đến lượng tính toán cần thiết để giải nén.



Hình 1: Chuẩn mã hóa H.265 (HEVC)

Trong bài này, em tham khảo cách xây dựng chuẩn mã hóa H.265 để xây dựng một bộ mã hóa thông tin hình ảnh đơn giản.

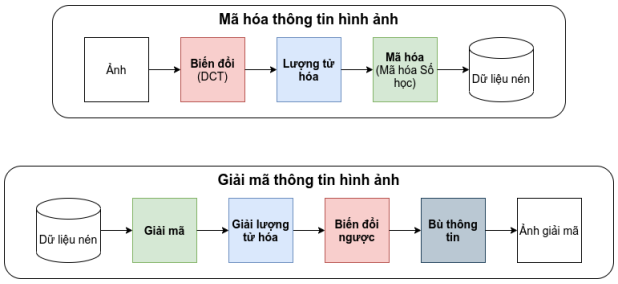
III. Mô hình Mã hóa hình ảnh đơn giản

Mô hình mã hóa hình ảnh của em được xây dựng dựa trên Chuẩn mã hóa H.265 bao gồm các khối đơn giản như khối Biến đổi, khối Lượng tự hóa, khối mã hóa entropy,...

Mô hình Mã hóa và Giải mã hình ảnh bao gồm 3 khối chính:

- **Khối Biến đổi**: Sử dụng phép biến đổi DCT 2 chiều với các khối 8x8 pixel.
- **Khối Lượng tử hóa**: Sử dụng ma trận lượng tử hóa được lấy từ ma trận lượng tử hóa của Chuẩn mã hóa H.265.
- **Khối Mã hóa**: Sử dụng Bộ mã hóa Số học.

Bên cạnh đó, phần Giải mã thông tin hình ảnh còn có thêm **khối Bù thông tin** để giảm thiểu tổn thất trong quá trình Mã hóa và Giải mã.



Hình 2: Sơ đồ Mã hóa và Giải nén thông tin hình ảnh đơn giản

A. Khối Biến đổi

Một biến đổi cosin rời rạc (DCT) biểu thị một chuỗi hữu hạn các điểm dữ liệu dưới dạng tổng các hàm cosin dao động ở các tần số khác nhau. DCT là một kỹ thuật biến đổi được sử dụng rộng rãi trong xử lý tín hiệu và nén dữ liệu (JPEG) video kỹ thuật số (MPEG và H.265),...

Nén DCT nén dữ liệu trong các tập hợp các khối DCT rời rạc. Các khối DCT có thể có một số kích thước, bao gồm 8x8 pixel cho DCT tiêu chuẩn và các kích thước DCT số nguyên khác nhau giữa 4x4 và 32x32 pixel. DCT có đặc tính "nén năng lượng" mạnh, và có khả năng đạt được chất lượng cao ở tỷ lệ nén dữ liệu cao. Tuy nhiên, các nhiễu khối có thể xuất hiện khi thực hiện mã hóa DCT theo khối.

Trong bài này, em sử dụng DCT cho phép biến đổi ảnh từ miền không gian pixel sang miền tần số, với các khối sử dụng đề là 8x8 pixel.

Phép biến đổi DCT 2 chiều được thể hiện thông qua Phương trình 1.

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i,j) \\ cos\left[\frac{\pi}{N}\left(i+\frac{1}{2}\right)u\right] cos\left[\frac{\pi}{M}\left(j+\frac{1}{2}\right)v\right] \quad (1)$$

B. Khối Lượng tử hóa

Trong phần này, em sử dụng ma trận lượng tử hóa cho lượng tử hóa hình ảnh. Ma trận lượng tử hóa được thiết kế để cung cấp độ phân giải cao hơn cho các thành phần tần số có thể cảm nhận được hơn so với các thành phần ít cảm nhận hơn (thường là tần số thấp hơn tần số cao) ngoài việc biến đổi nhiều thành phần thành 0, có thể được mã hóa với hiệu quả cao nhất.

Ma trận Lượng tử hóa của chuẩn H.265 được sử dụng trong bộ mã hóa của em:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 67 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Kết hợp khối Biến đổi DCT và khối Lượng tử hóa đồng thời lọc bỏ các thành phần tần số cao, em xây dựng một thuật toán biến đổi ảnh từ miền không gian pixel sang miền tần số được thể hiện bởi Thuật toán 1 và Thuật toán 2.

Algorithm 1: Biến đổi DCT sang miền tần số kết hợp lượng tử hóa

```
Result: Transform_matrix
Compute number of 8x8 block;
for each block in img do
    patch = DCT(block);
    patch = round(patch / QMatrix);
    Transfom_matrix.append(patch[:4, :4]);
end
```

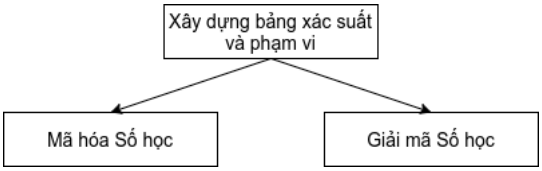
Algorithm 2: Biến đổi IDCT sang miền pixel kết hợp giải lượng tử hóa

```
Result: Inverse_Transform_matrix
Compute number of patch;
for each path in Transfom_matrix do
    block = zeros(8,8);
    block[:4, :4] = patch;
    block *= QMatrix;
    block = IDCT(block);
    Inverse_Transfom_matrix.append(block);
end
```

C. Khối Mã hóa - Mã hóa số học

Mã hóa số học (Arithmetic coding) là một dạng mã hóa entropy được sử dụng trong nén dữ liệu không mất dữ liệu. Khi một chuỗi được chuyển đổi sang mã hóa số học, các ký tự được sử dụng thường xuyên sẽ được lưu trữ với ít bit hơn và các ký tự không thường xuyên xuất hiện sẽ được lưu trữ với nhiều bit hơn, dẫn đến tổng số bit được sử dụng ít hơn. Mã hóa số học khác với các dạng mã hóa entropy khác, chẳng hạn như mã hóa Huffman, ở chỗ, thay vì tách đầu vào thành các ký hiệu thành phần và thay thế mỗi mã bằng một mã, mã hóa số học mã hóa toàn bộ thông điệp thành một số duy nhất, một phân số có độ chính xác tùy ý q, trong đó $0,0 \leq q < 1,0$. Nó đại diện cho thông tin hiện tại dưới dạng một phạm vi, được xác định bởi hai số. Một họ mã hóa entropy gần đây được gọi là hệ thống số không đối xứng cho phép triển khai

nhanh hơn nhờ hoạt động trực tiếp trên một số tự nhiên duy nhất đại diện cho thông tin hiện tại.



Hình 3: Quy trình thực hiện mã hóa Số học

Mã hóa số học được thực hiện theo sơ đồ được thể hiện tại Hình 3. Xây dựng bảng xác xuất và phạm vi là bước đầu tiên và rất quan trọng cho quá trình mã hóa / giải mã. Bảng xác suất và phạm vi được xây dựng theo Thuật toán 3:

Algorithm 3: Xây dựng Bảng xác suất và phạm vi

```
Result: Probability, Left_range, Right_range
total = length(array) ;
for item in array do
    if item in Probability then
        Probability[item] += 1;
    else
        Probability[item] = 1;
    end
end
start = 0.0;
for key, value in Probability.item() do
    Probability[item] = value / total;
    Left_range[item] = start;
    Right_range[item] = start + Probability[item];
    start = Right_range;
end
```

Sau khi xây dựng thành công Bảng xác suất và phạm vi từ hình ảnh đầu vào, tiếp theo có thể mã hóa ảnh đầu vào theo Thuật toán 4 và giải mã theo Thuật toán 5

Kết quả đầu ra sẽ là một chuỗi các số thập phân q với $0.0 \leq q < 1.0$.

D. Lọc nhiễu và hậu xử lý sau giải mã

Trong khối này, em áp dụng các thuật toán xử lý hình ảnh cơ bản để giảm nhiễu, và lọc bỏ nhiễu khối sau giải mã và tăng cường chất lượng hình ảnh sau giải mã. Các thuật toán được sử dụng bao gồm:

- Các thuật toán làm mờ ảnh: Trung bình, Gaussian,...
- Thuật toán tăng cường chất lượng ảnh.

IV. KẾT QUẢ VÀO THẢO LUẬN

Trong phần này, chúng em sử dụng 2 hình ảnh để thể hiện kết quả của bộ mã hóa thông tin hình ảnh của mình. Các hình ảnh được thể hiện tại Hình 4.

Algorithm 4: Mã hóa Số học cho ảnh

```
Result: Encode_result
block_size = 4;
rows, cols = Matrix.shape();
total = rows*cols;
Array = Matrix.reshape(total);
for i in range(0, total, block_size) do
    l = 0.0;
    r = 1.0;
    for j in range(i, i+block_size) do
        oldLeft = l;
        oldRight = r;
        l = oldLeft + (oldRight - oldLeft) *
          Left_range[Array[j]];
        r = oldLeft + (oldRight - oldLeft) *
          Right_range[Array[j]];
    end
    it = int(i/block_size);
    Encode_result[it] = (l+r)/2;
end
```

Algorithm 5: Giải mã Số học cho ảnh

```
Result: Decode_result
block_size = 4;
for i in range(0, total, block_size) do
    l = 0.0;
    r = 1.0;
    code = Encode[i//block_size];
    for j in range(i, i+block_size) do
        for k in Probability.keys() do
            oldLeft = l;
            oldRight = r;
            lCheck = oldLeft + (oldRight - oldLeft) *
              Left_range[k];
            rCheck = oldLeft + (oldRight - oldLeft) *
              Right_range[k];
            if code > lCheck and code < rCheck then
                Decode_result[j] = k;
                l = lCheck;
                r = rCheck;
                break;
            end
        end
    end
end
Decode_result = Decode_result.reshape((rows, cols));
```

1) Định tính: Trong phần này, kết quả định tính được thể hiện thông qua mối tương quan giữa ảnh đầu vào và ảnh sau giải mã, so sánh chất lượng thông qua cảm nhận của mắt nhìn.

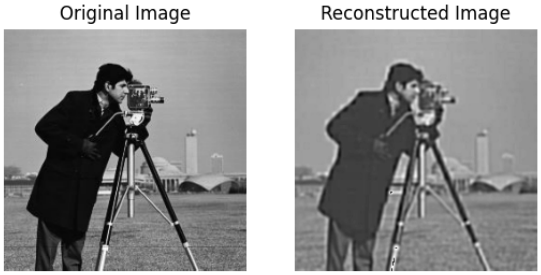
Các phương pháp được dùng để đánh giá chất lượng bộ mã hóa bào gồm hai phương pháp chính: Đánh giá định tính và định lượng.

Kết quả mã hóa và giải mã của bộ mã hóa của em được thể hiện trong Hình 5:

(a) Lena (b) Cameraman

Hình 4: Các hình ảnh để thử nghiệm bộ mã hóa

(a) Mã hóa và giải mã Lena



(b) Mã hóa và giải mã Cameraman

Hình 5: Kết quả mã hóa và giải mã

2) Định lượng: Trong phần này, em sử dụng các chỉ số bao gồm: thời gian mã hóa/giải mã, PSNR, tỉ số nén để đánh giá chất lượng của bộ mã hóa ảnh.

*** Thời gian mã hóa / giải mã**

Thời gian mã hóa / giải mã là thời gian mà bộ mã hóa dữ liệu nén/giải nén thành công một hình ảnh.

*** PSNR**

PSNR thể hiện tỷ số giữa công suất tối đa có thể có của tín hiệu và công suất của nhiễu làm hỏng ảnh hưởng đến độ trung thực của biểu diễn của nó. Vì nhiều tín hiệu có dải động rất rộng, PSNR thường được biểu thị dưới dạng đại lượng logarit sử dụng thang decibel.

PSNR được sử dụng để định lượng chất lượng tái tạo cho hình ảnh và video chịu nén mất dữ liệu.

PSNR được xác định dễ thông qua sai số bình phương trung bình (MSE). Cho một ảnh đơn sắc có kích thước mxn không nhiễu I và xấp xỉ nhiễu của nó là K, MSE được định nghĩa

Bảng I: Bảng kết quả định lượng cho bộ mã hóa hình ảnh

Ảnh	Kích thước	PSNR	Tỉ số nén	Thời gian mã hóa	Thời gian giải mã
Lena	512x512	31.47	15.6	0.17	0.35
Cameraman	256x256	30.29	17.2	0.04	0.10

theo Phương trình 2:

$$\text{MSE} = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{2}$$

PSNR được định nghĩa theo MSE theo Phương trình 3:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{3}$$

*** Tỉ số nén**

Tỷ số nén là tỷ số giữa dung lượng ảnh ban đầu và dung lượng dữ liệu sau khi mã hóa. Nó thể hiện mức độ nén dữ liệu của bộ mã hóa.

Kết quả thu được của bộ mã hóa của em cho hai hình ảnh thu được được thể hiển trong Bảng I.

V. Kết luận

Trong bài này, em nghiên cứu tổng quan về chuẩn mã hóa hình ảnh nói chung và chuẩn mã hóa H.265 nói riêng. Từ đó xây dựng được một bộ mã hóa hình ảnh đơn giản. Qua quá trình định tính và định lượng, bộ mã hóa chõ kết quả đầu ra tương đối giống ảnh gốc với thời gian mã hóa / giải mã rất nhỏ (< 0.5s). Source code được xuất bản tại https://github.com/duynamrcv/image_compressor2

Tài liệu

[1] Thomson, Gavin; Shah, Athar - "Introducing HEIF and HEVC".
[2] Bitmovin - "Video Developer Report 2019".
[3] Rishi Raj Sharma ; K. V. Arya - "Parameter optimization for HEVC/H.265 encoder using multi-objective optimization technique".
[4] Barbero, M.; Hofmann, H.; Wells, N. D. - "DCT source coding and current implementations for HDTV".
[5] Wang, Hanli; Kwong, S.; Kok, C. - "Efficient prediction algorithm of integer DCT coefficients for H.264/AVC optimization".
[6] Diego Coelho; Sushmabhargavi Nimmalapalli; Vassil Dimitrov; Arjuna Madanayake; Renato Cintra; Arnaud Tisserand - "Computation of 2D 8x8 DCT Based on the Loeffler Factorization Using Algebraic Integer Encoding"
[7] John Wiseman - "An Introduction to MPEG Video Compression".
[8] Ze-Nian Li; Mark S. Drew; Jiangchuan Liu - "Fundamentals of Multimedia".
[9] J. Duda, K. Tahboub, N. J. Gadil, E. J. Delp - "The use of asymmetric numeral systems as an accurate replacement for Huffman coding".
[10] Paul G. Howard, Jeffrey S. Vitter - "Arithmetic Coding for Data Compression".