XỬ LÝ ẢNH
&
THỊ GIÁC MÁY TÍNH

IMAGE PROCESSING AND COMPUTER VISION



CHƯƠNG 8: NÉN DỮ LIỆU ẢNH

VS

Original transparent PNG
File size: 57 KB

Shrunk transparent PNG
File size: 15 KB



CHƯƠNG 8: NÉN DỮ LIỆU ẢNH

Giới thiệu nén ảnh

Các phương pháp nén ảnh



GIỚI THIỆU

- Nén dữ liệu là phương pháp nhằm **giảm** thông tin **"dư thừa"** trong dữ liệu gốc nhằm thu được lượng **thông tin nhỏ hơn** dữ liệu gốc.
- Với dữ liệu ảnh, các thuật toán nén ảnh thường đạt hiệu quả **10:1**, một số cho kết quả cao hơn (VD: thuật toán **fratal** cho tỉ số nén **30:1**)



NÉN DỮ LIỆU ẢNH

- Có thể phân loại các phương pháp nén dữ liệu ảnh theo 3 hướng:
  - Phân loại theo nguyên lý
  - Phân loại theo cách thức thực hiện nén
  - Phân loại theo triết lý của sự mã hóa



NÉN DỮ LIỆU ẢNH

- Phân loại theo nguyên lý:
  - Nén chính xác (nén không mất thông tin)
    - Sau khi giải nén ta thu được dữ liệu gốc
  - Nén không bảo toàn (nén có mất thông tin)
    - Sau giải nén không thu được hoàn toàn dữ liệu gốc
    - Lợi dụng khả năng có hạn của mắt người để loại bỏ dữ liệu



NÉN DỮ LIỆU ẢNH

- Phân loại theo cách thức thực hiện nén
  - Phương pháp nén không gian (Spatial Data Compression)
    - Thực hiện nén bằng các mẫu ảnh trong không gian
  - Phương pháp sử dụng biến đổi (Transform Coding)
    - Bao gồm các phép biến đổi ảnh gốc



NÉN DỮ LIỆU ẢNH

- Phân loại theo triết lý của sự mã hóa
  - Phương pháp nén thế hệ thứ nhất
    - Bao gồm các phương pháp đơn giản (lấy mẫu, gán từ mã hóa)
  - Phương pháp nén thế hệ thứ hai
    - Dựa vào độ bão hòa của tỷ lệ nén



CÁC PHƯƠNG PHÁP NÉN DỮ LIỆU ẢNH

- Phương pháp nén thế hệ thứ nhất
- Phương pháp mã hóa loạt dài
- Phương pháp mã hóa Huffman
- Phương pháp LZW
- Phương pháp mã hóa khối
- Phương pháp thích nghi
- Biến đổi Cosin và chuẩn nén JPEG



CÁC PHƯƠNG PHÁP NÉN DỮ LIỆU ẢNH

- Phương pháp nén thế hệ thứ hai
- Phương pháp Kim tự tháp
  - Phương pháp Kim tự tháp Laplace (Laplacian pyramid)
- Phương pháp mã hóa dựa vào biểu diễn ảnh
  - Phương pháp mã hóa dựa vào vùng gia tăng
  - Phương pháp tách-hợp



NÉN DỮ LIỆU ẢNH

- Phương pháp mã hóa loạt dài (RLCRunLength Encoding)
  - Định nghĩa: một loạt dài là một dãy các ký hiệu lặp lại liên tục
  - Mục đích của mã hóa loạt dài là xác định các loạt dài, kích thước, và các ký hiệu trong loạt dài



VÍ DỤ

- K KKKK KKKK
- ABC DEFG
- ABA BBBC
- A bc12 3bbb bCDE
- Xác định các loạt dài:
- 1. KKKKKKKKK => Loạt dài = 9 ký hiệu K
- 2. ABCDEFG => Không có loạt dài nào.
- 3. ABABBBC => loạt dài = 3 ký hiệu B
- 4. abc123bbbbCDE => loạt dài = 4 ký hiệu b



NÉN DỮ LIỆU ẢNH

- Mã hóa loạt dài gán các từ mã cho các loạt dài thay vì mã hóa cho từng ký hiệu riêng biệt.
- Mỗi loạt dài được thay thế bởi 1 từ mã gồm 3 phần (r, l, s). Trong đó:
  - r: ký hiệu cờ lặp lại (r : repeat)
  - l : độ dài của loạt dài (l: length)
  - s: các ký hiệu có mặt trong loạt dài (s : symbol)



NÉN DỮ LIỆU ẢNH

- Với ví dụ trên:
  - 1. Loạt dài gồm 9 ký hiệu K được thay thế bởi mã (‘r’, ‘9’, ‘K’) hoặc r9K.
  - Dãy thứ 2 : ABCDEFG không phải là loạt dài được thay thế bằng dãy (‘n’, ‘7’, ABCDEFG) hoặc n7ABCDEFG.
    - N: cờ loạt dài không lặp lại (n: non - repeat)



NÉN DỮ LIỆU ẢNH

- Ví dụ: Mã loạt dài cho dữ liệu sau:
- A AAAA AAAA BBBB BBBC CCCC DDEF EDDC CCCC BBBB BBBA AAAA AAAA
- Tính tỉ số nén nếu mỗi ký hiệu sẽ được biểu diễn bởi 8 bit trong trường hợp không nén và các loạt dài có độ dài < 256.



NÉN DỮ LIỆU ẢNH

- Thực hiện RLE theo Gray code và Binary code cho ảnh xám
- Đối với ảnh nhị phân, các bước thực hiện:
  - Mã hóa từng dòng riêng biệt, bắt đầu với số lượng số 0
  - Mã hóa một chuỗi số 0 và số 1, bằng cách RLE, lặp lại các số 0 và 1 trong mỗi chuỗi.



NÉN DỮ LIỆU ẢNH

- Đối với ảnh xám các bước thực hiện như sau:
  - Chuyển ảnh mức xám thành nhiều ảnh nhị phân được gọi là plane
  - Tách ảnh
  - Thực hiện RLE, mã hóa Huffman một chuỗi số 0 và số 1, lặp lại các số 0 và 1 trong mỗi chuỗi.
  - Tính tỷ số nén (độ dài từ mã trước và sau nén)



NÉN DỮ LIỆU ẢNH

- Ví dụ: Cho ma trận ảnh I như sau
- Hãy chuyển ma trận ảnh trên sang gray code và binary code.
- Tách ảnh và mã hóa RLE
- Thực hiện RLE, mã hóa Huffman một chuỗi số 0 và số 1, lặp lại các số 0 và 1 trong mỗi chuỗi.
- Tính tỷ số nén (độ dài từ mã trước và sau nén)

$$I = \begin{bmatrix} 5 & 5 & 6 & 5 & 5 & 4 \\ 5 & 4 & 6 & 5 & 4 & 4 \\ 4 & 4 & 5 & 4 & 5 & 5 \\ 3 & 3 & 4 & 3 & 4 & 4 \\ 2 & 3 & 4 & 3 & 2 & 3 \\ 1 & 2 & 3 & 2 & 1 & 2 \end{bmatrix}$$



NÉN DỮ LIỆU ẢNH

- Kết luận : Tỷ số nén gray code nhỏ hơn tỷ số nén binary code, điều đó chứng tỏ gray code giảm được nhiều dư thừa hơn binary code. Đó là do các biểu diễn các số gần nhau chỉ khác nhau 1 bit nên đã tạo ra nhiều quá trình lặp thuận lợi khi thực hiện RLE



NÉN DỮ LIỆU ẢNH

- Phương pháp này được sử dụng để mã hóa ảnh trong ảnh PCX và BMP
- Ta có thể mã hóa sử dụng chiều dài cố định hoặc thích nghi kiểu Huffman



PHƯƠNG PHÁP MÃ HÓA HUFFMAN

- Mã hóa Huffman dựa vào mô hình thống kê
- Dựa vào dữ liệu gốc, tần suất xuất hiện của các ký tự được tính toán
- Sau đó gán cho ký tự tần suất cao mã ngắn và ký tự tần suất ít mã dài
- Được phát triển để mã hóa chung các loại dữ liệu khác nhau tuy nhiên chỉ một số loại dữ liệu mới mang lại hiệu quả mong muốn



PHƯƠNG PHÁP MÃ HÓA HUFFMAN

- Việc mã hóa này giúp giảm lượng dữ liệu cho ký tự xuất hiện nhiều hơn và có thể giảm lượng dữ liệu cần lưu trữ
- Tuy nhiên trong một số trường hợp mã hóa theo cách này có thể gây bất lợi chứ không có lợi (khi sự khác biệt về tần suất không nhiều)



PHƯƠNG PHÁP MÃ HÓA HUFFMAN

- Các mã Huffman được xây dựng từ dưới lên trên, bắt đầu với các nút lá của cây và lặp lại cho đến khi gặp nút gốc
- Để thực hiện mã hóa Huffman thì:
  - Các ký hiệu được sắp xếp thành 1 dãy các nút lá để tạo thành cây nhị phân.
  - Mỗi nút được gán 1 trọng số là tần suất xuất hiện của ký hiệu tương ứng.



CÁCH XÂY DỰNG CÂY MÃ HUFFMAN

1. Hai nút chưa được xét có trọng số nhỏ nhất sẽ được gắn vào 1 nút mới có trọng số bằng tổng trọng số của 2 nút này.
2. Nút mới này sẽ được thêm vào danh sách các nút chưa xét đến và loại bỏ 2 nút đã xét trong danh sách.
3. 1 trong 2 nút được gán mã là 0 (ví dụ bên trái), nút còn lại được gán mã là 1 (bên phải).
4. Lặp lại các bước trên cho đến khi chỉ còn 1 nút trong danh sách. Nút còn lại được xem là gốc của cây mã.



NÉN DỮ LIỆU ẢNH

- Ví dụ: Cho thông điệp

“BCA ACAD BDCA DAEE EABA CDBA CADC BADA BEAB EAAA”



NÉN DỮ LIỆU ẢNH

- Xây dựng cây mã Huffman với tần suất xuất hiện như sau:

Ký tự	Tần suất
A	24
B	12
C	10
D	8
E	8



NÉN DỮ LIỆU ẢNH

- Ưu điểm của phương pháp mã hoá Huffman là đạt được hệ số nén cao (Hệ số nén tuỳ thuộc vào cấu trúc của các tập tin).
- Nhược điểm của phương pháp này là bên nhận muốn giải mã được thông điệp thì phải có một bảng mã giống như bảng mã ở bên gửi, do đó khi nén các tập tin bé hệ số nén không được cao.



PHƯƠNG PHÁP LZW

- Được Abraham Lempel, Jacob Ziv phát triển trước rồi được Terry Welch nâng cấp (Lempel–Ziv–Welch)
- Thuật toán này là thuật toán nén từ điển, dựa vào việc lập một từ điển các ký tự có tần suất cao
- Điểm mạnh của kỹ thuật này là khả năng tổ chức từ điển để đạt hiệu quả cao
- Có thể được dùng để nén các loại file nhị phân khác nhau



PHƯƠNG PHÁP LZW

- Một chuỗi ký tự cần nén phải tuân thủ nguyên tắc sau:
  - Một tập hợp từ hai kí tự trở lên gọi một xâu kí tự
  - Nếu tìm thấy các xâu kí tự đã gặp, phải nhớ và gán cho nó một dấu hiệu (token) riêng.
  - Nếu lần sau gặp lại xâu kí tự đó, xâu kí tự sẽ được thay thế bằng dấu hiệu của nó.



PHƯƠNG PHÁP LZW

- Ví dụ có chuỗi sau:

0101010101000000111111010101010011100100111

- Quét qua toàn bộ chuỗi và tìm ra các đoạn giống nhau.

0101010101000000111111010101010011100100 111

- Ở đây đoạn “0101010101” xuất hiện hai lần nên ta định nghĩa nó bằng một từ mới là “X”. Sau bước này chuỗi trên của ta còn:



PHƯƠNG PHÁP LZW

- X000000111111X001110010011 1
- Tương tự lần quét tiếp và đặt vào “từ điển” ta có chuỗi kết quả như sau:
- X000000111111X001110010011 1
- Với: X = 0101010101; A = 00; B = 001; C = 11
- XAABCCXBCBBC
- Như vậy chuỗi 45 byte ban đầu được nén lại còn 12 byte tức là còn 12/45 = 26,67% dung lượng ban đầu.



NÉN DỮ LIỆU ẢNH

- Phần quan trọng nhất của phương pháp nén này là phải tạo một mảng rất lớn dùng để lưu giữ các xâu kí tự đã gặp, mảng này được gọi là "Từ điển".
- Khi các byte dữ liệu cần nén được đem đến, chúng liền được giữ lại trong một bộ đệm chứa (Accumulator) và đem so sánh với các chuỗi đã có trong "từ điển".



NÉN DỮ LIỆU ẢNH

- Nếu chuỗi dữ liệu trong bộ đệm chứa không có trong "từ điển" thì nó được bổ sung thêm vào "từ điển" và chỉ số của chuỗi ở trong "từ điển" chính là dấu hiệu của chuỗi.
- Nếu chuỗi trong bộ đệm chứa đã có trong "từ điển" thì dấu hiệu của chuỗi được đem ra thay cho chuỗi ở dòng dữ liệu ra.



NÉN DỮ LIỆU ẢNH

- Do kích thước bộ nhớ không phải vô hạn và để đảm bảo tốc độ tìm kiếm, từ điển chỉ giới hạn 4096 ở phần tử dùng để lưu lớn nhất là 4096 giá trị của các từ mã. Như vậy độ dài lớn nhất của từ mã là 12 bits ($4096 = 2^{12}$).



NÉN DỮ LIỆU ẢNH

- LZW dựa vào một từ điển lưu các mẫu có tần suất cao trong ảnh
- LZW sẽ luôn cập nhật từ điển mỗi khi đọc thêm được một ký tự mới
- Từ điển có tối đa 4096 từ vựng để đảm bảo hiệu quả (độ dài lớn nhất của từ mã là 12 bit)

NÉN DỮ LIỆU ẢNH



- Từ điển của LZW
- Cấu trúc từ điển

0	0	
1	1	
...	...	
255	255	(Clear Code)
256	Chuỗi	
257	Chuỗi	
258	Chuỗi	
259	Chuỗi	
...	...	
...	...	
4095	Chuỗi	



NÉN DỮ LIỆU ẢNH

- 256 từ đầu (0...255) là mã của ký tự ASCII
- Từ 256 là mã đặc biệt (CC – Clear Code)
  - Để khắc phục trường hợp số mẫu lặp trong ảnh lớn hơn 4096
  - Mã xóa sẽ chỉ báo việc kết thúc mã hóa với từ điển cũ và bắt đầu bộ tự điển mới
- Từ 257 là mã (EOI – End Of Information)
  - Giúp phân chia file thành nhiều cụm ảnh (file ảnh động GIF có nhiều ảnh gộp lại)



NÉN DỮ LIỆU ẢNH

- Còn lại là (258...4095) là các mẫu lặp lại trong ảnh.
  - 512 phần tử đầu tiên được biểu diễn bởi 9 bit
  - 512 đến 1023 biểu diễn bởi 10 bit
  - 1024 đến 2047 biểu diễn bởi 11 bit
  - 2048 đến 4095 biểu diễn bởi 12 bit



PHƯƠNG PHÁP LZW (GIẢI NÉN)



PHƯƠNG PHÁP LZW (GIẢI NÉN)