

# Bài thực hành 3 - Xử lý ảnh

Lớp N01 - Năm học 2021/2022

October 2021

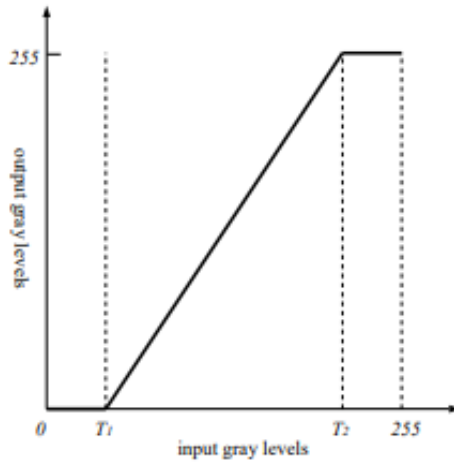
## 1 Hướng dẫn

- Tải file cần thiết để thực hành tại đây: [Click here](https://drive.google.com/file/d/1F3VPB49CbcPSbbUZZh_v0uEJKf2n4wPy/view?usp=sharing), hoặc copy link này vào trình duyệt: [https://drive.google.com/file/d/1F3VPB49CbcPSbbUZZh\\_v0uEJKf2n4wPy/view?usp=sharing](https://drive.google.com/file/d/1F3VPB49CbcPSbbUZZh_v0uEJKf2n4wPy/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, 4.2, 5.2 (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 03/10/2021. 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 Giãn độ tương phản (Contrast Stretching)

### 2.1 Nhắc lại kiến thức

Toán tử điểm là một hàm giúp ánh xạ cường độ của mỗi điểm ảnh đầu vào với một cường độ tương ứng trong điểm ảnh đầu ra. Một ví dụ đơn giản nhất của toán tử điểm là giãn độ tương phản. Giãn độ tương phản là một toán tử điểm cho phép giãn một khoảng độ xám  $[T_1, T_2]$  mà người dùng quan tâm và loại bỏ đi các khoảng độ xám không quan tâm. Cụ thể trong hình 1, các điểm có độ xám nằm trong khoảng  $[0, T_1]$  sẽ tương ứng với giá trị đầu ra là 0, các điểm có độ xám nằm trong khoảng  $[0, 255]$  sẽ tương ứng với giá trị đầu ra là 255, và các điểm có độ xám trong khoảng  $[T_1, T_2]$  sẽ được kéo giãn ra thành  $[0, 255]$ .



Hình 1: Đồ thị hàm giãn độ tương phản trong khoảng  $[T_1, T_2]$

Công thức cụ thể của hàm giãn độ tương phản là:

$$y = \begin{cases} 0 & \text{when } x < T_1 \\ \frac{255}{T_2 - T_1}(x - T_1) & \text{when } T_1 \leq x \leq T_2 \\ 255 & \text{when } x > T_2 \end{cases} \quad (1)$$

## 2.2 Bài tập

- Viết một hàm Matlab cho phép thực hiện hàm giãn độ tương phản được mô tả trong phần 2.1 và lưu vào file `imgStretch.m`. Lời khai báo hàm như sau:

```
output=imgStretch(input,T1,T2)
```

Gợi ý: Tạo một ma trận output có kích thước bằng ma trận input và tính giá trị từng điểm ảnh theo công thức (1) bằng vòng lặp. Ngoài cách này, sinh viên có thể sử dụng các toán tử ma trận của Matlab.

- Áp dụng hàm `imgStretch` cho ảnh `kids.tif` với các tham số  $T_1 = 70$  và  $T_2 = 180$ .

Yêu cầu trong báo cáo

- Mã lệnh của hàm `imgStretch`.
- Ảnh `kids.tif` trước và sau xử lý.
- Nhận xét về kết quả thu được.

## 3 Hiệu chỉnh Gamma (Gamma correction)

### 3.1 Nhắc lại kiến thức

Cường độ sáng hiển thị của một thiết bị thường không phải là hàm tuyến tính với tín hiệu nhận được. Ví dụ, một điểm ảnh có độ xám 200 sẽ không được hiển thị sáng gấp đôi một điểm ảnh có độ sáng 100. Hầu hết màn hình hiển thị độ sáng thực là một hàm mũ so với tín hiệu đầu vào. Với một màn hình CRT thông thường, độ sáng của điểm ảnh xấp xỉ với tín hiệu đầu vào mũ 2.5. Hệ số này gọi là  $\gamma$  của thiết bị.

Với một ảnh độ xám 8-bit, quan hệ giữa điểm ảnh hiển thị thực tế và điểm ảnh ban đầu như sau:

$$y = 255\left(\frac{x}{255}\right)^\gamma \quad (2)$$

Để màn hình có thể hiển thị lên độ xám đúng nhất, hình ảnh cần phải được tiền xử lý bằng một hiệu chỉnh gamma ( $\gamma$  correction), với một hệ số  $\gamma < 1$  sao cho giá trị của điểm ảnh hiển thị thực tế bằng với giá trị điểm ảnh ban đầu.

### 3.2 Bài tập

- Viết một hàm Matlab cho phép tiền xử lý hình ảnh bằng một hiệu chỉnh gamma sao cho ảnh sau xử lý được hiển thị đúng nhất. Hàm này nhận vào ảnh gốc và hệ số  $\gamma$  của màn hình.

`output=gammCorr(input,gamma)`

Chú ý: Hệ số  $\gamma$  mà hàm nhận vào là hệ số của màn hình hiển thị. Để tiền xử lý ảnh, chúng ta phải hiệu chỉnh ảnh với nghịch đảo của  $\gamma$ .

- Áp dụng hàm hiệu chỉnh vừa rồi với ảnh `dark.tif` và hệ số  $\gamma$  của màn hình là 2.2.

Yêu cầu trong báo cáo

- Mã lệnh của hàm `gammCorr`.
- Ảnh `dark.tif` trước và sau xử lý.
- Nhận xét về kết quả thu được.

## 4 Cắt bit-plane (Bit-plane slicing)

### 4.1 Nhắc lại kiến thức

Mức xám của mỗi điểm ảnh được mã hoá trên  $B$  bit và có thể biểu diễn các giá trị trong đoạn  $[0, 2^B - 1]$ . Với một ảnh độ xám 8 bit, chúng ta có thể trích chọn

từng bit của ảnh, ví dụ: với điểm ảnh có độ xám 243 (hệ nhị phân là 11110011), giá trị của nó ở bit-plane 1 là 1, bit-plane 2 là 1, bit-plane 3 là 0, ....

Trong các bit mã hoá, các bit bậc cao thường chứa nhiều thông tin hơn các bit bậc thấp.

## 4.2 Bài tập

1. Sử dụng hàm `bitPlaneSlicing` trong file `bitPlaneSlicing.m` được cung cấp để chia ảnh `dollar.tif` thành 8 bit-plane và hiển thị các bit-plane đó.
2. Nhận xét về các bit-plane thu được.
3. Thử khôi phục ảnh chỉ sử dụng bit-plane 8-7, sau đó sử dụng các bit-plane 8-7-6, và cuối cùng là các bit-plane 8-7-6-5. Nhận xét về kết quả thu được.

Hàm `bitPlaneSlicing` chia hình ảnh đầu vào `X` là một ảnh độ xám 8 bit thành 8 bit-plane riêng biệt `Y`. Để lấy được bit-plane thứ  $i$  chúng ta sử dụng câu lệnh:

```
Y(:, :, i)
```

Để tạo lại ảnh từ bit-plane 8-7, chúng ta có thể sử dụng câu lệnh:

```
b8*128+b7*64
```

Yêu cầu trong báo cáo

- Mã lệnh của tất cả các câu lệnh.
- Nhận xét.
- Ảnh được tạo lại từ các bit-plane được yêu cầu và nhận xét.

## 5 Cân bằng lược đồ xám (Histogram Equalization)

### 5.1 Nhắc lại kiến thức

Cân bằng lược đồ xám là một kỹ thuật tăng cường chất lượng ảnh thông dụng. Mục đích của kỹ thuật này là biến đổi hình ảnh đầu vào sao cho lược đồ xám của nó phân bố đều trên toàn bộ các mức xám. Với Matlab, chúng ta có thể sử dụng hàm `imhist` để xem lược đồ xám của ảnh và hàm `histeq` để thực hiện cân bằng lược đồ xám.

### 5.2 Bài tập

1. Hiển thị hình ảnh `kids.tif` cùng với lược đồ xám của nó. Nhận xét về lược đồ xám.

2. Cân bằng lược đồ xám của ảnh, hiển thị ảnh đầu ra và lược đồ xám của ảnh đầu ra. Nhận xét.

Yêu cầu trong báo cáo

- Mã lệnh của tất cả các câu lệnh.
- Ảnh đầu vào, ảnh đầu ra và hai lược đồ xám.
- Nhận xét về các lược đồ xám.