

XỬ LÝ ẢNH

GIỚI THIỆU MÔN HỌC

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

1 Mục tiêu môn học

2 Tài liệu tham khảo

3 Tài liệu tham khảo

4 Dánh giá

Mục tiêu môn học

Cung cấp cho sinh viên những kiến thức cơ bản:

- Khái niệm về ảnh số.
- Một số phương pháp nâng cao chất lượng ảnh.
- Một số phương pháp phân vùng ảnh.
- Một số phương pháp nén ảnh.
- Một số phương pháp nhận dạng ảnh.
- ...

Tài liệu tham khảo

- Lương Mạnh Bá, Nguyễn Thanh Thủy, Nhập môn xử lý ảnh số. Nhà xuất bản khoa học kỹ thuật, 2003.
- Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing, Prentice Hall, 3 rd Edition, 2016.
- Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, Prentice Hall, 3 rd Edition.

Các nguồn tham khảo khác

Các bạn có thể tìm các bài báo khoa học ở những trang sau:

- IEEE Xplore: <https://ieeexplore.ieee.org/Xplore/home.jsp>.
- ScienceDirect: <https://www.sciencedirect.com>.
- Springer: <https://link.springer.com>.

Dánh giá

Hình thức đánh giá sinh viên như sau:

- Điểm học đầy đủ: 10%.
- Kiểm tra giữa kỳ: 30%.

Cách 1:

- Hình thức đọc báo khoa học và thuyết trình + code demo
- Mỗi nhóm không quá 4 bạn, mỗi bạn đều phải có công việc cụ thể.

Cách 2:

- Hoàn thành 3 bài kiểm tra giữa kỳ.
- Kiểm tra cuối kỳ: 60%.
- Gồm 4 bài, thời gian làm bài 1 tiếng.

Chú ý: Những sinh viên nghỉ quá 20% tổng số tiết học sẽ bị cấm thi.

Một số bài toán

- Giấu tin trong ảnh.
- Thủy vân số.
- Tăng cường chất lượng ảnh.
- Tổng hợp hình ảnh (y học, viễn thám.)
- Nén ảnh.
- Phân loại ảnh và nhận dạng.
- ...

XỬ LÝ ẢNH

TỔNG QUAN XỬ LÝ ẢNH

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

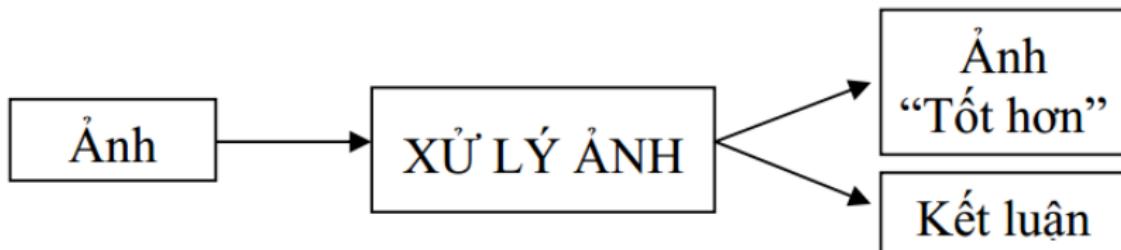
Nội Dung

- 1 Tổng quan về hệ thống xử lý ảnh
- 2 Những vấn đề cơ bản trong xử lý ảnh
- 3 Thu nhận và biểu diễn ảnh

Tổng quan về hệ thống xử lý ảnh

- Khái niệm về quá trình xử lý ảnh:

Là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Những vấn đề cơ bản trong xử lý ảnh

- Chỉnh mức xám.
- Khử nhiễu.
- Nhận dạng.
- Nén ảnh.
- ...

Thu nhận và biểu diễn ảnh

- Ảnh có thể được thu nhận thông qua các thiết bị thu nhận ảnh. Bao gồm 2 loại chính ứng với 2 loại ảnh thông dụng Raster, Vector và có thể cho ảnh đen trắng, ảnh xám hoặc ảnh màu.
- Ảnh trên máy tính là kết quả thu nhận theo các phương pháp số hoá. Các ảnh thường được biểu diễn theo 2 mô hình cơ bản:
 - Mô hình Raster.
 - Mô hình Vector.

Thu nhận và biểu diễn ảnh

Mô hình Raster

- Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh.
- Thường thu nhận qua các thiết bị như camera, scanner.
- Mỗi điểm ảnh được biểu diễn qua 1 hay nhiều bit.
- Thuận lợi cho hiển thị và in ấn.

Mô hình Vector

- Ảnh vector được tạo nên từ những yếu tố cốt lõi như điểm ảnh, đường thẳng, đường cong, những hình dạng, và đa giác.
- Hình ảnh vector có thể phóng to hay thu nhỏ tùy ý mà không bị vỡ, các đường viền cũng không bị răng cưa.
- Dữ liệu có trong ảnh vector ít hơn ảnh bitmap, do đó ít tốn dung lượng lưu trữ hơn.

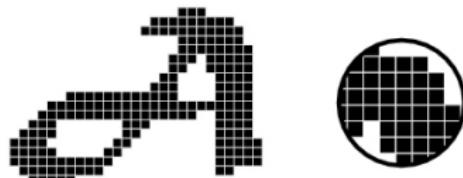
Thu nhận và biểu diễn ảnh

Ví dụ về ảnh Vector và ảnh Bitmap

Ảnh Vector



Ảnh Bitmap



Thu nhận và biểu diễn ảnh

Một số khái niệm cơ bản của ảnh Bitmap:

- **Phần tử ảnh:** Được gọi là pixel. Mỗi pixel gồm một cặp tọa độ x, y và màu.
- **Độ phân giải:** Là số lượng điểm ảnh (pixel) dùng để tập hợp thành hình ảnh.
- **Mức xám:** Là các giá trị có thể có của điểm ảnh hay còn gọi là cường độ sáng.

Một số kiểu ảnh

Ảnh đen trắng:

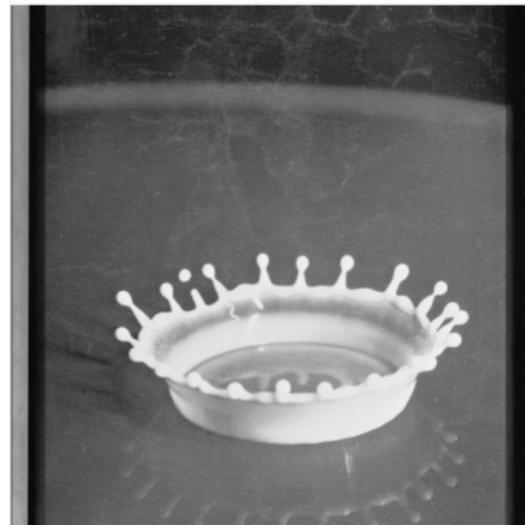
- Được biểu diễn bởi một ma trận ảnh.
- Mỗi phần tử của ma trận nhận các giá trị 0 hoặc 1.



Một số kiểu ảnh

Ảnh xám 8 bit:

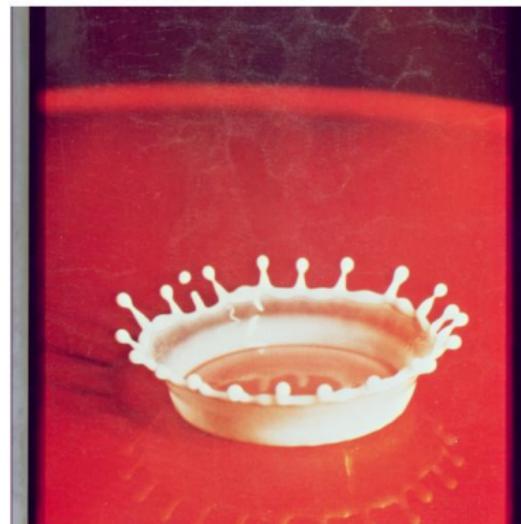
- Được biểu diễn bởi một ma trận ảnh.
- Mỗi phần tử của ma trận nhận các giá trị từ 0 đến 255.



Một số kiểu ảnh

Ảnh màu 24 bit:

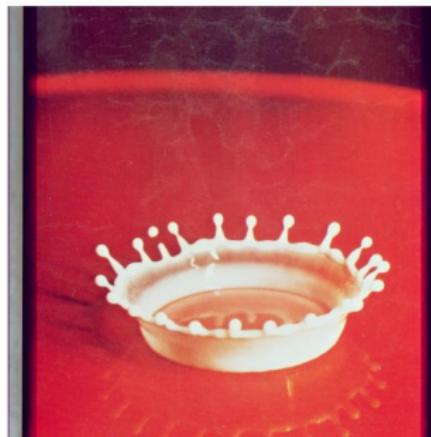
- Được biểu diễn bởi 3 ma trận ảnh (Red, Green, Blue).
- Mỗi phần tử của từng ma trận (R, G, B) nhận các giá trị từ 0 đến 255.



Một số kiểu ảnh

Ảnh màu 32 bit:

- Được biểu diễn bởi 3 ma trận ảnh (Red, Green, Blue).
- Có thêm ma trận alpha (A) có thể tạo ra sự trong suốt cho ảnh.
- Mỗi phần tử của từng ma trận (R, G, B) nhận các giá trị từ 0 đến 255.



Một số câu lệnh cơ bản trong Matlab

- Lệnh đọc ảnh:

$I = imread(\text{đường dẫn tới file ảnh}).$

- Lệnh ghi ảnh:

$imwrite(\text{ảnh}, \text{đường dẫn vị trí lưu ảnh} + \texttên file ảnh).$

- Lệnh chuyển từ ảnh màu sang dạng nhị phân:

$Ibw = im2bw(I, 0.5)$ hoặc $Ibw = imbinarize(I).$

- Lệnh chuyển từ ảnh màu sang dạng ảnh xám:

$Igray = rgb2gray(I).$

XỬ LÝ ẢNH

NÂNG CAO CHẤT LƯỢNG ẢNH

(MIỀN KHÔNG GIAN)

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

- 1 Tổng quan về nâng cao chất lượng ảnh
- 2 Một số chỉ số đánh giá
- 3 Kỹ thuật nâng cao chất lượng ảnh
- 4 Một số phép biến đổi

Tổng quan về nâng cao chất lượng ảnh

- Nâng cao chất lượng ảnh là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh.
- Nâng cao chất lượng ảnh gồm hai giai đoạn khác nhau: tăng cường ảnh và khôi phục ảnh.
- Mục đích nhằm hoàn thiện các đặc tính của ảnh như:
 - **Tăng độ tương phản, điều chỉnh mức xám của ảnh.**
 - Lọc nhiễu, hay làm trơn ảnh.
 - Làm nổi biên ảnh.

Một số chỉ số đánh giá chất lượng ảnh

Để có thể định lượng được chất lượng ảnh sau khi tăng cường có thực sự tốt hay không, có một số chỉ số đánh giá chất lượng ảnh thường được sử dụng:

- Độ sáng của ảnh (Cường độ sáng trung bình).
- Độ tương phản (Phương sai).
- Lượng thông tin (Entropy).
- Độ sắc nét (Trung bình biên ảnh).

Một số chỉ số đánh giá chất lượng ảnh

Cho một ảnh I có kích thước MxN:

Đô sáng của ảnh

$$\mu = \frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j)$$

Chú ý: Lệnh $M = \text{mean}(I, \text{'all'})$ dùng để tính giá trị trung bình của ma trận ảnh I trong matlab.

Ví dụ: Cho ma trận 8 bit kích thước 3×3 . Tính độ sáng của ảnh.

255	102	51
51	153	204
153	0	153

Một số chỉ số đánh giá chất lượng ảnh

Chuyển ma trận ảnh về miền $[0,1]$:

1	0.4	0.2
0.2	0.6	0.8
0.6	0	0.6

Độ sáng của ảnh:

$$\mu = \frac{1 + 0.4 + 0.2 + 0.2 + 0.6 + 0.8 + 0.6 + 0 + 0.6}{9} = 0.4889.$$

Một số chỉ số đánh giá chất lượng ảnh

Cho một ảnh I có kích thước MxN:

Độ tương phản của ảnh

$$\sigma^2 = \frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} l^2(i,j) - \left(\frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} l(i,j) \right)^2$$

Chú ý: Lệnh $V = \text{var}(I,1,'all')$ dùng để tính giá trị phương sai của ma trận ảnh I trong matlab.

Ví dụ: Cho ma trận 8 bit kích thước 3×3 . Tính độ tương phản của ảnh.

255	102	51
51	153	204
153	0	153

Một số chỉ số đánh giá chất lượng ảnh

Chuyển ma trận ảnh về miền $[0,1]$:

1	0.4	0.2
0.2	0.6	0.8
0.6	0	0.6

Độ tương phản của ảnh:

$$\sigma^2 = \left(\frac{1^2 + 0.4^2 + 0.2^2 + 0.2^2 + 0.6^2 + 0.8^2 + 0.6^2 + 0^2 + 0.6^2}{9} \right) - \left(\frac{1 + 0.4 + 0.2 + 0.2 + 0.6 + 0.8 + 0.6 + 0 + 0.6}{9} \right)^2 = 0.0899.$$

Một số chỉ số đánh giá chất lượng ảnh

Cho một ảnh I có kích thước MxN:

Lượng thông tin của ảnh

$$E = - \sum_{i=0}^{255} p_i * \log_2(p_i)$$

p_i là xác suất xuất hiện của điểm ảnh có mức xám i ($i \in [0, 255]$)

Chú ý: Lệnh $E = \text{entropy}(I)$ dùng để tính lượng thông tin của ma trận ảnh I trong matlab.

Ví dụ: Cho ma trận 8 bit kích thước 3×3 . Tính lượng thông tin của ảnh.

255	102	51
51	153	204
153	0	153

Một số chỉ số đánh giá chất lượng ảnh

Chuyển ma trận ảnh về miền $[0,1]$:

1	0.4	0.2
0.2	0.6	0.8
0.6	0	0.6

Bảng thống kê tần số xuất hiện của các điểm ảnh:

Mức xám	0	0.2	0.4	0.6	0.8	1
Số lần xuất hiện	1	2	1	3	1	1
Xác suất	$1/9$	$2/9$	$1/9$	$3/9$	$1/9$	$1/9$

Một số chỉ số đánh giá chất lượng ảnh

Lượng thông tin của ảnh:

$$E = -\left(\frac{1}{9} * \log_2\left(\frac{1}{9}\right) + \frac{2}{9} * \log_2\left(\frac{2}{9}\right) + \frac{1}{9} * \log_2\left(\frac{1}{9}\right) + \frac{3}{9} * \log_2\left(\frac{3}{9}\right)\right. \\ \left. + \frac{1}{9} * \log_2\left(\frac{1}{9}\right) + \frac{1}{9} * \log_2\left(\frac{1}{9}\right)\right) = 2.4194.$$

Một số chỉ số đánh giá chất lượng ảnh

Cho một ảnh I có kích thước MxN:

Đô sắc nét của ảnh

$$G = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sqrt{(Gx(i,j))^2 + (Gy(i,j))^2}$$

Trong đó:

- $Gx(i, j) = I(i + 1, j) - I(i, j)$.
 - $Gy(i, j) = I(i, j + 1) - I(i, j)$.

Chú ý: Lệnh `[Gx, Gy] = imgradientxy(I,'intermediate');` dùng để tính giá trị đạo hàm bậc nhất của ma trận ảnh I theo hai hướng x,y trong matlab.

Một số chỉ số đánh giá chất lượng ảnh

Ví dụ: Cho ma trận 8 bit kích thước 3×3 . Tính độ sắc nét của ảnh.

255	102	51
51	153	204
153	0	153

Chuyển ma trận ảnh về miền $[0,1]$:

1	0.4	0.2
0.2	0.6	0.8
0.6	0	0.6

Một số chỉ số đánh giá chất lượng ảnh

Tính giá trị đạo hàm theo hai hướng G_x, G_y :

$$Gx = \begin{array}{|c|c|c|} \hline & -0.6 & -0.2 & 0 \\ \hline & 0.4 & 0.2 & 0 \\ \hline & -0.6 & 0.6 & 0 \\ \hline \end{array}$$

$$G_y = \begin{array}{|c|c|c|} \hline & -0.8 & 0.2 & 0.6 \\ \hline 0.4 & & -0.6 & -0.2 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Độ sắc nét của ảnh:

$$G = (\sqrt{(-0.6)^2 + (-0.8)^2} + \sqrt{(-0.2)^2 + (0.2)^2} + \sqrt{(0)^2 + (0.6)^2} \\ + \sqrt{(0.4)^2 + (0.4)^2} + \sqrt{(0.2)^2 + (-0.6)^2} + \sqrt{(0)^2 + (-0.2)^2} \\ + \sqrt{(-0.6)^2 + (0)^2} + \sqrt{(0.6)^2 + (0)^2} + \sqrt{(0)^2 + (0)^2})/9 \\ = 0.4979.$$

Kỹ thuật nâng cao chất lượng ảnh

Kỹ thuật trong miền không gian:

- Một số phép biến đổi cơ bản.
- Biến đổi tuyến tính từng phần.
- Cân bằng Histogram.
- Bộ lọc trung bình.
- Bộ lọc trung vị.
- Bộ lọc sắc nét.

Kỹ thuật nâng cao chất lượng ảnh

Toán tử điểm là phép biến đổi đối với từng điểm ảnh đang xét, không liên quan đến các điểm lân cận khác. Có hai cách tiệm cận với phương pháp này:

- Dùng một hàm biến đổi thích hợp với mục đích hoặc yêu cầu đặt ra để biến đổi giá trị mức xám của điểm ảnh sang một giá trị mức xám khác.
 - Dùng lược đồ mức xám (Gray Histogram).

Về mặt toán học, toán tử **điểm** là một ánh xạ từ giá trị cường độ ánh sáng $u(m,n)$ sang giá trị cường độ ánh sáng khác $v(m,n)$ thông qua hàm $f(\cdot)$, tức là: $v(m,n) = f(u(m,n))$.

Một số phép biến đổi

Một số phép biến đổi cường độ sáng cơ bản bao gồm:

- Phép biến đổi âm bản.
 - Phép biến đổi logarit.
 - Phép biến đổi lũy thừa.

Phép biến đổi tuyến tính từng phần:

- Kéo dãn độ tương phản.
 - Cắt theo mức cường độ sáng.

Xử lý Histogram:

- Cân bằng Histogram.
 - Cân bằng Histogram theo lược đồ có sẵn (Histogram matching).
 - Xử lý Histogram cục bộ.

Phép biến đổi âm bản

Phép biến đổi âm bản:

Với một ảnh có các mức xám nằm trong khoảng $[0, \dots, L-1]$ ta có:

$$s = L-1-r$$

Trong đó:

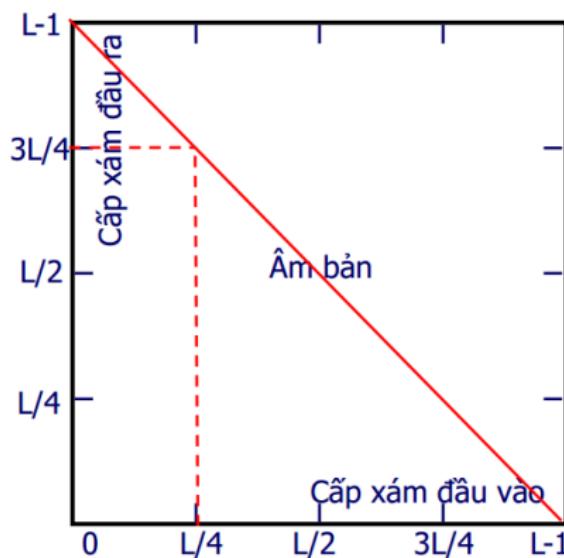
- r là các giá trị điểm ảnh trước khi xử lý.
 - s là các giá trị điểm ảnh sau khi xử lý.

Biến đổi âm bản:

- Làm nổi bật các chi tiết có màu sáng ở trong vùng tối.
 - Đặc biệt áp dụng đối với những ảnh có vùng tối lớn.

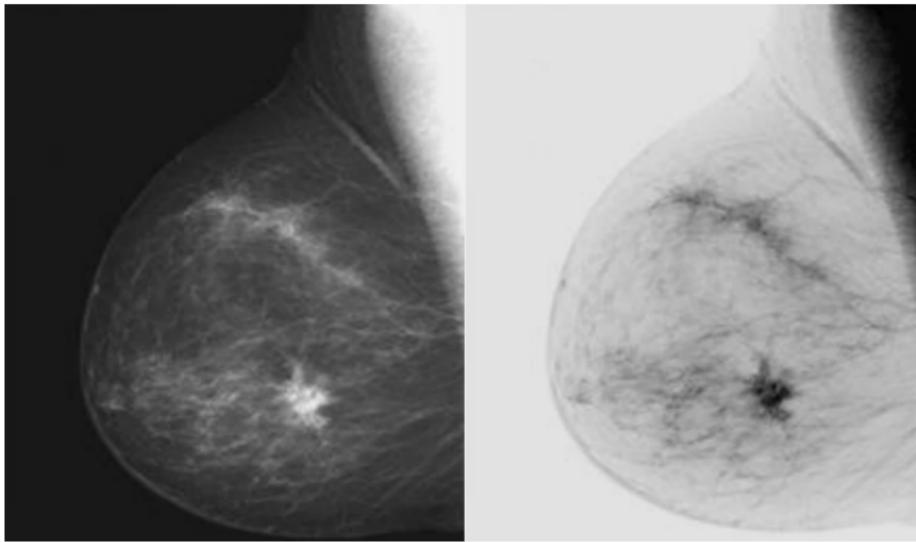
Phép biến đổi âm bản

Đồ thị của phép biến đổi âm bản:



Phép biến đổi âm bản

Ảnh trước và sau phép biến đổi âm bản:



Chú ý: Lệnh `I1 = imcomplement(I)` dùng để biến đổi ảnh âm bản

Phép biến đổi âm bản

Ví dụ: Cho ảnh xám đa cấp I với các cấp xám nằm từ [0,...,255].
Tìm ảnh âm bản của I.

25	26	45	18	230	229	210	237
15	2	25	214	240	253	230	41
18	154	14	201	237	101	241	54
19	254	13	201	236	1	242	54

Phép biến đổi Logarit

Phép biến đổi Logarit:

$$s = c^* \log(1+r)$$

Trong đó:

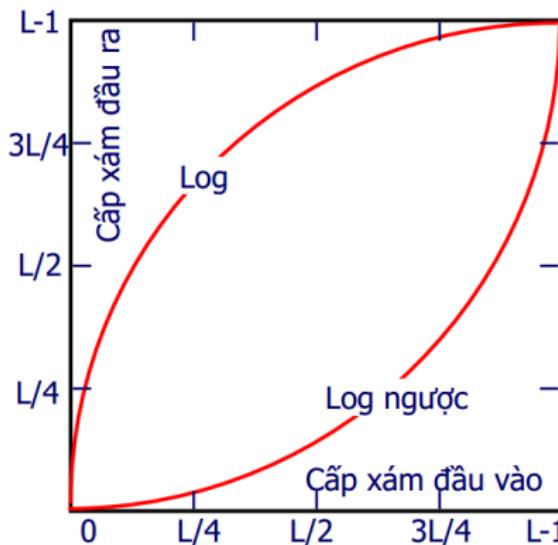
- r là các giá trị điểm ảnh trước khi xử lý.
 - s là các giá trị điểm ảnh sau khi xử lý.
 - c là một hằng số dương.

Biến đổi Logarit:

- Các giá trị mức xám thấp qua phép biến đổi sẽ tạo ra mức xám cao hơn.
 - Các giá trị mức xám cao sẽ nén lại thành mức xám thấp hơn.
 - Phép biến đổi này nhằm mục đích tăng chi tiết ở vùng tối.

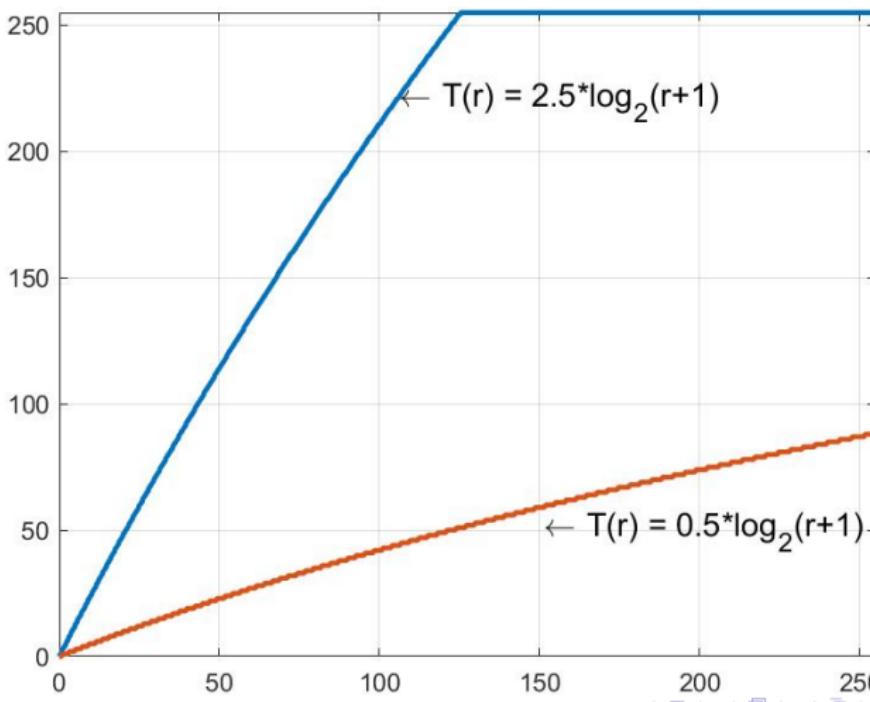
Phép biến đổi Logarit

Đồ thị của phép biến đổi logarit:



Phép biến đổi Logarit

Đồ thị của phép biến đổi logarit đối với $c>1$ và $0<c<1$:



Phép biến đổi Logarit

Nhận xét:

- Phép biến đổi Logarit ánh xạ một khoảng hẹp các giá trị cấp xám thấp trong ảnh đầu vào thành một khoảng rộng hơn các giá trị cấp xám của ảnh đầu ra.
- Ngược lại nó ánh xạ một khoảng rộng các giá trị cấp xám cao trong ảnh đầu vào thành một khoảng hẹp hơn các giá trị cấp xám của ảnh đầu ra.
- Phép biến đổi Logarit ngược ánh xạ một khoảng rộng các giá trị cấp xám thấp trong ảnh đầu vào thành một khoảng rộng hơn các giá trị cấp xám của ảnh đầu ra.
- Ngược lại nó ánh xạ một khoảng hẹp các giá trị cấp xám cao trong ảnh đầu vào thành một khoảng hẹp hơn các giá trị cấp xám của ảnh đầu ra.

Phép biến đổi Logarit

Ví dụ minh họa 1: Ảnh đầu vào quá tối.

(Biến đổi logarit kéo dãn các mức xám có giá trị nhỏ).



Ảnh trước và sau phép biến đổi logarit.

Phép biến đổi Logarit

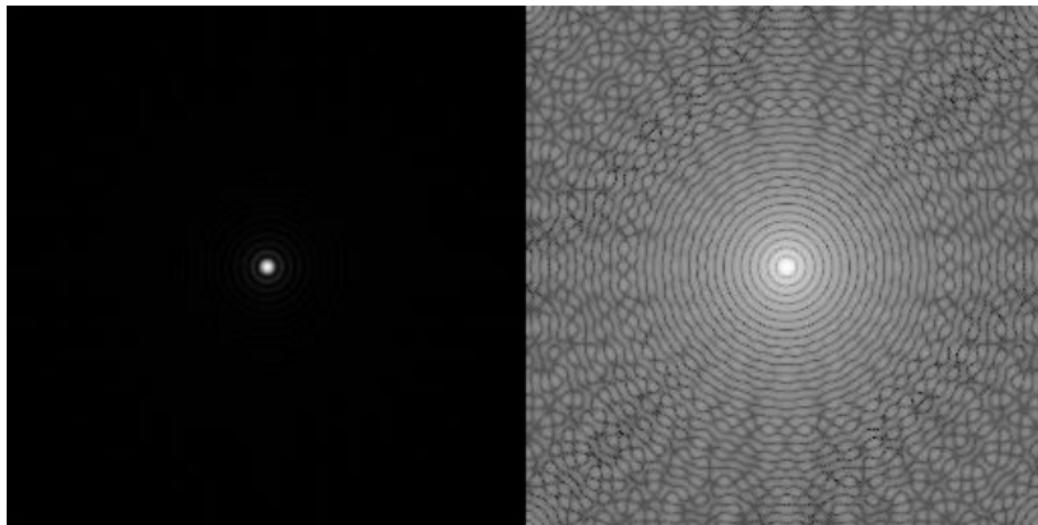
Ví dụ minh họa 2: Ảnh đầu vào quá sáng.
(Biến đổi logarit nén các mức xám có giá trị cao).



Ảnh trước và sau phép biến đổi logarit.

Phép biến đổi Logarit

Ví dụ minh họa 3: Ảnh đầu vào có khoảng giá trị rộng.
(Biến đổi logarit nén lại các mức xám có khoảng giá trị rộng).



Ảnh trước và sau phép biến đổi logarit.

Phép biến đổi Logarit

Ví dụ: Cho ảnh xám đa cấp I với các cấp xám nằm từ $[0, \dots, 255]$. Dùng biến đổi $s = 2 * \log(1 + r)$ để tìm ảnh đầu ra I.

$I =$	250	126	40
	39	10	240
	20	245	30

$$|1 = \begin{array}{|c|c|c|} \hline & 0.9804 & 0.4941 & 0.1569 \\ \hline & 0.1529 & 0.0392 & 0.9412 \\ \hline & 0.0784 & 0.9608 & 0.1176 \\ \hline \end{array}$$

$$|2 = \begin{array}{|c|c|c|} \hline & 1.3666 & 0.8031 & 0.2914 \\ \hline & 0.2846 & 0.0769 & 1.3266 \\ \hline & 0.1510 & 1.3467 & 0.2225 \\ \hline \end{array}$$

$I_3 =$	255	205	74
	73	20	255
	39	255	57

Phép biến đổi lũy thừa

Phép biến đổi Lũy thừa:

$$s = c * r^\gamma$$

Trong đó:

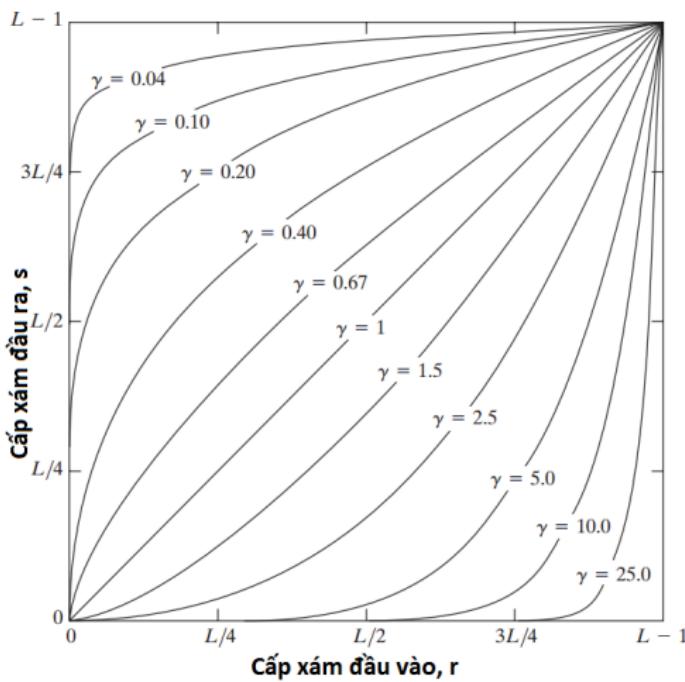
- r là các giá trị điểm ảnh trước khi xử lý.
- s là các giá trị điểm ảnh sau khi xử lý.
- γ (gamma) là một hằng số.

Biến đổi lũy thừa:

- Các giá trị mức xám thấp qua phép biến đổi sẽ tạo ra mức xám cao hơn.
- Các giá trị mức xám cao sẽ nén lại thành mức xám thấp hơn.
- Phép biến đổi này nhằm mục đích tăng chi tiết ở vùng tối.

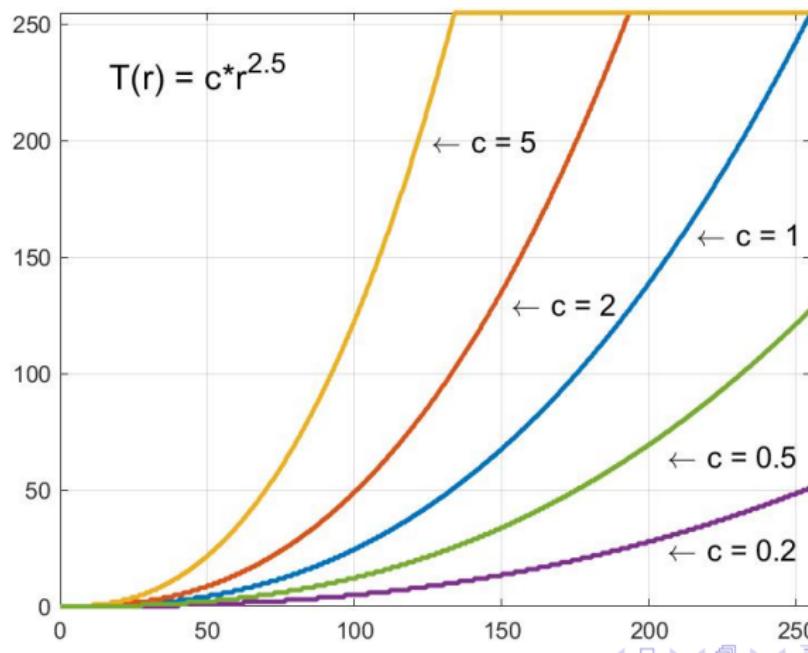
Phép biến đổi lũy thừa

Đồ thị của phép biến đổi lũy thừa:



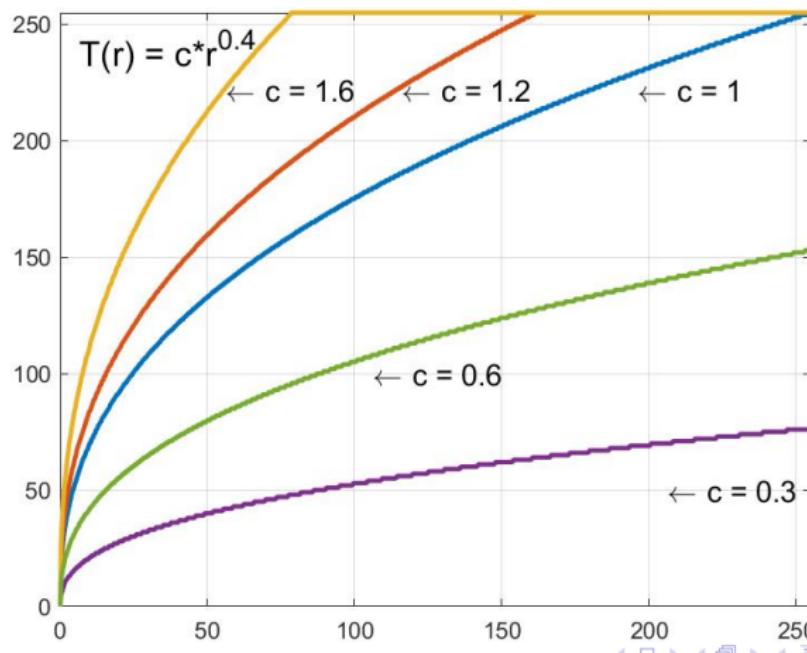
Phép biến đổi lũy thừa

Đồ thị của phép biến đổi lũy thừa $T(r) = c * r^{2.5}$ với các giá trị của c (0.2, 0.5, 1, 2 và 5):



Phép biến đổi lũy thừa

Đồ thị của phép biến đổi lũy thừa $T(r) = c * r^{0.4}$ với các giá trị của c (0.3, 0.6, 1, 1.2 và 1.6):



Phép biến đổi lũy thừa

Nhân xét:

- Với $\gamma < 1$ (nén gamma), các giá trị mức xám nhỏ qua phép biến đổi sẽ tạo ra các mức xám lớn hơn, trong khi đó các giá trị mức xám cao sẽ chuyển thành mức xám nhỏ hơn.
 - Với $\gamma = 1$, phép biến đổi là một hàm tuyến tính giữa đầu vào và đầu ra. Đặc biệt khi $c = \gamma = 1$, ảnh đầu ra và ảnh đầu vào là giống nhau.
 - Với $\gamma > 1$ (kéo dãn gamma), ta có phép biến đổi ngược so với $\gamma < 1$.

Phép biến đổi lũy thừa

Ví dụ minh họa 1: Ảnh quá sáng.

(Biến đổi hàm mũ thực hiện kéo dãn gamma, mức xám được nén lại).



Ảnh trước và sau phép biến đổi lũy thừa

Phép biến đổi lũy thừa

Ví dụ minh họa 2: Ảnh quá tối.

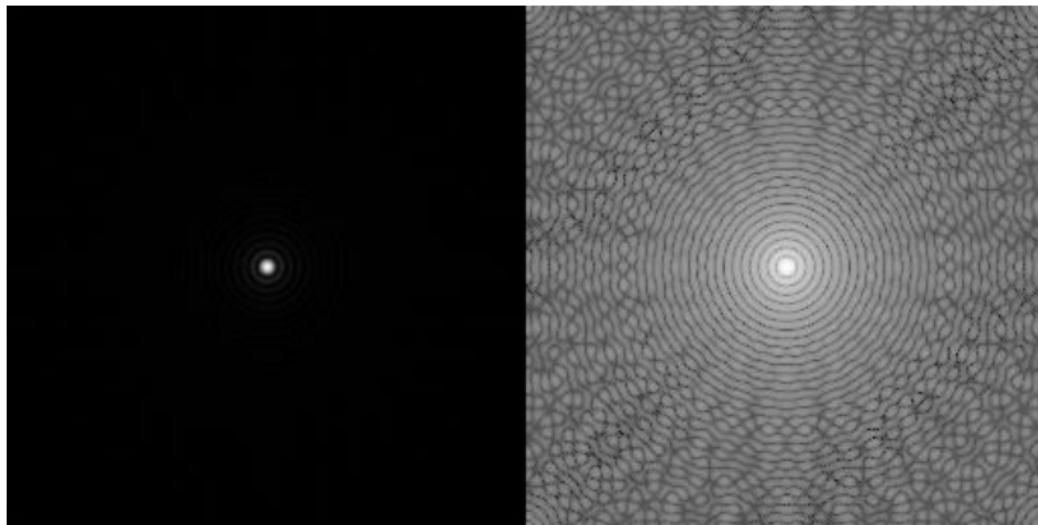
(Biến đổi hàm mũ thực hiện việc nén gamma, mức xám được kéo dãn).



Ảnh trước và sau phép biến đổi lũy thừa

Phép biến đổi lũy thừa

Minh họa trường hợp 3: Ảnh có khoảng xám rộng (thực hiện việc nén gamma).



Ảnh trước và sau phép biến đổi lũy thừa

Phép biến đổi lũy thừa

Ví dụ: Cho ảnh xám đa cấp I với các cấp xám nằm từ $[0, \dots, 255]$.
Dùng biến đổi $s = r^{0.3}$ để tìm ảnh đầu ra I.

$I =$	250	126	40
	39	10	240
	20	245	30

$$|1 = \begin{array}{|c|c|c|} \hline & 0.9804 & 0.4941 & 0.1569 \\ \hline & 0.1529 & 0.0392 & 0.9412 \\ \hline & 0.0784 & 0.9608 & 0.1176 \\ \hline \end{array}$$

$$l_2 = \begin{array}{|c|c|c|} \hline & 0.9941 & 0.8094 & 0.5737 \\ \hline & 0.5693 & 0.3785 & 0.9820 \\ \hline & 0.4660 & 0.9881 & 0.5262 \\ \hline \end{array}$$

$$I_3 = \begin{array}{|c|c|c|} \hline & 253 & 206 & 146 \\ \hline & 145 & 97 & 250 \\ \hline & 119 & 252 & 134 \\ \hline \end{array}$$

XỬ LÝ ẢNH

NÂNG CAO CHẤT LƯỢNG ẢNH

(MIỀN KHÔNG GIAN)

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

1 Tổng quan về nâng cao chất lượng ảnh

2 Biến đổi tuyến tính từng phần

Tổng quan về nâng cao chất lượng ảnh

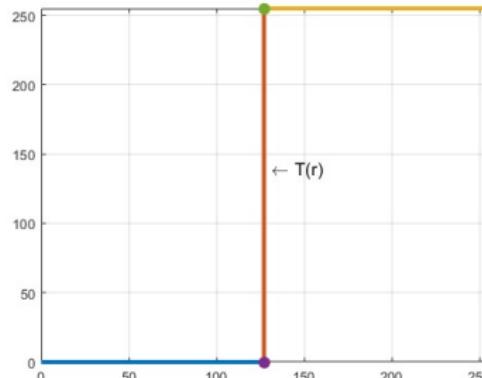
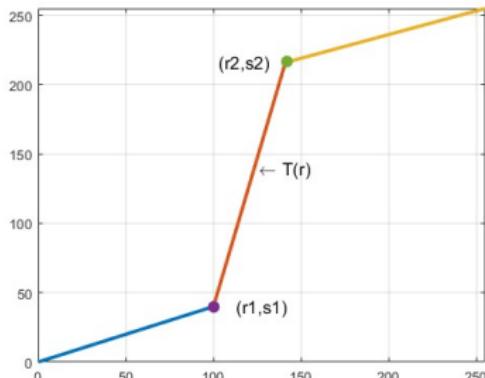
- Nâng cao chất lượng ảnh là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh.
- Nâng cao chất lượng ảnh gồm hai giai đoạn khác nhau: tăng cường ảnh và khôi phục ảnh.
- Mục đích nhằm hoàn thiện các đặc tính của ảnh như:
 - **Tăng độ tương phản, điều chỉnh mức xám của ảnh.**
 - Lọc nhiễu, hay làm trơn ảnh.
 - Làm nổi biên ảnh.

Biến đổi tuyến tính từng phần

- Kéo dãn độ tương phản.
 - Cắt theo mức cường độ.

Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Biến đổi tuyến tính từng phần là một kỹ thuật dùng để kéo dãn độ tương phản của ảnh.



Minh họa phép biến đổi tuyến tính từng phần

Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Vị trí của các điểm (r_1, s_1) và (r_2, s_2) có vai trò thay đổi hình dạng của hàm biến đổi.

- Nếu $r_1 = s_1$ và $r_2 = s_2$ thì việc biến đổi là hàm tuyến tính với mức xám đầu vào bằng chính mức xám đầu ra và do đó sẽ không tạo ra thay đổi cường độ sáng.
 - Nếu $r_1 = r_2, s_1 = 0$, và $s_2 = L - 1$ thì việc biến đổi là trở thành hàm ngưỡng, nó được ứng dụng để tạo ra hình ảnh nhị phân.

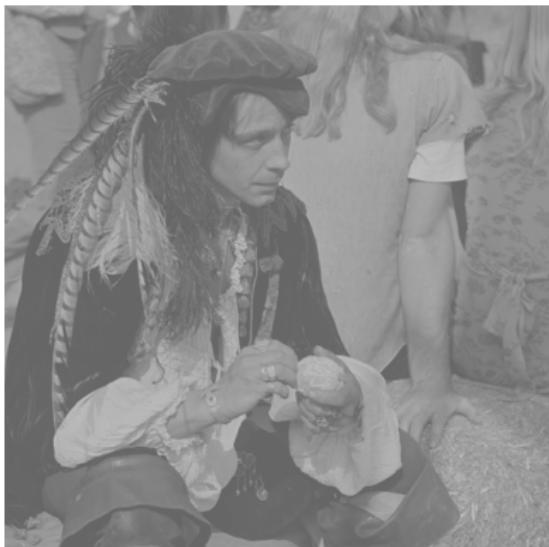
Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Kéo độ giãn tương phản bằng cách đặt $(r_1, s_1) = (r_{min}, 0)$ và $(r_2, s_2) = (r_{max}, L - 1)$ trong đó r_{min} và r_{max} biểu thị mức cường độ sáng tối thiểu và tối đa trong hình ảnh đầu vào tương ứng.



Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

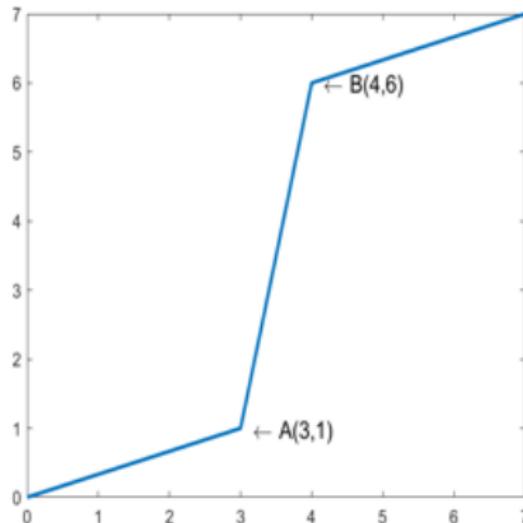
Tạo ảnh nhị phân bằng cách đặt $(r_1, s_1) = (m, 0)$ và $(r_2, s_2) = (m, L - 1)$ trong đó m là cường độ sáng trung bình trong ảnh.



Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Ví dụ: Cho ma trận ảnh I kích thước 5×5 và hai điểm A (3,1) và B(4,6). Hãy thực hiện biến đổi tuyến tính từng phần để kéo dãn độ tương phản cho ảnh I.

3	2	1	2	1
1	3	2	4	7
4	7	3	4	3
5	3	4	4	3
6	3	5	4	6



Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Với cường độ sáng nằm trong đoạn $[0, 3]$, xác định phương trình đường thẳng đoạn OA:

$$\frac{x - 0}{3 - 0} = \frac{y - 0}{1 - 0}$$

Như vậy:

$$y = \frac{x}{3}$$

Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Với cường độ sáng nằm trong đoạn [3, 4], xác định phương trình đường thẳng đoạn AB:

$$\frac{x - 3}{4 - 3} = \frac{y - 1}{6 - 1}$$

Như vậy:

$$y = 5x - 14$$

Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Với cường độ sáng nằm trong đoạn [4,7], xác định phương trình đường thẳng đoạn BC:

$$\frac{x - 4}{7 - 4} = \frac{y - 6}{7 - 6}$$

Như vậy:

$$y = \frac{x + 14}{3}$$

Biến đổi tuyến tính từng phần - Kéo dãn độ tương phản

Tóm lại ta có các cường độ sáng mới sẽ được tính theo công thức sau:

$$\begin{cases} y = \frac{x}{3} & x \in [0, 3] \\ y = 5x - 14 & x \in [3, 4] \\ y = \frac{x + 14}{3} & x \in [4, 7] \end{cases}$$

Áp dụng vào biến đổi cường độ sáng cho ảnh ta thu được ảnh mới như sau:

3	2	1	2	1
1	3	2	4	7
4	7	3	4	3
5	3	4	4	3
6	3	5	4	6

→

1	1	0	1	0
0	1	1	6	7
6	7	1	6	1
6	1	6	6	1
7	1	6	6	7

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

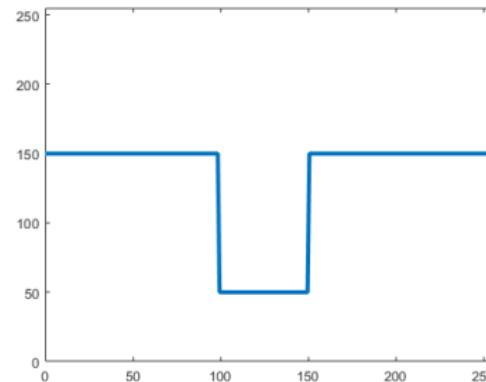
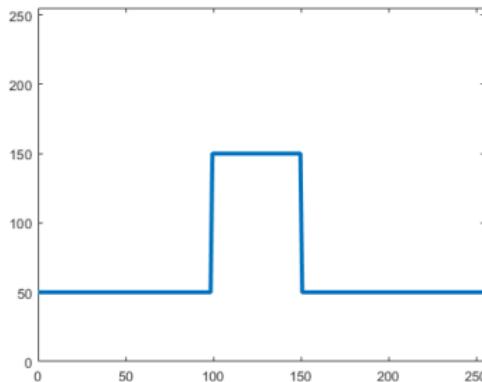
Là một phương pháp làm nổi bật một phạm vi cường độ sáng cụ thể trong một hình ảnh. Phương pháp này được ứng dụng trong việc phân đoạn các vùng mức xám nhất định từ phần còn lại của hình ảnh.



Hình minh họa cắt theo mức cường độ sáng

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

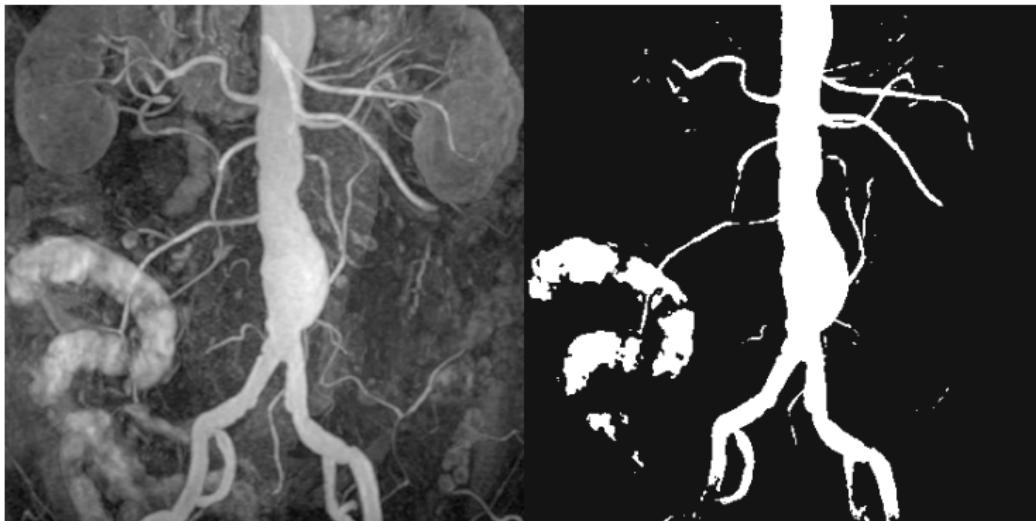
Loại 1: Hiển thị phạm vi cường độ sáng mong muốn bằng màu trắng và triệt tiêu tất cả các cường độ sáng khác thành màu đen hoặc ngược lại. Điều này dẫn đến một hình ảnh nhị phân.



Đồ thị minh họa trong trường hợp loại 1.

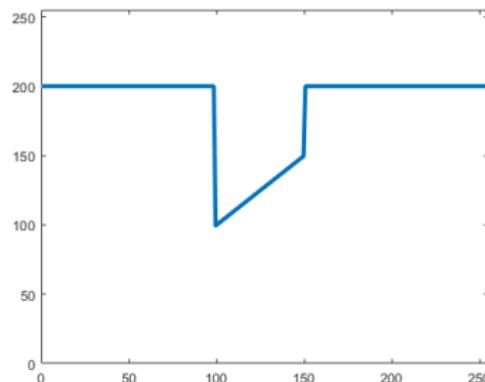
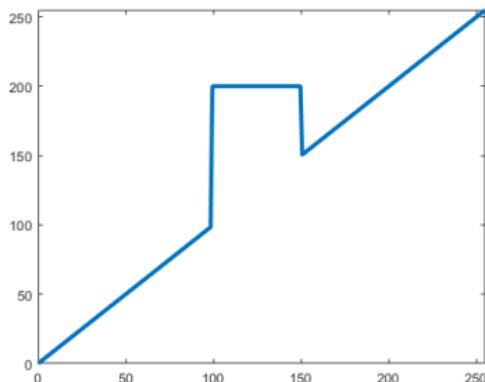
Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Minh họa áp dụng phương pháp cắt theo mức cường độ sáng loại 1 để làm nổi bật lên mạch máu chính biểu diễn bằng màu trắng và các thành phần còn lại được biểu diễn bằng màu đen.



Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

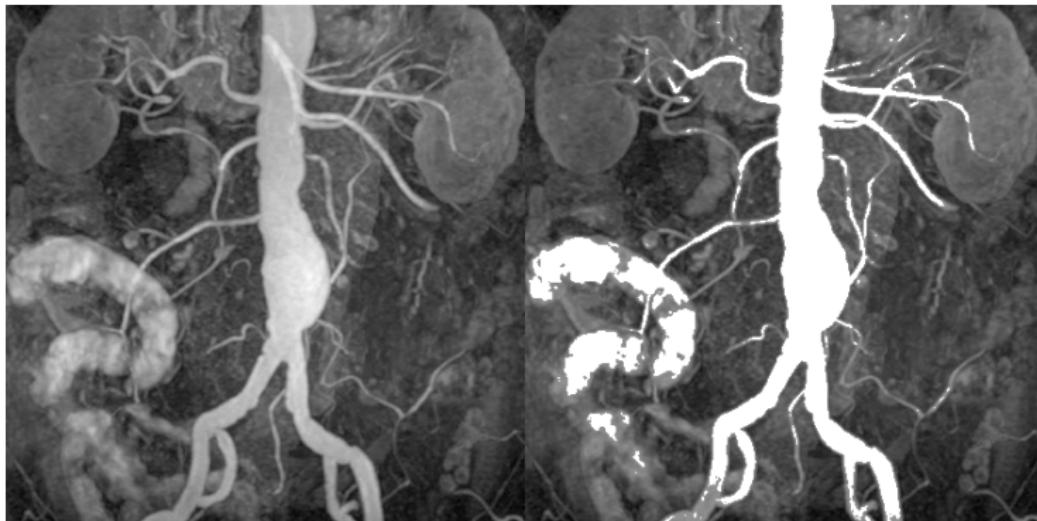
Loại 2: Làm sáng hoặc làm tối phạm vi cường độ sáng mong muốn và để các giá trị cường độ sáng khác không thay đổi hoặc ngược lại.



Đồ thị minh họa trong trường hợp loại 2.

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Minh họa áp dụng phương pháp cắt theo mức cường độ sáng loại 2 để làm nổi bật lên mạch máu chính màu biểu diễn bằng màu trắng và các thành phần còn lại được giữ nguyên như ảnh gốc.



Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Ví dụ 1: Cho ma trận ảnh I kích thước 5x5

1	2	5	2	3
4	3	6	5	3
6	5	4	6	6
1	4	4	7	4
3	5	6	7	2

Cắt theo mức cường độ theo công thức sau:

$$\begin{cases} y = 7 & x \in [4, 5] \\ y = 0 & x \in [0, 3] \quad \text{hoặc} \quad x \in [6, 7] \end{cases}$$

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Các giá trị cường độ sáng 4,5 được làm nổi bật (nhận giá trị tối đa là 7), các cường độ sáng còn lại được gán bằng 0. Tạo ra hình ảnh nhị phân.

1	2	5	2	3
4	3	6	5	3
6	5	4	6	6
1	4	4	7	4
3	5	6	7	2



0	0	7	0	0
7	0	0	7	0
0	7	7	0	0
0	7	7	0	7
0	7	0	0	0

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Ví dụ 2: Cho ma trận ảnh I kích thước 5x5

1	2	5	2	3
4	3	6	5	3
6	5	4	6	6
1	4	4	7	4
3	5	6	7	2

Cắt theo mức cường độ theo công thức sau:

$$\begin{cases} y = 7 & x \in [4, 5] \\ y = x & x \in [0, 3] \quad \text{hoặc} \quad x \in [6, 7] \end{cases}$$

Biến đổi tuyến tính từng phần - Cắt theo mức cường độ sáng

Các giá trị cường độ sáng 4, 5 được làm nổi bật (nhận giá trị tối đa là 7), các cường độ sáng còn lại được giữ nguyên.

1	2	5	2	3
4	3	6	5	3
6	5	4	6	6
1	4	4	7	4
3	5	6	7	2



1	2	7	2	3
7	3	6	7	3
6	7	7	6	6
1	7	7	7	7
3	7	6	7	2

XỬ LÝ ẢNH

NÂNG CAO CHẤT LƯỢNG ẢNH

(MIỀN KHÔNG GIAN)

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

1 Tổng quan về nâng cao chất lượng ảnh

2 Xử lý Histogram

Tổng quan về nâng cao chất lượng ảnh

- Nâng cao chất lượng ảnh là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh.
- Nâng cao chất lượng ảnh gồm hai giai đoạn khác nhau: tăng cường ảnh và khôi phục ảnh.
- Mục đích nhằm hoàn thiện các đặc tính của ảnh như:
 - Tăng độ tương phản, điều chỉnh mức xám của ảnh.
 - Lọc nhiễu, hay làm trơn ảnh.
 - Làm nổi biên ảnh.

Xử lý Histogram

- Cân bằng Histogram.
 - Khớp Histogram (Histogram Matching).
 - Xử Lý Histogram cục bộ.

Cân bằng Histogram

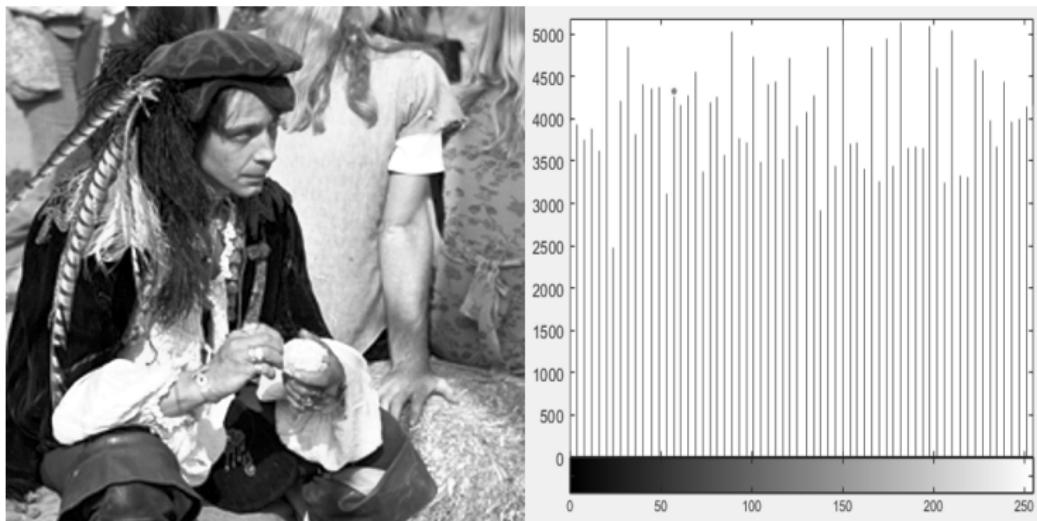
- Histogram của một ảnh với mức xám trong khoảng từ [0, L-1] là một hàm rời rạc $h(r_k) = n_k$, trong đó r_k là mức xám thứ k và n_k là số lượng pixel trong ảnh có mức xám r_k .
- Histogram là một đồ thị biểu diễn độ sáng của một bức ảnh với trục hoành là độ sáng và trục tung là số lượng điểm ảnh ở độ sáng tương ứng. Chiều cao của các cột trên đồ thị cũng thể hiện số lượng pixel ở mức sáng tương ứng.
- Histogram thường được chuẩn hóa. Với n là tổng số pixels của ảnh, histogram chuẩn hóa được tính qua biểu thức:

$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

- Có thể coi $p(r_k)$ là hàm mật độ xác suất của r_k , cho biết khả năng xuất hiện tương ứng của từng giá trị mức xám.

Cân bằng Histogram

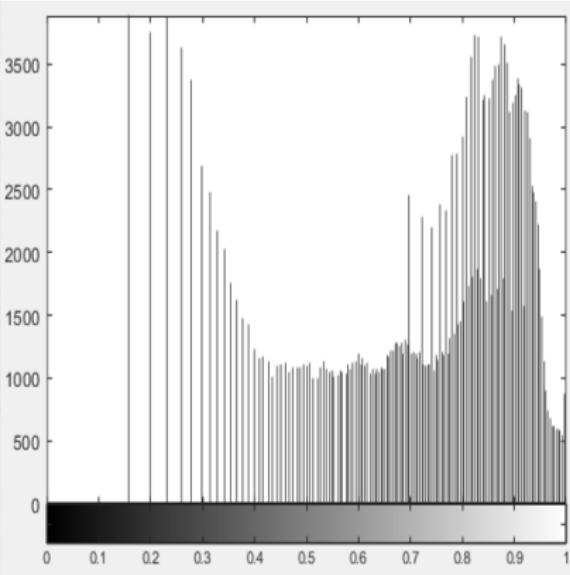
Ảnh và biểu đồ Histogram tương ứng:



Chú ý: Lệnh `imhist(I)` dùng để vẽ Histogram của ảnh trong matlab

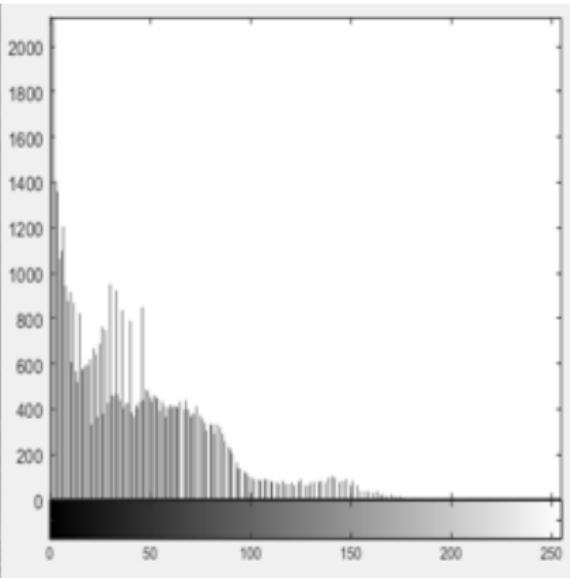
Cân bằng Histogram

Ảnh quá sáng và biểu đồ Histogram tương ứng:



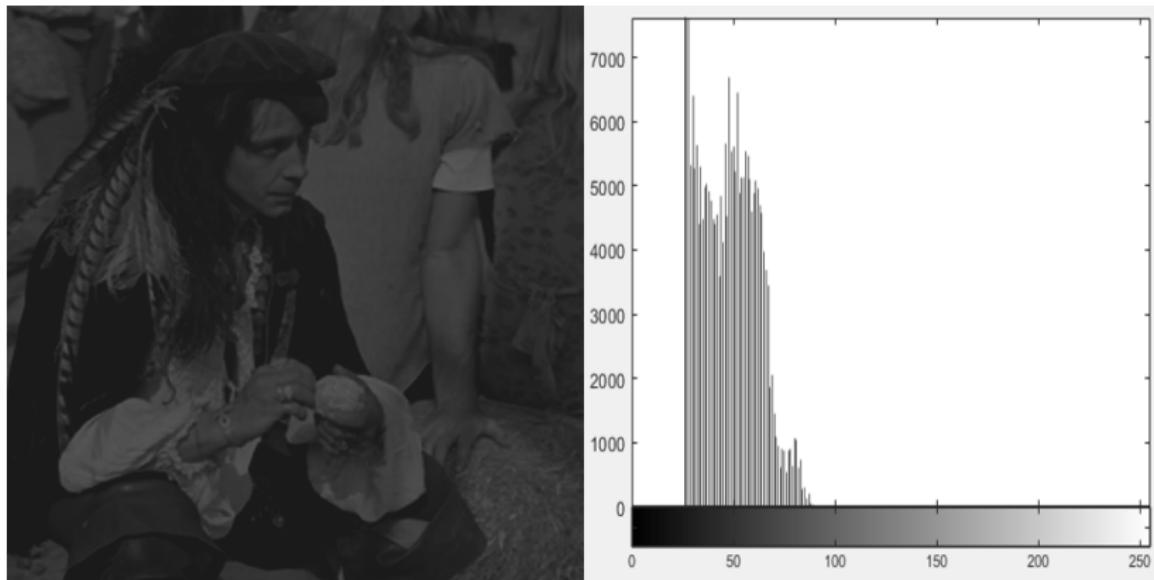
Cân bằng Histogram

Ảnh quá tối và biểu đồ Histogram tương ứng:



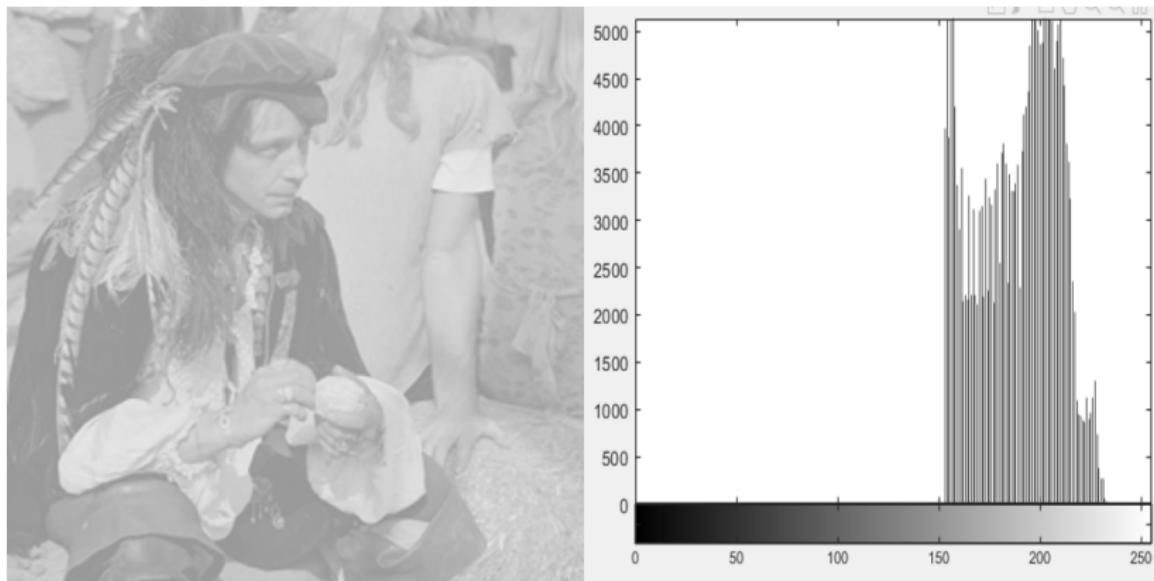
Cân bằng Histogram

Ảnh có độ tương phản thấp và biểu đồ Histogram tương ứng:



Cân bằng Histogram

Ảnh có độ tương phản thấp và biểu đồ Histogram tương ứng:



Cân bằng Histogram

- Mục đích cân bằng histogram là đưa ra một ảnh có mức xám được phân bố đồng đều.
- Kỹ thuật “cân bằng histogram” có thể cải thiện chất lượng ảnh tự động mà không cần làm thủ công với chức năng co/giãn mức xám.
- Ý tưởng:** Tối ưu hàm Entropy của ảnh để tìm hàm phân phối xác suất của các điểm ảnh. Từ đó xác định hàm biến đổi $g = T(f)$ áp dụng cho từng pixel của ảnh $f(x,y)$.

Cân bằng Histogram

Cho ảnh I có kích thước $M \times N$ có L mức xám. Giả sử các mức xám r_i là liên tục trong khoảng $[0, L-1]$. Gọi $p(r)$ và $p(s)$ lần lượt là các hàm phân phối xác suất của ảnh trước và sau khi cân bằng.

Hàm đo lượng thông tin của ảnh I (Entropy) được tính như sau:

$$E = - \int_0^{L-1} p(r) * \log_2(p(r)) dr = -\frac{1}{\ln(2)} \int_0^{L-1} p(r) * \ln(p(r)) dr$$

Với $p(r)$ là hàm phân phối xác suất của các điểm ảnh thỏa mãn điều kiện sau:

$$\int_0^{L-1} p(r) dr = 1$$

Cân bằng Histogram

Để tìm $p(s)$ cho ảnh sau khi cân bằng, áp dụng phương pháp nhân tử Lagrange để tối ưu cho hàm Entropy có ràng buộc điều kiện:

- **Bước 1:** Thêm một biến nhân tử Lagrange α và định nghĩa một hàm Lagrange $F(p(r), \alpha)$ như sau:

$$F(p(r), \alpha) = - \int_0^{L-1} p(r) * \ln(p(r)) dr + \alpha * \left(\int_0^{L-1} p(r) dr - 1 \right)$$

- **Bước 2:** Thực hiện tính đạo hàm riêng theo $p(r)$ và α ta thu được kết quả như sau:

$$\frac{\partial F(p(r), \alpha)}{\partial p(r)} = -(\ln(p(r)) + 1) + \alpha, \quad \frac{\partial F(p(r), \alpha)}{\partial \alpha} = \int_0^{L-1} p(r) dr - 1$$

Cân bằng Histogram

- Bước 3:** Giải các phương trình thu được ở bước 2, ta thu được các kết quả sau:

$$p(r) = e^{(\alpha-1)}$$

$$\alpha = 1 - \ln(L - 1)$$

Từ đó ta rút ra được giá trị của $p(r)$ tối ưu chính là $p(s)$:

$$p(s) = \frac{1}{L - 1}$$

Như vậy, hàm phân phối xác suất $p(s)$ của ảnh sau khi cân bằng chính là phân phối đồng nhất (Uniform distribution).

Cân bằng Histogram

Mặt khác, từ lý thuyết xác suất thống kê ta có:

$$p(r)dr = p(s)ds$$

$$\frac{ds}{dr} = \frac{p(r)}{p(s)} = \frac{p(r)}{(1/(L-1))}$$

Do $s = T(r)$ nên:

$$\frac{dT(r)}{dr} = (L-1) * p(r)$$

$$T(r_k) = (L-1) * \int_0^k p(r)dr$$

Cân bằng Histogram

Áp dụng trong trường hợp biến r_i là rìa rạc, ta có hàm chuyển đổi $T(r_k)$ xác định như sau:

$$T(r_k) = \text{round}((L - 1) * \sum_{i=0}^k p_i(r)) = \text{round}((L - 1) * cdf_k(p_k))$$

Trong đó $cdf_k(p_k)$ là hàm phân phối tích lũy (Cumulative distribution function) đến k theo các xác suất.

Cân bằng Histogram

Ta có thể áp dụng chuẩn hóa Min-max đối với hàm cdf_k , hàm $T(r_k)$ khi đó được viết lại như sau:

$$T(r_k) = \text{round}((L - 1) * \frac{cdf_k(p_k) - cdf_{min}(p)}{cdf_{max}(p) - cdf_{min}(p)})$$

Trong trường hợp tổng tích lũy cdf được tính theo số lần xuất hiện của các mức xám. Khi đó $T(r_k)$ cũng được viết lại như sau:

$$T(r_k) = \text{round}((L - 1) * \frac{cdf_k(n_k) - cdf_{min}(n)}{cdf_{max}(n) - cdf_{min}(n)})$$

Chú ý:

- $cdf_{max}(p) = 1$
- $cdf_{max}(n) = M \times N$

Cân bằng Histogram

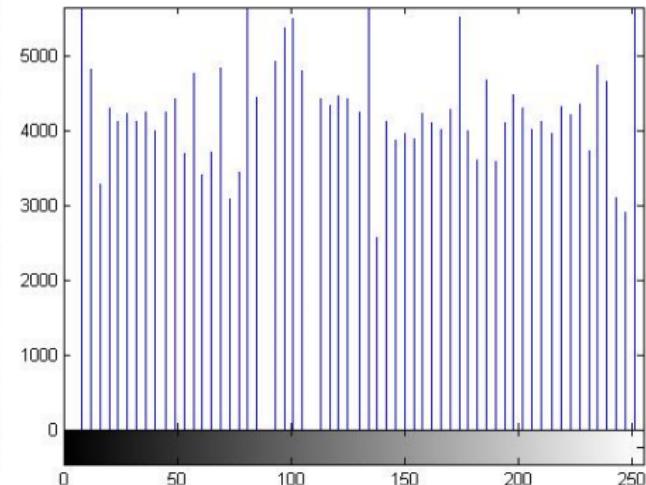
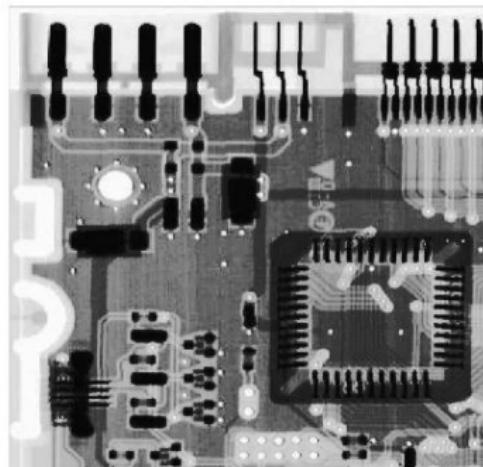
Cho ảnh I có kích thước MxN. Các bước thực hiện cân bằng Histogram cho ảnh I với số mức xám mới là L như sau:

- **Bước 1:** Thông kê được số lượng các pixel n_k tương ứng đối với từng mức xám r_k .
- **Bước 2:** Tính các hàm phân bố tích lũy cho (cdf) các điểm ảnh có mức xám nhỏ hơn hoặc bằng k: $cdf(r_k) = \sum_{i=0}^k n_i$.
- **Bước 3:** Tính giá trị các pixel mới s_k theo công thức sau:

$$s_k = round\left(\frac{cdf(r_k) - cdf_{min}}{M * N - cdf_{min}} * (L - 1)\right)$$

Cân bằng Histogram

Ảnh và đồ thị histogram sau khi cân bằng:



Chú ý: Lệnh $H = \text{histeq}(I)$ dùng để cân bằng histogram của ảnh trong matlab.

Cân bằng Histogram

Cho ảnh I được biểu diễn bởi ma trận như hình dưới. Hãy thực hiện cân bằng Histogram của ảnh I với số mức xám mới bằng 4.

1	2	4	6	7
2	1	3	4	5
7	2	6	9	1
4	1	2	1	2

Cân bằng Histogram

Bảng tính toán như sau:

r_k	1	2	3	4	5	6	7	9
n_k	5	5	1	3	1	2	2	1
cdf	5	10	11	14	15	17	19	20
s_k	0	1	1	2	2	2	3	3

Kết quả sau khi cân bằng Histogram:

1	2	4	6	7	0	1	2	2	3
2	1	3	4	5	1	0	1	2	2
7	2	6	9	1	3	1	2	3	0
4	1	2	1	2	2	0	1	0	1

Cân bằng Histogram

Bài tập: Cho ảnh I như hình vẽ. Hãy thực hiện:

- Vẽ đồ thị Histogram của ảnh I.
- Cân bằng Histogram của ảnh I với mức xám mới là 10.
- Vẽ đồ thị Histogram của ảnh I sau khi cân bằng.

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Cân bằng Histogram

Bảng tính toán như sau:

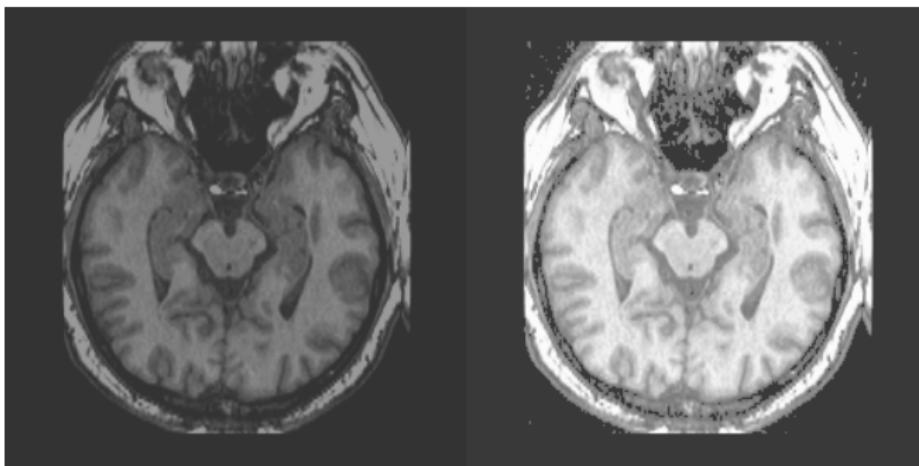
r_k	0	1	2	3	4	5	6	7	8	9
n_k	0	0	6	5	4	1	0	0	0	0
cdf	0	0	6	11	15	16	16	16	16	16
s_k	0	0	3	6	8	9	9	9	9	9

Kết quả sau khi cân bằng Histogram:

Trước khi cân bằng				Sau khi cân bằng			
2	3	3	2	3	6	6	3
4	2	4	3	8	3	8	6
3	2	3	5	6	3	6	9
2	4	2	4	3	8	3	8

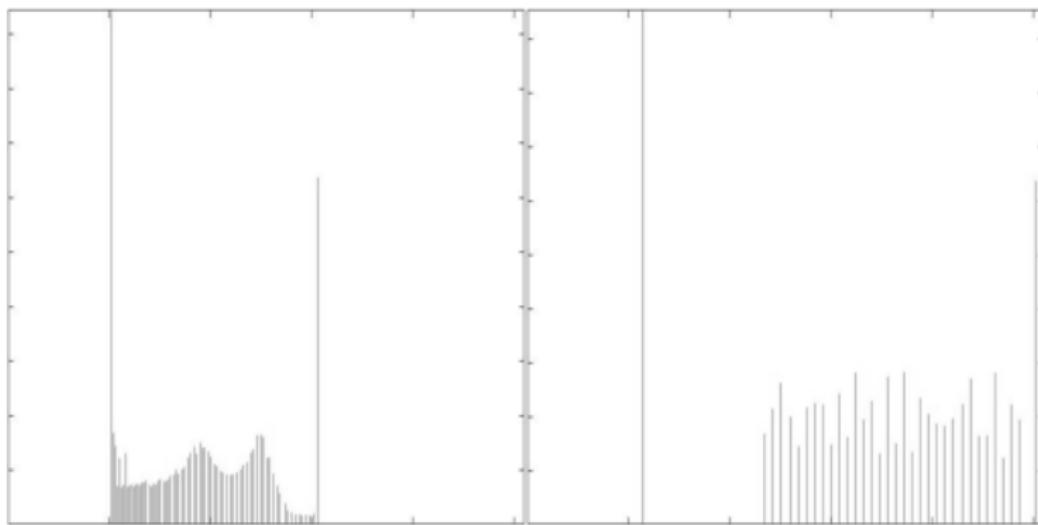
Histogram Matching

Cân bằng Histogram không phải lúc nào cũng hiệu quả trong mọi trường hợp. Quan sát hình ảnh sau khi áp dụng phương pháp cân bằng Histogram.



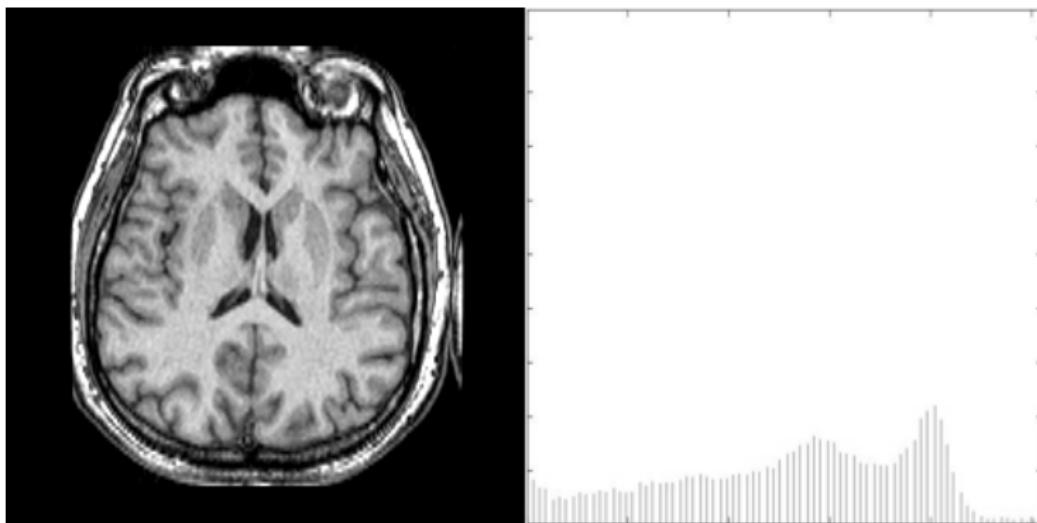
Hình ảnh trước và sau khi cân bằng Histogram

Histogram Matching



Biểu đồ Histogram của 2 ảnh trước và sau khi cân bằng

Histogram Matching



Ảnh và biểu đồ Histogram tương ứng.

Histogram Matching

Cho I_2 là ảnh tham chiếu và ảnh I_1 cần cân bằng giống với Histogram của I_2 . Gọi $p(r_i)$ là hàm phân phối xác suất các mức xám của ảnh I_1 và $p(z_i)$ là hàm phân phối xác suất các mức xám của ảnh I_2 , $r_i, z_i \in [0, L - 1]$. Mục tiêu cần tìm một hàm biến đổi X để $z_k = X(r_k)$.

Ý tưởng:

- Cân bằng Histogram của ảnh thứ nhất ($s_k = T(r_k)$)
- Cân bằng Histogram của ảnh thứ hai ($t_k = G(z_k)$)
- Do ($s_k = t_k$) nên ta có $z_k = G^{-1}(T(r_k))$

Histogram Matching

Các bước thực hiện cân bằng ảnh I_1 theo một biểu đồ Histogram của ảnh I_2 như sau.

Bước 1: Cân bằng Histogram cho ảnh I_1

$$s_k = T(r) = (L - 1) * \sum_{i=0}^{L-1} p_i(r)$$

Bước 2: Cân bằng Histogram cho ảnh I_2

$$t_k = G(z) = (L - 1) * \sum_{i=0}^{L-1} p_i(z)$$

Histogram Matching

Bước 3: Thực hiện xây dựng bảng ảnh xạ các mức xám mới bằng cách tìm các giá trị của j theo điều kiện sau:

$$G(z_j) - T(r_i) \geq 0$$

Bước 4: Dựa trên bảng ánh xạ tìm được ở bước 3, tiến hành cân bằng ma trận ảnh đầu vào.

Chú ý: Lệnh $J = imhistmatch(I, I_{tc})$ dùng để thực hiện Histogram matching trong Matlab. Trong đó:

- I là ảnh đầu vào.
- I_{tc} là ảnh tham chiếu.
- J là ảnh đầu ra.

Histogram Matching

Ví dụ: Cho 2 ảnh I_1 và I_2 được thống kê các mức xám theo bảng 1 và bảng 2. Hãy cân bằng ảnh I_1 theo một biểu đồ Histogram của ảnh I_2 .

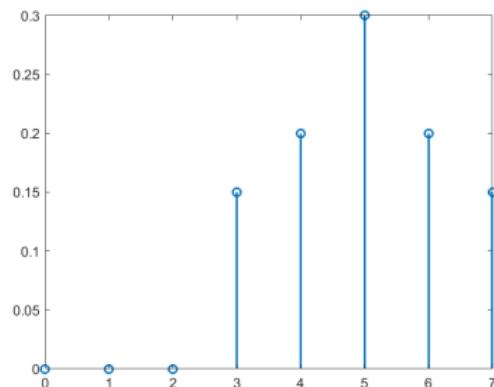
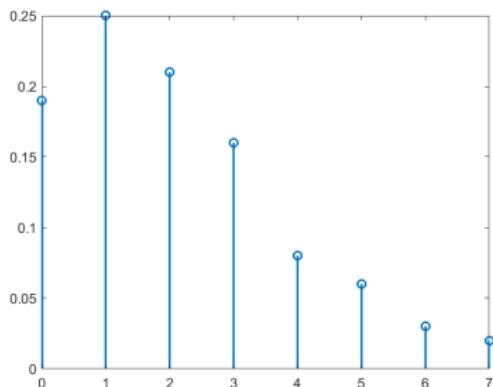
Bảng 1. Thống kê cho các mức xám trong ảnh I_1

r_i	0	1	2	3	4	5	6	7
$p(r_i)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

Bảng 2. Thống kê cho các mức xám trong ảnh I_2

z_j	0	1	2	3	4	5	6	7
$p(z_j)$	0.00	0.00	0.00	0.15	0.20	0.30	0.20	0.15

Histogram Matching



Biểu đồ Histogram của ảnh I_1 và ảnh I_2 .

Histogram Matching

Bước 1: Cân bằng Histogram cho ảnh thứ nhất (I_1)

$$H_1(r_k) = (L - 1) * cdf(r_k)$$

r_k	0	1	2	3	4	5	6	7
$p(r_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
$cdf(r_k)$	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
$H_1(r_k)$	1.33	3.08	4.55	5.67	6.23	6.65	6.68	7.00
$R(H_1(r_k))$	1	3	5	6	6	7	7	7

Histogram Matching

Bước 2: Cân bằng Histogram cho ảnh thứ hai (I_2)

$$H_2(z_k) = (L - 1) * cdf(z_k)$$

z_k	0	1	2	3	4	5	6	7
$p(z_k)$	0.00	0.00	0.00	0.15	0.20	0.30	0.20	0.15
$cdf(z_k)$	0.00	0.00	0.00	0.15	0.35	0.65	0.85	1.00
$H_2(z_k)$	0.00	0.00	0.00	1.05	2.45	4.55	5.95	7.00
$R(H_2(z_k))$	0	0	0	1	2	5	6	7

Histogram Matching

Bước 3: Xây dựng bảng ánh xạ các mức xám mới dựa trên điều kiện:

$$(H_2(z_j) - H_1(z_i) \geq 0)$$

Với $i = 0$, duyệt các giá trị của j từ 0 đến 7 để tìm j thỏa mãn điều kiện trên. Chọn được giá trị $j = 3$ thỏa mãn.

$$(H_2(z_3) - H_1(r_0) = 1 - 1 \geq 0)$$

Với $i = 1$, duyệt các giá trị của j từ 3 đến 7 để tìm j thỏa mãn điều kiện trên. Chọn được giá trị $j = 5$ thỏa mãn.

$$(H_2(z_5) - H_1(r_1) = 5 - 3 \geq 0)$$

Histogram Matching

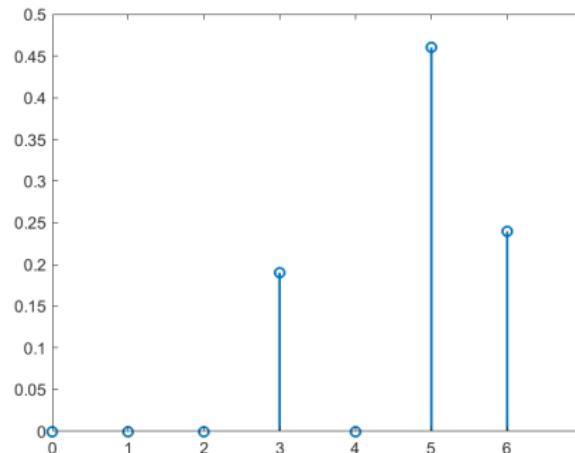
Làm tương tự cho các giá trị i còn lại, thu được bảng ánh xạ các mức xám như sau:

r_i	$H_1(r_i)$	z_j	$H_2(z_j)$	r_{new}
0	1	0	0	3
1	3	1	0	5
2	5	2	0	5
3	6	3	1	6
4	6	4	2	6
5	7	5	5	7
6	7	6	6	7
7	7	7	7	7

Histogram Matching

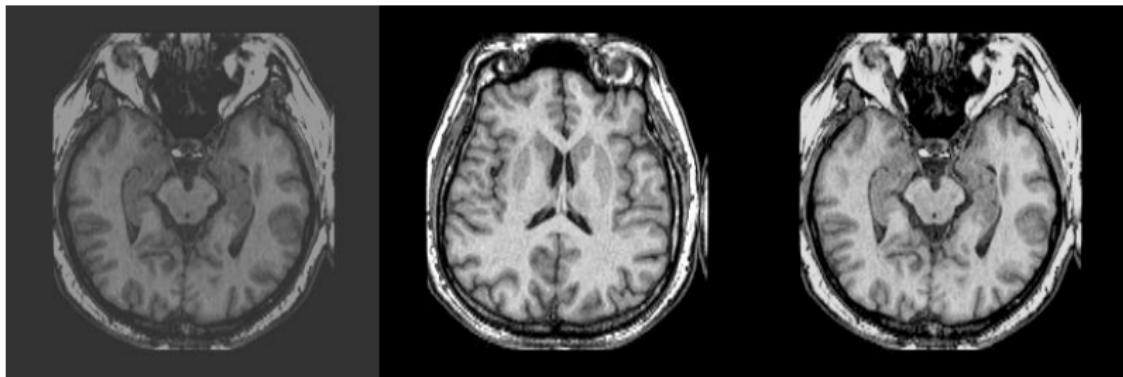
Bước 4: Bảng thống kê các mức xám của ảnh I_1 sau khi sử dụng Histogram matching như sau:

r	3	5	6	7
p(r)	0.19	0.46	0.24	0.11



Histogram của ảnh I_1 sau khi matching.

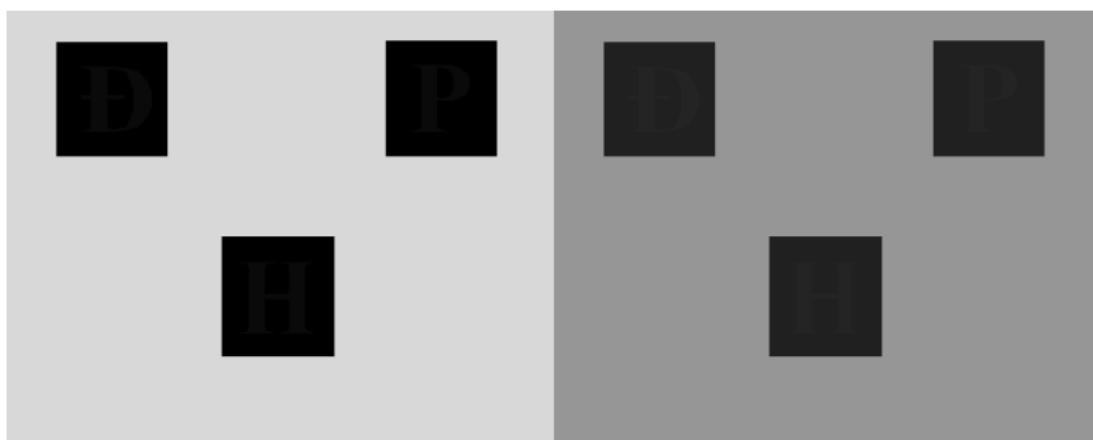
Histogram Matching



Minh họa việc sử dụng Histogram matching

Xử lý Histogram cục bộ

Các phương pháp cân bằng Histogram đã học thường không hiệu quả khi tăng cường chi tiết trên các vùng nhỏ trong ảnh.



Ảnh trước và sau cân bằng Histogram tương ứng.

Xử lý Histogram cục bộ

Điều này có thể được giải thích là do số lượng điểm ảnh trong các vùng nhỏ sẽ không ảnh hưởng đáng kể đến sự tính toán của các biến đổi toàn bộ ảnh.

Ý tưởng: Thực hiện hàm biến đổi dựa trên sự phân bố cường độ sáng các vùng lân cận của điểm ảnh. Sử dụng 2 chỉ số là cường độ sáng trung bình và độ tương phản của ảnh làm điều kiện để cập nhật lại cường độ sáng của mỗi điểm ảnh (x,y) nằm trong 1 vùng lân cận của điểm ảnh (x,y) .

Xử lý Histogram cục bộ

Cho một hình ảnh I , có các mức xám r_i nằm trong phạm vi $[0, L - 1]$ và đặt $p(r_i)$ là xác suất xuất hiện của cường độ sáng r_i trong ảnh.

Bước 1: Tính cường độ sáng trung bình và chỉ số tương phản của ảnh I .

- Cường độ sáng trung bình của ảnh I :

$$m = \sum_{i=0}^{L-1} r_i * p(r_i)$$

- Chỉ số tương phản của ảnh I (Phương sai của ảnh):

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 * p(r_i)$$

Xử lý Histogram cục bộ

Gọi S_{xy} là một vùng lân cận của tọa độ (x,y) có kích thước $U \times V$.

Gọi $p_s(r_i)$ là xác suất xuất hiện của mức xám r_i trong vùng S_{xy} .

Bước 2: Tính cường độ sáng trung bình và chỉ số tương phản của tất cả các vùng ảnh S_{xy} ($x = \overline{0, U - 1}$ và $y = \overline{0, V - 1}$).

- Cường độ sáng trung bình của vùng S_{xy} :

$$m_s = \sum_{i=0}^{L-1} r_i * p_s(r_i)$$

- Chỉ số tương phản của vùng S_{xy} (Phương sai của vùng S_{xy}):

$$\sigma_s^2 = \sum_{i=0}^{L-1} (r_i - m_s)^2 * p_s(r_i)$$

Xử lý Histogram cục bộ

Bước 3: Cập nhật lại cường độ sáng tại các vị trí (x,y).

Điều kiện cập nhật cường độ sáng tại các điểm (x, y) như sau:

$$\begin{cases} k_0m \leq m_s \leq k_1m \\ k_2\sigma^2 \leq \sigma_s^2 \leq k_3\sigma^2 \end{cases}$$

Khi đó cường độ sáng tại (x, y) được cập nhật theo công thức:

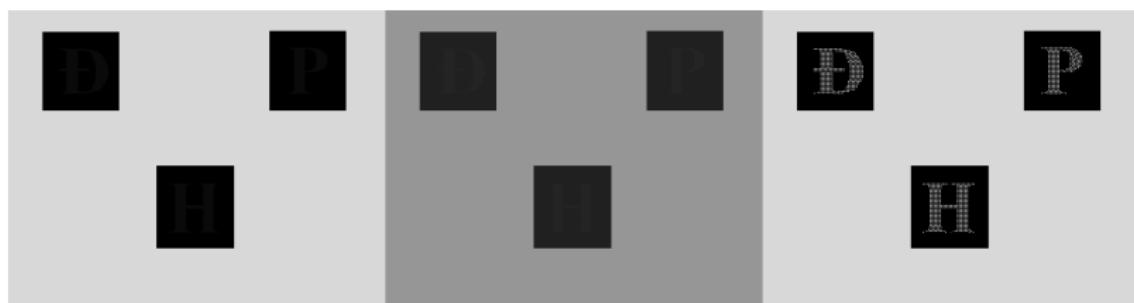
$$f(x, y) = \text{round}(C * f(x, y))$$

Trong đó, k_0, k_1, k_2, k_3, C là các hằng số.

Xử lý Histogram cục bộ

Thực hiện việc xử lý Histogram cục bộ, với vùng cục bộ lựa chọn có kích thước 3×3 và các tham số lựa chọn là:

$$k_0 = 0, k_1 = 0.25, k_2 = 0, k_3 = 0.1, C = 50$$



Ảnh trước và sau xử lý Histogram cục bộ.

Xử lý Histogram cục bộ

Ví dụ: Cho ma trận I kích thước 3×3 . Hãy xử lý Histogram cục bộ với vùng cục bộ kích thước 3×3 và các tham số $k_0 = 0, k_1 = 0.8, k_2 = 0, k_3 = 4, C = 1.8$

4	3	0
1	0	1
4	2	3

Xử lý Histogram cục bộ

Bảng thống kê các mức xám:

r_i	0	1	2	3	4
n_i	2	2	1	2	2
p_i	2/9	2/9	1/9	2/9	2/9

Bước 1: Tính cường độ sáng trung bình và phương sai của ảnh I:

- Cường độ sáng trung bình của ảnh I:

$$m = \sum_{i=0}^{L-1} r_i * p(r_i) = \frac{0 * 2 + 1 * 2 + 2 * 1 + 3 * 2 + 4 * 2}{9} = 2$$

.

Xử lý Histogram cục bộ

- Chỉ số tương phản của ảnh I (Phương sai của ảnh):

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 * p(r_i)$$

$$\begin{aligned}
 \sigma^2 &= (0-2)^2 * \frac{2}{9} + (1-2)^2 * \frac{2}{9} + (2-2)^2 * \frac{1}{9} + (3-2)^2 * \frac{2}{9} + (4-2)^2 * \frac{2}{9} \\
 &= \frac{8}{9} + \frac{2}{9} + \frac{0}{9} + \frac{2}{9} + \frac{8}{9} \\
 &= \frac{20}{9} = 2.2222
 \end{aligned}$$

Xử lý Histogram cục bộ

Bước 2: Tính cường độ sáng trung bình và chỉ số tương phản của tất cả các vùng $S(x, y)$.

Đặt cửa sổ kích thước 3×3 tại vị trí $(x, y) = (0, 0)$. Vùng cục bộ $S(0, 0)$ kích thước 3×3 thu được như sau:

0	0	0
0	4	3
0	1	0

Bảng thống kê tần số xuất hiện các mức xám trong vùng $S(0, 0)$

r_i	0	1	3	4
n_i	6	1	1	1
p_i	$6/9$	$1/9$	$1/9$	$1/9$

Xử lý Histogram cục bộ

- Cường độ sáng trung bình của vùng $S(0, 0)$:

$$m_s(0, 0) = \frac{0 * 6 + 1 * 1 + 3 * 1 + 4 * 1}{9} = \frac{8}{9} = 0.8889.$$

- Chỉ số tương phản của vùng $S(0, 0)$:

$$\begin{aligned}\sigma_s^2(0, 0) &= (0 - \frac{8}{9})^2 * \frac{6}{9} + (1 - \frac{8}{9})^2 * \frac{1}{9} + (3 - \frac{8}{9})^2 * \frac{1}{9} + (4 - \frac{8}{9})^2 * \frac{1}{9} \\ &= \frac{170}{81} = 2.0988\end{aligned}$$

Xử lý Histogram cục bộ

Bước 3: Kiểm tra điều kiện để cập nhật cường độ sáng tại $S(0, 0)$.

$$\begin{cases} k_0 m \leq m_s \leq k_1 m \\ k_2 \sigma^2 \leq \sigma_s^2 \leq k_3 \sigma^2 \end{cases}$$

$$\begin{cases} 0 * 2 \leq \frac{8}{9} \leq 0.8 * 2 \\ 0 * \frac{20}{9} \leq \frac{170}{81} \leq 4 * \frac{20}{9} \end{cases}$$

Nhận thấy điều kiện thỏa mãn, khi đó cập nhật lại cường độ sáng tại vị trí (0,0):

$$f(0, 0) = \text{round}(C * f(x, y)) = \text{round}(1.8 * 4) = 7$$

Xử lý Histogram cục bộ

Thực hiện tương tự tại tất cả các vị trí (x, y) còn lại ta thu được ma trận sau khi xử lý Histogram cục bộ như sau:

7	5	0
1	0	2
7	2	5

XỬ LÝ ẢNH

NÂNG CAO CHẤT LƯỢNG ẢNH

(MIỀN KHÔNG GIAN)

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa học Máy tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

- 1 Tổng quan về nâng cao chất lượng ảnh
- 2 Phép tương quan và phép nhân chập
- 3 Phân tách ma trận mặt nạ
- 4 Một số bộ lọc cơ bản

Tổng quan về nâng cao chất lượng ảnh

- Nâng cao chất lượng ảnh là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh.
- Nâng cao chất lượng ảnh gồm hai giai đoạn khác nhau: tăng cường ảnh và khôi phục ảnh.
- Mục đích nhằm hoàn thiện các đặc tính của ảnh như:
 - Tăng độ tương phản, điều chỉnh mức xám của ảnh.
 - **Lọc nhiễu, hay làm trơn ảnh (Đối với ảnh bị nhiễu).**
 - Làm nổi biên ảnh (Đối với ảnh không sắc nét, bị mờ).

Một số phương pháp lọc ảnh

Một số phương pháp lọc nhiễu (làm trơn ảnh) có thể phân loại như sau:

- Lọc tuyến tính
 - Lọc trung bình.
 - Lọc nhị thức.
 - Lọc Gaussian.
- Lọc phi tuyến.
 - Lọc min-max.
 - Lọc trung vị.

Phép tương quan và phép nhân chập

Cho ma trận ảnh I ($M \times N$), gọi K ($m \times n$) là một cửa sổ (mặt nạ) với ($m = 2a+1$, $n = 2b+1$). Khi đó phép tương quan (Correlation) giữa mặt nạ K và ma trận ảnh I được tính theo công thức sau:

$$C(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b K(i, j) * I(x + i, y + j)$$

- Tâm của ma trận mặt nạ trượt trên từng điểm ảnh của ma trận I, giá trị tương quan tại từng điểm được tính bằng tổng của các tích giữa hệ số của mặt nạ K và giá trị của điểm ảnh I tại vị trí tương ứng.
 - Thường được sử dụng để khớp 1 mẫu cho trước với 1 vùng trong 1 ảnh.

Phép tương quan và phép nhân chập

Ví dụ 1: Cho ma trận ảnh I kích thước 4×4 và ma trận mặt nạ K kích thước 3×3 . Hãy tính tương quan giữa K và I.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	1	1
1	1	1
1	1	1

Ma trận ảnh I và ma trận mặt nạ K

Kết quả thực hiện phép tương quan:

14	24	30	22
33	54	63	45
57	90	99	69
46	72	78	54

Chú ý: Lệnh `xcorr2(I,K)` để tính tương quan giữa hai ma trận.

Phép tương quan và phép nhân chập

Ví dụ 2: Cho ma trận ảnh I kích thước 5×5 và ma trận mặt nạ kích thước 3×3 . Hãy tìm vùng trong ảnh I khớp với mẫu K nhất.

2	3	4	6	7
6	7	5	5	3
4	5	1	2	1
2	4	2	1	2
1	3	1	1	2

4	7	7
5	5	3
1	2	2

Tính ma trận tương quan giữa mặt nạ K và ma trận ảnh I:

45	67	80	92	76
104	153	168	173	117
138	170	144	102	61
93	103	81	53	35
56	73	60	45	33

Phép tương quan và phép nhân chập

Ví dụ 3: Cho ma trận ảnh I kích thước 5×5 và ma trận mặt nạ K kích thước 3×3 . Hãy tìm vùng trong ảnh I khớp với mẫu K nhất.

2	3	4	6	7
6	7	5	5	3
4	5	1	2	1
2	4	2	1	2
1	3	1	1	2

1	2	1
2	2	2
1	1	2

Phép tương quan và phép nhân chập

Tính ma trận tương quan giữa mặt nạ K và ma trận ảnh I:

30	41	48	50	34
47	59	61	54	39
47	55	46	33	20
32	37	29	22	13
16	22	19	14	11

Nhận xét:

- Kết quả tìm được không chính xác. Lý do là khi vùng ảnh chứa các pixel có giá trị cao thì phép tương quan cũng cao.
- Cần phải chuẩn hóa giá trị tương quan.

Phép tương quan và phép nhân chập

Công thức tính tương quan theo Cosin:

$$C(x, y) = \frac{\sum_{i=-a}^a \sum_{j=-b}^b K(i, j) * I(x + i, y + j)}{\sqrt{\sum_{i=-a}^a \sum_{j=-b}^b K^2(i, j)} * \sqrt{\sum_{i=-a}^a \sum_{j=-b}^b I^2(x + i, y + j)}}$$

Áp dụng tính tương quan giữa ma trận K và ảnh I:

0.62	0.71	0.77	0.81	0.64
0.81	0.9	0.9	0.86	0.71
0.79	0.85	0.77	0.78	0.62
0.78	0.86	0.86	0.98	0.69
0.6	0.76	0.69	0.74	0.71

Phép tương quan và phép nhân chập

Cho ma trận ảnh I ($M \times N$), gọi K ($m \times n$) là một cửa sổ (mặt nạ) với ($m = 2a+1$, $n = 2b+1$). Khi đó phép nhân chập (Convolution) giữa ma trận ảnh I và mặt nạ K được tính theo công thức sau:

$$C(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b K(i, j) * I(x - i, y - j)$$

- Mặt nạ K được xoay 180 độ trước khi thực hiện.
- Thực hiện cách tính như phép tương quan.
- Thường được sử dụng để lọc ảnh.

Phép nhân chập

Ví dụ: Cho ma trận ảnh I và ma trận mặt nạ K như hình dưới. Hãy thực hiện tính nhân chập giữa ma trận I và K ($I \otimes K$).

Ma trận ảnh I

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Ma trận K

1	1	1
1	1	1
1	1	1

Phép tương quan và phép nhân chập

Kết quả phép nhân chập ma trận ảnh I và ma trận mặt nạ K như hình dưới:

14	24	30	22
33	54	63	45
57	90	99	69
46	72	78	54

Chú ý: Dùng lệnh `C = conv2(I, K, 'same')` để thực hiện nhân chập trong matlab

Phép nhân chập

Một số tính chất của phép nhân chập và phép tương quan:

Tính chất	Phép nhân chập	Phép tương quan
Giao hoán	$K \otimes I = I \otimes K$	Không có
Kết hợp	$K_1 \otimes (K_2 \otimes I) = (K_1 \otimes K_2) \otimes I$	Không có
Phân phôi	$K \otimes (I_1 + I_2) = K \otimes I_1 + K \otimes I_2$	$K \otimes (I_1 + I_2) = K \otimes I_1 + K \otimes I_2$

Phân tách ma trận mặt nạ

Một ma trận mặt nạ K kích thước $m \times n$ có thể phân tách được nếu nó biểu diễn được dưới dạng tích của 2 ma trận K_1 (kích thước $m \times 1$) và K_2 (kích thước $1 \times n$) như sau:

$$K = K_1 * K_2$$

Ví dụ: Ma trận K :

1	1	1
1	1	1
1	1	1

được phân tách thành 2 ma trận K_1 và K_2 như sau:

$$K_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{và} \quad K_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Phân tách ma trận mặt nạ

Với một ảnh I có kích thước $M \times N$, và một ma trận mặt nạ K có kích thước $m \times n$.

- Để thực hiện phép nhân chập I và K cần: $M*N*m*n$ phép tính.
- Nếu ma trận K được phân tách thành 2 ma trận K_1 và K_2 thì phép nhân chập giữa I và K_1 cần $M*N*m$ phép tính, và phép nhân chập giữa I và K_2 cần $M*N*n$ phép tính. Như vậy phép nhân chập giữa I và K cần $M*N*(m+n)$ phép tính.

Do đó, ưu thế tính toán trong trường hợp ma trận mặt nạ có thể phân tách được so với không phân tách được có thể biểu diễn theo công thức sau:

$$C = \frac{M * N * m * n}{M * N * (m + n)} = \frac{m * n}{m + n}$$

Bộ lọc không gian làm mịn

Bản chất chính của các thao tác lọc ảnh là việc thực hiện các phép tính nhân chập giữa ma trận ảnh và mặt nạ lọc.

- **Ý tưởng chính:** Thay thế giá trị mỗi pixel trong ảnh bằng trung bình các mức xám lân cận khi xử lý bằng mặt nạ lọc. Kết quả giảm độ sắc nét trong ảnh.
- **Nhược điểm:** Các cạnh (hầu hết là dấu hiệu không thể thiếu trong ảnh) cũng bị làm mờ đi, đây là 1 điều không mong muốn xảy ra.

Bộ lọc trung bình

Mặt nạ lọc trung bình (mặt nạ hộp) là các ma trận mặt nạ thường được biểu diễn là trung bình của các ma trận đơn vị lẻ. Ví dụ như:

$$K = 1/9 * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Chú ý: Dùng lệnh `K = fspecial('average',[m,n])` để tạo mặt nạ lọc trung bình kích thước mxn trong matlab. Mặc định là m = n = 3.

Bộ lọc trung bình

Tổng quát, với ảnh I kích thước $M \times N$, mặt nạ K kích thước $m \times n$:

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b K(s, t)I(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b K(s, t)}$$

- m và n lẻ, $a = (m-1)/2$, $b = (n-1)/2$.

Bộ lọc trung bình

Ví dụ: Cho ảnh I có ma trận ảnh như sau. Hãy thực hiện lọc trung bình với ma trận mặt nạ 3x3.

131	137	141	60	90	60
139	133	133	90	135	89
138	132	127	60	89	58

Chú ý: Dùng lệnh $B = imboxfilt(I,n,'Padding',0)$, hoặc $B = imfilter(I,K)$, hoặc $B = conv2(I,K,'same')$ để lọc trung bình trong matlab. Với I là ảnh cần lọc, K là mặt nạ trung bình, n là kích thước mặt nạ (lé).

Bộ lọc trung bình

Ví dụ: Ảnh có nhiễu được lọc với bộ lọc trung bình 7x7:



Chú ý: Dùng lệnh $K = \text{imnoise}(I, \text{'salt \& pepper'}, 0.02)$ để tạo nhiễu muối tiêu cho ảnh I.

Bộ lọc nhị thức (Binomial filter)

Một bộ lọc nhị thức 1 chiều đơn giản nhất có thể làm mịn là trung bình cộng của 2 hàng xóm lân cận:

$$K = \frac{1}{2} * [1 \ 1]$$

Chúng ta có thể áp dụng bộ lọc này n lần theo hàng:

$$K_n = \frac{1}{2^n} * [1 \ 1] * [1 \ 1] * \dots * [1 \ 1]$$

Một số mặt nạ lọc trong trường hợp cụ thể:

$$K_2 = \frac{1}{4} * [1 \ 2 \ 1], \quad K_3 = \frac{1}{8} * [1 \ 3 \ 3 \ 1], \quad \dots$$

Bộ lọc nhị thức

Bộ lọc nhị thức 2 chiều kích thước bằng 3×3 như sau:

$$B = \frac{1}{2} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \frac{1}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

Kết quả áp dụng bộ lọc Nhị thức 3×3 :

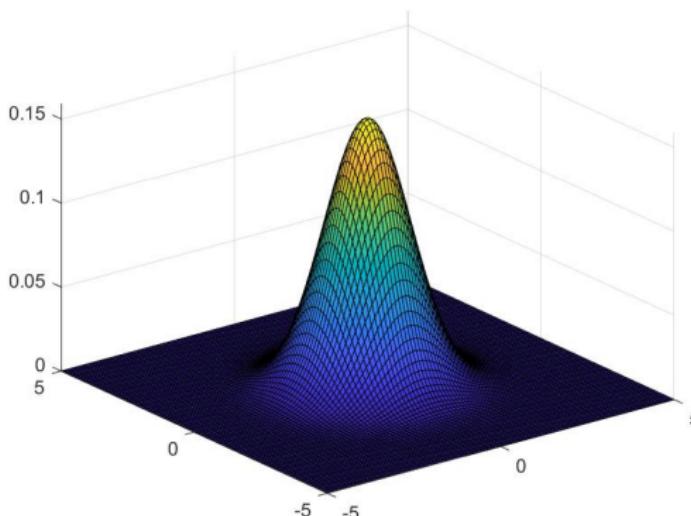


Bộ lọc Gaussian

Hàm Gaussian 2 chiều với trung bình $(0,0)$ và độ lệch chuẩn bằng σ có công thức sau:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Đồ thị Gaussian có hình dạng sau:



Bộ lọc Gaussian

Áp dụng hàm Gaussian với $\sigma = 1$ để tính mặt nạ kích thước 5x5:

0.0029	0.0131	0.0215	0.0131	0.0029
0.0131	0.0585	0.0965	0.0585	0.0131
0.0215	0.0965	0.1592	0.0965	0.0215
0.0131	0.0585	0.0965	0.0585	0.0131
0.0029	0.0131	0.0215	0.0131	0.0029

Chuẩn hóa bằng cách chia các phần tử cho tổng các phần tử:

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0596
0.0030	0.0133	0.0219	0.0133	0.0030

Chú ý: Dùng lệnh **K = fspecial('gaussian',5,1)** trong Matlab.

Bộ lọc Gaussian

Ảnh trước và sau khi được khử nhiễu bằng mặt nạ Gaussian kích thước 5×5 và $\sigma = 3$:



Bộ lọc không gian làm mịn

Bộ lọc trung vị:

- Thay thế giá trị của pixel bằng median của các mức xám lân cận (giá trị ban đầu của pixel đó bao gồm cả tính toán giá trị median).
- Lọc median khá phổ biến giảm nhiễu khá tốt mà ảnh ít bị mờ hơn bộ lọc mịn với mặt nạ cùng kích thước.
- Lọc median còn đặc biệt hiệu quả đối với việc giảm nhiễu muối tiêu.
- Median m của 1 tập hợp là 1 nửa các giá trị trong tập hợp đó lớn hơn hoặc bằng m , nửa còn nhỏ hơn hoặc bằng m .

Bộ lọc không gian làm mịn

Bộ lọc trung vị:

- Để biểu diễn bộ lọc median của 1 điểm trong ảnh, đầu tiên phải sắp xếp các giá trị pixel và lân cận của nó, chọn median, và thay giá trị này cho pixel đó.
- Ví dụ: 1 mặt nạ 3x3 có các giá trị điểm lân cận: 10, 20, 20, 20, 15, 20, 20, 25, 100. Sắp xếp lại các giá trị: 10, 15, 20, 20, 20, 20, 20, 25, 100. Median = 20.
- Mục đích: bộ lọc này dùng để làm cho các pixel có giá trị sai khác sẽ giống với các giá trị hàng xóm hơn.

Bộ lọc không gian làm mịn

Bộ lọc trung vị:

Ví dụ: Cho ảnh I có ma trận ảnh như sau. Hãy thực hiện lọc trung vị với ma trận mặt nạ 3x3.

131	137	141	0	133	0
139	133	133	132	133	132
138	132	127	0	132	0

Bộ lọc không gian làm mịn

Ví dụ: Ảnh có nhiễu được lọc với bộ lọc trung vị 3×3 :



Chú ý: Có thể dùng lệnh $Tv = medfilt2(I)$ trong matlab để thực hiện lọc trung vị với ảnh I (mặc định là mặt nạ 3×3)

XỬ LÝ ẢNH

MỘT SỐ PHƯƠNG PHÁP DÒ BIÊN ẢNH

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa học Máy tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

- 1 Tổng quan về dò biên ảnh
- 2 Phân loại kỹ thuật dò biên
- 3 Một số phương pháp dò biên
- 4 Bộ lọc không gian sắc nét

Tổng quan về dò biên ảnh

Khái niệm về biên

- Biên là một vấn đề chủ yếu trong phân tích ảnh vì các kỹ thuật phân đoạn ảnh chủ yếu dựa vào biên.
 - Một điểm ảnh có thể coi là điểm biên nếu ở đó có sự thay đổi đột ngột về mức xám. Tập hợp các điểm biên tạo thành biên hay đường bao ảnh của ảnh .
 - Về mặt toán học người ta coi điểm biên của ảnh là điểm có sự biến đổi đột ngột về độ xám.
 - Nếu cường độ sáng các điểm ảnh lân cận xấp xỉ cường độ sáng điểm ảnh trung tâm, thì điểm ảnh đó không là điểm thuộc biên.

Phân loại kỹ thuật dò biển

Người ta thường sử dụng 2 phương pháp dò biên sau:

- Phương pháp dò biên trực tiếp.
 - Phương pháp dò biên gián tiếp.

Phương pháp dò biên trực tiếp

- Làm nổi biên dựa vào sự biến thiên về giá trị độ sáng của điểm ảnh.
- Nếu lấy đạo hàm bậc nhất ta có phương pháp Gradient.
- Nếu lấy đạo hàm bậc hai ta có phương pháp Laplace.
- Phương pháp dò biên trực tiếp tỏ ra khá hiệu quả vì ít chịu ảnh hưởng của nhiễu. Nếu sự biến thiên độ sáng không đột ngột, phương pháp này lại kém hiệu quả.

Phương pháp dò biên trực tiếp

Quy trình dò biên trực tiếp:

- Khử nhiễu ảnh.
 - Làm nổi biên.
 - Định vị điểm biên.
 - Liên kết và trích chọn biên.

Phương pháp dò biên gián tiếp

- Nếu ta phân được ảnh thành các vùng thì đường phân ranh giữa các vùng đó chính là biên.
- Việc phân vùng ảnh thường dựa vào kết cấu (texture) bề mặt của ảnh.
- Phương pháp dò biên gián tiếp tuy có khó cài đặt tuy nhiên lại áp dụng khá tốt khi sự biến thiên độ sáng nhỏ.

Một số phương pháp phát hiện biên cục bộ

Phương pháp đạo hàm bậc nhất (Gradient)

- Phương pháp Robert.
 - Phương pháp Sobel.
 - Phương pháp Prewitt.
 - Phương pháp Kirsh.

Phương pháp đạo hàm bậc hai (Laplace)

- Phương pháp Laplacian.
 - Phương pháp LoG (Laplacian of Gaussian).
 - Phương pháp DoG (Difference of Gaussian).

Phương pháp tối ưu

- Phương pháp Canny.

Khái niệm Gradient

- Gradient của một hàm $f(x_1, x_2, \dots, x_n)$, được ký hiệu ∇f là một vector n chiều, mà mỗi thành phần trong vector đó là một đạo hàm riêng theo từng biến của hàm đó:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

- Vector gradient tại một điểm cho biết từ điểm đó, hướng nào làm tăng giá trị f nhiều nhất có thể.
 - Độ lớn của vector gradient là mức độ thay đổi lớn nhất

Phương pháp Gradient

- Là phương pháp dò biên cục bộ dựa vào cực đại của đạo hàm.
 - Phương pháp dựa trên gradient hiệu quả khi cường độ sáng thay đổi rõ nét trên biên. Nghĩa là biên thiên độ sáng là đột ngột. .
 - Gradient là một véctơ có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh theo hai hướng x và y.
 - Các thành phần của gradient được tính bởi:
$$\frac{\partial f}{\partial x} = f(x + 1, y) - f(x, y)$$
$$\frac{\partial f}{\partial y} = f(x, y + 1) - f(x, y)$$
 - Kỹ thuật này sử dụng một cặp mặt nạ H_1 và H_2 trực giao (theo 2 hướng vuông góc).

Phương pháp Gradient

- Gọi g_1, g_2 là gradient tương ứng theo 2 hướng x và y , thì biên độ của gradient, ký hiệu là g tại điểm (m,n) và góc giữa 2 gradient được tính theo công thức:

$$g(m, n) = \sqrt{g_1^2(m, n) + g_2^2(m, n)}$$

$$\varphi(m, n) = \arctan\left(\frac{g_2(m, n)}{g_1(m, n)}\right)$$

- Để đơn giản trong tính toán, biên độ của gradient trên có thể tính như sau:

$$g(m, n) = |g_1(m, n)| + |g_2(m, n)|$$

- Các toán tử đạo hàm được áp dụng là khá nhiều. Một số toán tử tiêu biểu: toán tử Robert, Sobel, Prewitt. . .

Toán tử Robert

- Được Robert đề xuất vào năm 1965.
 - Nó áp dụng trực tiếp của các công thức đạo hàm tại điểm (x,y) .
 - Với mỗi điểm ảnh $I(x,y)$ của I , đạo hàm theo x , theo y được ký hiệu tương ứng bởi g_x, g_y được tính:

$$g_x = I(x+1, y) - I(x, y)$$

$$g_y = I(x, y+1) - I(x, y)$$

- Robert đề xuất nhân chập ảnh với 2 mặt nạ H_1 và H_2 :

$$H_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

Toán tử Robert

Ví dụ: Cho ma trận ảnh I, dùng toán tử Robert để tìm biên ảnh I.

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Sử dụng toán tử Robert, tính toán được như sau:

$$|G_1| = \begin{bmatrix} 5 & 5 & 5 & 4 \\ 5 & 5 & 5 & 8 \\ 5 & 5 & 5 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad |G_2| = \begin{bmatrix} 3 & 3 & 3 & 8 \\ 3 & 3 & 3 & 12 \\ 3 & 3 & 3 & 16 \\ 14 & 15 & 16 & 0 \end{bmatrix} \quad (2)$$

Toán tử Robert

Kết quả biên ảnh I, dùng toán tử Robert:

$$G = |G_1| + |G_2| = \begin{bmatrix} 8 & 8 & 8 & 12 \\ 8 & 8 & 8 & 20 \\ 8 & 8 & 8 & 28 \\ 27 & 29 & 31 & 16 \end{bmatrix}$$

Toán tử Robert

Ví dụ: Ảnh và sau khi dò biên bằng toán tử Robert.



Chú ý: Dùng hàm `E = edge(I,'roberts')` để dò biên theo phương pháp Robert trong matlab

Toán tử Sobel

- Được Irwin Sobel đề xuất năm 1968.
 - Sobel đề xuất nhân chập ảnh với 2 măt nạ H_1 và H_2 :

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad H_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3)$$

Do nhiễu có ảnh hưởng đến việc dò biên, nên mặt nạ Sobel thu được bằng cách khử nhiễu (làm mịn) bằng bộ lọc nhị thức (binomial filter). Sau đó, tiếp tục được dò biên bằng cách sử dụng phương pháp đạo hàm bậc nhất (gradient). Các bước để thu được cặp mặt nạ Sobel như sau:

Toán tử Sobel

Bước 1: Làm trơn ảnh (khử nhiễu) với bộ lọc nhị thức (3×3) theo cột:

$$I_1 = [1 \ 2 \ 1]^T \otimes I$$

Bước 2: Thực hiện dò biên I_1 bởi đạo hàm bậc nhất theo hàng ngang:

$$I_2 = [-1 \ 0 \ 1] \otimes I_1$$

Như vậy, việc dò biên theo chiều ngang để thu được I_2 có thể viết lại như sau:

$$I_2 = [1 \ 2 \ 1]^T \otimes [-1 \ 0 \ 1] \otimes I$$

$$l_2 = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \otimes l$$

Toán tử Sobel

Bước 3: Làm trơn ảnh (khử nhiễu) với bộ lọc nhị thức (3×3) theo hàng:

$$I_3 = [1 \ 2 \ 1] \otimes I$$

Bước 4: Thực hiện dò biên I_3 bởi đạo hàm bậc nhất theo hàng cột:

$$I_4 = [-1 \ 0 \ 1]^T \otimes I_3$$

Như vậy, việc dò biên theo chiều dọc để thu được I_4 có thể viết lại như sau:

$$I_4 = [-1 \ 0 \ 1]^T \otimes [1 \ 2 \ 1] \otimes I$$

$$l_4 = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \otimes l$$

Toán tử Sobel

Ví dụ: Cho ma trận ảnh I, dùng toán tử Sobel để tìm biên ảnh I.

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Sử dụng toán tử Sobel, tính toán được như sau:

$$|G_1| = \begin{bmatrix} 10 & 6 & 6 & 13 \\ 24 & 8 & 8 & 28 \\ 40 & 8 & 8 & 44 \\ 38 & 6 & 6 & 41 \end{bmatrix} \quad |G_2| = \begin{bmatrix} 16 & 24 & 28 & 23 \\ 24 & 32 & 32 & 24 \\ 24 & 32 & 32 & 24 \\ 28 & 40 & 44 & 35 \end{bmatrix} \quad (4)$$

Toán tử Sobel

Theo công thức tính biên độ G:

$$G = |G_1| + |G_2|$$

Như vậy giá trị trị của G là:

$$G = \begin{bmatrix} 26 & 30 & 34 & 36 \\ 48 & 40 & 40 & 52 \\ 64 & 40 & 40 & 68 \\ 66 & 46 & 50 & 76 \end{bmatrix}$$

Toán tử Sobel

Ví dụ: Ảnh và sau khi dò biên bằng toán tử Sobel.



Chú ý: Dùng hàm `E = edge(I,'sobel')` để dò biên theo phương pháp Sobel trong matlab

Toán tử Prewitt

- Được Judith M. S. Prewitt đề xuất năm 1970.
 - Prewitt đề xuất nhân chập ảnh với 2 mặt na H_1 và H_2 :

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

Do nhiễu có ảnh hưởng đến việc dò biên, nên mặt nạ Prewitt thu được bằng cách khử nhiễu (làm mịn) bằng bộ lọc trung bình (box filter). Sau đó, tiếp tục được dò biên bằng cách sử dụng phương pháp đạo hàm bậc nhất (gradient). Các bước để thu được cặp mặt nạ Prewitt như sau:

Toán tử Prewitt

Bước 1: Làm trơn ảnh (khử nhiễu) với bộ lọc trung bình (3×3) theo cột:

$$I_1 = [1 \ 1 \ 1]^T \otimes I$$

Bước 2: Thực hiện dò biên I_1 bởi đạo hàm bậc nhất theo hàng ngang:

$$I_2 = [-1 \ 0 \ 1] \otimes I_1$$

Như vậy, việc dò biên theo chiều ngang để thu được I_2 có thể viết lại như sau:

$$I_2 = [1 \ 1 \ 1]^T \otimes [-1 \ 0 \ 1] \otimes I$$

$$I_2 = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \otimes I$$

Toán tử Prewitt

Bước 3: Làm trơn ảnh (khử nhiễu) với bộ lọc trung bình (3×3) theo hàng:

$$I_3 = [1 \ 1 \ 1] \otimes I$$

Bước 4: Thực hiện dò biên I_3 bởi đạo hàm bậc nhất theo hàng cột:

$$I_4 = [-1 \ 0 \ 1]^T \otimes I_3$$

Như vậy, việc dò biên theo chiều dọc để thu được I_4 có thể viết lại như sau:

$$I_4 = [-1 \ 0 \ 1]^T \otimes [1 \ 1 \ 1] \otimes I$$

$$l_4 = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \otimes l$$

Toán tử Prewitt

Ví dụ: Cho ma trận ảnh I, dùng toán tử Prewitt để tìm biên ảnh I.

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Sử dụng toán tử Prewitt, tính toán được như sau:

$$|G_1| = \begin{bmatrix} 8 & 4 & 4 & 10 \\ 18 & 6 & 6 & 21 \\ 30 & 6 & 6 & 33 \\ 24 & 4 & 4 & 26 \end{bmatrix} \quad |G_2| = \begin{bmatrix} 11 & 18 & 21 & 15 \\ 16 & 24 & 24 & 16 \\ 16 & 24 & 24 & 16 \\ 19 & 30 & 33 & 23 \end{bmatrix} \quad (6)$$

Toán tử Prewitt

Theo công thức tính biên độ G:

$$G = |G_1| + |G_2|$$

Như vậy giá trị trị của G là:

$$G = \begin{bmatrix} 19 & 22 & 25 & 25 \\ 34 & 30 & 30 & 37 \\ 46 & 30 & 30 & 49 \\ 43 & 34 & 37 & 49 \end{bmatrix}$$

Toán tử Prewitt

Ví dụ: Ảnh và sau khi dò biên bằng toán tử Prewitt.



Chú ý: Dùng hàm $E = \text{edge}(I, \text{'prewitt'})$ để dò biên theo phương pháp Prewitt trong matlab

Toán tử La bàn Kirsh

- Được Russell A. Kirsch đề xuất năm 1971.
 - Kỹ thuật sử dụng 8 mặt nạ nhân chập theo 8 hướng: 0^0 , 45^0 , 90^0 , 135^0 , 180^0 , 225^0 , 270^0 , 315^0 :

-3	-3	-3
-3	0	5
-3	5	5

-3	-3	-3
-3	0	-3
5	5	5

-3	-3	-3
5	0	-3
5	5	-3

-3	-3	5
-3	0	5
-3	-3	5

5	-3	-3
5	0	-3
5	-3	-3

-3	5	5
-3	0	5
-3	-3	-3

5	5	5
-3	0	-3
-3	-3	-3

5	5	-3
5	0	-3
-3	-3	-3

Toán tử La bàn Kirsh

Các bước tính toán theo toán tử La bàn:

- **Bước 1:** Tính nhân chập ảnh I với mặt nạ H_i : $G_i = I \oplus H_i$ với $i = \overline{1, 8}$.
 - **Bước 2:** Tính giá trị $G(x, y) = \text{Max}|G_i(x, y)|$ với $i = \overline{1, 8}$.

Toán tử La bàn Kirsh

Ví dụ: Cho ma trận I, hãy sử dụng toán tử la bàn để tìm biên.

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Toán tử La bàn Kirsh

Lần lượt nhân chập ma trận ảnh I với mặt nạ la bàn ta được:

$$|G_1| = |I \oplus H_1| = \begin{bmatrix} 65 & 62 & 71 & 10 \\ 116 & 80 & 80 & 15 \\ 152 & 80 & 80 & 43 \\ 13 & 54 & 61 & 114 \end{bmatrix}$$

$$|G_2| = |I \oplus H_2| = \begin{bmatrix} 49 & 78 & 87 & 66 \\ 68 & 96 & 96 & 73 \\ 72 & 96 & 96 & 77 \\ 99 & 174 & 189 & 114 \end{bmatrix}$$

Toán tử La bàn Kirsh

Lần lượt nhân chập ma trận ảnh I với mặt nạ la bàn ta được:

$$|G_3| = |I \oplus H_3| = \begin{bmatrix} 1 & 30 & 39 & 90 \\ 12 & 48 & 48 & 129 \\ 40 & 48 & 48 & 165 \\ 99 & 70 & 77 & 6 \end{bmatrix}$$

$$|G_4| = |I \oplus H_4| = \begin{bmatrix} 39 & 18 & 17 & 26 \\ 84 & 24 & 24 & 57 \\ 144 & 24 & 24 & 93 \\ 99 & 2 & 3 & 94 \end{bmatrix}$$

Toán tử La bàn Kirsh

Lần lượt nhân chập ma trận ảnh I với mặt nạ la bàn ta được:

$$|G_5| = |I \oplus H_5| = \begin{bmatrix} 39 & 58 & 65 & 30 \\ 76 & 80 & 80 & 1 \\ 104 & 80 & 80 & 37 \\ 27 & 82 & 91 & 190 \end{bmatrix}$$

$$|G_6| = |I \oplus H_6| = \begin{bmatrix} 39 & 66 & 81 & 54 \\ 60 & 96 & 96 & 55 \\ 56 & 96 & 96 & 51 \\ 53 & 66 & 75 & 70 \end{bmatrix}$$

Toán tử La bàn Kirsh

Lần lượt nhân chập ma trận ảnh I với mặt nạ la bàn ta được:

$$|G_7| = |I \oplus H_7| = \begin{bmatrix} 23 & 42 & 49 & 54 \\ 12 & 48 & 48 & 79 \\ 24 & 48 & 48 & 107 \\ 165 & 114 & 123 & 18 \end{bmatrix}$$

$$|G_8| = |I \oplus H_8| = \begin{bmatrix} 25 & 14 & 15 & 54 \\ 60 & 24 & 24 & 111 \\ 96 & 24 & 24 & 171 \\ 93 & 34 & 35 & 114 \end{bmatrix}$$

Toán tử La bàn Kirsh

Xác định ma trận biên ảnh:

$$|G| = \max |G_i| = \begin{bmatrix} 65 & 78 & 87 & 90 \\ 116 & 96 & 96 & 129 \\ 152 & 96 & 96 & 171 \\ 165 & 174 & 189 & 190 \end{bmatrix}$$

Phương pháp Laplace

- Các phương pháp gradient ở trên làm việc khá tốt khi mà độ sáng thay đổi rõ nét.
 - Khi mức xám thay đổi chậm, miền chuyển tiếp trải rộng, phương pháp cho hiệu quả hơn đó là phương pháp sử dụng đạo hàm bậc hai Laplace.
 - Kỹ thuật Laplace thường cho biên mảnh, tức là đường biên có độ rộng bằng một pixel. Tuy nhiên, kỹ thuật này rất nhạy cảm với nhiễu vì đạo hàm bậc hai thường không ổn định.
 - Toán tử Laplace được định nghĩa như sau:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Phương pháp Laplace

Trong đó:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Như vậy, có thể tính xấp xỉ đạo hàm bậc hai bằng cách nhân chập ảnh I với ma trận mặt na sau:

$$H_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Phương pháp Laplace

Trong thực tế, một số mặt nạ Laplace thường dùng:

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Phương pháp Laplace

Ví dụ, Cho ảnh I, mặt nạ H dùng toán tử Laplace phát hiện biên:

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Phương pháp Laplace

Biên của ảnh thu được:

$$G = |I \oplus H| = \begin{bmatrix} 3 & 2 & 1 & 5 \\ 4 & 0 & 0 & 9 \\ 8 & 0 & 0 & 13 \\ 29 & 18 & 19 & 37 \end{bmatrix}$$

Phương pháp Laplace

Ví dụ: Ảnh và sau khi dò biên bằng phương pháp Laplace.



Bộ lọc không gian sắc nét

- Làm sáng lên các chi tiết tốt trong ảnh hoặc tăng cường các chi tiết bị mờ.
 - **Ý tưởng chính:** Lấy vi phân ảnh (tức lấy đạo hàm cấp 1, cấp 2) nhằm tăng cường mép ảnh và điểm gián đoạn khác.

Bộ lọc không gian sắc nét

Phép biến đổi Laplacian:

- Được đề xuất bởi Rosenfeld và Kak năm [1982].
- Là một đạo hàm có hướng với ảnh (với ảnh là một hàm $f(x,y)$). Công thức tổng quát:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Trong đó:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Bộ lọc không gian sắc nét

Mặt nạ Laplacian:

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Bộ lọc không gian sắc nét

Phép biến đổi Laplacian:

- Làm sáng tỏ mức xám không liên tục trong ảnh.
 - Làm mờ những vùng có cấp độ xám khác nhau.
 - Phục hồi những đặc tính căn bản.
 - Độ sắc nét được bảo toàn.

Bộ lọc không gian sắc nét

Phép biến đổi Laplacian:

Dùng để nâng cao chất lượng ảnh như sau:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & (1) \\ f(x, y) + \nabla^2 f(x, y) & (2) \end{cases}$$

Trong đó:

- Phương trình (1) dùng khi hệ số trung tâm của mặt nạ Laplacian là âm.
- Phương trình (2) dùng khi hệ số trung tâm của mặt nạ Laplacian là dương.

Bộ lọc không gian sắc nét

Rút gọn công thức $g(x,y)$:

$$\begin{aligned}g(x, y) &= f(x, y) - \nabla^2 f(x, y) \\&= f(x, y) - [f(x + 1, y) + f(x - 1, y) \\&\quad + f(x, y + 1) + f(x, y - 1)] + 4f(x, y) \\&= 5f(x, y) - [f(x + 1, y) + f(x - 1, y) \\&\quad + f(x, y + 1) + f(x, y - 1)]\end{aligned}$$

Ma trận mặt nạ:

0	-1	0
-1	5	-1
0	-1	0

Bộ lọc không gian sắc nét

Ví dụ: Cho ảnh I và mặt nạ Laplacian K có ma trận như sau. Hãy thực hiện phép biến đổi Laplacian để tăng cường ảnh.

Ma trận ảnh I

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Mặt nạ K

0	-1	0
-1	4	-1
0	-1	0

Bộ lọc không gian sắc nét

Kết quả sau khi thực hiện tăng cường ảnh bằng phép biến đổi Laplacian:

0	0	2	9
9	6	7	17
17	10	11	25
42	32	34	53

Bộ lọc không gian sắc nét

Ví dụ ảnh được lọc với bộ lọc Laplacian



XỬ LÝ ẢNH LỌC ẢNH TRÊN MIỀN TẦN SỐ

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa học Máy tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

1 Giới thiệu phép biến đổi Fourier

Giới thiệu phép biến đổi Fourier

- Công thức biến đổi Fourier cho hàm 2 biến:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

- Công thức biến đổi Fourier ngược cho hàm 2 biến:

$$f(x, y) = \frac{1}{M * N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

- Công thức Euler:

$$e^{j\omega} = \cos(\omega) + j\sin(\omega)$$

Chú ý: Hàm **fft2(I)** và **ifft2(I)** dùng để biến đổi Fourier và Fourier ngược trong matlab

Giới thiệu phép biến đổi Fourier

Ví dụ 1: Biến đổi Fourier cho ma trận ảnh I sau:

$$I = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Các bước thực hiện như sau:

$$\begin{aligned} F(u, v) = & f(0, 0)e^{-j2\pi(\frac{u*0}{2} + \frac{v*0}{2})} + f(0, 1)e^{-j2\pi(\frac{u*0}{2} + \frac{v*1}{2})} \\ & + f(1, 0)e^{-j2\pi(\frac{u*1}{2} + \frac{v*0}{2})} + f(1, 1)e^{-j2\pi(\frac{u*1}{2} + \frac{v*1}{2})} \end{aligned} \quad (1)$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $F(0,0)$:

$$\begin{aligned} F(0,0) &= f(0,0)e^{-j2\pi(\frac{0*0}{2} + \frac{0*0}{2})} + f(0,1)e^{-j2\pi(\frac{0*0}{2} + \frac{0*1}{2})} \\ &\quad + f(1,0)e^{-j2\pi(\frac{0*1}{2} + \frac{0*0}{2})} + f(1,1)e^{-j2\pi(\frac{0*1}{2} + \frac{0*1}{2})} \\ &= f(0,1)e^{-j2\pi(\frac{0*0}{2} + \frac{0*1}{2})} + f(1,0)e^{-j2\pi(\frac{0*1}{2} + \frac{0*0}{2})} \\ &= 2 * e^0 = 2. \end{aligned} \tag{2}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $F(0,1)$:

$$\begin{aligned} F(0,1) &= f(0,0)e^{-j2\pi(\frac{0*0}{2} + \frac{1*0}{2})} + f(0,1)e^{-j2\pi(\frac{0*0}{2} + \frac{1*1}{2})} + \\ &\quad f(1,0)e^{-j2\pi(\frac{0*1}{2} + \frac{1*0}{2})} + f(1,1)e^{-j2\pi(\frac{0*1}{2} + \frac{1*1}{2})} \\ &= f(0,1)e^{-j2\pi(\frac{0*0}{2} + \frac{1*1}{2})} + f(1,0)e^{-j2\pi(\frac{0*1}{2} + \frac{1*0}{2})} \\ &= 2 * (e^{-j\pi} + e^0) = 0. \end{aligned} \tag{3}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $F(1,0)$:

$$\begin{aligned} F(1,0) &= f(0,0)e^{-j2\pi(\frac{1*0}{2} + \frac{0*0}{2})} + f(0,1)e^{-j2\pi(\frac{1*0}{2} + \frac{0*1}{2})} + \\ &\quad f(1,0)e^{-j2\pi(\frac{1*1}{2} + \frac{0*0}{2})} + f(1,1)e^{-j2\pi(\frac{1*1}{2} + \frac{0*1}{2})} \\ &= f(0,1)e^{-j2\pi(\frac{1*0}{2} + \frac{0*1}{2})} + f(1,0)e^{-j2\pi(\frac{1*1}{2} + \frac{0*0}{2})} \\ &= 2 * (e^0 + e^{-j\pi}) = 0. \end{aligned} \tag{4}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $F(1,1)$:

$$\begin{aligned} F(1,1) &= f(0,0)e^{-j2\pi(\frac{1*0}{2} + \frac{1*0}{2})} + f(0,1)e^{-j2\pi(\frac{1*0}{2} + \frac{1*1}{2})} + \\ &\quad f(1,0)e^{-j2\pi(\frac{1*1}{2} + \frac{1*0}{2})} + f(1,1)e^{-j2\pi(\frac{1*1}{2} + \frac{1*1}{2})} \\ &= f(0,1)e^{-j2\pi(\frac{1*0}{2} + \frac{1*1}{2})} + f(1,0)e^{-j2\pi(\frac{1*1}{2} + \frac{1*0}{2})} \\ &= 2 * (e^{-j\pi}) = -2. \end{aligned} \tag{5}$$

Giới thiệu phép biến đổi Fourier

Kết quả thực hiện biến đổi Fourier cho ma trận ảnh $f(x,y)$:

$$F(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

Giới thiệu phép biến đổi Fourier

Ví dụ 2: Biến đổi Fourier ngược cho ma trận ảnh sau:

$$F(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

Các bước thực hiện như sau:

$$\begin{aligned} f(x, y) = & \frac{1}{2 * 2} (F(0, 0)e^{j2\pi(\frac{0*x}{2} + \frac{0*y}{2})} + F(0, 1)e^{j2\pi(\frac{0*x}{2} + \frac{1*y}{2})} \\ & + F(1, 0)e^{j2\pi(\frac{1*x}{2} + \frac{0*y}{2})} + F(1, 1)e^{j2\pi(\frac{1*x}{2} + \frac{1*y}{2})}) \end{aligned} \quad (6)$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $f(0,0)$:

$$\begin{aligned} f(0,0) &= \frac{1}{2*2}(F(0,0)e^{j2\pi(\frac{0*0}{2}+\frac{0*0}{2})} + F(0,1)e^{j2\pi(\frac{0*0}{2}+\frac{1*0}{2})} \\ &\quad + F(1,0)e^{j2\pi(\frac{1*0}{2}+\frac{0*0}{2})} + F(1,1)e^{j2\pi(\frac{1*0}{2}+\frac{1*0}{2})}) \\ &= \frac{1}{4}(F(0,0)e^{j2\pi(\frac{0*0}{2}+\frac{0*0}{2})} + F(1,1)e^{j2\pi(\frac{1*0}{2}+\frac{1*0}{2})}) \\ &= \frac{1}{4} * (2e^0 - 2e^0) = 0. \end{aligned} \tag{7}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $f(0,1)$:

$$\begin{aligned} f(0,1) &= \frac{1}{2*2}(F(0,0)e^{j2\pi(\frac{0*0}{2}+\frac{0*1}{2})} + F(0,1)e^{j2\pi(\frac{0*0}{2}+\frac{1*1}{2})} \\ &\quad + F(1,0)e^{j2\pi(\frac{1*0}{2}+\frac{0*1}{2})} + F(1,1)e^{j2\pi(\frac{1*0}{2}+\frac{1*1}{2})}) \\ &= \frac{1}{4}(F(0,0)e^{j2\pi(\frac{0*0}{2}+\frac{0*1}{2})} + F(1,1)e^{j2\pi(\frac{1*0}{2}+\frac{1*1}{2})}) \\ &= \frac{1}{4} * (2e^0 - 2e^{j\pi}) = 1. \end{aligned} \tag{8}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $f(1,0)$:

$$\begin{aligned} f(1,0) &= \frac{1}{2*2}(F(0,0)e^{j2\pi(\frac{0*1}{2} + \frac{0*0}{2})} + F(0,1)e^{j2\pi(\frac{0*1}{2} + \frac{1*0}{2})} \\ &\quad + F(1,0)e^{j2\pi(\frac{1*1}{2} + \frac{0*0}{2})} + F(1,1)e^{j2\pi(\frac{1*1}{2} + \frac{1*0}{2})}) \\ &= \frac{1}{4}(F(0,0)e^{j2\pi(\frac{0*1}{2} + \frac{0*0}{2})} + F(1,1)e^{j2\pi(\frac{1*1}{2} + \frac{1*0}{2})}) \\ &= \frac{1}{4} * (2e^0 - 2e^{j\pi}) = 1. \end{aligned} \tag{9}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho $f(1,1)$:

$$\begin{aligned}f(1,1) &= \frac{1}{2*2}(F(0,0)e^{j2\pi(\frac{0*1}{2}+\frac{0*1}{2})} + F(0,1)e^{j2\pi(\frac{0*1}{2}+\frac{1*1}{2})} \\&\quad + F(1,0)e^{j2\pi(\frac{1*1}{2}+\frac{0*1}{2})} + F(1,1)e^{j2\pi(\frac{1*1}{2}+\frac{1*1}{2})}) \\&= \frac{1}{4}(F(0,0)e^{j2\pi(\frac{0*1}{2}+\frac{0*1}{2})} + F(1,1)e^{j2\pi(\frac{1*1}{2}+\frac{1*1}{2})}) \tag{10} \\&= \frac{1}{4} * (2e^0 - 2e^{j2\pi}) = 0.\end{aligned}$$

Giới thiệu phép biến đổi Fourier

Kết quả thực hiện biến đổi Fourier ngược cho ma trận ảnh F:

$$f(x, y) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Giới thiệu phép biến đổi Fourier

Biến đổi Fourier cho ma trận ảnh $f(x,y)$ cũng có thể thực hiện thông qua việc tính tích các ma trận như sau:

$$F = U^T * f * V$$

Trong đó:

- $U(x,u) = e^{-j2\pi \frac{ux}{M}}$, với $x,u = \overline{0, M-1}$
- $V(y,v) = e^{-j2\pi \frac{vy}{N}}$, với $v,y = \overline{0, N-1}$

Giới thiệu phép biến đổi Fourier

Biến đổi Fourier ngược cho ma trận ảnh $F(x,y)$ cũng có thể thực hiện thông qua việc tính tích các ma trận như sau:

$$f(x,y) = U^{-1} * F * V^{-1}$$

Trong đó:

- $U(x,u) = e^{-j2\pi \frac{ux}{M}}$, với $x,u = \overline{0, M-1}$
- $V(y,v) = e^{-j2\pi \frac{vy}{N}}$, với $v,y = \overline{0, N-1}$

Giới thiệu phép biến đổi Fourier

Ví dụ 3: Biến đổi Fourier cho ma trận ảnh sau:

$$f(x, y) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Tính giá trị cho hai ma trận U và V:

$$U = \begin{bmatrix} e^{-j2\pi * \frac{0*0}{2}} & e^{-j2\pi * \frac{0*1}{2}} \\ e^{-j2\pi * \frac{1*0}{2}} & e^{-j2\pi * \frac{1*1}{2}} \end{bmatrix} = \begin{bmatrix} e^0 & e^0 \\ e^0 & e^{-j\pi} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$V = \begin{bmatrix} e^{-j2\pi * \frac{0*0}{2}} & e^{-j2\pi * \frac{0*1}{2}} \\ e^{-j2\pi * \frac{1*0}{2}} & e^{-j2\pi * \frac{1*1}{2}} \end{bmatrix} = \begin{bmatrix} e^0 & e^0 \\ e^0 & e^{-j\pi} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho ma trận Fourier F:

$$F = U^T * f * V$$

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \end{aligned} \tag{11}$$

Giới thiệu phép biến đổi Fourier

Ví dụ 4: Biến đổi Fourier ngược cho ma trận ảnh $F(x,y)$ sau:

$$F(x,y) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

Tính giá trị cho hai ma trận U và V :

$$U = \begin{bmatrix} e^{-j2\pi * \frac{0*0}{2}} & e^{-j2\pi * \frac{0*1}{2}} \\ e^{-j2\pi * \frac{1*0}{2}} & e^{-j2\pi * \frac{1*1}{2}} \end{bmatrix} = \begin{bmatrix} e^0 & e^0 \\ e^0 & e^{-j\pi} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$V = \begin{bmatrix} e^{-j2\pi * \frac{0*0}{2}} & e^{-j2\pi * \frac{0*1}{2}} \\ e^{-j2\pi * \frac{1*0}{2}} & e^{-j2\pi * \frac{1*1}{2}} \end{bmatrix} = \begin{bmatrix} e^0 & e^0 \\ e^0 & e^{-j\pi} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Giới thiệu phép biến đổi Fourier

Tính giá trị cho ma trận $f(x,y)$ sau khi biến đổi Fourier ngược:

$$f(x, y) = U^{-1} * F * V^{-1}$$

$$\begin{aligned} F &= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} * \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} * \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{aligned} \tag{12}$$

Giới thiệu phép biến đổi Fourier

Ví dụ 5: Hình ảnh trước và sau khi biến đổi Fourier:



XỬ LÝ ẢNH

LỌC ẢNH TRÊN MIỀN TẦN SỐ

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa học Máy tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

1 Giới thiệu lọc ảnh trên miền tần số

2 Một số bộ lọc trên miền tần số

Bộ lọc trên miền tần số

- Khi chụp ảnh, ta có thể thu được các ảnh có tần số thấp (sự thay đổi mức xám của ảnh ít, ví dụ như ảnh một bức tường) hay ảnh có tần số cao (ví dụ như biên của vật thể).
- Bộ lọc thông thấp làm giảm đi thành phần tần số cao trong khi cho các thành phần tần số thấp đi qua.
- Bộ lọc thông cao làm giảm đi thành phần tần số thấp trong khi cho các thành phần tần số cao đi qua.

Các bước lọc ảnh trên miền tần số

- **Bước 1:** Chuyển ảnh cần lọc I sang miền tần số bởi biến đổi Fourier, thu được $F(u,v)$.
- **Bước 2:** Dịch chuyển $F(u,v)$ từ góc ảnh ra tâm thu được $F_1(u, v)$
- **Bước 3:** Tính bộ lọc $H(u,v)$ trên miền tần số.
- **Bước 4:** Thực hiện tích vô hướng giữa $F_1(u, v)$ và $H(u,v)$ thu được $G(u,v)$.
- **Bước 5:** Dịch chuyển $G(u,v)$ từ tâm về góc ảnh thu được $G_1(u, v)$.
- **Bước 6:** Chuyển đổi $G_1(u, v)$ về miền không gian bằng cách sử dụng biến đổi ngược Fourier, sau đó lấy ra phần thực thu được ảnh sau khi lọc là I_1 .

Một số bộ lọc trên miền tần số

Một số loại bộ lọc trong miền tần số như sau:

- Bộ lọc thông thấp lý tưởng.
- Bộ lọc thông thấp Gaussian.
- Bộ lọc thông thấp Butterworth. (Butterworth lowpass filters).
- Bộ lọc thông cao lý tưởng.
- Bộ lọc thông cao Gaussian.
- Bộ lọc thông cao Butterworth (Butterworth highpass filters).

Bộ lọc thông thấp lý tưởng

Bộ lọc thông thấp 2 chiều không suy giảm tất cả các tần số trong một vòng tròn bán kính từ gốc, và cắt tất cả các tần số ở bên ngoài vòng tròn. Bộ lọc này được định nghĩa như sau:

$$H(u, v) = \begin{cases} 1 & \text{nếu } D(u, v) \leq D_0 \\ 0 & \text{nếu } D(u, v) > D_0 \end{cases}$$

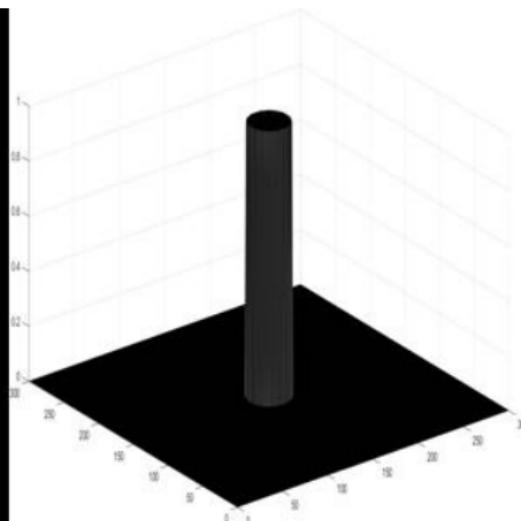
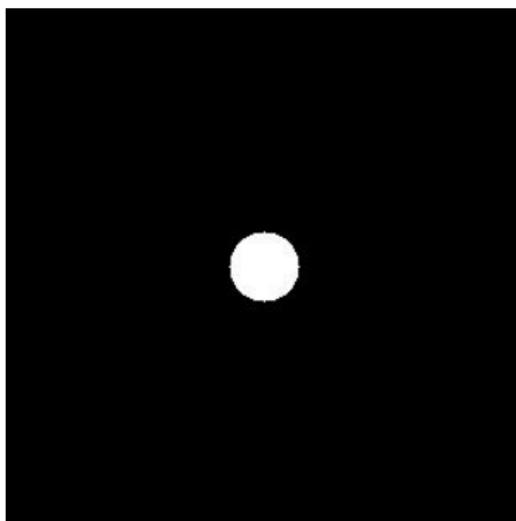
Trong đó, D_0 là một hằng số dương và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

Bộ lọc thông thấp lý tưởng

Hình minh họa bộ lọc thông thấp lý tưởng biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông thấp lý tưởng

Hình minh họa ảnh trước và sau khi lọc thông thấp lý thường:



Bộ lọc thông thấp Gaussian

Bộ lọc thông thấp 2 chiều Gaussian được định nghĩa như sau:

$$H(u, v) = e^{\left(\frac{-D^2(u, v)}{2D_0^2}\right)}$$

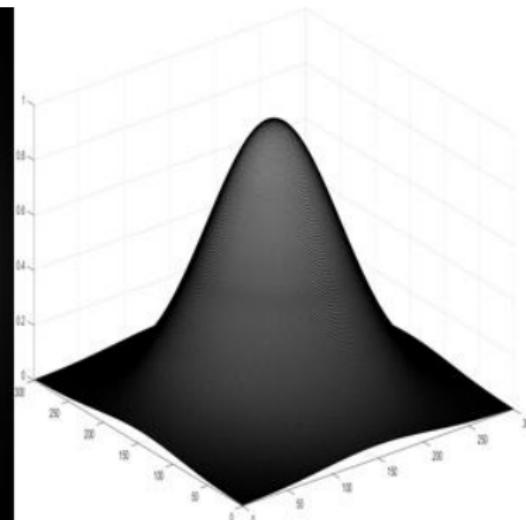
Trong đó, D_0 là một hằng số dương (được gọi là tần số cắt) và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

Bộ lọc thông thấp Gaussian

Hình minh họa bộ lọc thông thấp Gaussian biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông thấp Gaussian

Hình minh họa ảnh trước và sau khi lọc thông thấp Gaussian:



Bộ lọc thông thấp Butterworth

Bộ lọc thông thấp 2 chiều Butterworth được định nghĩa như sau:

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}}$$

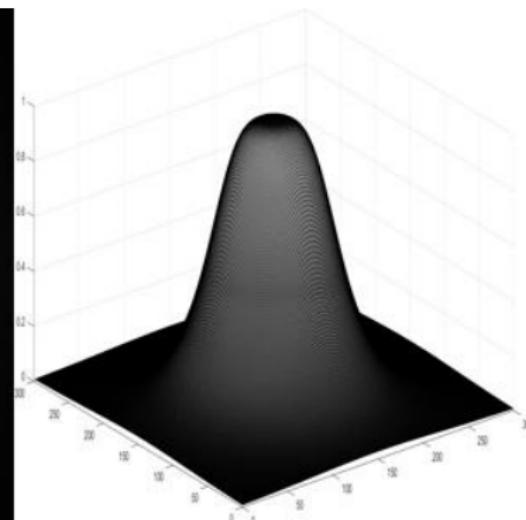
Trong đó, D_0 là một hằng số dương (được gọi là tần số cắt) và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

Bộ lọc thông thấp Butterworth

Hình minh họa bộ lọc thông thấp Butterworth biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông thấp Butterworth

Hình minh họa ảnh trước và sau khi lọc thông thấp Butterworth:



Bộ lọc thông cao lý tưởng

Bộ lọc thông cao 2 chiều không suy giảm tất cả các tần số bên ngoài một vòng tròn bán kính từ gốc, và cắt tất cả các tần số ở bên trong vòng tròn. Bộ lọc này được định nghĩa như sau:

$$H(u, v) = \begin{cases} 0 & \text{nếu } D(u, v) \leq D_0 \\ 1 & \text{nếu } D(u, v) > D_0 \end{cases}$$

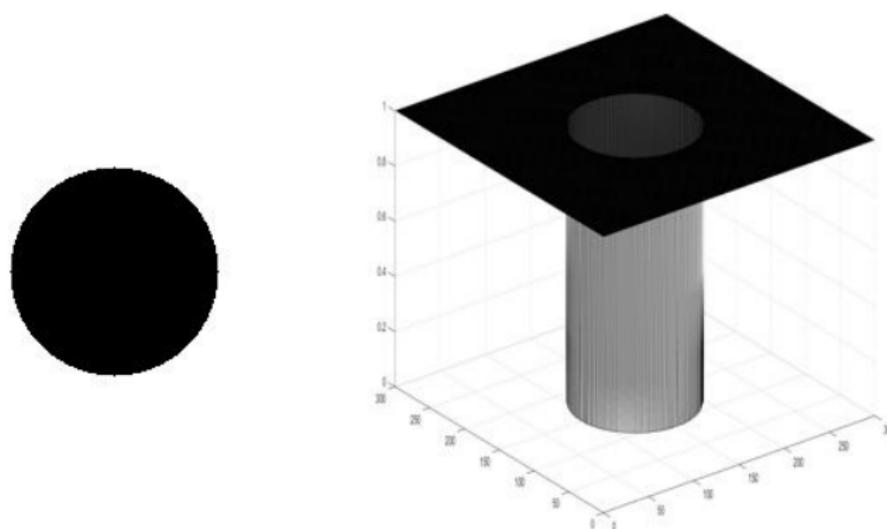
Trong đó, D_0 là một hằng số dương và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

Bộ lọc thông cáo lý tưởng

Hình minh họa bộ lọc thông cao lý tưởng biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông cáo lý tưởng

Hình minh họa ảnh trước và sau khi lọc thông cao lý thường:



Bộ lọc thông cao Gaussian

Bộ lọc thông cao 2 chiều Gaussian được định nghĩa như sau:

$$H(u, v) = 1 - e^{\left(\frac{-D^2(u, v)}{2D_0^2}\right)}$$

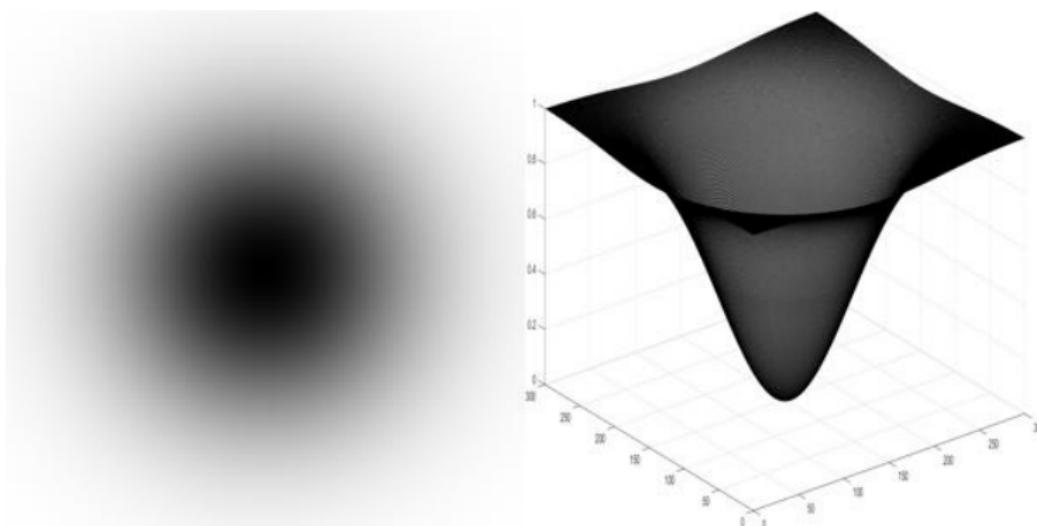
Trong đó, D_0 là một hằng số dương (được gọi là tần số cắt) và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

Bộ lọc thông cao Gaussian

Hình minh họa bộ lọc thông cao Gaussian biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông cao Gaussian

Hình minh họa ảnh trước và sau khi lọc thông cao Gaussian:



Bộ lọc thông cao Butterworth

Bộ lọc thông cao 2 chiều Butterworth được định nghĩa như sau:

$$H(u, v) = 1 - \frac{1}{1 + [\frac{D(u, v)}{D_0}]^{2n}}$$

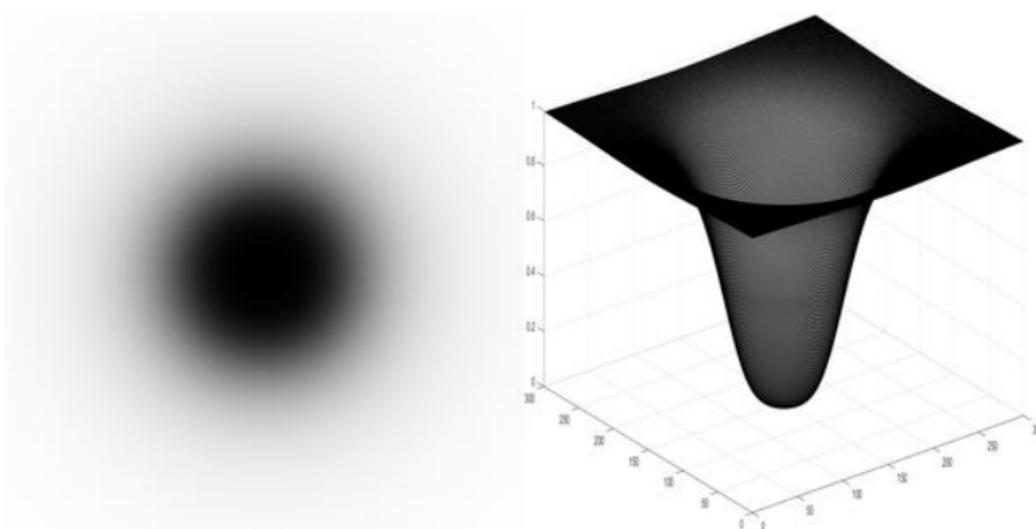
Trong đó, D_0 là một hằng số dương (được gọi là tần số cắt) và $D(u, v)$ được tính như sau:

$$D(u, v) = \sqrt{(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2}$$

P, Q là chiều dài và chiều rộng của ảnh I.

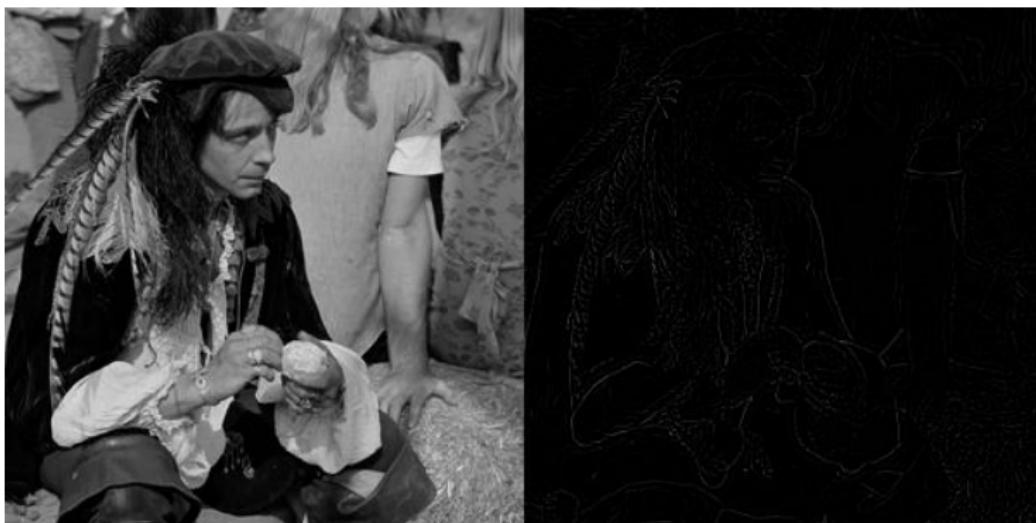
Bộ lọc thông cao Butterworth

Hình minh họa bộ lọc thông cao Butterworth biểu diễn trong không gian 2 chiều và 3 chiều:



Bộ lọc thông cao Butterworth

Hình minh họa ảnh trước và sau khi lọc thông cao Butterworth:



XỬ LÝ ẢNH

CÁC PHƯƠNG PHÁP NÉN ẢNH

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2020

Nội Dung

- 1 Tổng quan về nén ảnh
- 2 Phân loại các phương pháp nén ảnh
- 3 Một số phương pháp nén ảnh

Tổng quan về nén ảnh

Một số khái niệm cơ bản:

- Dư thừa dữ liệu và thông tin.
- Nén dữ liệu.
- Tỉ lệ nén.
- Tính dung lượng ảnh.

Dư thừa dữ liệu và thông tin

- Dữ liệu và thông tin không tương đương với nhau.
- Dữ liệu là phương tiện để thể hiện thông tin. Số lượng dữ liệu có thể nhiều hơn số lượng thông tin.
- Những dữ liệu dư thừa không cung cấp thêm thông tin.
- Nén ảnh mục đích là để giảm bớt lượng dữ liệu dư thừa trong khi vẫn giữ nguyên được lượng thông tin.

Một số kiểu dư thừa dữ liệu trong ảnh

- **Kiểu 1: Dư thừa mă**

Một số mức xám trong ảnh phổ biến hơn các mức xám còn lại.

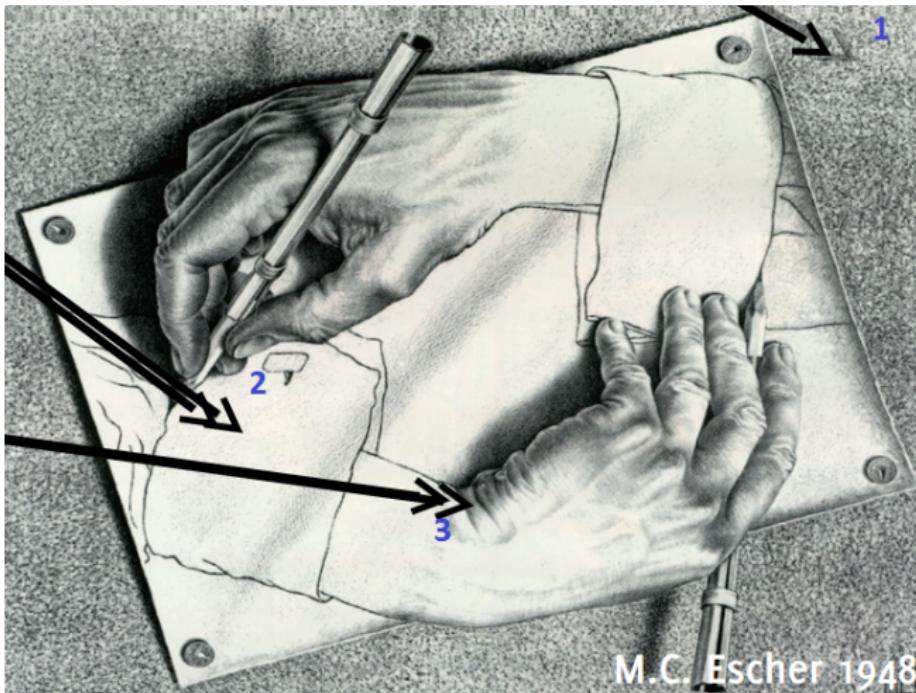
- **Kiểu 2: Dư thừa trong pixel**

Giá trị của các pixel trong ảnh có thể được dự đoán hợp lý từ các pixel láng giềng.

- **Kiểu 3: Dư thừa trong thị giác mắt**

Mắt người chỉ có thể phân biệt được một số lượng màu sắc nhất định.

Một số kiểu dư thừa dữ liệu trong ảnh



Nén dữ liệu

- Nén dữ liệu nhằm làm giảm lượng thông tin “dư thừa” trong dữ liệu gốc.
- Với dữ liệu ảnh, kết quả thường là 10:1. Một số phương pháp còn cho kết quả cao hơn.
- Tạo ra sự hiệu quả trong việc lưu trữ và truyền tải dữ liệu.

Tỉ lệ nén

- Là một trong các đặc trưng quan trọng nhất của mọi phương pháp nén.
- Gọi n_1 là dữ liệu ban đầu, n_2 là dữ liệu sau khi nén. Tỉ lệ nén C_r được tính như sau:

$$C_r = \frac{n_1}{n_2}$$

- Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r}$

Dung lượng ảnh

Cho một ảnh I có kích thước $M \times N$. Gọi L là số mức xám của ảnh I .

Dung lượng của ảnh được tính theo công thức: $M * N * L_{tb}$.

Trong đó:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k) p(r_k)$$

$I(r_k)$ là độ dài bit biểu diễn cho mức xám r_k .

$p(r_k)$ là xác suất của mức xám r_k trong ảnh I . $p(r_k) = \frac{n_k}{M * N}$

Dung lượng ảnh

Ví dụ: Giả sử ảnh 3 bit I có thống kê về các mức xám như bảng sau. Tính dung lượng của ảnh I.

r_k	$n(r_k)$	$p(r_k)$	mã
0	113	0.051	000
1	139	0.063	001
2	142	0.064	010
3	145	0.066	011
4	181	0.082	100
5	105	0.047	101
6	52	0.023	110
7	1323	0.601	111

- Giá trị trung bình mỗi pixel:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k)p(r_k) = 3 \text{ bit}$$

- Dung lượng của ảnh I:

$$C = L_{tb} * \sum_{k=0}^{L-1} n(r_k)$$

$$C = 3 * 2200 = 6600 \text{ bit}$$

Phân loại các phương pháp nén ảnh

Có nhiều cách phân loại các phương pháp nén khác nhau:

- Phương pháp nén không mất mát thông tin.
 - Ảnh nén sau khi giải nén có thể thu được dữ liệu ảnh ban đầu.
 - Tỉ lệ nén thấp từ 2 đến 10 lần.
- Phương pháp nén có mất mát thông tin.
 - Ảnh nén sau khi giải nén không thu được dữ liệu như ảnh gốc ban đầu.
 - Tỉ lệ nén cao từ 10 đến 30 lần.

Phân loại các phương pháp nén ảnh

Dựa vào quá trình phát triển của các phương pháp nén. Có thể chia các phương pháp nén thành hai nhóm:

- Các phương pháp nén thế hệ thứ nhất: Gồm các phương pháp mà mức độ tính toán là đơn giản, thí dụ việc lấy mẫu, gán từ mã,...
- Các phương pháp nén thế hệ thứ hai: Dựa vào độ bão hòa của tỷ lệ nén.

Một số phương pháp nén ảnh

- Phương pháp nén ảnh Huffman.
- Phương pháp nén ảnh Run-Length Encoding.
- Phương pháp nén ảnh LZW

Phương pháp nén ảnh Huffman

- Được đề xuất bởi Huffman vào năm 1952.
- Là phương pháp dựa vào mô hình thống kê.
- Ý tưởng của phương pháp: gán cho các mức xám có tần suất cao trong ảnh một từ mã ngắn, các mức xám có tần suất thấp từ mã dài.

Phương pháp nén ảnh Huffman

Ví dụ: Cho ảnh l 3 bit, kích thước 44x50 có các mức xám được biểu diễn bởi các mã trong bảng sau. Tính dung lượng ảnh l ứng với các mã đó.

r_k	$n(r_k)$	$p(r_k)$	mã 1	mã 2
0	113	0.051	000	1011
1	139	0.063	001	1010
2	142	0.064	010	1001
3	145	0.066	011	1000
4	181	0.082	100	110
5	105	0.047	101	1110
6	52	0.023	110	1111
7	1323	0.601	111	0

Phương pháp nén ảnh Huffman

- Đối với mã 1, dung lượng ảnh I:

$$C_1 = L_{tb} * M * N = 3 * 44 * 50 = 6600 \text{ bit}$$

- Đối với mã 2:

- Độ dài trung bình mỗi pixel:

$$L_{tb} = \sum_{k=0}^{L-1} l(r_k)p(r_k) = 2.1 \text{ bit}$$

- Dung lượng ảnh I:

$$C_2 = L_{tb} * M * N = 2.1 * 44 * 50 = 4620 \text{ bit}$$

Phương pháp nén ảnh Huffman

Đầu vào: Ảnh I có kích thước MxN và có K mức xám.

Đầu ra: Ảnh I* được nén theo phương pháp Huffman.

Phương pháp này gồm 2 giai đoạn:

Giai đoạn 1: Xây dựng cây nhị phân Huffman

- Duyệt toàn bộ ảnh để tính xác suất các mức xám có trong ảnh.
- Khởi tạo rừng có K cây, mỗi cây chỉ có một nút gốc, mỗi nút gốc tương ứng với một mức xám và có trọng số là tần số (hoặc xác suất) của mức xám đó trong ảnh.
- Thực hiện bước lặp cho đến khi rừng chỉ còn 1 cây:
 - Chọn hai cây có trọng số ở gốc nhỏ nhất hợp thành một cây bằng cách thêm một gốc mới nối với hai gốc đã chọn.
 - Trọng số của gốc mới bằng tổng trọng số của hai gốc tạo thành nó.

Phương pháp nén ảnh Huffman

Giai đoạn 2: Xây dựng mã cho các mức xám

- Bắt đầu từ 2 nút con ở gốc, gán giá trị 0 cho nút có xác suất cao và giá trị 1 cho nút có xác suất thấp.
- Đi dần xuống các nút lá cho đến nút cuối cùng của cây để gán giá trị cho tất cả các nút trên cây.

Phương pháp nén ảnh Huffman

Cho ảnh 3 bit I có các thống kê mức xám như sau. Thực hiện nén ảnh I theo phương pháp Huffman.

r_k	$n(r_k)$	$p(r_k)$	mã
0	113	0.051	000
1	139	0.063	001
2	142	0.064	010
3	145	0.066	011
4	181	0.082	100
5	105	0.047	101
6	52	0.023	110
7	1323	0.601	111

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$
7	0.601
4	0.082
3	0.066
2	0.064
1	0.063
0	0.051
5	0.047
6	0.023

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1
7	0.601	0.601
4	0.082	0.082
3	0.066	0.070
2	0.064	0.066
1	0.063	0.064
0	0.051	0.063 } 0.051 }
5	0.047 } 6	0.023 }

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2
7	0.601	0.601	0.601
4	0.082	0.082	0.114
3	0.066	0.070	0.082
2	0.064	0.066	0.070
1	0.063	0.064	0.066
0	0.051	0.063	0.064
5	0.047	0.051	
6	0.023		

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3
7	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130
3	0.066	0.070	0.082	0.114
2	0.064	0.066	0.070	0.082 }
1	0.063	0.064	0.066 }	0.070 }
0	0.051	0.063 }	0.064 }	
5	0.047 }	0.051 }		
6	0.023 }			

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4
7	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152
3	0.066	0.070	0.082	0.114	0.130
2	0.064	0.066	0.070	0.082	0.114
1	0.063	0.064	0.066	0.070	
0	0.051	0.063	0.064		
5	0.047	0.051			
6	0.023				

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5
7	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244
3	0.066	0.070	0.082	0.114	0.130	0.152
2	0.064	0.066	0.070	0.082	0.114	
1	0.063	0.064	0.066	0.070		
0	0.051	0.063	0.064			
5	0.047	0.051				
6	0.023					

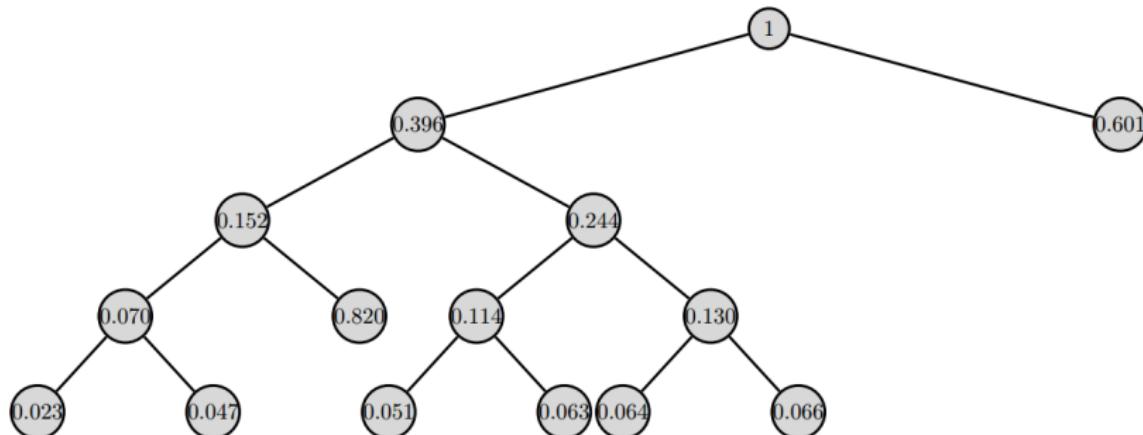
Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 1 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244	0.396
3	0.066	0.070	0.082	0.114	0.130	0.152	
2	0.064	0.066	0.070	0.082	0.114		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047						
6	0.023						

Phương pháp nén ảnh Huffman

Xây dựng cây Huffman sau bước 1 như sau:



Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601	0.601	0.601 0
4	0.082	0.082	0.114	0.130	0.152	0.244	0.396 1
3	0.066	0.070	0.082	0.114	0.130	0.152	
2	0.064	0.066	0.070	0.082	0.114		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244	0.396
3	0.066	0.070	0.082	0.114	0.130	0.152	0.396
2	0.064	0.066	0.070	0.082	0.114		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601 0	0.601 0	0.601 0
4	0.082	0.082	0.114	0.130	0.152 11	0.244 10	0.396 1
3	0.066	0.070	0.082	0.114	0.130 100	0.152 11	
2	0.064	0.066	0.070	0.082	0.114 101		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601 0	0.601 0	0.601 0	0.601 0
4	0.082	0.082	0.114	0.130 100	0.152 11	0.244 10	0.396 1
3	0.066	0.070	0.082	0.114 101	0.130 100	0.152 11	
2	0.064	0.066	0.070	0.082 110	0.114 101		
1	0.063	0.064	0.066	0.070 111			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6	
7	0.601	0.601	0.601	0	0.601	0	0.601	0
4	0.082	0.082	0.114	101	0.130	100	0.152	11
3	0.066	0.070	0.082	110	0.114	101	0.130	100
2	0.064	0.066	0.070	111	0.082	110	0.114	101
1	0.063	0.064	0.066	1000	0.070	111		
0	0.051	0.063	0.064	1001				
5	0.047	0.051						
6	0.023							

Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601 0	0.601 0	0.601 0	0.601 0	0.601 0	0.601 0
4	0.082	0.082 110	0.114 101	0.130 100	0.152 11	0.244 10	0.396 1
3	0.066	0.070 111	0.082 110	0.114 101	0.130 100	0.152 11	
2	0.064	0.066 1000	0.070 111	0.082 110	0.114 101		
1	0.063	0.064 1001	0.066 1000	0.070 111			
0	0.051	0.063 1010	0.064 1001				
5	0.047	0.051 1011					
6	0.023						

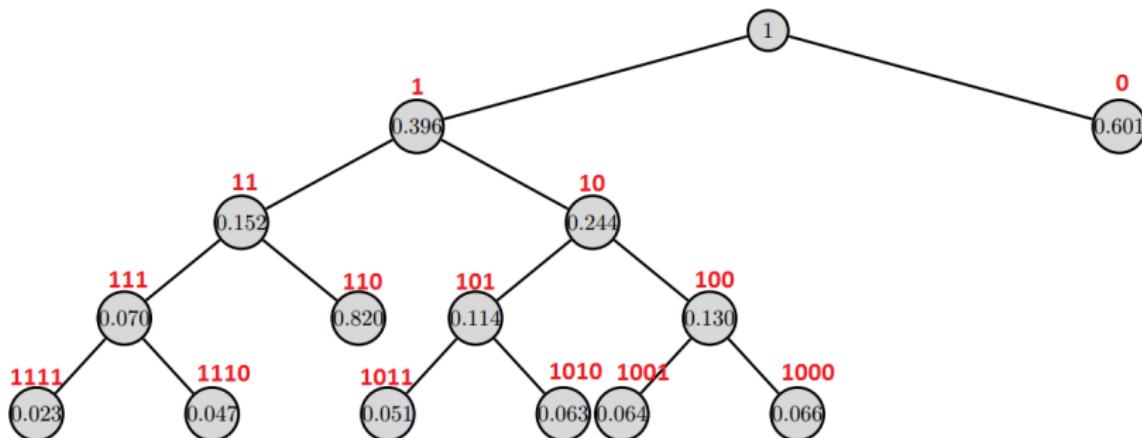
Phương pháp nén ảnh Huffman

Quá trình thực hiện bước 2 của phương pháp Huffman như sau:

r_k	$p(r_k)$	code	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0	0.601	0	0.601	0	0.601	0
4	0.082	110	0.082	110	0.114	101	0.130	100
3	0.066	1000	0.070	111	0.082	110	0.114	101
2	0.064	1001	0.066	1000	0.070	111	0.082	110
1	0.063	1010	0.064	1001	0.066	1000	0.070	111
0	0.051	1011	0.063	1010	0.064	1001		
5	0.047	1110	0.051	1011				
6	0.023	1111						

Phương pháp nén ảnh Huffman

Gán mã cho cây Huffman như sau:



Phương pháp nén ảnh Huffman

Giá trị trung bình biểu diễn cho mỗi mức xám là:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k)p(r_k) = 2.1bit$$

Dung lượng của ảnh I:

$$C = L_{tb} * \sum_{k=0}^{L-1} n(r_k) = 2.1 * 2200 = 4620bit$$

Phương pháp nén ảnh Huffman

Bài tập áp dụng: Cho ảnh I 2 bit, kích thước 5×5 được biểu diễn bởi ma trận sau. Hãy nén ảnh I theo phương pháp Huffman, tính dung lượng ảnh I trước và sau khi nén.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 & 0 \\ 0 & 2 & 3 & 1 & 0 \\ 0 & 1 & 0 & 0 & 2 \end{bmatrix}$$

Phương pháp nén ảnh Huffman

Kết quả sau khi thực hiện:

r_k	$n(r_k)$	$p(r_k)$	mã 1	mã 2
0	13	0.52	00	0
1	7	0.28	01	10
2	3	0.12	10	110
3	2	0.08	11	111

- Dung lượng ảnh I trước khi nén:

$$C_1 = L_{tb} * M * N = 2 * 5 * 5 = 50\text{bit}$$

- Dung lượng ảnh I sau khi nén:

$$C_2 = L_{tb} * M * N = 1.68 * 5 * 5 = 42\text{bit}$$

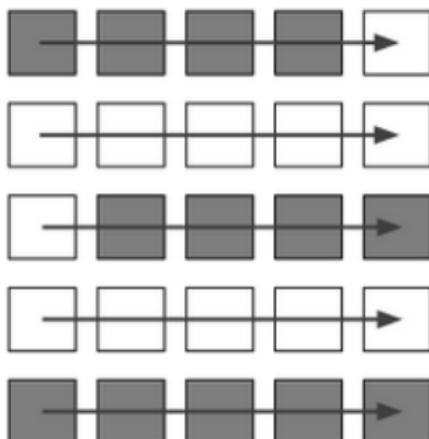
Phương pháp nén mã mạch dài

- Được đề xuất từ những năm 1950, lúc đầu được phát triển cho nén ảnh đen trắng.
- Nguyên tắc của phương pháp là phát hiện một loạt các bít lặp lại. Sau đó, thay thế chuỗi cũ bởi một chuỗi mới gồm 2 thông tin: chiều dài chuỗi và bít lặp (ký tự lặp).
- Phương pháp này chỉ có hiệu quả khi chiều dài dãy lặp lớn hơn một ngưỡng nào đó.
- Chuỗi thay thế sẽ có chiều dài ngắn hơn chuỗi cần thay.
- Phương pháp này được sử dụng trong việc mã hóa lưu trữ các ảnh Bitmap theo dạng PCX, BMP.

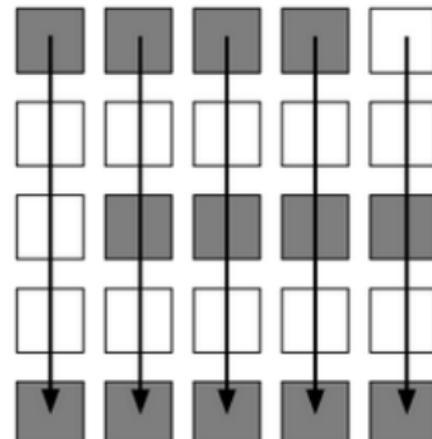
Phương pháp nén mã mạch dài

- Đối với ảnh, có thể có các thứ tự duyệt sau:

Duyệt hiệu quả



Duyệt không hiệu quả



Phương pháp nén mã mạch dài

Ví dụ: Cho ảnh nhị phân I có ma trận ảnh (10×10) như sau. Dùng phương pháp mã hóa mạch dài để nén ảnh. Cho biết tỷ số nén.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Phương pháp nén mã mạch dài

- Ảnh trước khi nén có kích thước là 100 bit.
- Sử dụng mã hóa mạch dài, ta có chuỗi sau khi nén:
(23,0), (4,1), (6,0), (1,1), (2,0), (1,1), (6,0), (4,1), (9,0), (1,1),
(9,0), (1,1), (6,0), (4,1), (23,0).
- Có tất cả 15 cặp. Dùng 5 bit để mã hóa số lượng các bit lặp, dùng 1 bit để mã hóa 2 giá trị 0 hoặc 1. Như vậy, độ dài sau khi mã hóa là: $15 * (5 + 1) = 90$ bit.
- Tỷ số nén: $C_r = \frac{100}{90} = 1.1$
- Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r} = 1 - \frac{90}{100} = 0.1 = 10\%$

Phương pháp nén mã mạch dài

Bài tập áp dụng: Cho ánh xám 3 bit l kích thước 8×8 có ma trận như sau. Dùng phương pháp mã hóa mạch dài để nén ảnh. Cho cho biết tỷ số nén.

0	0	1	1	1	1	1	1
1	1	1	2	2	2	2	2
2	2	2	2	3	3	3	3
4	4	5	5	5	6	6	6
6	6	6	6	6	6	6	6
6	7	7	7	7	7	7	7
7	0	7	0	7	0	7	0
0	0	0	0	0	0	0	0

Phương pháp nén mã mạch dài

- Ảnh trước khi nén có kích thước là $8*8*3 = 192$ bit.
- Sử dụng mã hóa mạch dài, ta có chuỗi sau khi nén:
 $(2,0), (9,1), (9,2), (4,3), (2,4), (3,5), (12,6), (8,7), (1,0), (1,7),$
 $(1,0), (1,7), (1,0), (1,7), (9,0).$
- Có tất cả 15 cặp. Dùng 4 bit để mã hóa số lượng các bit lặp, dùng 3 bit để mã hóa các mức xám từ 0 đến 7. Như vậy, độ dài sau khi mã hóa là: $15*(4+3) = 105$ bit.
- Tỷ số nén: $C_r = \frac{192}{105} = 1.829$
- Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r} = 1 - \frac{105}{192} = 0.453 = 45.3\%$

Phương pháp nén LZW

- Khái niệm nén dựa trên từ điển được Jocob Lempe và Abraham Ziv đưa ra vào năm 1977.
- Năm 1984 Welch đã cải tiến giải thuật LZ thành giải thuật hiệu quả hơn đặt tên là LZW. Là phương pháp nén không mất thông tin.
- Phương pháp này xây dựng từ điển các chuỗi pixel có tần suất lặp lại cao và thay thế bằng từ mã mỗi khi lặp lại chúng.
- LZW được dùng làm chuẩn nén cho các dạng ảnh GIF và TIFF.

Phương pháp nén LZW

- Phần quan trọng nhất của phương pháp nén này là phải tạo một mảng rất lớn dùng để lưu giữ các chuỗi pixel đã gấp, mảng này được gọi là "Từ điển".
- Từ mã từ 0 đến 255 chứa các số nguyên từ 0 đến 255.
- Từ mã thứ 256 là mã đặc biệt. Để khắc phục trường hợp số mẫu lặp trong ảnh lớn hơn 4096. Mã xóa sẽ cho biết việc kết thúc mã hóa với từ điển cũ và bắt đầu bộ tự điển mới.
- Từ mã thứ 257 chứa mã kết thúc thông tin. Giúp phân chia file thành nhiều cụm ảnh.
- Các từ mã còn lại từ 258 đến 4095 là các mẫu lặp lại trong ảnh.

Phương pháp nén LZW

Cấu trúc từ điển:

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

Phương pháp nén LZW

- 0 đến 255 được biểu diễn bởi 8 bit.
- 256 đến 511 được biểu diễn bởi 9 bit.
- 512 đến 1023 được biểu diễn bởi 10 bit.
- 1024 đến 2047 được biểu diễn bởi 11 bit.
- 2048 đến 4095 được biểu diễn bởi 12 bit.

Phương pháp nén LZW

Quá trình nén ảnh:

Bước 1: Đọc ảnh, lưu ma trận ảnh vào mảng 1 chiều

Bước 2: Khởi tạo dãy pixel S, từ điển D và dãy pixel đầu ra O

Duyệt mảng chứa các pixel ảnh

- Nếu pixel đang xét là dấu hiệu kết thúc
 - o Đẩy các pixel của S ra ngoài (gán cho dãy O)
- Ngược lại dãy pixel đó vào cuối dãy S
 - o Nếu S chứa pixel chưa có trong từ điển D
 - Thêm S vào trong từ điển D
 - Đẩy pixel (hoặc dãy pixel trừ pixel cuối có trong dãy S) ra ngoài (gán cho dãy O)
 - o Nếu S chứa pixel đã có trong từ điển
 - Quay lén, tiếp tục vòng lặp

Phương pháp nén LZW

Ví dụ: Cho ảnh xám I 8 bit kích thước 3×5 có ma trận như sau.
Dùng phương pháp LZW để nén ảnh I. Cho biết tỷ số nén.

$$I = \begin{bmatrix} 72 & 69 & 76 \\ 76 & 79 & 72 \\ 69 & 76 & 76 \\ 79 & 72 & 69 \\ 76 & 76 & 79 \end{bmatrix}$$

Phương pháp nén LZW

- Ảnh trước khi nén có dung lượng $3*5*8 = 120$ bit.
- Chuỗi mức xám của ảnh ban đầu:
72-69-76-76-79-72-69-76-76-79-72-69-76-76-79.
- Xây dựng từ điển theo LZW như sau:

Phương pháp nén LZW

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	72			
2	72	69	72-69	258	72
3	69	76	69-76	259	69
4	76	76	76-76	260	76
5	76	79	76-79	261	76
6	79	72	79-72	262	79
7	72	69	72-69	đã có 258	
8	72-69	76	72-69-76	263	72-69 (258)
9	76	76	76-76	đã có 260	
10	76-76	79	76-76-79	264	76-76 (260)
11	79	72	79-72	đã có 262	
12	79-72	69	79-72-69	265	79-72 (262)
13	69	76	69-76	đã có 259	
14	69-76	76	69-76-76	266	69-76 (259)
15	76	79	76-79	đã có 261	
16	76-79	#			76-79 (261)

Phương pháp nén LZW

- Chuỗi mức xám của ảnh ban đầu:
72-69-76-76-79-**72-69-76-76-79-72-69-76-76-79**.
- Chuỗi mức xám của ảnh ban đầu sau khi nén như sau:
72-69-76-76-79-**258-260-262-259-261**.
- Dung lượng sau khi nén: $5*8 + 5*9 = 85$ bit.
- Tỷ số nén: $C_r = \frac{120}{85} = 1.412$
- Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r} = 1 - \frac{85}{120} = 0.2916 = 29.17\%$

Phương pháp nén LZW

Quá trình giải nén:

Số thứ tự	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	72			
2	72	69	72-69	258	72
3	69	76	69-76	259	69
4	76	76	76-76	260	76
5	76	79	76-79	261	76
6	79	258	79-72	262	79
7	258	260	72-69-76	263	72-69
8	260	262	76-76-79	264	76-76
9	262	259	79-72-69	265	79-72
10	259	261	69-76-76	266	69-76
11	261	#			76-79

Phương pháp nén LZW

Bài tập áp dụng 1: Cho ảnh xám 8 bit l kích thước 5×5 có ma trận như sau. Dùng phương pháp LZW để nén ảnh. Cho cho biết tỷ số nén, dữ liệu dư thừa.

$$\begin{bmatrix} 72 & 79 & 74 & 76 & 72 \\ 79 & 74 & 76 & 72 & 79 \\ 74 & 76 & 72 & 79 & 74 \\ 76 & 72 & 79 & 74 & 76 \\ 72 & 79 & 74 & 76 & 72 \end{bmatrix}$$

Phương pháp nén LZW

Stt	Dãy hiện tại	Pixel kế tiếp	Tử điển		Đầu ra
			Tử	Mã	
1	Null	72			
2	72	79	72-79	258	72
3	79	74	79-74	259	79
4	74	76	74-76	260	74
5	76	72	76-72	261	76
6	72	79	72-79	đã có 258	
7	72-79	74	72-79-74	262	72-79 (258)
8	74	76	74-76	đã có 260	
9	74-76	72	74-76-72	263	74-76 (260)
10	72	79	72-79	đã có 258	
11	72-79	74	72-79-74	đã có 262	
12	72-79-74	76	72-79-74-76	264	72-79-74 (262)
13	76	72	76-72	đã có 261	
14	76-72	79	76-72-79	265	76-72 (261)
15	79	74	79-74	đã có 259	
16	79-74	76	79-74-76	266	79-74 (259)
17	76	72	76-72	đã có 261	
18	76-72	79	76-72-79	đã có 265	
19	76-72-79	74	76-72-79-74	267	76-72-79 (265)
20	74	76	74-76	đã có 260	
21	74-76	72	74-76-72	đã có 263	
22	74-76-72	79	74-76-72-79	268	74-76-72 (263)
23	79	74	79-74	đã có 259	
24	79-74	76	79-74-76	đã có 266	
25	79-74-76	72	79-74-76-72	269	79-74-76 (266)
26	72	#			72

Phương pháp nén LZW

- Chuỗi mức xám của ảnh ban đầu:
72-79-74-76-**72-79**-**74-76**-**72-79**-**74-76-72**-**79-74-76**-**72-79-74-76**-
76-72-79-74-76-72.
 - Chuỗi mức xám của ảnh ban đầu sau khi nén như sau:
72-79-74-76-**258-260**-**262-261**-**259-265**-**263-266**-**72**.
 - Dung lượng ảnh trước khi nén: $5*5*8 = 200$ bit.
 - Dung lượng ảnh sau khi nén: $5*8 + 8*9 = 112$ bit.
 - Tỷ số nén: $C_r = \frac{200}{112} = 1.786$
 - Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r} = 1 - \frac{112}{200} = 0.44 = 44\%$

Phương pháp nén LZW

Quá trình giải nén:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	72			
2	72	79	72-79	258	72
3	79	74	79-74	259	79
4	74	76	74-76	260	74
5	76	258	76-72	261	76
6	258	260	72-79-74	262	72-79
7	260	262	74-76-72	263	74-76
8	262	261	72-79-74-76	264	72-79-74
9	261	259	76-72-79	265	76-72
10	259	265	79-74-76	266	79-74
11	265	263	76-72-79-74	267	76-72-79
12	263	266	74-76-72-79	268	74-76-72
13	266	#			79-74-76

Phương pháp nén LZW

Bài tập áp dụng 2: Cho ánh xám 8 bit l kích thước 4×4 có ma trận như sau. Dùng phương pháp LZW để nén ảnh. Cho cho biết tỷ số nén, dữ liệu dư thừa.

$$\begin{bmatrix} 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \end{bmatrix}$$

Phương pháp nén LZW

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	39			
2	39	39	39-39	258	39
3	39	126	39-126	259	39
4	126	126	126-126	260	126
5	126	39	126-39	261	126
6	39	39	39-39	đã có 258	
7	39-39	126	39-39-126	262	39-39 (258)
8	126	126	126-126	đã có 260	
9	126-126	39	126-126-39	263	126-126 (260)
10	39	39	39-39	đã có 258	
11	39-39	126	39-39-126	đã có 262	
12	39-39-126	126	39-39-126-126	264	39-39-126 (262)
13	126	39	126-39	đã có 261	
14	126-39	39	126-39-39	265	126-39 (261)
15	39	126	39-126	đã có 259	
16	39-126	126	39-126-126	266	39-126 (259)
17	126	#			126

Phương pháp nén LZW

- Chuỗi mức xám của ảnh ban đầu:
39-39-126-126-39-39-126-126-39-39-126-126-39-39-126-126.
- Chuỗi mức xám của ảnh ban đầu sau khi nén như sau:
39-39-126-126-258-260-262-261-259-126.
- Dung lượng ảnh trước khi nén: $4*4*8 = 128$ bit.
- Dung lượng ảnh sau khi nén: $5*8 + 5*9 = 85$ bit.
- Tỷ số nén: $C_r = \frac{128}{85} = 1.506$
- Dữ liệu dư thừa: $D_r = 1 - \frac{1}{C_r} = 1 - \frac{85}{128} = 0.336 = 33.6\%$

Phương pháp nén LZW

Quá trình giải nén:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	39			
2	39	39	39-39	258	39
3	39	126	39-126	259	39
4	126	126	126-126	260	126
5	126	258	126-39	261	126
6	258	260	39-39-126	262	39-39
7	260	262	126-126-39	263	126-126
8	262	261	39-39-126-126	264	39-39-126
9	261	259	126-39-39	265	126-39
10	259	126	39-126-126	266	39-126
11	126	#			126