

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
THÀNH PHỐ HỒ CHÍ MINH**

-----\*\*-----  
\*

**TS. NGUYỄN THANH HẢI**

**GIÁO TRÌNH  
XỬ LÝ ẢNH**

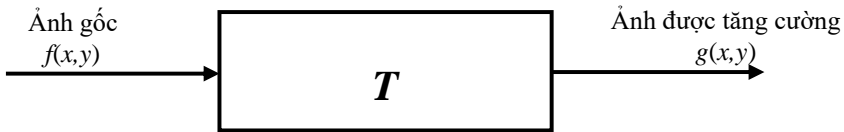
**(Ngành Điện - Điện Tử)**

**NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA  
THÀNH PHỐ HỒ CHÍ MINH – 2014**

# Chương 5

## TĂNG CƯỜNG ẢNH

Tăng cường ảnh là làm cho ảnh có chất lượng tốt hơn theo ý đồ sử dụng. Ảnh thu được thường bao gồm nhiều cần phải loại bỏ hoặc ảnh không sắc nét, bị mờ hoặc cần làm rõ các đặc trưng của ảnh như: thay đổi độ tương phản, giảm nhiễu, làm trơn biên, khuếch đại ảnh, điều khiển mức xám... Những kỹ thuật tăng cường và tái tạo được áp dụng cho một ảnh ngõ vào thông qua những thuật toán và cho ra ảnh tăng cường hay tái tạo ở ngõ ra, như trong hình 5.1.



**Hình 5.1.** Sơ đồ của kỹ thuật tăng cường ảnh

Những kỹ thuật tăng cường và tái tạo ảnh có thể được thực hiện trong miền điểm, miền không gian hay trong miền tần số. Những kỹ thuật xử lý điểm được thực hiện cho một điểm ảnh (pixel) trong ảnh gốc tại tọa độ  $(x,y)$  để cho ra một điểm ảnh tương ứng tại tọa độ  $(x,y)$  trong ảnh tăng cường. Điều này có nghĩa chỉ pixel trong ảnh gốc giữ vai trò trong việc tính toán giá trị pixel tương ứng trong ảnh tăng cường và những pixel này được xác định ở cùng một vị trí tọa độ trong ảnh gốc. Kỹ thuật tăng cường ảnh trong không gian hay xử lý mặt nạ được phân nhóm theo công dụng: làm trơn nhiễu, nổi biên. Để làm trơn ảnh hay tách nhiễu, người ta sử dụng các bộ lọc tuyến tính (lọc trung bình, thông thấp) hay lọc phi tuyến (trung vị, giả trung vị, lọc đồng hình). Để làm nổi biên (ứng với tần số cao), người ta dùng các bộ lọc thông cao, bộ lọc Laplace. Một số phép biến đổi có tính toán phức tạp được chuyển sang miền tần số để thực hiện, kết quả cuối cùng được chuyển trở lại miền không gian nhờ các biến đổi ngược.

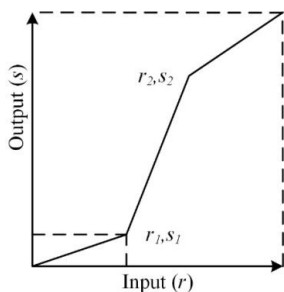
### 5.1. XỬ LÝ ĐIỂM

Xử lý điểm ảnh là biến đổi giá trị một điểm ảnh dựa vào giá trị của chính nó mà không hề dựa vào các điểm ảnh khác. Sự lựa chọn chính xác phép biến đổi nào phụ thuộc vào mục đích của vấn đề xử lý điểm. Xử lý điểm thường thực hiện trên một ảnh để cải thiện chất lượng của nó bằng việc đảm bảo dãy mức xám (gray-level) của nó.

### 5.1.1. Mở rộng độ tương phản

Một trong những hàm tuyến tính từng đoạn đơn giản nhất là biến đổi dạng mở rộng độ tương phản. Những ảnh có độ tương phản thấp là kết quả của quá trình chụp trong môi trường thiếu sáng hay có thể do chất lượng của cảm biến hình ảnh, hay thậm chí do lỗi của người sử dụng trong quá trình thiết lập thông số. Mở rộng độ tương phản là tiến trình làm tăng khoảng giá trị mức xám trong ảnh giúp ảnh hiển thị tốt hơn trên các thiết bị hiển thị trong môi trường ánh sáng mạnh.

Hình 5.2(a) biểu diễn ánh xạ cường độ mức xám của dạng mở rộng độ tương phản, trong đó có hai điểm  $(r_1, s_1)$  và  $(r_2, s_2)$  cho phép điều chỉnh hình dạng của hàm biến đổi. Nếu trường hợp  $r_1 = s_1$  và  $r_2 = s_2$  thì không có bất kỳ sự thay đổi nào tại ảnh ngõ ra. Còn nếu  $r_1 = r_2$ ,  $s_1 = 0$  và  $s_2 = L - 1 = 255$  thì ta có hàm lấy ngưỡng. Việc thay đổi các giá trị của hai điểm nêu trên cho phép tạo ra ảnh có độ tương phản thay đổi khác nhau. Trong hình 5.2(b) trình bày ảnh có độ tương phản thấp và tương ứng là ảnh sau khi được mở rộng trong hình 5.2(c) với thông số  $(r_1, s_1) = (r_{\min}, 0)$  và  $(r_2, s_2) = (r_{\max}, L - 1)$  với  $r_{\min}$  và  $r_{\max}$  đánh dấu giá trị mức xám nhỏ nhất và lớn nhất của ảnh ngõ ra. Do đó, hàm này sẽ mở rộng khoảng giá trị mức xám của ảnh trên một khoảng lớn nhất có thể  $[0, L-1]$ . Cuối cùng là ảnh trong hình 5.2(d) là kết quả khi dùng hàm lấy ngưỡng với  $(r_1, s_1) = (m, 0)$  và  $(r_2, s_2) = (m, L - 1)$  với  $m$  là giá trị mức xám trung bình của ảnh ngõ vào.



(a)



(b)



(c)



(d)

**Hình 5.2.** Phép mở rộng độ tương phản

(a) Hàm biến đổi; (b) Ảnh gốc; (c) Ảnh sau khi dùng phép mở rộng độ tương phản; (d) Ảnh sau khi dùng hàm lấy ngưỡng

**Ví dụ 5.1:** Thực hiện phép mở rộng độ tương phản trong hình 5.2

```
clear all;
f=imread('pout.tif');
r1=min(min(f));
r2=max(max(f));
for i=1:size(f,1)
    for j=1:size(f,2)
        if f(i,j)==r1
            g1(i,j)=0;
        elseif f(i,j)==r2
            g1(i,j)=255;
        else
            g1(i,j)=255/(r2-r1)*(f(i,j)-r1);
```

```

end
    end
end
g1=uint8(g1);
m=mean2(f);
for i=1:size(f,1)
    for j=1:size(f,2)
        if f(i,j) <= m
g2(i,j)=0;
        else
            g2(i,j)=255;
        end
    end
end
g2=uint8(g2);

```

Trong MATLAB, hàm `stretchlim` cho phép tìm các giới hạn (cận trên và cận dưới giá trị mức xám) dùng trong phép mở rộng độ tương phản cùng với hàm `imadjust`.

**Ví dụ 5.2:** Mở rộng độ tương phản dùng hàm `imadjust`

```

clear all;
f=imread('pout.tif');
g=imadjust(f,stretchlim(f));
figure;
subplot(1,2,1)
imshow(f)
xlabel(' (a) ')
subplot(1,2,2)
imshow(g)
xlabel(' (b) ')

```



(a)



(b)

**Hình 5.3.** Tăng cường ảnh kết hợp hàm *stretchlim* và *imadjust*: (a) Ảnh gốc; (b) Ảnh đã mở rộng độ tương phản

### 5.1.2. Xử lý lược đồ (Histogram)

Histogram của một ảnh xám có  $L$  mức xám khác nhau, là một hàm rời rạc, có biểu thức:

$$h(r_k) = n_k \quad (5.1)$$

trong đó  $r_k$  là giá trị mức xám thứ  $k$  trong đoạn  $[0, L-1]$  và  $n_k$  là số pixels có giá trị mức xám là  $r_k$ . Ví dụ với đoạn  $[0, 255]$ ,  $r_0 = 0$ ,  $r_1 = 1 \dots$

Histogram thường được chuẩn hóa. Với  $n$  là tổng số pixels của ảnh, histogram chuẩn hóa được tính bằng công thức:

$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n} \quad (5.2)$$

Có thể xem  $p(r_k)$  là hàm mật độ xác suất của  $r_k$ , cho biết khả năng xuất hiện tương ứng của từng giá trị mức xám. Sử dụng hàm `imhist` để thực hiện histogram của ảnh:

$$h = \text{imhist}(f, b)$$

Trong đó,  $f$  là ảnh cần thực hiện histogram,  $b$  là số đoạn biểu diễn (mặc định là 256). Nếu  $b$  nhỏ, giả sử  $b = 2$ , thì thang cường độ sẽ chia làm 2 khoảng: 0 đến 127, 128 đến 255, trong đó  $h(1)$  là số pixels có giá trị trong đoạn  $[0, 127]$  và  $h(2)$  là số pixels có giá trị trong đoạn  $[127, 255]$ . Nếu không có thông số đầu ra thì hàm `imhist` biểu diễn cho histogram của ảnh.

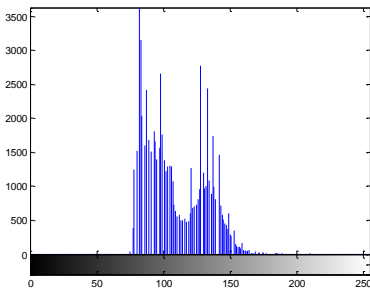
Ngoài ra có thể có được hàm  $p(r_k)$  qua dòng lệnh:

```
p = imhist(f,b)/numel(f);
```

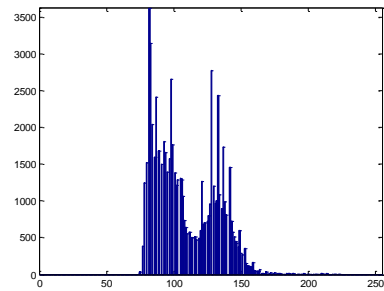
Với `numel` là tổng số pixels có trong ảnh `f`.

**Ví dụ 5.3:** Một số dạng đồ thị dùng để biểu diễn histogram

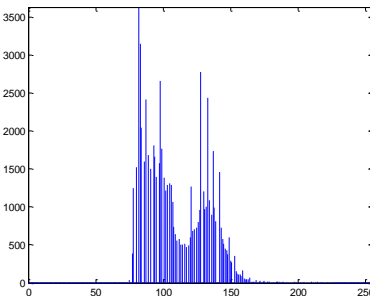
```
clear all;
f=imread('pout.tif');
h=imhist(f);
imhist(f);
axis([0 255 min(h) max(h)]);
figure;bar(0:255,h);
axis([0 255 min(h) max(h)]);
figure;stem(0:255,h,'marker','none');
axis([0 255 min(h) max(h)]);
figure;plot(0:255,h);
axis([0 255 min(h) max(h)]);
```



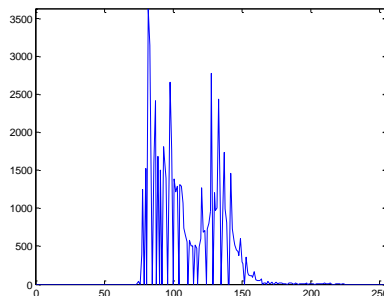
(a)



(b)



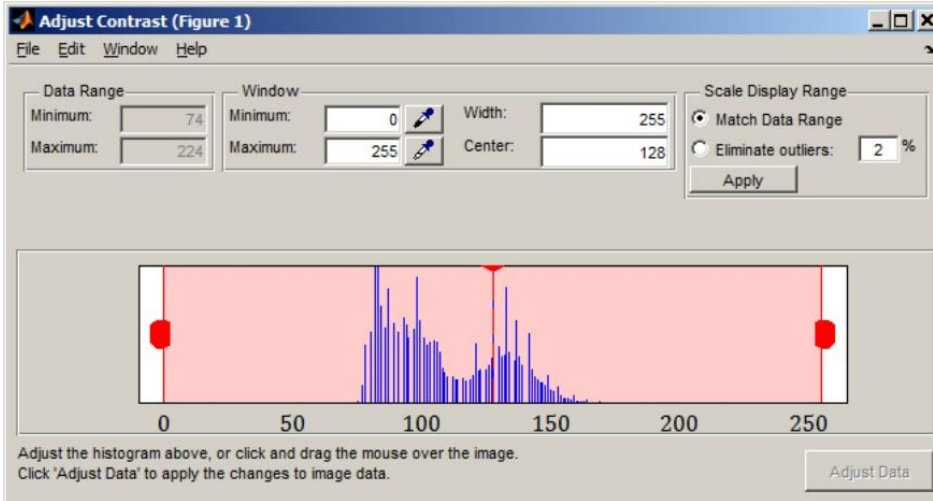
(c)



(d)

**Hình 5.4.** Một số dạng đồ thị có thể được dùng để biểu diễn histogram(a) `imhist` (dạng mặc định); (b) `bar`; (c) `stem`; (d) `plot`

Một công cụ trực quan cho phép thay đổi độ tương phản thông qua việc thay đổi cận trên và cận dưới giá trị mức xám trong biểu đồ histogram được xây dựng sẵn là `imcontrast` với cú pháp: `imcontrast(f)`



**Hình 5.5.** Mô tả giao diện cho phép thay đổi độ tương phản thông qua hàm `imcontrast`

**Lưu ý:** Hàm `imcontrast` có thể dùng cho ảnh đang hiển thị trên hình thông qua cú pháp sau:

```
imshow(f)
```

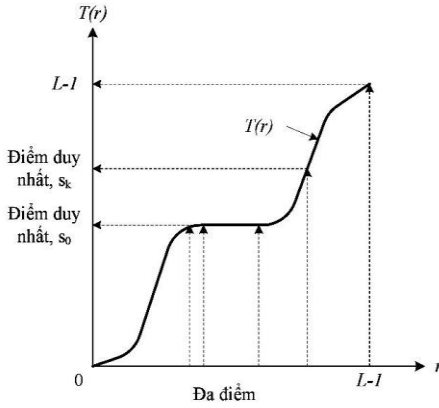
```
imcontrast(gca)
```

#### 5.1.2.1. Cân bằng Histogram

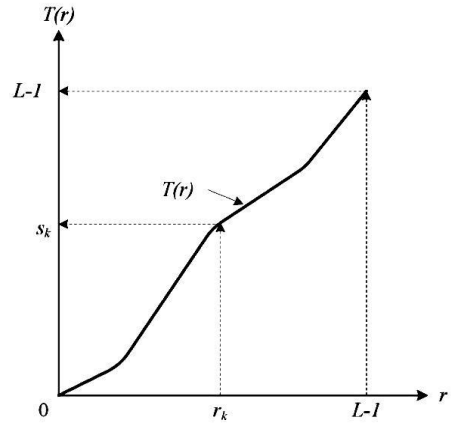
Như đã biết, hàm biến đổi mức xám có dạng tổng quát  $s = T(r)$  với  $0 \leq r \leq L-1$  sẽ luôn tạo ra giá trị mức xám ở ngõ ra là  $s$  ứng với mỗi giá trị mức xám ngõ vào là  $r$ . Chúng ta giả định rằng hàm  $T(r)$  luôn tăng đơn điệu trong khoảng  $0 \leq r \leq L-1$  và tất nhiên là  $0 \leq T(r) \leq L-1$ . Do đó, hoàn toàn có thể suy luận ngược lại rằng  $r = T^{-1}(s)$  với  $0 \leq s \leq L-1$ .

Những điều kiện ràng buộc nêu trên với  $T(r)$  luôn tăng để đảm bảo rằng giá trị mức xám ngõ ra không bao giờ nhỏ hơn ngõ vào tương ứng, điều này hạn chế các thành phần lạ xuất hiện trong ảnh ngõ ra, như trong hình 5.6(a). Ngoài ra, có thể nhận thấy rằng ảnh ngõ ra luôn có khoảng giá trị mức xám bằng với ngõ vào, và cũng như ánh xạ từ  $s$  đến  $r$  là dạng một – một ở hình 5.6(b).





(a)



(b)

**Hình 5.6.** (a) Hàm biến đổi tăng đơn điệu với hiện tượng nhiều giá trị mức xám có thể ánh xạ đến duy nhất một giá trị. (b) Hàm tăng đơn điệu hoàn toàn với ánh xạ một – một.

Các giá trị mức xám trong ảnh có thể được biểu diễn như là các biến ngẫu nhiên trong khoảng  $[0, L-1]$  thông qua hàm mật độ xác suất PDF của chính nó. Với việc đánh dấu  $p_r(r)$  và  $p_s(s)$  là các hàm PDF của  $r$  và  $s$  tương ứng, nếu  $p_r(r)$  và  $T(r)$  được biết trước, và  $T(r)$  là liên tục và khả vi trên khoảng đang xét thì PDF của biến  $s$  sau khi được ánh xạ có thể tìm được theo công thức sau:

$$p_s(s) = p_r(r) \left| \frac{ds}{dr} \right| \quad (5.3)$$

Hàm chuyển đổi dạng tổng quát cho trường hợp này được đưa ra có dạng:

$$s = T(r) = (L-1) \int_0^r p_r(w) dw \quad (5.4)$$

Với  $w$  là một biến hình thức của tích phân. Bên phải của biểu thức được xem như là hàm phân bố tích lũy (CDF) của các biến ngẫu nhiên  $r$ . Biểu thức (5.4) đã thỏa mãn tất cả các điều kiện nêu trên bao gồm hàm biến đổi là tăng đơn điệu và nằm trong khoảng cho phép  $[0, L-1]$ .

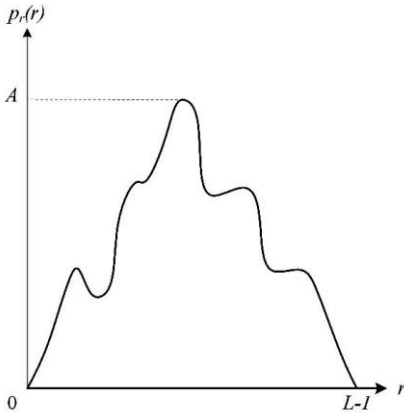
Để tìm thành phần  $p_s(s)$  tương ứng với hàm chuyển đổi đang được xem xét, áp dụng định luật Leibniz cho biểu thức (5.4)

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1) \frac{d}{dr} \left[ \int_0^r p_r(w) dw \right] = (L-1) p_r(r) \quad (5.5)$$

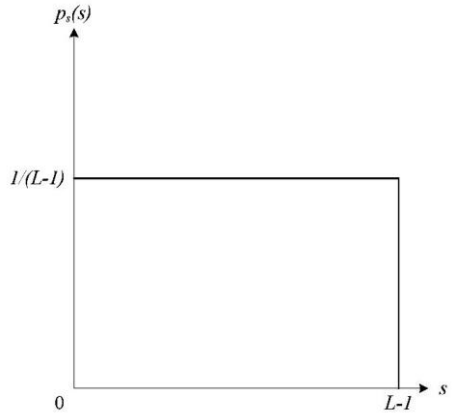
Thế kết quả này vào biểu thức (5.3):

$$p_s(s) = p_r(r) \left| \frac{ds}{dr} \right| = p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| = \frac{1}{L-1}; 0 \leq s \leq L-1 \quad (5.6)$$

Với kết quả ở biểu thức (5.6), thành phần  $p_s(s)$  có dạng hàm phân bố mật độ xác suất đều (uniform). Cần lưu ý là trong biểu thức (5.3),  $T(r)$  phụ thuộc vào  $p_r(r)$ , nhưng trong biểu thức (5.6) thì  $p_s(s)$  hoàn toàn độc lập với thành phần  $p_r(r)$ . Hình 5.7 làm rõ hơn về khái niệm này.



(a)



(b)

**Hình 5.7.** (a) Hàm PDF được cho bất kỳ. (b) Kết quả của việc áp dụng của chuyển đổi trong biểu thức (5.6) cho tất cả các mức xám  $r$ . Kết quả ngỏ ra  $p_s(s)$  có dạng PDF đồng nhất trên tất cả các giá trị mức xám và hoàn toàn độc lập với PDF của  $r$ .

**Ví dụ 5.4:** Cho hàm PDF:

$$p_r(r) = \frac{2r}{(L-1)^2}; 0 \leq r \leq L-1 \quad (5.7)$$

Từ biểu thức (5.4), ta có

$$s = T(r) = (L-1) \int_0^r p_r(w) dw = \frac{2}{L-1} \int_0^r w dw = \frac{r^2}{L-1} \quad (5.8)$$

Khi đó, giá trị mức xám ngỏ ra bằng giá trị ngỏ vào bình phương chia cho  $L-1$ . Hoàn toàn có thể kiểm chứng PDF của các giá trị mức xám trong ảnh mới là dạng đồng nhất bằng cách thế  $p_r(r)$  vào biểu thức (5.6):

$$\begin{aligned}
p_s(s) &= p_r(r) \left| \frac{dr}{ds} \right| = \frac{2r}{(L-1)^2} \left| \left[ \frac{ds}{dr} \right]^{-1} \right| = \frac{2r}{(L-1)^2} \left| \left[ \frac{d}{dr} \frac{r^2}{L-1} \right]^{-1} \right| \\
&= \frac{2r}{(L-1)^2} \left| \frac{L-1}{2r} \right| = \frac{1}{L-1}
\end{aligned} \tag{5.9}$$

Với điều kiện là  $r$  không âm và  $L > 1$ .

Như đã được đề cập trước đó, xác suất xuất hiện của giá trị mức xám  $r_k$  trong ảnh được xấp xỉ theo:

$$p_r(r_k) = \frac{n_k}{MN}; k = 0, 1, 2, \dots, L-1 \tag{5.10}$$

Với  $MN$  là tổng số điểm ảnh,  $n_k$  là số điểm ảnh có cùng giá trị  $r_k$ , và  $L$  là số lượng mức xám có thể biểu diễn (thường là 256, tương ứng với 8 bit). Hàm chuyển đổi dạng rời rạc trong biểu thức (5.4) có dạng như sau:

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j) = \frac{L-1}{MN} \sum_{j=0}^k n_j; k = 0, 1, 2, \dots, L-1 \tag{5.11}$$

Do đó, ảnh ngõ ra thu được bằng cách ánh xạ từng pixel trong ảnh ngõ vào với mức xám  $r_k$  thành điểm ảnh có mức xám  $s_k$  tương ứng ở ngõ ra theo như biểu thức (5.6), và ánh xạ  $T(r_k)$  trong biểu thức này được gọi là cân bằng histogram.

**Ví dụ 5.5:** Viết thuật toán cân bằng histogram

```

clear all;
f=imread('pout.tif');
h=histogram(f);
L=256;
for i=1:L
    s(i)=(L-1)*sum(h(1:i));
end
s=uint8(s);
for i=1:size(f,1)
    for j=1:size(f,2)
        g(i,j)=s(f(i,j)+1);
    end
end
end

```

Hay có thể viết hàm `hist_equa` cho phép thực hiện nhanh thuật toán cân bằng histogram cho các ứng dụng về sau:

```
function g=hist_equa(f);  
h=histogram(f);  
L=256;  
for i=1:L  
    s(i)=(L-1)*sum(h(1:i));  
end  
s=uint8(s);  
for i=1:size(f,1)  
    for j=1:size(f,2)  
        g(i,j)=s(f(i,j)+1);  
    end  
end  
end
```

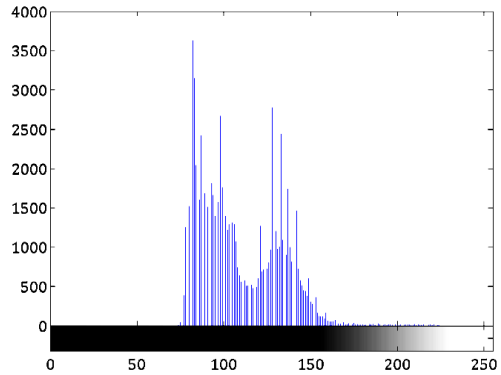
Trong MATLAB, hàm `histeq` có sẵn trong Toolbox cho phép thực hiện cân bằng histogram của ảnh với cú pháp sau:

$$J=\text{histeq}(I, \text{hgram})$$

Lưu ý, hàm `histeq` vẫn có thể được dùng cho ảnh màu (xem thêm phần Help trong MATLAB).



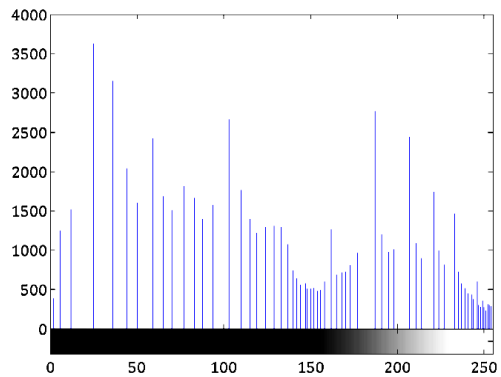
(a)



(b)



(c)



(d)

**Hình 5.8.** Minh họa kết quả trong ví dụ 5.5 (a) – (b) Ảnh gốc và histogram (c) – (d) Ảnh sau khi cân bằng và histogram tương ứng.

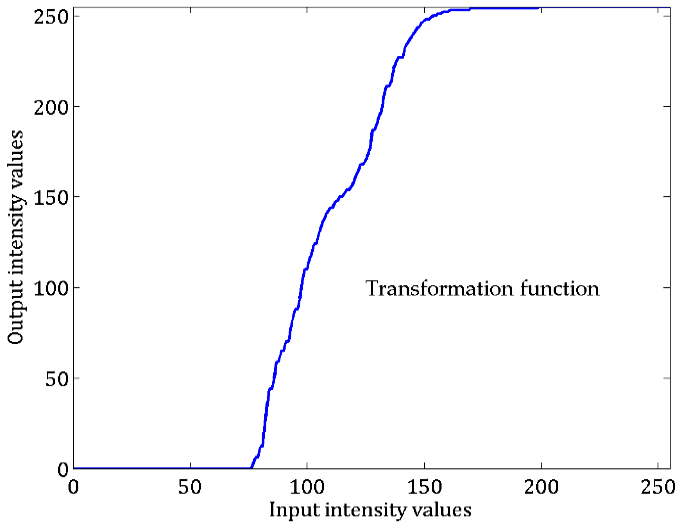
**Ví dụ 5.7:** Biểu diễn hàm chuyển đổi (ánh xạ) của thuật toán cân bằng histogram

```
clear all;
f=imread('pout.tif');
imhist(f);
h=histogram(f);
cdf=cumsum(h);
for i=1:256
    s(i)=255*cdf(i);
```

```

end
s=uint8(s);
plot(0:255,s,'linewidth',2);
axis([0 255 0 255]);
xlabel('Input intensity values');
ylabel('Output intensity values');
text(125,100,'Transformation function');

```



**Hình 5.9.** Hàm chuyển đổi được dùng để ánh xạ các giá trị mức xám từ ảnh ngõ vào trong hình 5.8(a) đến ảnh ngõ ra trong hình 5.8(c)

Một điều cần lưu ý là những ảnh sau khi tăng cường bằng thuật toán cân bằng histogram có biểu đồ histogram như nhau. Đây cũng là một trong những hạn chế của thuật toán, đặc biệt là một ảnh được chụp trong các điều kiện sáng khác nhau sẽ luôn cho ra cùng một ảnh.

#### 5.1.2.2. Kỹ thuật phối hợp lược đồ (Histogram matching)

Như đã biết, kỹ thuật cân bằng histogram sẽ tự động thực hiện chuyển đổi cường độ mức xám của ảnh để tạo ra ảnh mới có histogram đồng nhất. Khi việc tăng cường ảnh được yêu cầu, thì đây là kỹ thuật được áp dụng do có thể dự tính trước được kết quả cũng như thực hiện dễ dàng. Tuy nhiên, trong một vài ứng dụng thì kỹ thuật cân bằng dựa trên histogram đồng nhất không phải lúc nào cũng tốt nhất. Và kỹ thuật được dùng để tạo ra một ảnh với histogram đã được chỉ rõ gọi là histogram matching hay histogram specification.

Nói một cách dễ hiểu, histogram matching là kỹ thuật điều chỉnh độ sáng, độ tương phản của hai ảnh bằng cách sử dụng histogram. Thông thường, nó được sử dụng như bộ cân bằng, được dùng để chuẩn hóa hai ảnh khi chúng được chụp trong cùng một môi trường ánh sáng cục bộ, cùng một địa điểm nhưng bởi các cảm biến, trong các điều kiện khí quyển hay ánh sáng toàn cục khác nhau.

Trong kỹ thuật này, hàm phân bố tích lũy CDF của ảnh cần chỉnh sửa sẽ được thay đổi sao cho càng giống với CDF của ảnh tham chiếu. Kỹ thuật histogram matching đầu tiên sẽ tính histogram của ảnh vào (ảnh sẽ được biến đổi) và của ảnh tham chiếu, sau đó CDF của từng ảnh được tính dựa trên hàm PDF của chúng. Ứng với mỗi giá trị mức xám ngõ vào sẽ được ánh xạ đến giá trị tương ứng tại hàm CDF của ảnh tham chiếu sao cho chúng có cùng giá trị mật độ tích lũy.

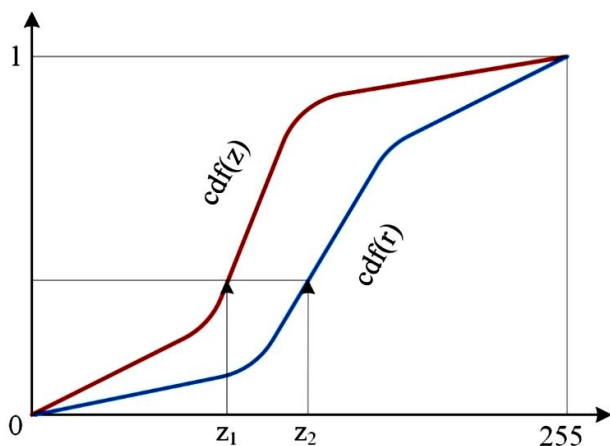
Gọi hàm phân bố xác suất của ảnh cần chỉnh sửa và ảnh tham chiếu lần lượt là  $p_z(z)$  và  $p_r(r)$ . Hàm phân bố tích lũy được xác định.

$$cdf_z(z_k) = \sum_{i=0}^k p_z(z_i); \quad cdf_r(r_k) = \sum_{i=0}^k p_r(r_i) \quad (5.12)$$

Ứng với mỗi giá trị mức xám  $r_k \in [0, 255]$  sẽ tìm thấy tương ứng một giá trị mức xám  $z_k$  sao cho

$$cdf_r(r_k) = cdf_z(z_k) \quad (5.13)$$

Từ đó, kết quả thu được là hàm ánh xạ giá trị mức xám từ ảnh tham chiếu sang. Hình 5.10 minh họa cho thuật toán histogram matching.



**Hình 5.10.** Biểu đồ hàm  $cdf$  minh họa cho thuật toán histogram matching

**Lưu ý:**

- Kích thước của hai ảnh không cần phải giống nhau.
- Thuật toán có thể tạo ra những khoảng trống trong lược đồ histogram. Điều này có thể đoán được do histogram sẽ bị biến dạng. Điều này khá phổ biến với ảnh có mức giá trị nguyên.
- Kỹ thuật này không thích hợp với ảnh có chứa dữ liệu phân loại và ảnh màu RGB, trong trường hợp này, các kênh màu nên được phân tách riêng để xử lý.

**Ví dụ 5.8:** Thực hiện thuật toán histogram matching với hai ảnh: ảnh tham chiếu (ảnh gốc pout), ảnh cần được chỉnh sửa.

```
clear all;
f=imread('pout.tif');
g=imadjust(f,stretchlim(f));
z=imhistmatch(g,f,256);
```

Kết quả ví dụ 5.8 được trình bày trong hình 5.11, ảnh cần được chỉnh sửa là ảnh đã bị biến đổi dùng hàm `imadjust` với sự hỗ trợ từ hàm `stretchlim` trong việc tìm cận trên và dưới của giá trị mức xám. Hàm thực hiện thuật toán histogram matching được xây dựng sẵn trong MATLAB là `imhistmatch` có cú pháp như sau:

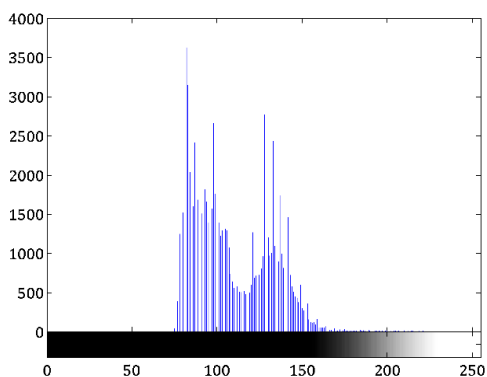
$$B = \text{imhistmatch}(A, \text{Ref}, N)$$



Với  $A$  là ảnh sẽ được biến đổi,  $Ref$  là ảnh tham chiếu và  $N$  là số lượng giá trị mức xám được chuyển đổi. Mặc định của giá trị  $N$  là 64 nếu không được khai báo.



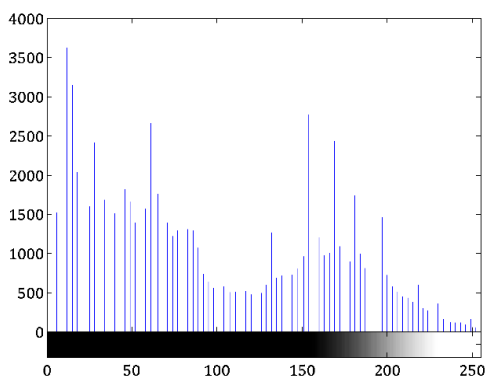
(a)



(b)



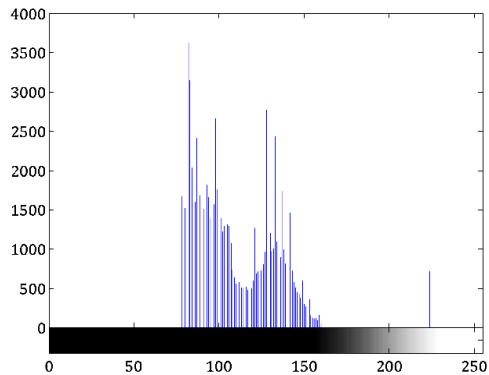
(c)



(d)



(e)



(f)

**Hình 5.11.** Kết quả ví dụ 5.8: (a) Ảnh gốc; (c) Ảnh cần chỉnh sửa; (e) Ảnh sau khi thực hiện histogram matching và histogram tương ứng.

## 5.2. XỬ LÝ MẶT NẠ

Nhiều do nhiều nguyên nhân khác nhau làm cho chất lượng ảnh bị giảm xuống. Để giảm nhiễu và nâng cao chất lượng ảnh ta sử dụng những phương pháp lọc với các mặt nạ khác nhau, phù hợp với mỗi loại nhiễu cụ thể. Trong phần này sẽ xét đến các bộ lọc ảnh trong không gian.

Lọc không gian (spatial filtering) cũng là một quá trình xử lý trên các điểm ảnh, dựa trên một phép toán với các điểm ảnh xung quanh. Phương pháp lọc ảnh bao gồm các bước:

- Xác định điểm ảnh trung tâm  $(x, y)$ .
- Thực hiện các phép toán với các điểm xung quanh  $(x, y)$ .
- Kết quả thu được đáp ứng của quá trình lọc tại  $(x, y)$ .
- Lặp lại các bước trên với tất cả các điểm ảnh khác.

### 5.2.1. Các bộ lọc làm mịn ảnh

Các bộ lọc làm mịn ảnh được sử dụng để làm mờ và giảm nhiễu. Làm mờ ảnh được sử dụng trong quá trình tiền xử lý ảnh, nhằm mục đích loại bỏ các chi tiết nhỏ ra khỏi ảnh trước khi tiến hành tách các thành phần lớn hơn khỏi ảnh, làm mờ còn được sử dụng để làm liền lại những đứt quãng nhỏ của đường thẳng hoặc đường cong.

Như đã biết, lọc tuyến tính là một quá trình trong đó mỗi điểm ảnh có giá trị bằng trung bình của các điểm ảnh lân cận xác định bởi bộ lọc. Do đó, ảnh sau xử lý trở nên mượt hơn, giảm độ sắc nét so với ảnh gốc. Kết quả là các thành phần nhiễu ngẫu nhiên, thường có mức xám khác biệt

với các vùng lân cận sẽ được loại bỏ. Tuy nhiên, một hạn chế dễ thấy khi tiến hành làm mượt ảnh là tại các vị trí biên (chỉ tiết được sử dụng nhiều trong xử lý ảnh), nơi có sự thay đổi nhanh chóng của các mức xám, lại bị làm mờ đi, ảnh hưởng đến các bước tiếp theo trong xử lý ảnh. Tuy nhiên, nếu sử dụng cửa sổ lọc thích hợp, có thể vừa giảm được nhiễu mà ít ảnh hưởng đến biên ảnh.

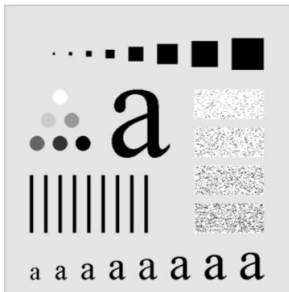
Một cửa sổ lọc chuẩn hóa thường thấy và đã được đề cập là cửa sổ có các hệ số giống nhau:

$$w = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5.14)$$

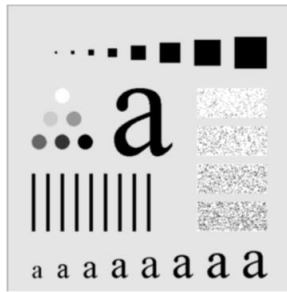
Một loại cửa sổ lọc chuẩn hóa khác có các hệ số khác nhau:

$$w = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (5.15)$$

Bộ lọc trên tiến hành lấy trung bình có trọng số đối với các điểm ảnh lân cận, tức là mỗi điểm ảnh được nhân với một hệ số khác nhau, hệ số càng lớn thì điểm ảnh đó có trọng số càng lớn. Ở bộ lọc trên, điểm ảnh trung tâm có vai trò quan trọng nhất trong phép toán tính trung bình; càng ra xa điểm ảnh trung tâm, trọng số của các điểm ảnh cũng giảm dần. Mục đích của việc làm này là hạn chế ảnh bị mờ khi tiến hành làm mượt. Tuy nhiên, ta rất khó thấy sự khác biệt giữa hai bộ lọc trên do cửa sổ lọc có kích thước nhỏ hơn nhiều so với kích thước ảnh.



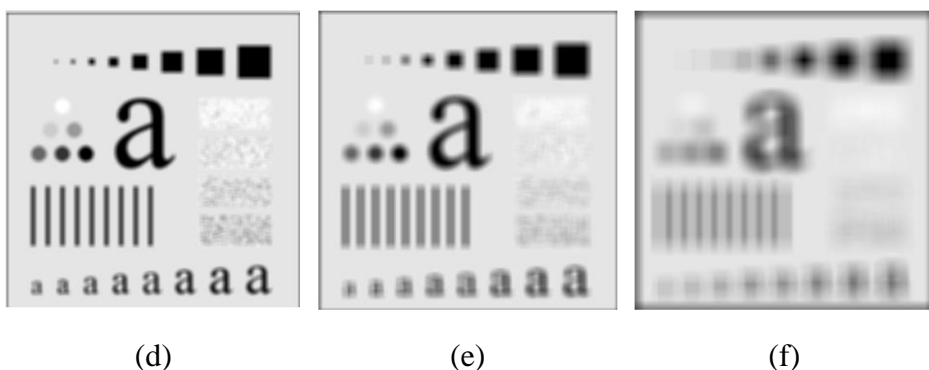
(a)



(b)



(c)

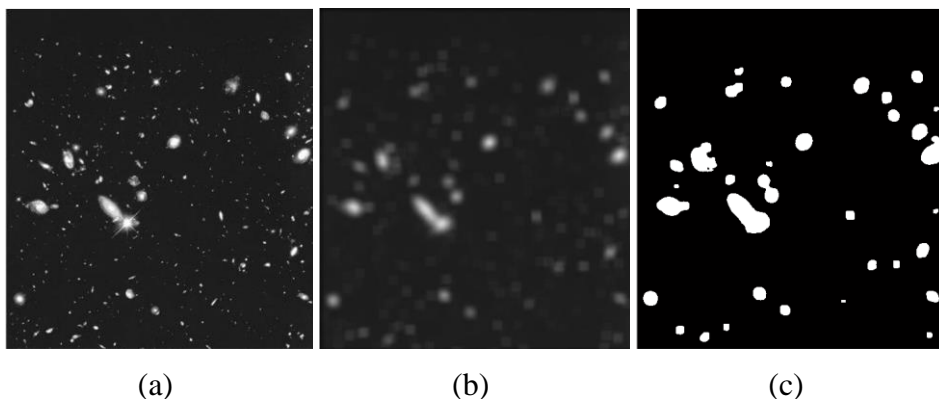


**Hình 5.12.** Áp dụng cửa sổ lọc khác nhau vào ảnh cần làm mịn

Xét ví dụ trên với việc sử dụng các bộ lọc trung bình có kích cỡ khác nhau, ở đây sử dụng padding là các giá trị 0. Hình 5.12(a) là ảnh gốc. Hình 5.12(b) sử dụng bộ lọc 3x3, hình bị mờ đi chút ít, với các chi tiết nhỏ như chữ a nhỏ và thành phần nhiễu bị mờ nhiều hơn so với các chi tiết khác.

Kết quả tương tự cũng xảy ra với hình 5.12(c) sử dụng bộ lọc kích thước 5x5. Các chi tiết nhỏ như nhiễu đã giảm dần, các mép hình răng cưa cũng đã được làm mượt hơn, nhưng vẫn đảm bảo các thành phần kích thước lớn không bị ảnh hưởng nhiều.

Hình 5.12(d) dùng cửa sổ 9x9, ảnh mờ hơn, các chi tiết nhiễu đã được giảm khá nhiều. Hình 5.12(e) và hình 5.12(f) dùng các bộ lọc tương ứng 16x16 và 35x35, các chi tiết nhỏ gần như đã bị loại khỏi ảnh, do đó có thể dễ dàng lấy được các thành phần có kích thước lớn.



**Hình 5.13.** Làm mịn ảnh sử dụng bộ lọc có kích thước 15x15

Hình 5.13(a) là ảnh gốc. Hình 5.13(b) là ảnh sau lọc với bộ lọc kích thước 15x15, cho thấy các chi tiết nhỏ gần như bị loại bỏ. Để thu các thành phần kích thước lớn của ảnh, có thể biến đổi ảnh trong hình 5.13(b) thành ảnh nhị phân. Ở đây cho mức ngưỡng là 25% giá trị mức xám lớn nhất của

ảnh 5.13(b). Kết quả được ảnh 5.13(c), tất cả các chi tiết nhỏ bị loại bỏ, chỉ còn các thành phần kích thước lớn.

Việc thực hiện làm mịn ảnh có thể được thực hiện nhờ các bộ lọc đã trình bày trong chương 4. Sau đây, ta sẽ xem xét việc ứng dụng các bộ lọc này trong tăng cường ảnh.

**Ví dụ 5.9:** Khảo sát các tùy chọn dùng lọc trung bình với cửa sổ lọc 15x15

```
clear all;
f=imread('check.tif');
h=ones(15)/(15^2);
g1=imfilter(f,h);
g2=imfilter(f,h,'replicate');
g3=imfilter(f,h,'symmetric');
g4=imfilter(f,h,'circular');
```

**Hình 5.14.** Kết quả dùng cửa sổ lọc 15x15 với các tùy chọn

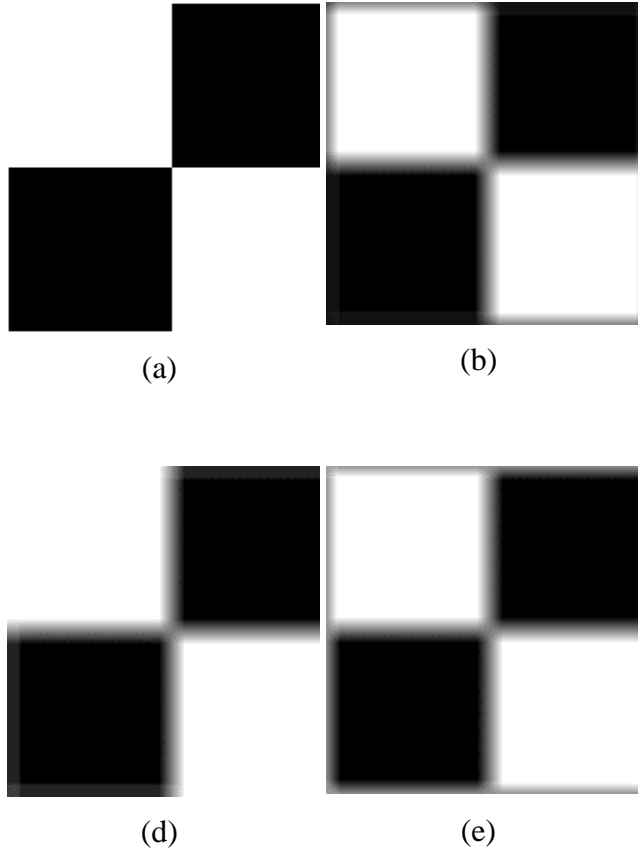
(a) Ảnh gốc

(b) Đường bao 'P' (mặc định)

(c) Replicate

(d) Symmetric

(e) Circular



Trong hình 5.14(b), do sử dụng tùy chọn mặc định với đường bao được mở rộng bằng cách thêm vào một điểm ảnh có giá trị là 0 nên ảnh

ngõ ra sẽ có một đường viền màu đen. Để giải quyết việc xuất hiện đường bao không mong muốn này, tùy chọn 'replicate' được sử dụng trong hình 5.14(c). Kết quả tương tự cũng đạt được với tùy chọn 'symmetric'. Tuy nhiên ở tùy chọn cuối cùng là 'circular', kết quả ảnh ngõ ra gặp phải vấn đề tương tự như trong hình 5.14(b).

Các mặt nạ lọc đã trình bày trong chương 4 có thể được áp dụng vào việc lọc ảnh. Tùy theo đặc điểm ảnh khác nhau mà loại mặt nạ cụ thể được sử dụng.

**Ví dụ 5.10:** Thực hiện lọc ảnh dùng mặt nạ Sobel

```
I=imread('peppers.png');
h=fspecial('average',[5 5]);
G=imfilter(I,h);
figure;
subplot(1,2,1)
imshow(I)
xlabel(' (a) ')
subplot(1,2,2)
imshow(G)
xlabel(' (b) ')
```



(a)



(b)

**Hình 5.15.** Thực hiện mặt nạ lọc trung bình: (a) Ảnh gốc; (b) Ảnh đã lọc

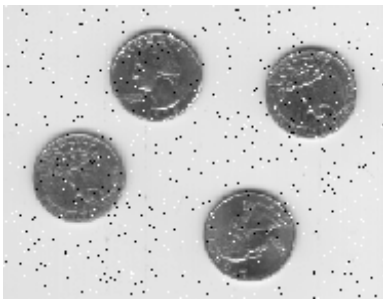
Một trong những bộ lọc cũng được sử dụng để loại nhiễu muối tiêu hay làm trơn ảnh là lọc trung vị. Bộ lọc này hiệu quả hơn các bộ chập khác nhờ việc đồng thời giảm nhiễu và bảo tồn biên. Trong MATLAB, bộ lọc này được khai báo:

```
B=medfilt2(A,[m n])
```

Trong đó A là ảnh ngõ vào, mỗi điểm ảnh trong ảnh ngõ ra B là trung bình của  $m \times n$  điểm ảnh lân cận.

**Ví dụ 5.11:** Thực hiện gây nhiễu muối tiêu lên ảnh và dùng `medfilt2` để lọc nhiễu này

```
I = imread('eight.tif');  
J = imnoise(I, 'salt & pepper', 0.02);  
K = medfilt2(J);  
figure,  
subplot(1,2,1)  
imshow(J), xlabel(' (a) ')  
subplot(1,2,2)  
imshow(K), xlabel(' (b) ')
```



(a)



(b)

**Hình 5.16.** Thực hiện bộ lọc trung vị: (a) Ảnh bị nhiễu muối tiêu; (b) Ảnh sau khi lọc trung vị

### 5.2.2. Các bộ lọc làm sắc nét ảnh

Mục đích của việc làm sắc nét ảnh là làm nổi bật các chi tiết trong ảnh hoặc làm sắc nét các chi tiết bị mờ bởi quá trình làm mịn ảnh. Quá trình làm mịn ảnh là thực hiện phép lấy trung bình các giá trị lân cận điểm ảnh cần xử lý, tương tự như phép toán tích phân, trong khi đó quá trình làm sắc nét ảnh tập trung vào sự sai lệch giữa các chi tiết trong ảnh, giống như phép toán vi phân. Kết quả là biên ảnh và các chi tiết nhiễu, những nơi có sự khác biệt về mức xám với các điểm ảnh xung quanh, được làm nổi bật lên.

Trong phần này sẽ thực hiện làm sắc nét ảnh dựa trên đạo hàm bậc một và bậc hai của hàm rời rạc.

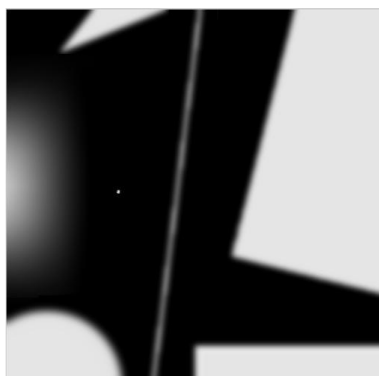
- Đạo hàm bậc một của hàm rời rạc một chiều:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \quad (5.16)$$

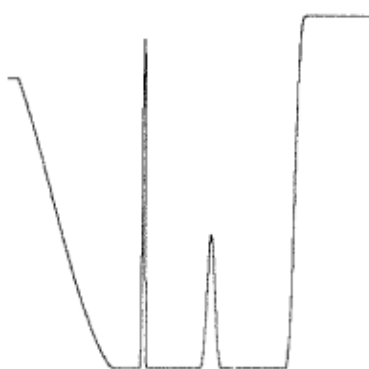
- Đạo hàm bậc hai của hàm rời rạc một chiều:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - f(x-1) - 2f(x) \quad (5.17)$$

Để thấy được sự giống cũng như khác nhau về cơ bản giữa hai phương pháp sử dụng đạo hàm bậc một và bậc hai, ta xét một ví dụ cụ thể sau:

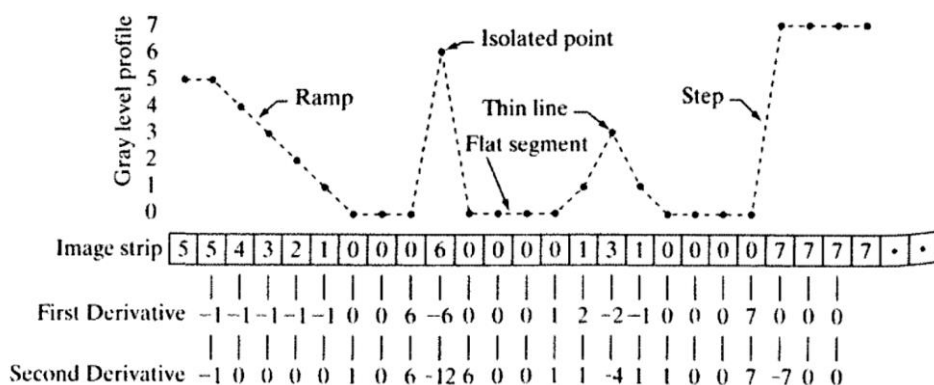


(a)



(b)

**Hình 5.17.** Biểu diễn mức xám của các pixel nằm trên đường ngang qua điểm giữa của ảnh



**Hình 5.18.** Đơn giản hóa các giá trị mức xám của hình 5.16(b)

Hình 5.17(a) có một đường chéo và một điểm nhiễu. Hình 5.17(b) biểu diễn mức xám của các pixel nằm trên đường ngang qua điểm giữa của ảnh, bao gồm cả điểm nhiễu. Hình 5.18 đơn giản hóa các giá trị mức xám



của hình 5.17(b), chỉ gồm 8 mức xám khác nhau. Từ hình 5.18, có thể phân tích ảnh hưởng của phương pháp đạo hàm bậc một và bậc hai đối với điểm nhiễu, với đường chéo và cạnh biên giữa đối tượng và nền.

Các đoạn có mức xám không đổi thì đạo hàm bậc một và hai đều cho đáp ứng là 0. Với đoạn dốc thoải, đạo hàm bậc một cho các mức khác 0 trên toàn đoạn, đạo hàm bậc hai chỉ cho các giá trị khác 0 ở đầu và cuối đoạn, điều đó chứng tỏ với các đoạn chuyển tiếp thì đạo hàm bậc một tạo ra cạnh dày hơn và đạo hàm bậc hai tạo ra cạnh sắc hơn. Với điểm nhiễu, đáp ứng xung quanh và tại điểm nhiễu đạo hàm bậc hai lớn hơn so với đạo hàm bậc một, do đó đạo hàm bậc hai tạo ra chi tiết sắc hơn đối với nhiễu và các giá trị mức xám thay đổi nhanh. Đường chéo cũng tương tự với điểm nhiễu. Còn lại với bước nhảy thì đáp ứng của đạo hàm bậc một và hai là tương tự nhau. Do đó, có thể kết luận:

- Đạo hàm bậc một tạo ra các cạnh dày hơn so với đạo hàm bậc hai.
- Đạo hàm bậc hai ảnh hưởng nhiều hơn đến các chi tiết tinh.

Đạo hàm bậc hai được sử dụng nhiều hơn trong xử lý ảnh do nó làm nổi bật các chi tiết sắc, do đó chúng ta sẽ tập trung vào phương pháp này để làm sắc nét ảnh.

Do ảnh là một hàm rời rạc hai chiều nên ta cần có đạo hàm bậc hai của hàm hai chiều. Toán tử Laplace của hàm hai biến là:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5.18)$$

Công thức tính đạo hàm bậc hai cho hàm rời rạc hai chiều thường sử dụng là:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (5.19)$$

Và:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (5.20)$$

Toán tử Laplace trong không gian rời rạc hai chiều là:

$$\partial^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (5.21)$$

Biểu thức trên có thể thực hiện đối với các phần tử của ảnh bằng cách nhân chập ảnh với ma trận:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{hoặc} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (5.22)$$

Một định nghĩa khác về đạo hàm bậc hai có sử dụng thêm các phần tử đường chéo:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{hoặc} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5.23)$$

Do Laplace là toán tử đạo hàm, nó làm sắc nét các chi tiết tinh hay các thành phần mức xám biến đổi nhanh, nhưng lại dẫn đến giá trị 0 cho những vùng có cùng một mức xám cũng như giảm giá trị của các thành phần mức xám ít biến đổi. Một cách đơn giản phục hồi lại các vùng này nhưng vẫn giữ cho các chi tiết sắc nét là cộng ảnh gốc với ảnh thực hiện phép lọc bằng toán tử Laplace.

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)] \quad (5.24)$$

Trong đó  $c = 1$  nếu hệ số trung tâm của mặt nạ lọc là dương,  $c = -1$  nếu ngược lại.



(a)



(b)



(c)



(d)

**Hình 5.19.** Xử lý ảnh dùng mặt nạ Laplace

Hình 5.19(a) là ảnh gốc. Hình 5.19(b) là ảnh sau xử lý dùng mặt nạ Laplace. Ở đây, vùng có thành phần mức xám biến đổi chậm sau khi lọc sẽ bị giảm giá trị. Trong khi đó phần cạnh biên và các chi tiết có mức xám biến đổi nhanh được thể hiện rõ nét hơn. Hình 5.18(c) là kết quả của việc cộng ảnh gốc với ảnh Laplace, phục hồi lại phần bị giảm mức xám nhưng vẫn giữ được sự sắc nét của các chi tiết. Hình 5.19(d) sử dụng mặt nạ có thêm các giá trị đường chéo, cho hình ảnh sắc nét hơn hình 5.19(c).

Mặt nạ Laplace có thể tạo ra trong Matlab nhờ hàm `fspecial`.

```
>> fspecial('laplacian', alpha)
```

Trong đó hệ số `alpha` cho phép điều chỉnh mức độ sắc nét của hình ảnh. Có thể dùng mặt nạ lọc trực tiếp như sau:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{hoặc} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5.25)$$

#### ▪ Bộ lọc tăng cường (*high-boost filter*)

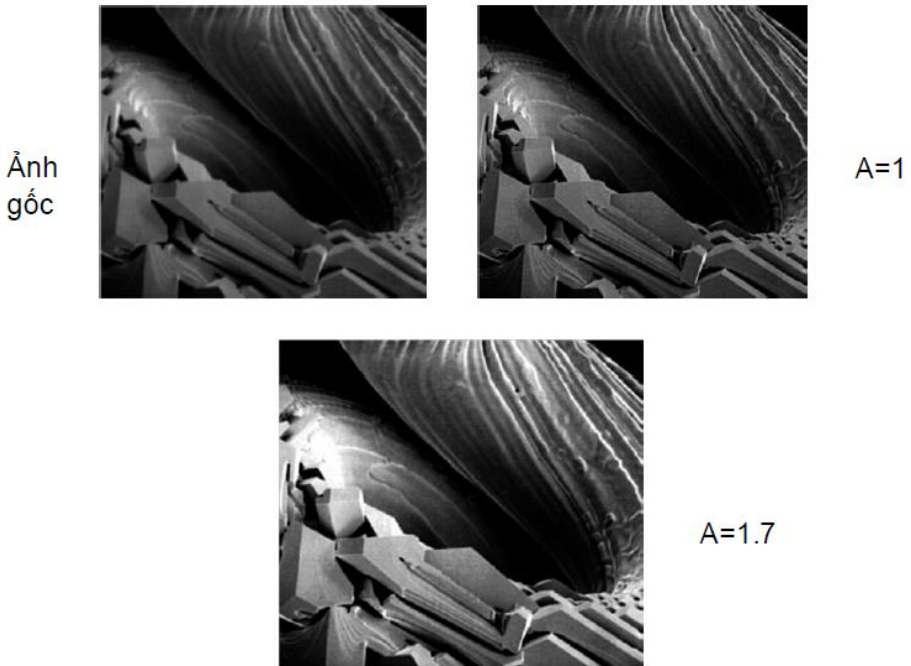
Bộ lọc tăng cường cũng là một ứng dụng của toán tử Laplace:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{hoặc} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \text{với} \quad (5.26)$$

$$A \geq 1$$

Với  $A = 1$ , bộ lọc tăng cường giống như mặt nạ lọc trực tiếp ở trên.  $A > 1$  thì khả năng làm sắc nét ảnh giảm dần, nếu  $A$  đủ lớn thì ảnh sau xử lý gần như giống ảnh ban đầu nhân với một hằng số.

Một ứng dụng của bộ lọc tăng cường là làm sáng ảnh vì nó nâng mức xám trung bình của ảnh ban đầu mà vẫn giữ được tính chất làm nét ảnh.



**Hình 5.20.** Ứng dụng của bộ lọc tăng cường với hệ số  $A$  khác nhau

## BÀI TẬP CHƯƠNG 5

- 5.1. Trình bày cách tính histogram và các dạng đồ thị biểu diễn histogram.
- 5.2. Mở ảnh 'trees.tif' và thực hiện cân bằng histogram cho ảnh này.
- 5.3. Mở ảnh 'coins.png'. Tạo nhiều muối tiêu lên ảnh này với các mức độ khác nhau. Dùng lọc trung vị để loại nhiễu với các mức độ khác nhau này.
- 5.4. Dùng mặt nạ Laplace để tăng độ nét cho ảnh 'ngc6543a.jpg'.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Quang Hoan, *Xử lý ảnh*, lưu hành nội bộ, Học viện Công nghệ Bưu chính Viễn thông, 2006.
2. Đỗ Năng Toàn, Phạm Việt Bình, *Xử lý ảnh*, Giáo trình môn học, Đại học Thái Nguyên, Khoa Công nghệ thông tin, 2007.
3. Maria Petrou, Panagiota Bosdogianni, *Image Processing: The Fundamentals*, John Wiley & Sons Ltd, 1999.
4. William K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 2001.
5. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, The Third Edition, Prentice Hall, 2008.
6. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing Using Matlab*, Prentice Hall, 2004.
7. Kayvan N., Robert S., *Biomedical Signal and Image Processing*, Taylor and Francis Group, 2006.

# GIÁO TRÌNH XỬ LÝ ẢNH

TS. NGUYỄN THANH HẢI

---

**NHÀ XUẤT BẢN**  
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
Khu Phố 6, Phường Linh Trung, Quận Thủ Đức, TPHCM  
Số 3, Công trường Quốc tế, Quận 3, TP Hồ Chí Minh  
ĐT: 38239171 – 38225227 – 38239172  
Fax: 38239172 - Email: [vnuhp@vnuhcm.edu.vn](mailto:vnuhp@vnuhcm.edu.vn)

\*\*\*\*\*

**PHÒNG PHÁT HÀNH NHÀ XUẤT BẢN**  
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
Số 3 Công trường Quốc tế - Quận 3 – TPHCM  
ĐT: 38239170 – 0982920509 – 0913943466  
Fax: 38239172 – Website: [www.nxbdhqghcm.edu.vn](http://www.nxbdhqghcm.edu.vn)

\*\*\*\*\*

*Chịu trách nhiệm xuất bản:*  
**NGUYỄN HOÀNG DŨNG**

*Chịu trách nhiệm nội dung:*  
**NGUYỄN HOÀNG DŨNG**

*Tổ chức bản thảo và chịu trách nhiệm về tác quyền*  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

*Biên tập:*  
**PHẠM ANH TÚ**

*Sửa bản in:*  
**THÙY DƯƠNG**

*Trình bày bìa*  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

<b>Mã số ISBN: 978-604-73-2582-5</b>
--------------------------------------

---

Số lượng 300 cuốn; khổ 16 x 24cm.

Số đăng ký kế hoạch xuất bản: 1007-2014/CXB/05-13/ĐHQGTPHCM.

Quyết định xuất bản số: 107 ngày 28/05/2014 của NXB ĐHQGTPHCM.

In tại Công ty TNHH In và Bao bì Hưng Phú.

Nộp lưu chiểu quý III năm 2014.