1. Cho ảnh “plumage.png”, chuyển ảnh này thành ảnh xám, gọi là I:

a) Gọi h = [1 2 4 2 1]/10.

Thực hiện các phép toán sau

I1 = I \* h (\* thể hiện phép tích chập)

I2 = I1 \* h’, với h’ là chuyển vị của h.

Hiển thị ảnh I2.

b) Gọi

Thực hiện các phép toán sau

I3 = I\*H

Hiển thị ảnh I3.

So sánh I2 và I3. Nhận xét.

c) So sánh thời gian thực hiện giữa hai cách làm ở câu a và b (có thể chạy nhiều lần và tính thời gian trung bình). Nhận xét và giải thích.

Nên thực hiện theo cách nào trong hai cách thực hiện a và b?

d) Vẽ phổ biên độ của ảnh I theo tần số (u,v) với . Từ đó, nhận xét về kết quả câu a, b.

**Code:**

import cv2

import numpy as np

from matplotlib import pyplot as plt

import time

# Đọc ảnh đầu vào

I = cv2.imread("plumage.png", 0)

# Kernel h

h = np.array([[1, 2, 4, 2, 1]]) / 10

h\_phay = np.transpose(h)

# Kernel H

H = np.array([[1, 2, 4, 1, 2], [2, 4, 8, 4, 2], [4, 8, 16, 8, 4], [2, 4, 8, 4, 2], [1, 2, 4, 2, 1]]) / 100

# câu a

I1 = cv2.filter2D(I, -1, h)

I2 = cv2.filter2D(I1, -1, h\_phay)

# câu b

I3 = cv2.filter2D(I, -1, H)

# Biến đổi Fourier của ảnh

f\_img = np.fft.fft2(I)

f\_img\_shifted = np.fft.fftshift(f\_img)

# Tính phổ biên độ

magnitude\_spectrum = 20 \* np.log(np.abs(f\_img\_shifted))

# Lấy kích thước của ảnh

rows, cols = I.shape

# Tính tần số u, v

u = np.fft.fftfreq(rows)

v = np.fft.fftfreq(cols)

# Vẽ phổ biên độ

plt.subplot(2, 2, 1), plt.imshow(I, cmap='gray')

plt.title('I'), plt.xticks([]), plt.yticks([])

plt.subplot(2, 2, 2), plt.imshow(I1, cmap='gray')

plt.title('I1'), plt.xticks([]), plt.yticks([])

plt.subplot(2, 2, 3), plt.imshow(I2, cmap='gray')

plt.title('I2'), plt.xticks([]), plt.yticks([])

plt.subplot(2, 2, 4), plt.imshow(I3, cmap='gray')

plt.title('I3'), plt.xticks([]), plt.yticks([])

# Vẽ phổ biên độ

plt.figure()

plt.imshow(magnitude\_spectrum, cmap='gray')

plt.title('Magnitude Spectrum')

plt.xlabel('u')

plt.ylabel('v')

plt.xticks([0, cols], [-0.5, 0.5])

plt.yticks([0, rows], [-0.5, 0.5])

plt.colorbar()

plt.show()

**Câu a:**

Close-up of a bird's feather

Description automatically generated

Kết quả ảnh I2

**Câu b:**

A close-up of a grey feather

Description automatically generated

Close-up of a bird's wing

Description automatically generated

Kết quả ảnh I3

**Nhận xét:**

− I2 và I3 là 2 kết quả khác nhau sau khi áp dụng bộ lọc. I2 được tạo ra bằng

cách áp dụng bộ lọc giữa h và h’ lần lượt trên ảnh gốc, trong khi I3 được

tạo ra bằng cách áp dụng bộ lọc H.

− Ảnh I2 là kết quả của việc lọc thông tin bên trong ảnh để giảm nhiễu và làm mờ

cạnh, trong khi ảnh I3 là kết quả của việc áp dụng bộ lọc trung bình trên toàn bộ

ảnh để làm mịn ảnh.

− Nhìn chung, I2 được áp dụng để giảm nhiễu và I3 để làm mịn ảnh. Tuy

nhiên, I2 có xu hướng làm mất một số chi tiết nhỏ và cạnh, trong khi I3 làm

mất nhiều chi tiết hơn.

**Câu c:**

|  |  |  |
| --- | --- | --- |
| Thời gian (s) Câu | a | b |
| Lần 1 | 0.004448413848876953 | 0.0010039806365966797 |
| Lần 2 | 0.008519649505615234 | 0.0 |
| Lần 3 | 0.0030069351196289062 | 0.0005178451538085938 |
| Lần 4 | 0.0 | 0.0 |
| Lần 5 | 0.0010051727294921875 | 0.001984119415283203 |
| Lần 6 | 0.0 | 0.0 |
| Lần 7 | 0.0 | 0.004530429840087891 |
| Lần 8 | 0.0 | 0.004386425018310547 |
| Lần 9 | 0.0 | 0.004903078079223633 |
| Lần 10 | 0.0007328987121582031 | 0.0007977485656738281 |
| Lần 11 | 0.0004162788391113281 | 0.001558065414428711 |
| Lần 12 | 0.0 | 0.0 |
| Lần 13 | 0.0003895759582519531 | 0.0 |
| Lần 14 | 0.0004181861877441406 | 0.0010066032409667969 |
| Lần 15 | 0.000514984130859375 | 0.0 |
| Trung bình | 0.001296806335 | 0.001379219691 |

*Bảng kết quả thời gian chạy 2 bộ lọc*

Dựa trên kết quả thời gian thực hiện và kết quả hiển thị của hai cách thực hiện a và b, ta có thể đưa ra nhận xét và giải thích như sau:

Thời gian thực hiện:

* Phương pháp a: Thời gian thực hiện cho phép tích chập theo cách a là thời gian tích chập hai lần với các kernel h và h’. Do đó, thời gian thực hiện có thể tăng khi phải thực hiện hai phép tích chập liên tiếp.
* Phương pháp b: Thời gian thực hiện cho phép tích chập theo cách b là thời gian tích chập một lần với kernel H có kích thước lớn hơn. Điều này có thể giảm thời gian thực hiện so với phương pháp a.

Hiệu suất và đơn giản của mã:

* Phương pháp a: Tích chập hai lần với các kernel h và h’ có thể làm tăng độ phức tạp của mã và làm cho mã trở nên phức tạp hơn.
* Phương pháp b: Sử dụng một kernel H lớn có thể làm cho mã trở nên đơn giản hơn vì chỉ cần thực hiện một phép tích chập duy nhất.

Kết quả hiển thị:

Kết quả hiển thị của cả hai phương pháp có thể khá giống nhau, vì cả hai đều là kết quả của phép tích chập trên ảnh đầu vào. Tuy nhiên, kết quả có thể có sự khác biệt nhỏ do các đặc điểm của kernel.

Dựa trên những nhận xét trên, nếu hiệu suất là ưu tiên, phương pháp b có thể là lựa chọn tốt hơn vì có thể giảm thời gian thực hiện và làm cho mã trở nên đơn giản hơn. Tuy nhiên, nếu đơn giản hóa mã là ưu tiên hoặc nếu kích thước của kernel có sự biến đổi, phương pháp a có thể được ưa chuộng hơn vì linh hoạt hơn trong việc xử lý các kernel khác nhau.

**Câu d:**

A greyscale shot of a black and white photo

Description automatically generated

Phổ biên độ của ảnh I

**Nhận xét:**

Kết quả cho ta thấy:

* Về hình dạng phổ thì phổ có hình dạng chủ yếu tựa như cánh chim.
* Qua hiển thị phổ biên độ của ảnh, có thể thấy rằng phổ biên độ tập trung chủ yếu vào ở các tần số thấp.

1. Phương pháp Democaising của Alleysson:

Trong các máy ảnh CCD mà chỉ sử dụng một chip cảm biến màu, chỉ có một độ nhạy màu duy nhất tại mỗi vị trí không gian. Với loại cảm biến này, thông tin không gian và màu sắc được kết hợp với nhau.

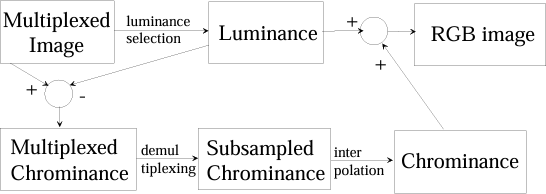
Spatial-chromatic sampling được đề xuất bởi David Alleysson, là một phương pháp tiếp cận để tách luminance và chrominance từ hình ảnh màu bằng cách sử dụng tính chất của các cảm biến màu. Phương pháp demosacing của Alleyson được đề xuất để tái tạo ảnh màu từ dữ liệu ảnh đơn sắc (ảnh grayscale) được bắt đầu bằng việc chuyển đổi dữ liệu ảnh màu từ các cảm biến hình ảnh màu (RGB) thành dữ liệu ảnh đơn sắc. Điểm chính của phương pháp demosaicing của Alleyson là khả năng kết hợp hiệu quả giữa việc sử dụng thông tin từ các pixel lân cận và việc áp dụng các kỹ thuật nâng cao để tái tạo màu sắc chính xác. Thuật toán này sử dụng một số kỹ thuật như trung bình hóa, nội suy, hoặc hồi quy để ước lượng màu sắc cho các pixel trung gian.. Bằng cách này, phương pháp demosaicing của Alleyson có khả năng tái tạo hình ảnh màu chân thực và có độ chính xác cao, dù chỉ có thông tin màu sắc hạn chế từ cảm biến ảnh

Một điểm đáng chú ý của thuật toán Alleyson là khả năng xử lý các vấn đề phức tạp như nhiễu, biến dạng mờ và thiếu màu, giúp cải thiện chất lượng hình ảnh sau khi được tái tạo màu sắc. Điều này làm cho thuật toán trở thành một lựa chọn phổ biến trong việc xử lý hình ảnh số và trong các ứng dụng như nhiếp ảnh số và xử lý ảnh y tế.

Luminance và Chrominance là hai thành phần của màu sắc được sử dụng để mô tả màu sắc trong hệ thống màu sắc. Luminance là thành phần độ sáng của màu sắc, trong khi chrominance là thành phần độ bão hòa màu sắc. Luminance và chrominance được sử dụng để xác định màu sắc của một điểm ảnh trong hệ thống màu sắc, như RGB (Red, Green, Blue) hoặc YUV (Lu- minance, Chrominance Blue, Chrominance Red). Việc tách luminance và chrominance rất quan trọng trong xử lý ảnh và video để thực hiện các bước tiền xử lý như đồng bộ hóa màu sắc, cân bằng màu sắc, nâng cao chất lượng ảnh và video.

Trong hình ảnh màu, mỗi điểm ảnh có chứa thông tin về ba thành phần màu sắc (R, G, B) tại mỗi vị trí không gian. Tuy nhiên, các thông tin này bị trộn lẫn vào nhau và khó phân tách, làm cho việc xử lý hình ảnh màu trở nên phức tạp.

Phương pháp này giải quyết vấn đề này bằng cách sử dụng hai bộ lọc khác nhau để tách riêng luminance và chrominance từ ảnh gốc. Phương pháp này có 4 bước thực hiện là:



Hình 1: Sơ đồ giải thuật Alleysson

# Tách luminance (tính toán luminance):

Đầu vào để xử lý tách thành phần luminance là ảnh mosaic, với mô hình sắp xếp không gian của bộ lọc màu được gọi là Bayer CFA (xem hình 1). Để tách luminance theo phương pháp Spatial- chromatic sampling với ảnh theo mô hình Bayer CFA, ta cần làm theo các bước sau:

* + - Đọc ảnh màu theo mô hình Bayer CFA và tách nó thành các channel màu tương ứng. Trong mô hình Bayer CFA, mỗi pixel chỉ có thông tin về một trong ba màu đỏ, xanh hoặc lục.
    - Áp dụng bộ lọc trung bình trên ảnh màu để tính toán độ sáng trung bình của từng pixel. Bộ lọc trung bình được mô tả như phương trình (2). Kết quả sẽ tạo ra một ảnh độ sáng trung bình (luminance) với kích thước giống với ảnh gốc.

# /64 (2)

# Tách Chrominance:

Để tách chrominance theo phương pháp Spatial-chromatic sampling với ảnh theo mô hình Bayer CFA, ta cần làm theo các bước sau:

* + - * Đọc ảnh màu theo mô hình Bayer CFA và tách nó thành các channel màu tương ứng. Trong mô hình Bayer CFA, mỗi pixel chỉ có thông tin về một trong ba màu đỏ, xanh hoặc lục.
      * Thành phần chrominance được tính bằng cách lấy ảnh màu theo mô hình Bayer CFA trừ đi thành phần Luminance (L) đã tính được ở bước đầu tiên 2.1:

Chrominance = {R-L;G-L;B-L} (3)

# Nội suy tái tạo lớp Chrominance:

Sau khi tính toán được các giá trị chrominance, ta cần khôi phục lại các giá trị chrominance còn thiếu trong ảnh theo mô hình Bayer CFA. Quá trình này được thực hiện bằng cách sử dụng một thuật toán tối ưu hóa để dự đoán các giá trị màu bị thiếu dựa trên các giá trị màu có sẵn của các pixel láng giềng. Thuật toán này tương tự phương pháp nội suy tuyến tính, sử dụng các bộ lọc tích chập ở phương trình sau để khôi phục các giá trị chrominance theo mô hình CFA:

FR,B = /4; FG = /4;

Trong đó:

FR,B và FG là hai bộ lọc nội suy tuyến tính cho các lưới màu đỏ và xanh dương, và lưới màu xanh lục, tương ứng.

# Kết hợp Luminance và chrominance sau khi tái tạo để tạo ảnh màu hoàn chỉnh:

Các giá trị Chrominance sau khi khôi phục sẽ bố trí theo mô hình Bayer CFA, để tái tạo lại ảnh màu hoàn hình, cần kết hợp thành phần luminance và thành phần chrominance sau khi khôi phục.

**Kết quả thực nghiệm:**

A white lighthouse with a white fence

Description automatically generated

1. Giải thuật Democaising của Laroche:
2. Mô tả ngắn gọn giải thuật Democaising của Laroche: Phương pháp của Laroche phu thuộc vào loại hướng cạnh trong việc phục hồi màu từ dữ liệu Bayer. Thực tế, nó xem xét hướng của cạnh để giảm thiểu sai số nội suy, tức là nội suy được thực hiện dọc theo cạnh. Sự nội suy như vậy được áp dụng trước tiên vào kênh màu Xanh lá. Khi kênh màu Xanh lá được hoàn thành, các kênh màu Đỏ/ Xanh lá cây được tái tạo dựa trên giả thuyết về sự khác biệt màu sắc không đổi, mà quy định rằng sự khác biệt màu (ví dụ giữa màu Xanh lá và Đỏ hoặc màu Xanh lá và Xanh lam) của một đối tượng là không đổi.

A red and white arrow pointing to a red square

Description automatically generated

A screenshot of a computer

Description automatically generated

A blue arrow pointing to a line

Description automatically generated

A blue arrow pointing to a line

Description automatically generated