

Bài thực hành 2 - Xử lý ảnh

Lớp N01 - Năm học 2021/2022

September 2021

1 Hướng dẫn

- Tải file cần thiết để thực hành tại:

https://drive.google.com/file/d/1lv2nHTdQLdRrYW7w0y6x0gm_JcpnrIFc/view?usp=sharing

- Mở MATLAB và tạo một thư mục mới để làm việc. Giải nén thư mục vừa tải vào thư mục mới đó.
- Sau khi hoàn thành bài tập, lưu kết quả tất cả bài tập của phần 2.2, 3.2 (mỗi câu hỏi sẽ có hướng dẫn cụ thể) vào một báo cáo (định dạng docx hoặc PDF) và nộp bài tập của mình trên Teams. Sinh viên nhớ kiểm tra đầy đủ các thao tác nộp bài.
- Sinh viên làm bài thực hành theo từng cá nhân.
- Hạn nộp bài: 23h59' ngày 19/09/2021. Các trường hợp nộp muộn, không hoàn thành thao tác nộp bài hoặc sao chép bài làm của sinh viên khác coi như không nộp bài.

2 Lọc trong miền không gian - Phát hiện đường viền

2.1 Nhắc lại kiến thức

Việc phát hiện đường viền trong ảnh thường liên quan đến việc xác định những sự thay đổi đột ngột về độ sáng giữa các điểm ảnh lân cận. Một phương pháp phổ biến trong phát hiện đường viền là sử dụng bộ lọc (tuyến tính hoặc phi tuyến), để làm nổi bật lên đường viền của ảnh.

Trong phần đầu của bài thực hành hôm nay, chúng ta sẽ tập trung vào các phương pháp làm nổi đường viền sử dụng các bộ lọc trong miền không gian $h(m, n)$ có kích cỡ $M_h \times N_h$. Trái ngược với các bộ lọc Gausse, lọc trung bình hay lọc trung vị ở bài thực hành trước, các bộ lọc sử dụng để làm nổi đường

viền là các bộ lọc thông cao (cho phép các tần số cao đi qua và làm giảm các tín hiệu tần số thấp).

Các bộ lọc được sử dụng trong bài hôm nay sẽ phát hiện sự thay đổi độ sáng đột ngột trong ảnh theo một hướng nhất định. Một vài bộ lọc cho biên độ lớn với các thay đổi từ sáng sang tối, trong khi một số khác có thể phát hiện thay đổi từ tối sang sáng, hoặc cả hai. Khi sử dụng các bộ lọc này, ảnh đầu vào đã chuẩn hoá (độ sáng nằm trong khoảng 0-1) có thể cho ảnh đầu ra có các giá trị âm hoặc lớn hơn 1. Để hiển thị các ảnh này chúng ta thường phải chuẩn hoá lại chúng bằng cách lấy giá trị tuyệt đối của điểm ảnh, hoặc bằng hàm `mat2gray`.

2.2 Bài tập

Lưu ý: Sinh viên nên lưu các câu lệnh của mình vào một script `.m` để tiện sử dụng lại khi cần.

1. Tạo một ảnh **X** có kích thước 256×256 pixels gồm một hình vuông có kích thước 50×50 pixels màu xám sáng trên nền màu xám đậm. Hình vuông được căn chỉnh sao cho nằm chính giữa ảnh. Chuyển **X** về dạng double và chuẩn hoá **X**. Đưa code và hình ảnh vào báo cáo.

2. Lọc Gradients

$$h1 = \begin{bmatrix} 1 & -1 \end{bmatrix} \quad h2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- (a) Áp dụng bộ lọc **h1** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y1**. Hiển thị `mat2gray(Y1)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h1** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?
- (b) Áp dụng bộ lọc **h2** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y2**. Hiển thị `mat2gray(Y2)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h2** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?

3. Bộ lọc Sobel

$$h3 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad h4 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad h5 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h6 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

- (a) Áp dụng bộ lọc **h3** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y3**. Hiển thị `mat2gray(Y3)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h3** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?
- (b) Áp dụng bộ lọc **h4** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y4**. Hiển thị `mat2gray(Y4)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h4** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?

- (c) Áp dụng bộ lọc **h5** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y5**. Hiển thị `mat2gray(Y5)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h5** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?
- (d) Áp dụng bộ lọc **h6** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y6**. Hiển thị `mat2gray(Y6)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h6** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?

4. Lọc Laplace

$$h7 = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Áp dụng bộ lọc **h7** đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y7**. Hiển thị `mat2gray(Y7)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h7** này làm nổi đường viền theo hướng nào?

5. Lọc LOG (Lọc Marr-Hildreth) Để tạo một bộ lọc LOG trong MATLAB chúng ta sử dụng hàm `fspecial` như sau:

```
h = fspecial('log', [5 5], 1); % Bộ lọc LOG có
    kích thước 5x5 và phương sai 1
```

- (a) Áp dụng bộ lọc LOG **h8** có kích cỡ 15×15 và phương sai 1 đối với ảnh **X** và lưu ảnh đầu ra trong ma trận **Y8**. Hiển thị `mat2gray(Y8)`. Đưa code và đồ thị vào báo cáo. Bộ lọc **h8** này làm nổi đường viền theo hướng nào? Bộ lọc này sử dụng tốt cho thay đổi từ sáng sang tối/ từ tối sang sáng hay cả hai?
 - (b) Thử thay đổi phương sai và nhận xét.
 - (c) So sánh kết quả của bộ lọc LOG và lọc Laplace.
6. Thử áp dụng toàn bộ các bộ lọc trên với ảnh đầu vào là ảnh **cameraman.tif** đã được chuẩn hoá. Nhận xét kết quả thu được.
7. Sử dụng hàm `ShowFilter(X,h)` được cung cấp trong file `ShowFilter.m` để hiển thị lên màn hình phổ tần số của ảnh đầu vào, ảnh đầu ra và bộ lọc. Nhận xét phổ tần số của các bộ lọc từ **h1** đến **h8**. Lưu ý: ảnh đầu ra hiển thị ở đây chưa được hậu xử lý (chưa đi qua hàm `abs` và `mat2gray`) nên các điểm ảnh có giá trị âm sẽ được hiển thị màu đen và các điểm ảnh có giá trị lớn hơn 1 sẽ được hiển thị màu trắng).

Hàm `ShowFilter(X,h)` thực hiện những công việc sau:

- Đầu vào là một ảnh X đã chuẩn hoá và một bộ lọc tích chập h .
- Tính ảnh đã lọc Y .
- Hiển thị trên 1 đồ thị các ảnh sau (từ trái sang phải, từ trên xuống dưới):
 - Ảnh đầu vào X .
 - Phổ tần số của X trên thang tuyến tính.
 - Phổ tần số của Y trên thang tuyến tính.
 - Phổ tần số của bộ lọc h trên thang tuyến tính.
 - Ảnh đầu ra Y .
 - Phổ tần số của X trên thang log.
 - Phổ tần số của Y trên thang log.
 - Phổ tần số của bộ lọc h trên thang log.

3 Lọc trong miền tần số

3.1 Nhắc lại kiến thức

Việc khai triển Fourier cho ảnh có thể cho chúng ta biết thông tin về biên độ và tần số của các tín hiệu xuất hiện trên ảnh. Từ phổ tần số đó, chúng ta có thể dễ dàng nhận ra và lọc đi các tần số không mong muốn, cũng như thiết kế ra các bộ lọc thông thấp, lọc thông cao cho miền tần số.

Trong matlab, để có được phổ tần số của một ảnh X và hiển thị biên độ của phổ tần số đó lên màn hình trên thang log, ta có thể dùng khai triển Fourier nhanh như sau:

```
F = fftshift(fft2(X))
imshow(mat2gray(log(abs(F))))
```

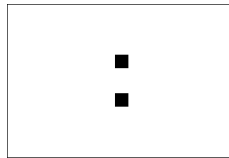
Để biến đổi ngược phổ tần số F thành ảnh, chúng ta dùng hai hàm ngược như sau:

```
X = ifft2(ifftshift(F))
```

3.2 Bài tập

1. Đọc file ảnh `saturn.png` và lưu file ảnh vào ma trận X . Chuyển X sang dạng double và chuẩn hoá X . Đưa code vào báo cáo.
2. Gọi hàm `AddPeriodicNoise` được cung cấp trong file `AddPeriodicNoise.m` với X để thêm nhiễu chu kỳ và lưu kết quả vào ảnh Y . Hàm này sẽ thêm một nhiễu chu kỳ theo phương dọc. Đưa code và ảnh Y vào báo cáo. Tính PSNR của Y so với X và đưa vào báo cáo.

3. Khai triển Fourier với Y , lưu kết quả vào ma trận FY và hiển thị biên độ của phổ tần số đó trên thang log. Đưa code và phổ tần số vào báo cáo. Nhận xét về điểm nhiễu trên phổ tần số.
4. Tạo một ma trận Z có cùng kích thước với Y và FY . Các giá trị của Z đều là 1. Đưa code vào báo cáo.
5. Thay đổi một số giá trị của ma trận Z thành 0 sao cho các giá trị đó che được đi điểm nhiễu trên phổ tần số đã nhìn thấy trên FY . Z sẽ được sử dụng như bộ lọc để lọc nhiễu. Đưa code và ảnh Z vào báo cáo. Sinh viên có thể thiết kế bộ lọc Z như hình dưới hoặc tự thiết kế một bộ lọc tùy ý.



Hình 1: Một ví dụ về bộ lọc Z . Các giá trị 0 có màu đen và giá trị 1 có màu trắng. Ô màu đen dùng để che đi tần số nhiễu không mong muốn trong phổ tần số của ảnh.

6. Nhân từng phần tử của phổ tần số FY với bộ lọc Z và lưu kết quả vào ma trận $FY2$. Trong matlab để thực hiện phép nhân từng phần tử của hai ma trận chúng ta sử dụng câu lệnh:

$$FY2 = FY .* Z$$

7. Biến đổi ngược phổ tần số $FY2$ để lấy lại ảnh và lưu ảnh đó vào ma trận $Y2$. Hiển thị ảnh cuối cùng $Y2$. Đưa ảnh và code vào báo cáo. Nhận xét về ảnh cuối cùng thu được. Tính PSNR của ảnh $Y2$ đó so với X và đưa vào báo cáo.