

MỤC LỤC

LỜI MỞ ĐẦU	1
PHẦN 1. NÂNG CAO CHẤT LƯỢNG ẢNH TRONG MIỀN KHÔNG GIAN	2
1.1 Giới thiệu	2
1.2 Khái quát nâng cao chất lượng hình ảnh trong miền không gian	2
1.3 Các phép biến đổi âm cơ bản	4
<i>a) Ảnh âm bản</i>	<i>4</i>
<i>b) Chuyển đổi logarit</i>	<i>5</i>
<i>c) Biến đổi hàm mũ</i>	<i>6</i>
<i>d) Hàm khoảng tuyến tính – tuyến tính từng đoạn</i>	<i>7</i>
1.4 Xử lý Histogram	9
<i>a) Khái niệm</i>	<i>9</i>
<i>b) Cân bằng Histogram</i>	<i>11</i>
<i>c) Thay đổi Histogram theo yêu cầu</i>	<i>12</i>
<i>d) Nâng cao chất lượng cục bộ</i>	<i>13</i>
1.5 Nâng cao chất lượng hình ảnh sử dụng toán tử logic và số học	14
<i>a) Toán tử trừ hình ảnh</i>	<i>15</i>
<i>b) Trung bình hình ảnh</i>	<i>16</i>
1.6 Lọc ảnh trong không gian	17
<i>a) Giới thiệu chung</i>	<i>17</i>
<i>b) Lọc tuyến tính</i>	<i>17</i>
<i>c) Lọc phi tuyến tính</i>	<i>21</i>
1.7 Ứng dụng lọc ảnh trong không gian	22
<i>a) Các bộ lọc làm mịn ảnh</i>	<i>22</i>
<i>b) Các bộ lọc làm sắc hình ảnh</i>	<i>23</i>
PHẦN 2. MÔ PHỎNG ỨNG DỤNG	24
Chương 1. Giới thiệu cơ bản về MATLAB	24
Chương 2. Chương trình mô phỏng	25

LỜI MỞ ĐẦU

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng cho nó.

Hình ảnh là một dạng dữ liệu đóng vai trò quan trọng trong việc trao đổi, xử lý, lưu trữ thông tin... Trong chúng ta có lẽ không ai là không từng sử dụng hình ảnh cho một mục đích nào đấy. Trong nhiều ngành nghề, trong một số loại hình công việc, người ta đều cần đến hình ảnh để mô tả, minh chứng hay diễn đạt những điều mà đôi khi chữ viết hay ngôn ngữ nói không lột tả hết được. Đặc biệt trong các ngành công nghiệp như: cơ khí chế tạo, chế biến, sản xuất,... việc đọc hình ảnh có thể nói là thường xuyên và cực kì quan trọng. Bản vẽ kỹ thuật (một dạng của hình ảnh) chính là kết quả ngôn ngữ kỹ thuật, mà qua nó, một quy trình công nghệ phải được xây dựng trong quá trình sản xuất, cũng như nó chính là cơ sở cho việc nghiệm thu bất kì sản phẩm nào. Nhu cầu lưu ảnh của các tài liệu, các bản vẽ, hoặc sửa đổi chúng và chuyển chúng sang các dạng đồ họa khác tiện cho việc nhận dạng, đối sánh mẫu để sử dụng sau này là điều cần thiết. Nhưng phải tổ chức việc lưu các dạng hình ảnh này như thế nào? Có cần xử lý gì trước khi lưu chúng không? Câu trả lời là có.

Do vậy tiền xử lý ảnh là việc cần làm. Có nhiều phương pháp, nhiều công cụ và nhiều phần mềm xử lý ảnh đã ra đời. Tăng cường chất lượng ảnh, mà công đoạn đầu tiên là một bước tiền xử lý nhằm loại bỏ nhiễu, khắc phục những khiếm khuyết do bước thu nhận ảnh không tốt là việc làm quan trọng. Có nhiều phương pháp cho việc nâng cao chất lượng ảnh nói chung và tiền xử lý nói riêng. Trong đồ án này, mục tiêu chính của nhóm là tìm hiểu về phương pháp để nâng cao chất lượng ảnh dựa trên các kỹ thuật lọc không gian.

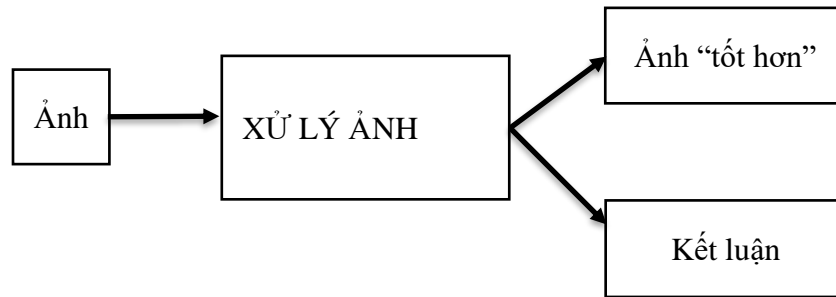
STT	Nội dung thực hiện	Sinh viên thực hiện
1	Dịch bài phần 1.1, 1.2, 1.3	Hồ Sỹ Đạt
2	Dịch bài phần 1.4, 1.5	Dương Anh Tuấn
3	Dịch bài phần 1.6, 1.7	Nguyễn Anh Tuấn
4	Trình bày word, slide	Hồ Sỹ Đạt, Nguyễn Anh Tuấn
5	Viết code chương trình	Dương Anh Tuấn
6	Giao diện chương trình (Guide)	Hồ Sỹ Đạt, Nguyễn Anh Tuấn
7	Hướng dẫn sử dụng chương trình	Dương Anh Tuấn

Bảng 1. Bảng phân công công việc

PHẦN 1. NÂNG CAO CHẤT LƯỢNG ẢNH TRONG MIỀN KHÔNG GIAN

1.1 Giới thiệu

Quá trình xử lý ảnh được xem như 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 có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1.1. Quá trình xử lý ảnh

Có nhiều phương pháp nhằm tăng cường chất lượng của ảnh, nhưng tập trung vào hai nhánh chính là xử lý ảnh trong miền không gian và xử lý ảnh trong miền tần số. Trong miền không gian, ảnh được xử lý trực tiếp trên các pixels. Miền tần số sử dụng biến đổi Fourier để xử lý.

1.2 Khái quát nâng cao chất lượng hình ảnh trong miền không gian

Miền không gian là tập hợp các pixels trong một bức ảnh. Chúng ta sẽ tiến hành xử lý trực tiếp trên các pixels này. Quá trình xử lý miền không gian được xác định theo hàm sau:

$$g(x, y) = T[f(x, y)] \quad (1)$$

Trong đó:

$f(x, y)$: là ảnh đầu vào/điểm ảnh đầu vào.

$g(x, y)$: là ảnh đầu ra/điểm ảnh được xử lý.

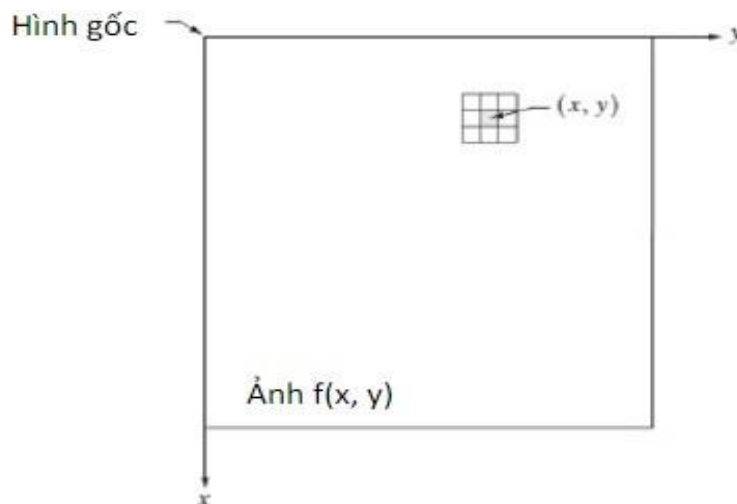
T : là một hàm của f , được định nghĩa là lân cận của (x, y) .

Thêm nữa T có thể thao tác trên một tập hợp những điểm ảnh đầu vào như là thực hiện tổng pixel-by-pixel của K điểm ảnh để giảm nhiễu - vấn đề này sẽ được đề cập ở dưới.

Cách thức chính của phương pháp lân cận một điểm là sử dụng một vùng vuông hoặc hình chữ nhật nhỏ mà có tâm điểm tại (x, y) như hình 1. Tâm của vùng ảnh phụ này được di chuyển thứ tự từng điểm ảnh mà bắt đầu ở điểm trên cùng bên trái.

Toán tử T được áp dụng tại từng điểm (x,y) để tạo ra điểm “g” ở vị trí đó. Quá trình xử lý chỉ sử dụng những điểm ảnh nằm trong vùng ảnh được giới hạn bởi vùng lân cận. Mặc dù các dạng vùng lân cận khác như hình tròn, hình thoi được sử dụng nhưng hình vuông và hình chữ nhật được dùng chủ yếu vì dễ áp dụng.

Miền không gian được định nghĩa là tập hợp của các điểm ảnh thành nên hình ảnh. Phương pháp miền không gian là kỹ thuật tác động trực tiếp lên điểm ảnh. Nguyên lý chính của việc định nghĩa vùng lân cận tại 1 điểm có tọa độ (x,y) là một hình phụ/phụ ảnh/hạ ảnh (sub-image) hình vuông hay hình chữ nhật ở chính giữa (x,y) . Hình dưới đây sẽ biểu diễn điều này.



Hình 1.2.1. Ví dụ về vùng lân cận

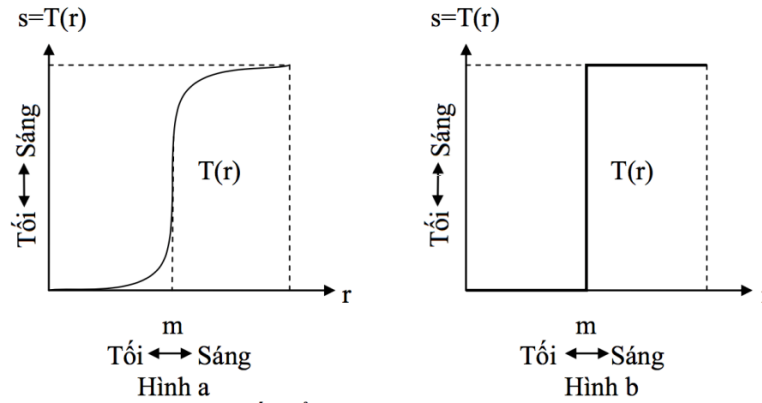
Trung tâm của ảnh phụ sẽ chạy từ pixel này đến pixel khác theo thứ tự từ trái qua phải và từ trên xuống dưới. Thuật toán T sẽ áp dụng cho mỗi vùng (x,y) để tìm đầu ra “g”. Dạng đơn giản nhất của T là khi vùng lân cận có kích thước 1×1 (có nghĩa là một điểm ảnh). Trong trường hợp này, g chỉ phụ thuộc vào giá trị của f tại (x,y) và T trở thành một hàm truyền mức xám có dạng:

$$s = T(r)$$

Trong đó r và s là hai biến thể hiện mức xám của $f(x,y)$ và $g(x,y)$.

Gọi m là một mức nào đó nếu $T(r)$ có dạng như trong hình (1.2.2-a) thì nó sẽ tạo ra một ảnh có độ tương phản cao hơn bằng cách làm đậm hơn những mức xám trên m và nhạt hơn những mức xám dưới m , đây là phương pháp kéo dẫn độ tương phản. Đặc biệt ở hình (1.2.2-b) ta thấy s đã được xác định bằng 0 nếu $r < m$ và ngược lại s sẽ bằng 1, đây có thể hiểu là phép biến đổi ảnh thành ảnh nhị phân.

Ảnh xạ của $T(r)$ được gọi là hàm ngưỡng. Bởi vì trong quá trình nâng cao chất lượng ảnh mỗi điểm sẽ phụ thuộc vào mức xám của chính nó nên ta còn gọi đây là phương pháp xử lý điểm.



Hình 1.2.2. Hàm biến đổi mức xám làm tăng độ tương phản.

Vùng lân cận lớn hơn có tính linh động hơn. Phương pháp tổng quát là sử dụng một hàm chứa giá trị ban đầu của vùng lân cận (x,y) để xác định giá trị của g tại điểm (x,y) . Một trong những phương thức chính của cách thức này là sử dụng mặt nạ (hay còn gọi là bộ lọc, nòng cốt). Hiểu một cách cơ bản thì 1 mặt nạ là mảng 2 chiều (3×3) , như hình 1.2.1, trong đó những giá trị của hệ số mặt nạ xác định bản chất của quá trình ví dụ như độ sắc nét của hình ảnh. Kỹ thuật này được gọi là xử lý mặt nạ hoặc lọc.

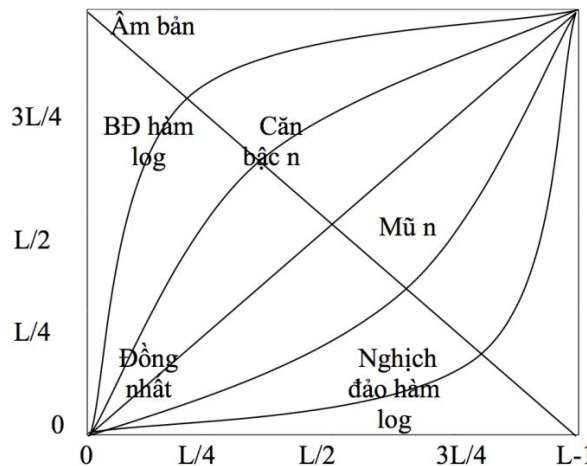
1.3 Các phép biến đổi âm cơ bản

Chúng ta bắt đầu nghiên cứu kỹ thuật nâng cấp ảnh bằng cách sử dụng những hàm chuyển đổi mức xám. Đây là phần cơ bản nhất của kỹ thuật nâng cấp ảnh.

a) Ảnh âm bản

Ảnh âm bản có mức xám trong dải $[0, L-1]$ được thể hiện bởi hàm biến đổi âm bản (hình 1.3.1), hàm này được lược diễn dưới dạng:

$$s = L - 1 - r \quad (2)$$



Hình 1.3.1. Biểu diễn hàm biến đổi âm bản

Để dễ hình dung ta xét ví dụ sau: Cho một ảnh có kích thước $R \times C$ với R là số hàng và C là số cột. $I(R,C)$ là ảnh ban đầu và $N(R,C)$ là ảnh âm bản, thì N được tính như sau:

$$N(r, c) = 255 - I(r, c)$$

Trong đó $0 \leq r \leq R$ và $0 \leq c \leq C$

Chúng ta có thể nhận thấy rằng giá trị của mỗi pixel ảnh gốc là dưới 255.

Ảnh qua xử lý có giá trị đối lại so với ảnh ban đầu.

Đảo ngược mức xám của một hình ảnh theo cách này tạo ra một ảnh âm bản. Loại xử lý này phù hợp để tăng cường chi tiết màu trắng hoặc xám ẩn trong vùng tối của hình ảnh, đặc biệt là khi các khu vực màu tối chiếm ưu thế về số lượng.

Trong matlab dùng hàm:

$$g = \text{imcomplement}(w);$$

Ví dụ:



Hình 1.3.2. Hình biến đổi âm bản.

b) Chuyển đổi logarit

Công thức tổng quát chuyển đổi logarit như sau:

$$s = c \cdot \log(1 + r) \quad (3)$$

Trong đó c là một hằng số và người ta cho $r \geq 0$. Hình dạng của đường cong hàm log trong hình (1.3.1) cho thấy sự biến đổi này ánh xạ một phạm vi hẹp của các giá trị màu xám ở mức độ thấp trong hình ảnh đầu thành một phạm vi rộng hơn của đầu ra. Ngược lại cũng đúng với các giá trị đầu vào cao hơn. Chúng ta sẽ sử dụng một chuyển đổi loại hình này để mở rộng các giá trị điểm ảnh tối trong một hình ảnh trong khi nén các mức xám giá trị cao hơn.

Trong matlab dùng hàm:

```
g3 = im2double(a1);
g3 = c*log(1 +(g3))
imshow(g3);
```

Ví dụ: biến đổi ảnh với hàm log có giá trị c thay đổi là 3:



Hình 1.3.3. Hình biến đổi ảnh với hàm log.

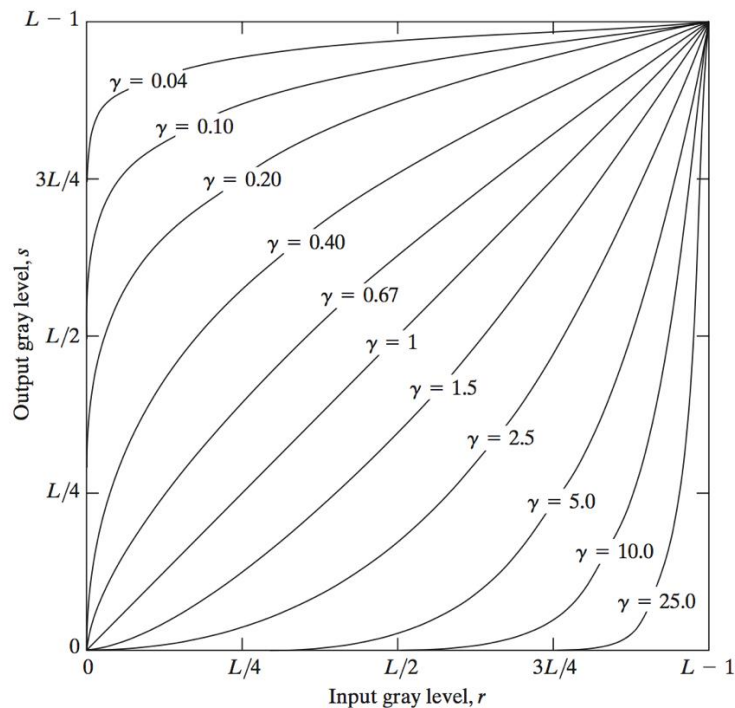
c) Biến đổi hàm mũ

Có dạng cơ bản là:

$$\begin{aligned} S &= cr^\gamma \\ \text{hoặc} \\ S &= c(r + \epsilon)^\gamma \end{aligned} \quad (4)$$

Trong đó: c và γ là hằng số dương.

Do có một khoảng chênh lệch, tuy nhiên thường bị bỏ qua vì đây là vấn đề canh chuẩn màn hình.



Hình 1.3.4. Đồ thị của hàm với giá trị khác nhau và với $c=1$.

Đồ thị của s theo r với có giá trị khác nhau như ta thấy ở hình 1.3.4. Như trong trường hợp của biến đổi hàm log, các đường cong hàm mũ với các giá trị của ánh xạ một phạm vi hẹp của đầu vào có giá trị đậm vào một phạm vi rộng lớn hơn của giá trị đầu vào, với điều ngược lại là đúng.

Không giống như các hàm log, ta thấy đây là một họ của các đường cong biến đổi có thể thu được từ khác nhau. Theo phán đoán, chúng ta thấy trong hình (6) các đường cong được tạo ra với giá trị của $\gamma > 1$ có hình dạng đối xứng với những đường cong tạo ra với các giá trị của $\gamma < 1$ qua đường thẳng $\gamma = 1$.

Các thiết bị sử dụng để chụp ảnh, in ấn, và màn hình hiển thị đáp ứng theo một định luật hàm mũ. Theo quy ước, số mũ trong phương trình hàm mũ được gọi là gamma (γ). Quá trình sử dụng để chỉnh sửa hàm mũ này được gọi là chỉnh sửa gamma.

Trong matlab dùng hàm:

```
g3 = im2double(a1);
g3 = c*h(g3.^gamma);
imshow(g3);
```

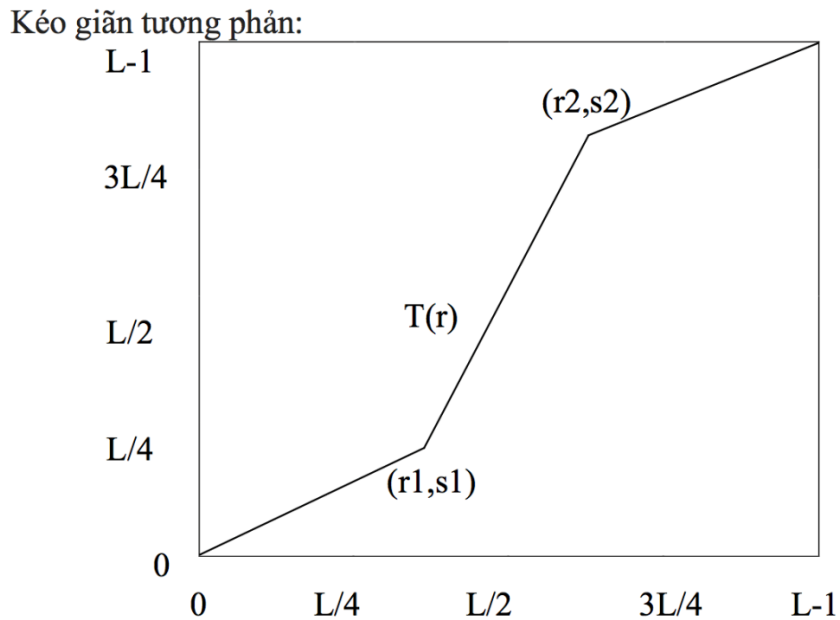
Ví dụ: giá trị c và γ thay đổi có giá trị $=3$.



Hình 1.3.5. Hình biến đổi với hàm mũ.

d) Hàm khoảng tuyến tính – tuyến tính từng đoạn

Một phương pháp khác là sử dụng hàm phân đoạn tuyến tính. Ưu điểm chính của hàm phân đoạn tuyến tính là có thể được tùy biến linh hoạt. Trong thực tế, một số biến đổi quan trọng có thể được thiết lập chỉ bằng hàm phân đoạn. Những bất lợi chính của hàm phân đoạn là đặc điểm kỹ thuật của nó yêu cầu người dùng nhập vào các thông số nhiều hơn đáng kể.



Hình 1.3.6. Đồ thị hàm chuyển đổi

Một trong những hàm phân đoạn tuyến tính đơn giản nhất là biến đổi kéo giãn tương phản. Hình ảnh tương phản thấp có thể là kết quả của việc chiếu sáng kém, các bộ cảm biến hình ảnh có dải động kém, hoặc thậm chí thiết lập sai khẩu độ ống kính trong quá trình chụp hình.

Từ đồ thị trong hình (8) ta có thể thấy:

Nếu $r_1 = r_2$ và $s_1 = s_2$, việc chuyển đổi là một hàm tuyến tính tạo ra mức xám không có thay đổi.

Nếu $r_1 = r_2$, $s_1 = 0$ và $s_2 = L - 1$, việc chuyển đổi sẽ trở thành một hàm tạo ngưỡng tạo ra một hình ảnh nhị phân.

Giá trị giữa của (r_1, s_1) và (r_2, s_2) tạo ra mức độ khác nhau của sự lây lan mức xám của hình ảnh đầu ra, làm ảnh hưởng đến độ tương phản của nó. Nói chung: $r_1 \leq r_2$ và $s_1 \leq s_2$ được giả định để các hàm là đơn trị và đơn điệu tăng. Điều kiện nhằm đảm bảo không tạo ra các thành phần có mức xám “lạ” trong hình ảnh xử lý.

Trong matlab dùng :

```
E=str2num(E);
g2=im2double(a1);
m=mean2(g2);
g3=1./(1+m./(g2+eps)).^E
imshow(a1);
```

Ví dụ: độ sáng cao có giá trị gần bằng 0 sẽ làm ảnh sáng hơn



Hình 1.3.7. Hình biến đổi với độ tương phản gần về 0.

Ví dụ: độ sáng thấp có giá trị xa 0 hơn sẽ làm ảnh tối hơn



Hình 1.3.8. Hình biến đổi với độ tương phản có giá trị >0 .

1.4 Xử lý Histogram

a) Khái niệm

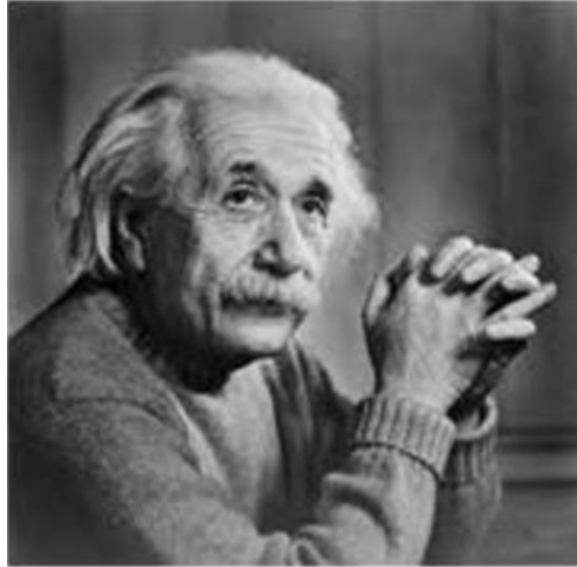
Histogram là một đồ thị, histogram thể hiện tần số của mọi thứ. Thông thường histogram có các thanh đại diện cho tần số xuất hiện của dữ liệu hay nói cách khác là histogram hiện thị sự phân bố tần số của một tập dữ liệu. Histogram của một hình ảnh cũng giống như histogram khác cũng cho thấy sự phân bố tần số. Nhưng biểu hình ảnh hiện thị tần số của các giá trị cường độ điểm ảnh.

Một histogram có hai trục:

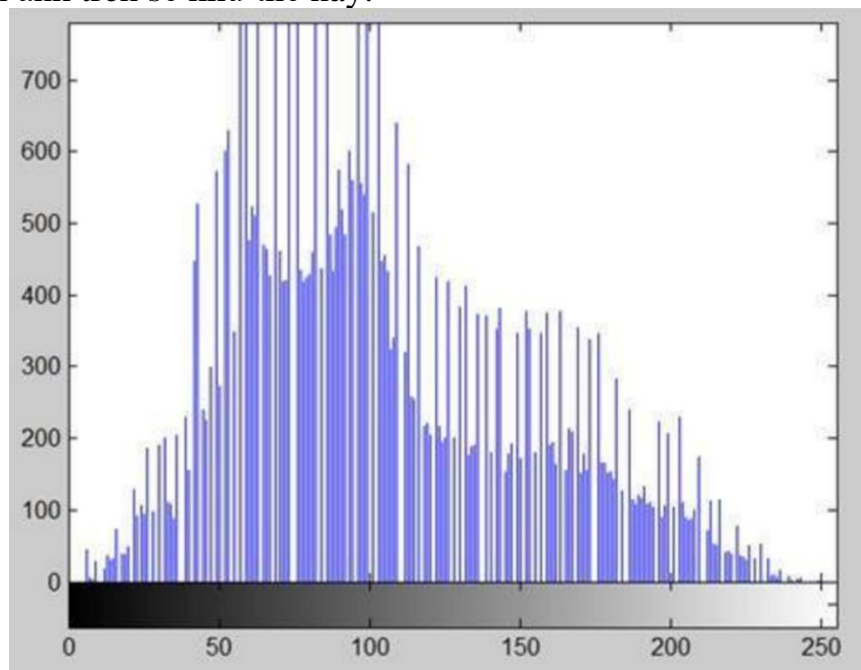
- + Trục x: chứa các điểm có sự phân bố tần số hay hiện thị cường độ mức xám.
- + Trục y: chứa tần số hay hiện thị tần số của cường độ.

Các độ cao khác nhau của thanh cho thấy tần số khác nhau của sự xuất hiện của dữ liệu.

Ví dụ:



Biểu đồ hình ảnh trên sẽ như thế này:



Từ biểu đồ thì ta thấy: trục x của biểu đồ thể hiện phạm vi của các giá trị pixel. Từ một hình ảnh 8 bpp có nghĩa là nó có 256 mức xám, đó là lý do trục x bắt đầu từ 0 và kết thúc tại 255 với mỗi khoảng cách là 50. Trong khi trục y thể hiện giá trị cường độ.

Xử lý histogram là bước đầu tiên trong tiền xử lý. Cân bằng histogram (histogram equalization: HE) và thay đổi histogram theo yêu cầu là hai phương pháp được sử dụng phổ biến để thay đổi histogram của một hình ảnh. Histogram là

một hàm rời rạc đại diện cho tần số xuất hiện của tất cả các mức xám trong hình ảnh. Biểu đồ được cho là:

$$h(r_k) = \frac{n_k}{N} \quad (5)$$

Trong đó : r_k là cường độ mức xám và n_k là số pixel có mức xám tương ứng với mỗi giá trị r_k

b) Cân bằng Histogram

Cân bằng histogram là một kỹ thuật phổ biến để tăng cường sự diện mạo của hình ảnh. Cân bằng histogram kéo dài biểu đồ trên toàn bộ quang phổ của các điểm ảnh (0-255). Nó làm tăng độ tương phản của hình ảnh và có thể được áp dụng để bình thường hóa sự chiếu sáng.

Quá trình này khá đơn giản và với từng cấp độ sáng j trong hình ảnh ban đầu, ta sẽ có giá trị mức xám mới của điểm ảnh (k) được tính như trong phương trình sau:

$$k = \sum_{j=0}^J \left(\frac{N_j}{T} \right) \quad (6)$$

Trong đó K là tổng số điểm ảnh trong hình ảnh có độ sáng bằng hoặc nhỏ hơn j , và T là tổng số điểm ảnh. Mục đích chính của cân bằng biểu đồ là tìm hàm biến đổi mức xám T để chuyển đổi hình ảnh f nhờ đó mà biểu đồ của $T(f)$ "cân bằng". Cho phương trình sau $s=T(r)$ trong đó r là mức độ xám $[0,1]$ (0 là đen nhất và 1 là sáng nhất). Qua phép biến đổi T ra luôn có 1 giá trị của s tương ứng với 1 giá trị r . Để thỏa điều kiện ta tìm được mỗi s từ 1 giá trị của r tương ứng thì cần thỏa 2 điều kiện sau:

$T(r)$ có giá trị duy nhất và tăng đồng nhất trên khoảng $r = [0, 1]$.

$T(r) = [0, 1]$ với $r = [0, 1]$.

Chú giải: Điều kiện đầu tiên $T(r)$ có giá trị duy nhất nhằm hàm nghịch T^{-1} tồn tại, và điều kiện tăng đồng nhất nhằm đảm bảo thứ tự tăng dần từ đen \rightarrow trắng trong hình ảnh ngõ ra. Điều kiện thứ 2 nhằm đảm bảo hình ngõ ra sẽ có chung kho ảnh mức xám đồng nhất với đầu vào.

Ta biết rằng mức xám trong một ảnh có thể coi như một biến ngẫu nhiên trong khoảng $[0,1]$. Để diễn tả thích hợp nhất cho sự phân bố của biến ngẫu nhiên này người ta dùng Hàm mật độ xác suất (PDF: Probability Density Function). Cho $p_r(r)$ và $p_s(s)$ thể hiện PDF của biến ngẫu nhiên r và s . Nếu $p_r(r)$ và $T(r)$ biết trước và $T^{-1}(s)$ thỏa điều kiện (a) thì ta sẽ có công thức liên hệ giữa p_r và p_s như sau:

$$P_s(s) = P_r(r) \left| \frac{dr}{ds} \right| \quad (7)$$

Ngoài ra ta còn có:

$$s = T(r) = \int_0^r p_r(w)dw \quad (8)$$

Trong đó : $\int_0^r p_r(w)dw$ được xem là hàm phân bố tích lũy của biến ngẫu nhiên r .

Một phiên bản cải tiến của Cân bằng Histogram là Cân bằng Histogram thích ứng.

Cân bằng mức xám trong matlab:

```
a1=rgb2gray(a1);  
g1=histeq(a1,nlev);  
imshow(a1);
```

Ví dụ với mức xám bằng 200:



Hình 1.4.1. Hình biến đổi với mức xám bằng 200

c) Thay đổi Histogram theo yêu cầu

Trong phần trình bày trước về cân bằng Histogram như là một phương pháp chỉnh sửa ảnh bằng việc tự động xác định hàm truyền để tạo ra hình ảnh sau xử lý có histogram đồng nhất và CDF thẳng dốc. Tuy nhiên phương pháp này không hẳn không có hạn chế, như trong ví dụ sau:



Hình 1.4.2. Ví dụ về việc thất bại khi dùng cân bằng Histogram

Bức tranh chụp cảnh hoàng hôn bên trái rất đẹp và màu hài hòa nhưng khi qua xử lý ảnh thì ra 1 ảnh không như mong muốn và mất hẳn tính thẩm mỹ. Phần này sẽ trình bày về phương pháp tạo ra hình ảnh dựa trên histogram đã được định hình. Phương pháp này có tên là: thay đổi histogram theo yêu cầu.

Sự hình thành của phương pháp thay đổi histogram theo yêu cầu được tóm tắt qua các bước sau:

Bước 1: Tìm Histogram của ảnh đầu vào $p_r(r)$ và xác định biến đổi cân bằng của nó:

$$S = T(r) = (L - 1) \int_0^r p_r(w)dw \quad (9)$$

Bước 2: Sử dụng PDF $p_z(r)$ của ảnh đầu ra để tìm hàm truyền.

$$G(z) = (L - 1) \int_0^z p_z(t)dt \quad (10)$$

Bước 3: Tìm hàm truyền nghịch đảo $z=G^{-1}(s)$, ánh xạ từ s qua z .

$$z = G^{-1}[T(r)] = G^{-1}(s) \quad (11)$$

Bước 4: Tìm ảnh đầu ra bằng việc cân bằng ảnh đầu vào, sau đó cân bằng từng pixel, thực hiện ánh xạ ngược để tìm điểm ảnh đầu ra tương ứng.

Chúng ta cần lưu ý thêm rằng ngay cả trong trường hợp rời rạc thì hàm nghịch của G là G^{-1} cũng phải thỏa 2 điều kiện sau:

- + Đơn điệu (luôn tăng hoặc giảm).
- + Đơn trị.

Hai điều kiện trên nhằm mục đích không có giá trị nào trong Histogram đã được match $P_z(z)$ trong phương trình sau bằng 0:

$$V_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k \quad (12)$$

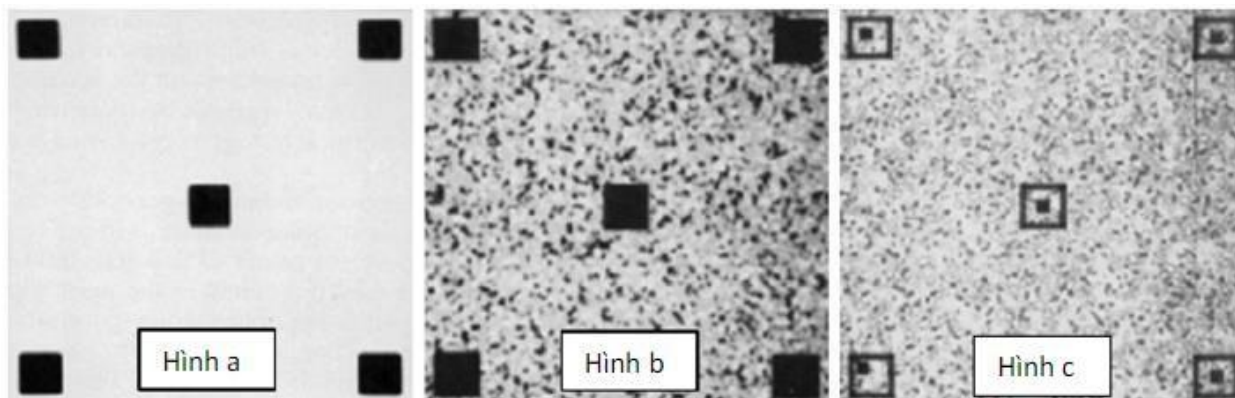
Thay đổi histogram theo yêu cầu giúp chúng ta tương ứng phân bố thang màu xám của ảnh làm mẫu cho một ảnh khác.

d) Nâng cao chất lượng cục bộ

Các phương pháp xử lý histogram được thảo luận đề cập trong hai phần trước mang tính toàn cục, nghĩa là điểm ảnh được sửa đổi bởi một hàm chuyển đổi mức xám cấp toàn bộ hình ảnh. Mặc dù cách tiếp cận toàn cục phù hợp để nâng cao

tổng thể, nhưng có những trường hợp ta chỉ cần thiết để tăng cường thông tin chi tiết trên những vùng nhỏ trong một số hình ảnh. Số điểm ảnh trong các khu vực này có thể có ảnh hưởng không đáng kể đến toàn cục của bức ảnh. Giải pháp là để đưa ra hàm chuyển đổi dựa trên phân phối mức xám tính trong vùng lân cận.

Ví dụ sau sẽ minh họa:



Hình 1.4.3. Ví dụ minh họa về việc nâng cao chất lượng ảnh cục bộ.

Hình a: Ảnh gốc, đã được làm mờ và giảm nhiễu.

Hình b: Kết quả của việc cân bằng Histogram toàn cục.

Hình c: Kết quả của việc cân bằng Histogram cục bộ sử dụng vùng lân cận có kích thước 7x7 pixel.

Nhận xét: Hình (1.4.3 b) cho thấy tăng cường nhiễu đáng kể, cộng thêm vào đó là tương phản nhẹ. Lưu ý rằng không có chi tiết cấu trúc mới đã được đưa ra bằng phương pháp này. Tuy nhiên, cân bằng histogram cục bộ sử dụng một vùng lân cận 7x7 tiết lộ sự hiện diện của hình vuông nhỏ bên trong các hình vuông màu lớn hơn. Các ô vuông nhỏ là quá gần trong mức xám, và kích thước của chúng quá nhỏ để ảnh hưởng đến cân bằng histogram toàn cục. Cũng lưu ý các kết cấu nhiễu tốt hơn trong hình (1.4.3c) là kết quả của chế biến địa phương sử dụng các vùng lân cận tương đối nhỏ.

1.5 Nâng cao chất lượng hình ảnh sử dụng toán tử logic và số học

Có hai nhóm thao tác đại số áp dụng lên ảnh là: số học và logic. Thao tác logic thực hiện trên từng điểm ảnh bao gồm: AND, OR, NOT. Để sử dụng toán tử logic trên thang màu của ảnh, thì các giá trị điểm ảnh cần phải chuyển về chuỗi giá trị nhị phân. Ví dụ thực hiện toán tử NOT cho màu đen, điểm ảnh 8-bit (một chuỗi 8 số 0) cho ra một điểm ảnh màu trắng (11111111). Toán tử logic NOT thực hiện giống với hàm chuyển đổi âm. Toán tử AND và OR được sử dụng cho làm mặt nạ. Trong những phần mặt nạ che khuất của ảnh AND và OR màu sáng là nhị phân 1, màu tối là 0. Mặt nạ masking còn được gọi là xử lý vùng lợi ích. Toán tử logic thường được dùng với các toán tử hình thái học.

Trong 4 phép tính toán học, phép trừ và cộng thường được sử dụng trong nâng cao chất lượng ảnh. Ta chia 2 bức ảnh một cách đơn giản bằng cách nhân một hình ảnh với nghịch đảo một hình khác. Bỏ qua toán tử nhân một hình ảnh với một hằng số để tăng mức xám trung bình, thì phép nhân ảnh sử dụng trong nâng cao chất lượng ảnh trở nên phổ biến hơn phép masking logic ở phần trên.

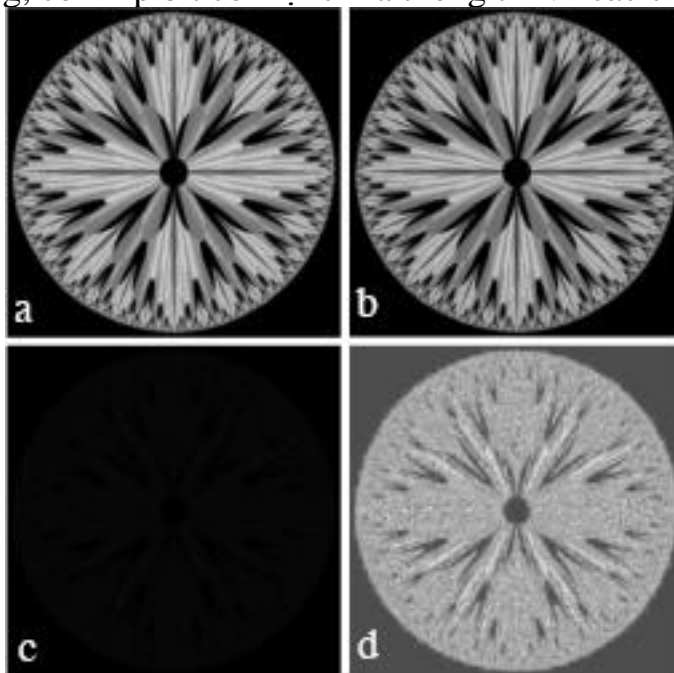
Chú ý: Với 2 biên nhị phân a và b , $a \text{ AND } b = 1$ khi cả $a=b=1$, nếu không thì kết quả là 0, tương tự kết quả $a \text{ OR } b = 0$ khi cả $a = b = 0$, nếu không thì kết quả bằng 1. Cuối cùng, $a=1$, $\text{NOT } a$ cho kết quả là 0 và ngược lại.

a) Toán tử trừ hình ảnh

Sự khác nhau giữa 2 điểm ảnh $f(x,y)$ và $h(x,y)$ là:

$$g(x,y) = f(x,y) - h(x,y) \quad (13)$$

Công thức trên được áp dụng để tính toán sự khác nhau giữa những cặp điểm tương ứng. Điều quan trọng của phép trừ hai ảnh là nó làm tăng sự khác nhau giữa 2 bức ảnh. Như đã biết, lớp có số thứ tự bit cao chứa một lượng lớn những thông tin hình ảnh quan trọng, còn lớp bit còn lại chứa thông tin về các chi tiết mịn.



Hình 1.5.1. a - ảnh gốc, b - ảnh sau khi thiết lập giá trị 4 lớp bit về giá trị 0, c - khác nhau giữa a và b, d - điều chỉnh biểu đồ ảnh c

Hình 1.5.1a là hình ảnh có dạng gấp khúc được dùng để miêu tả lý thuyết lớp bit ở phần trước. Hình 1.5.1b là hình đã bỏ đi 4 lớp bit quan trọng nhất trên hình gốc, hình 1.5.1a có kết quả gần như gốc nhưng đường nét trong hình hơi vỡ vì có ít giá trị màu xám hơn so với hình gốc. Vì sự chênh lệch giá trị của từng điểm ảnh

trên 2 bức ảnh rất nhỏ nên có kết quả hình 1.5.1c. Để làm rõ các chi tiết hơn, ta có thể thực hiện kéo dài mức phản xạ. Ta chọn điều chỉnh lược đồ nhưng sử dụng hàm chuyển đổi lũy thừa để có thể là cách chọn hợp lý hình 1.5.1d. Hình 1.5.1d được dùng để đánh giá hiệu ứng khi đặt các lớp bit có thứ tự thấp bằng 0.

b) Trung bình hình ảnh

Xét mức nhiễu ảnh $g(x)$ được hình thành bằng cách thêm mức nhiễu $\eta(x,y)$ vào ảnh gốc $f(x,y)$ ta có:

$$g(x, y) = f(x, y) + \eta(x, y) \quad (14)$$

Trong đó, giả định tại mọi cặp tọa độ (x,y) mức nhiễu ảnh là không tương quan trọng và có giá trị trung bình là 0. Mục tiêu phương pháp này để giảm nhiễu của ảnh bằng cách thêm vào một tập hợp mức nhiễu ảnh $\{g_i(x, y)\}$.

Nếu mức nhiễu thỏa mãn yêu cầu trên, ta có thể thấy rằng mức nhiễu trung bình của ảnh được tính bằng trung bình $\bar{g}(x, y)$ của bức ảnh có K mức nhiễu khác nhau.

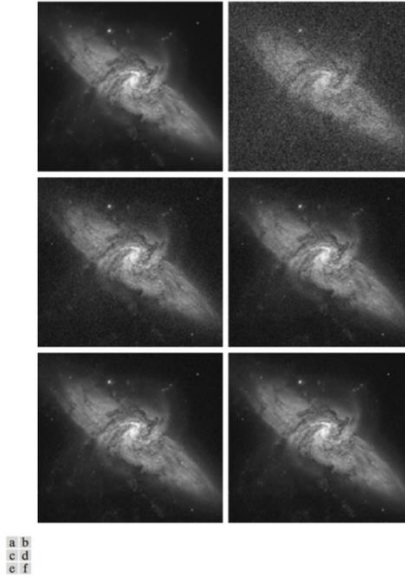
$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y) \quad (15)$$

Và

$$E\{\bar{g}(x, y)\} = f(x, y) \quad (16)$$

$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{K} \sigma_{\eta(x,y)}^2$$

Khi K tăng, theo công thức và độ biến thiên giá trị điểm ảnh ở cùng từng vị trí (x,y) giảm. Vì đó $E\{\bar{g}(x, y) = f(x, y)\}$ điều đó nghĩa là $\bar{g}(x, y)$ tiến gần tới $f(x,y)$ khi giá trị mức nhiễu ảnh được sử dụng trong phương pháp này tăng lên.



Hình 1.5.2. a: ảnh gốc, b: ảnh được điều chỉnh nhiễu với trung bình 0 và độ lệch chuẩn mức xám, c-f: kết quả áp dụng khử nhiễu bằng phương pháp trung bình hình ảnh

1.6 Lọc ảnh trong không gian

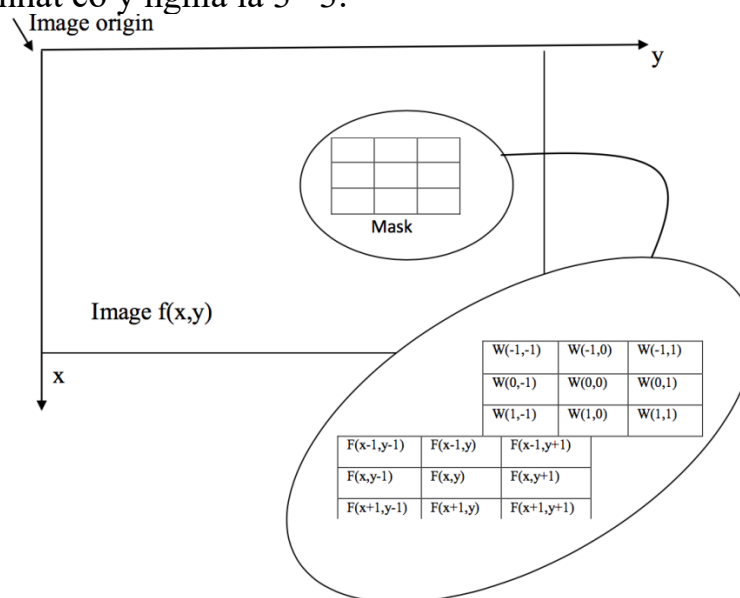
a) Giới thiệu chung

Nhiều thường xuất hiện trên ảnh do nhiều nguyên nhân khác nhau. Để giảm nhiễu và nâng cao chất lượng ảnh ta sử dụng những phương pháp lọc khác nhau, phù hợp với mỗi loại nhiễu cụ thể. Trong chương này ta sẽ xét đến 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: (1) xác định điểm ảnh trung tâm (x,y); (2) thực hiện các phép toán với các điểm xung quanh (x,y); (3) kết quả ta được đáp ứng của quá trình lọc tại (x,y); (4) lặp lại các bước trên với tất cả các điểm ảnh khác.

b) Lọc tuyến tính

Khái niệm: Lọc tuyến tính là phương pháp lọc trong đó mức xám mỗi pixel của ảnh mới là tổ hợp tuyến tính của các mức xám của các pixels lân cận, tức là mỗi pixel lân cận sẽ được nhân với một hệ số tương ứng rồi được cộng lại để được đáp ứng tại điểm ảnh trung tâm. ($m \times n$). Trong Matlab, các hệ số này được sắp xếp trong một ma trận kích thước $m \times n$, gọi là bộ lọc. Cơ chế lọc được thực hiện bằng cách di chuyển tâm của mặt nạ qua lần lượt từng điểm ảnh và thực hiện tính tổng các tích của mức xám các điểm ảnh xung quanh với hệ số bộ lọc. Kích thước bộ lọc là lẻ. Kích thước nhỏ nhất có ý nghĩa là 3×3 .



Hình 1.6.1. Sơ đồ lọc không gian

- Ta có biểu thức tính mức xám tại (x,y) công thức:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (17)$$

Có hai khái niệm khi chúng ta thực hiện phương pháp lọc tuyến tính, đó là tương quan (correlation) và tích chập (convolution).

Tương quan là quá trình dịch bộ lọc qua từng điểm ảnh như ta đã đề cập.

Tích chập cũng sử dụng quá trình tương tự, ngoại trừ bộ lọc w quay 180 trước khi tiến hành dịch bộ lọc.

Trong Matlab: cung cấp cho ta hàm `imfilter` để thực hiện lọc tuyến tính có cú pháp của hàm như sau:

```
>> g = imfilter(f, w, filtering_mode, boundary_options, size_options)
```

Trong đó f là ảnh gốc, g là ảnh sau xử lý.

- Các thông số tùy định: `filtering_mode` có gồm:
 - + “corr” thực hiện phép toán tương quan
 - + “conv” thực hiện phép chập.
- `Size_options` có thể là: “same” hoặc “full”.
- Thông số `boundary_options` cho ta cách thức chèn đệm(padding).
- Matlab cho ta 4 cách thức:
 - + **P**: Biên ảnh gốc sẽ được thêm padding có giá trị mức xám là P , mặc định là 0
 - + “replicate”: các mức xám bên ngoài lặp lại giá trị của biên.
 - + “symmetric”: các mức xám bên ngoài đối xứng gương với các điểm ảnh bên trong qua biên.
 - + “circular”: ảnh được thêm padding trên cơ sở giả thiết ảnh đầu vào là một hàm tuần hoàn.
- Các bộ lọc tuyến tính trong matlab:
 - + 'average' `fspecial('average', [r c]).`
 - + 'disk' `fspecial('disk', r).`
 - + 'gaussian' `fspecial('gaussian', [r c], sig).`
 - + 'laplacian' `fspecial('laplacian', alpha).`
 - + 'log' `fspecial('log', [r c], sig).`
 - + 'motion' `fspecial('motion', len, theta).`
 - + 'prewitt' `fspecial('prewitt').`
 - + 'sobel' `fspecial('sobel').`
 - + 'unsharp' `fspecial('unsharp', alpha).`

Một vài ví dụ minh họa lọc tuyến tính trong matlab:

Ví dụ 1:

```
A = imread(hinh.jpg); k=fspecial("motion",50,54);
```

$B = \text{imfilter}(a, k, \text{'symmetric'})$;



Hình 13.1. Ảnh trước và sau khi lọc bộ lọc $\text{fspecial}(\text{'motion'}, 50, 54)$ và padding là 'symmetric'

Ví dụ 2:

```
f = imread('peppers.png');
w4 = fspecial('laplacian', 0);
f = im2double(f);
g4 = f - imfilter(f, w4, 'replicate');
```



Hình 13.2. Ảnh trước và sau khi lọc bộ lọc $\text{fspecial}(\text{'laplacian'}, 0)$ và padding là 'replicate'

Ví dụ 3:

```
I = imread('peppers.png');
g = fspecial('gaussian', [5 5], 1.5);
subplot(2,3,1), imshow(I), title('Anh goc');
Is = imfilter(I, g);
```



```
subplot(2,3,2), imshow(Is),title('Anh da loc gauss');
```



Hình 1.6.2. Ảnh trước và sau khi lọc bộ lọc `fspecial('gaussian',[5 5],1.5)`

Ví dụ 4:

```
I=imread('peppers.png');
k = fspecial('log', [5 5], 0.1);
B=imfilter(I,k);
subplot(1,2,1), imshow(I),title('Anh goc');
subplot(1,2,2), imshow(B),title('Anh da loc');
```



Hình 1.6.3. Ảnh trước và sau khi lọc bộ lọc `k = fspecial('log', [5 5], 0.1)`

Ví dụ 5:

```
A=imread('peppers.png');
h2=fspecial('average',[7 10]);
subplot(1,2,1); imshow(A),title('Anh goc');
B=imfilter(A,h2);
subplot(1,2,2); imshow(B),title('Anh goc');
```



Hình 1.6.4. Ảnh trước và sau khi lọc bộ lọc $k = fspecial('average',[710])$

c) Lọc phi tuyến tính

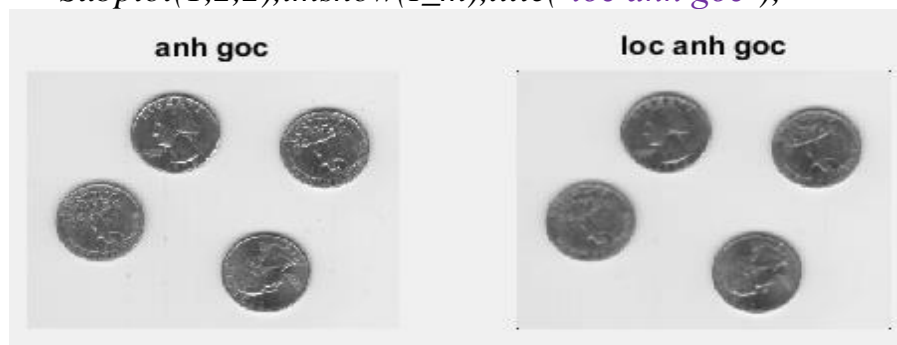
Khái niệm: lọc phi tuyến sử dụng một cửa sổ lọc và trượt qua các pixels của ảnh gốc. Tuy nhiên nếu lọc tuyến tính dựa theo việc lấy tổng có trọng số các pixels lân cận thì lọc phi tuyến sẽ thực hiện một phép toán phi tuyến với các pixels đó.

- Lọc trung vị (Median filtering): thay thế điểm ảnh đích bằng giá trị trung vị của các điểm ảnh lân cận và chính điểm ảnh đó. Bộ lọc Median sẽ tìm giá trị trung vị trong tất cả các điểm ảnh và thay vào điểm ảnh trung tâm.

$$v(m,n) = \text{trungvi}(y(m-k, n-l)) \text{ với } \{k,l\} \in W \quad (19)$$

Trong matlab:

```
I=imread('eight.tif');
I_m=medfilt2(I,[3 3]);
Subplot(1,2,1),imshow(I),title('anh goc');
Subplot(1,2,2),imshow(I_m),title('loc anh goc');
```

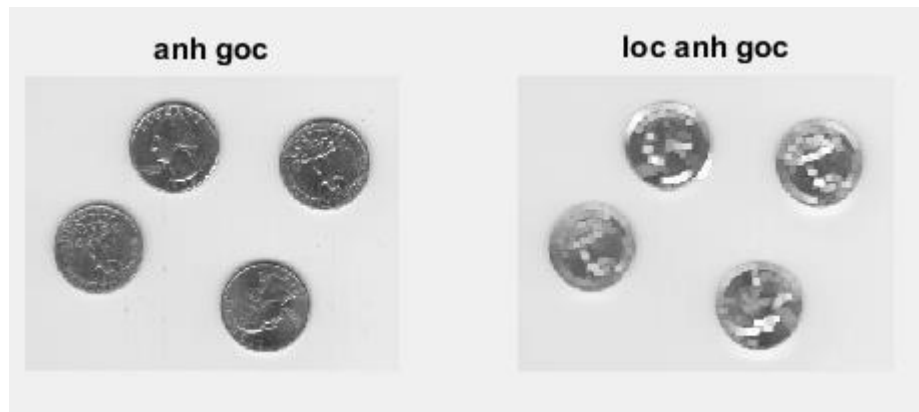


Hình 1.6.5. Ảnh trước và sau khi lọc bộ lọc trung vị

- Lọc hạng(Rank) : Tùy vào bậc bộ lọc mà chúng ta đã chọn có giá trị tối đa hay tối thiểu trong vùng lân cận được gọi là các bộ lọc tối đa hay tối thiểu.

Trong matlab:

```
I=imread('eight.tif');
I_m = ordfilt2(I,(5,5));
subplot(1,2,1), imshow(I),title('anh goc');
subplot(1,2,2), imshow(I_m), title('loc anh goc');
```



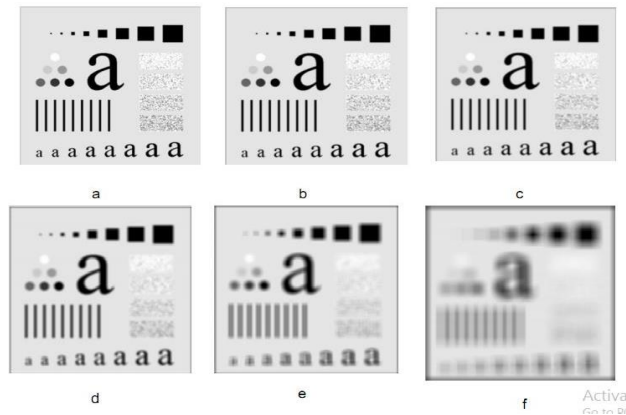
Hình 1.6.6. Ảnh trước và sau khi lọc bộ lọc hạng (Rank)

1.7 Ứng dụng lọc ảnh trong không gian

a) Các bộ lọc làm mịn ảnh

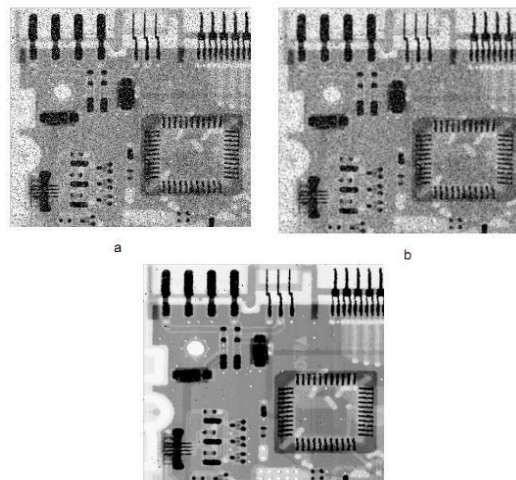
Các bộ lọc làm mịn ảnh được sử dụng để làm mờ và giảm nhiễu

Ví dụ 1:



Hình 1.7.1. Lọc tuyến tính làm mịn ảnh: a - ảnh gốc, b – 3x3, c – 5x5, d – 9x9, e – 16x16, f – 35x35

Ví dụ 2:



Hình 15. Bộ lọc hạng làm mịn ảnh: a - ảnh bị nhiễu, b - ảnh dùng bộ lọc trung bình 3x3, c - ảnh dùng bộ lọc trung vị

b) Các bộ lọc làm sắc hình ảnh

Mục đích của việc làm sắc ảnh là nổi bật các chi tiết trong ảnh hoặc làm sắc các chi tiết bị mờ bởi quá trình làm mờ ảnh

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

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

Trong hệ số alpha cho phép chỉnh mức độ sắc nét của hình ảnh.

Bộ lọc tăng cường (high-boost filter): Bộ lọc tăng cường cũng là một ứng dụng của toán tử Laplace.

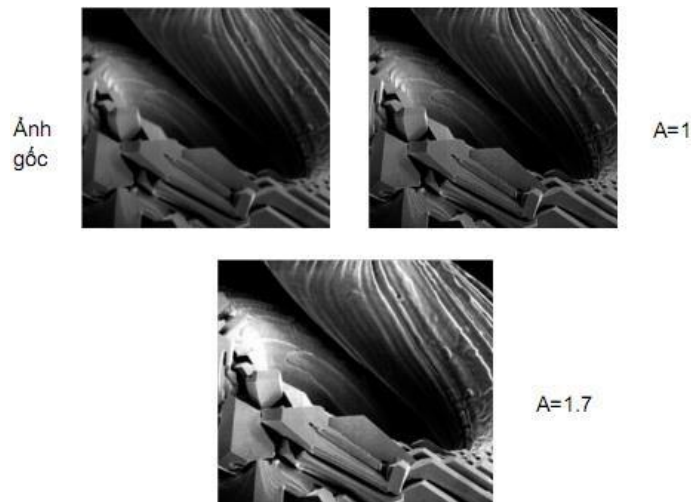
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

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 1 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.

Ví dụ:



Hình 1.7.2 Bộ lọc tăng cường ảnh

PHẦN 2. MÔ PHỎNG ỨNG DỤNG

Chương 1. Giới thiệu cơ bản về MATLAB

MATLAB viết tắt của Matrix Laboratory là một bộ phần mềm toán học của hãng Mathworks để lập trình, tính toán số và có tính trực quan rất cao.

MATLAB là một công cụ tính toán trong kỹ thuật, đặc biệt là các bài toán về ma trận. Ma trận cỡ “ $m \times n$ ” là bảng hình chữ nhật gồm m hàng và n cột. MATLAB có thể làm việc với nhiều kiểu dữ liệu khác nhau. Với chuỗi kí tự cũng xem là một dãy các kí tự hay là dãy mã số của các ký tự.

MATLAB dùng để giải quyết các bài toán về giải tích số, xử lý tín hiệu số, xử lý đồ họa,... mà không phải lập trình cổ điển

Hiện nay, MATLAB có đến hàng ngàn lệnh và hàm tiện ích. Ngoài các hàm cài sẵn trong chính ngôn ngữ thì MATLAB còn có các lệnh và hàm ứng dụng chuyên biệt trong các Toolbox để mở rộng môi trường MATLAB nhằm giải quyết các bài toán thuộc các phạm trù riêng. Các Toolbox khá quan trọng và tiện ích cho người dùng như toán sơ cấp, xử lý tín hiệu số, xử lý ảnh, xử lý âm thanh, logic mờ,...

Chương 2. Chương trình mô phỏng

2.1 Giao diện chương trình

a) Màn hình chờ



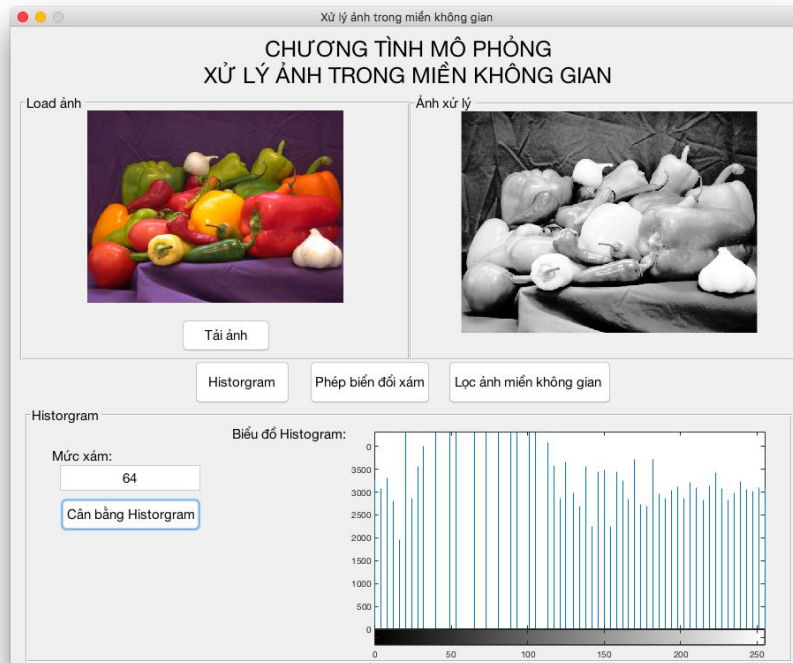
Hình 2.1.a. Màn hình welcome

b) Giao diện chính



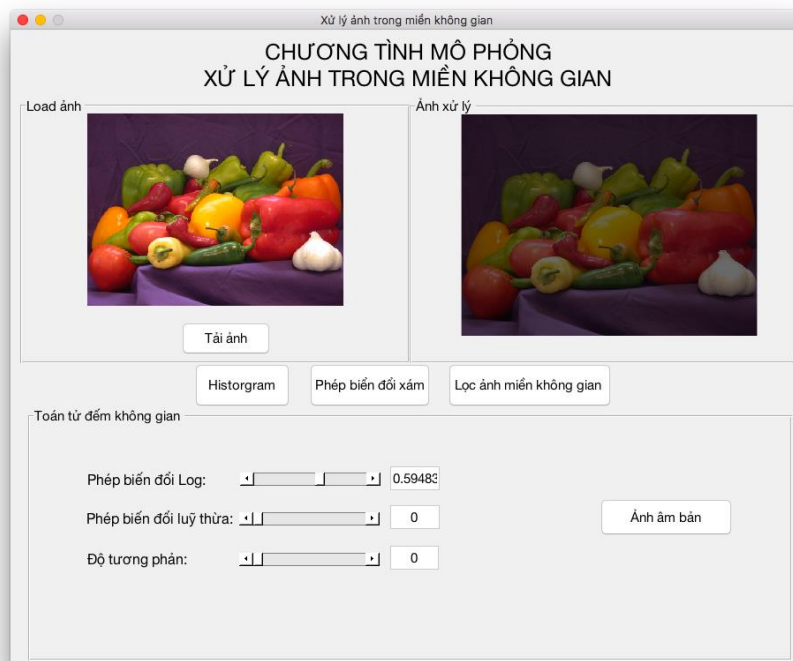
Hình 2.1.b. Giao diện chính của chương trình

c) Cân bằng Histogram



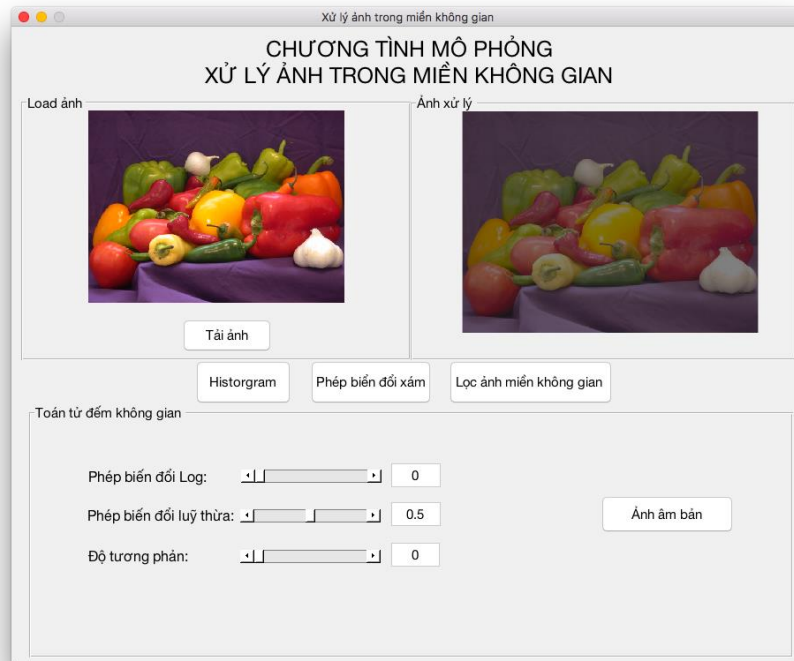
Hình 2.1.c. Cân bằng Histogram. Đầu vào là mức xám.

d) Phép biến đổi Log



Hình 2.1.d. Phép biến đổi Log. Đầu vào là một hằng số c

e) Phép biến đổi Luỹ thừa



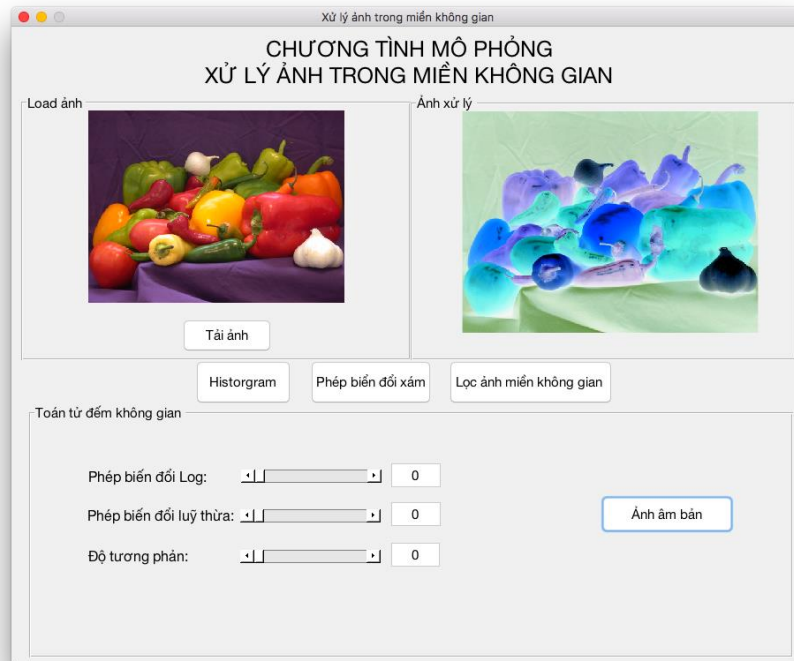
Hình 2.1.e Phép biến đổi Luỹ thừa. Đầu vào là một hằng số c

f) Độ tương phản



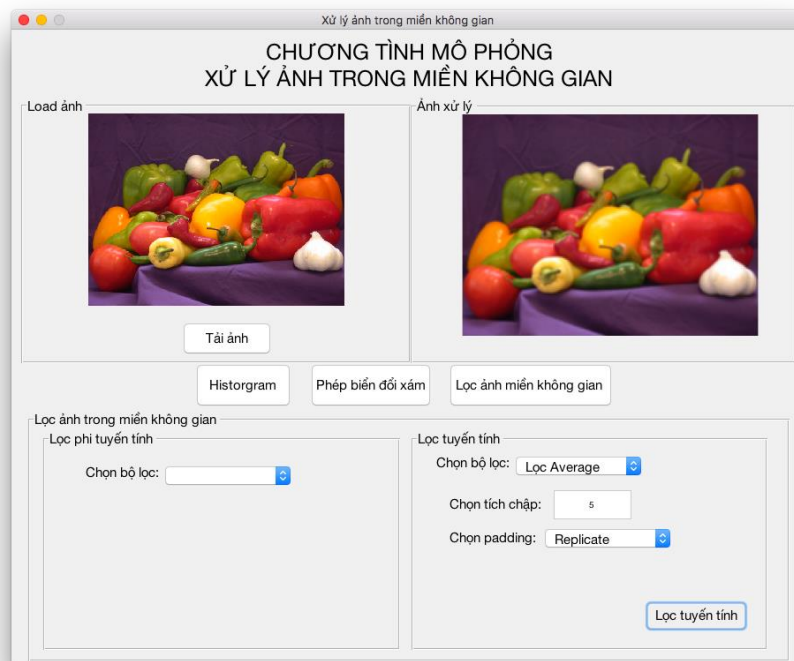
Hình 2.1.f. Độ tương phản ảnh. Đầu vào là một hằng số

g) Ảnh âm bản



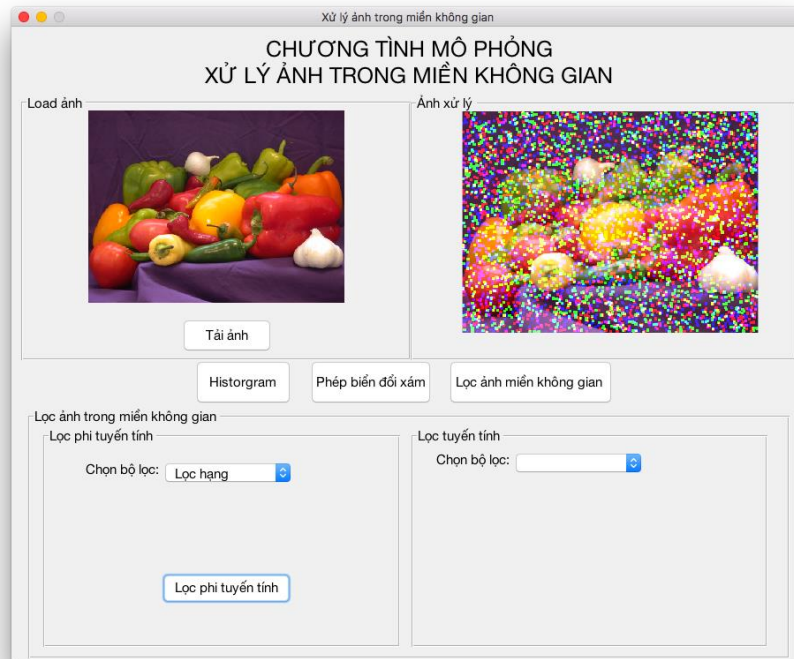
Hình 2.1.g. Ảnh âm bản.

h) Lọc tuyến tính Average



Hình 2.1.h. Lọc tuyến tính Average. Đầu vào là tích chập, tùy chọn Padding.

i) Lọc phi tuyến Rank



Hình 2.1.i. Lọc phi tuyến tính Rank.

TÀI LIỆU THAM KHẢO.

Lương Mạnh Bá, Nguyễn Thanh Thủy. Nhập môn xử lý ảnh số. Nhà xuất bản Khoa học Kỹ thuật, 2003

Võ Đức Khánh, Hoàng Văn Kiém. Giáo trình xử lý ảnh số. Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh.

Nguyễn Kim Sách. Xử lý ảnh và Video số. Nhà xuất bản Khoa học và Kỹ thuật, 1997.

Nguyễn Quốc Trung. Xử lý tín hiệu và lọc số. Nhà xuất bản Khoa học và Kỹ thuật, 2004.

Bhabatosh Chanda, Dwijesh Dutta Maumder. Digital Imge Processing and Analysis.

KẾT LUẬN

Xử lý ảnh bằng các thuật toán đại số giúp chúng ta có nhiều cách tiếp cận đa dạng trong việc chỉnh sửa ảnh để đạt được những mong muốn. Có nhiều phương pháp và mỗi phương pháp sẽ có thuận lợi và bất thuận lợi riêng, vậy nên việc chọn phương pháp hợp lý là điều quan trọng.

- Âm ảnh giúp chúng ta phát hiện những chi tiết trắng ẩn chứa trong nền màu đen, điều này rất ý nghĩa trong y khoa.
- Biến đổi hàm mũ thì có ý nghĩa trong việc điều chỉnh tương phản của hình.
- Biến đổi hàm log có hữu ích trong việc tăng thêm chi tiết vùng tối của hình ảnh. Một hình ảnh bị loá thì khu vực mức xám tập trung nhiều sẽ được tìm thấy khi ta cho $\gamma > 1$.
- Histogram cung cấp những thông tin khá hữu ích về một bức ảnh. Cân bằng histogram thực hiện việc tăng độ tương phản bằng cách phân bổ lại các mức xám sao cho trải dài các biên độ xám từ 1-255.
- Thay đổi histogram theo yêu cầu cho phép ta chỉnh sửa một bức ảnh dựa trên một bức ảnh chuẩn (đã được chỉnh sửa) một cách nhanh chóng. Nguyên lý là tương ứng histogram của ảnh xử lý với ảnh mẫu dùng để đối chiếu.
- Theo lý thuyết mỗi phương pháp đều có những điểm mạnh và điểm yếu nên trong thực tế người ta thường kết hợp nhiều phương pháp thông qua nhiều quá trình chỉnh sửa để thu được hình mong muốn.