

**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 1

GIỚI THIỆU

1.1. HỆ THỐNG XỬ LÝ ẢNH

Trong những năm gần đây, mặc dù còn rất mới mẻ trong lĩnh vực khoa học và công nghệ nhưng xử lý ảnh đang được nghiên cứu và phát triển với tốc độ nhanh chóng bởi các trung tâm nghiên cứu, trường đại học và viện... với rất nhiều ứng dụng trên các lĩnh vực khác nhau. Xử lý ảnh chỉ mới được đưa vào giảng dạy ở bậc đại học Việt Nam trong những năm gần đây dành cho một số ngành học nhất định. Đây là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác: đại số tuyến tính, xác suất thống kê, xử lý số tín hiệu, trí tuệ nhân tạo... cũng được đề cập và phân tích trong nội dung.

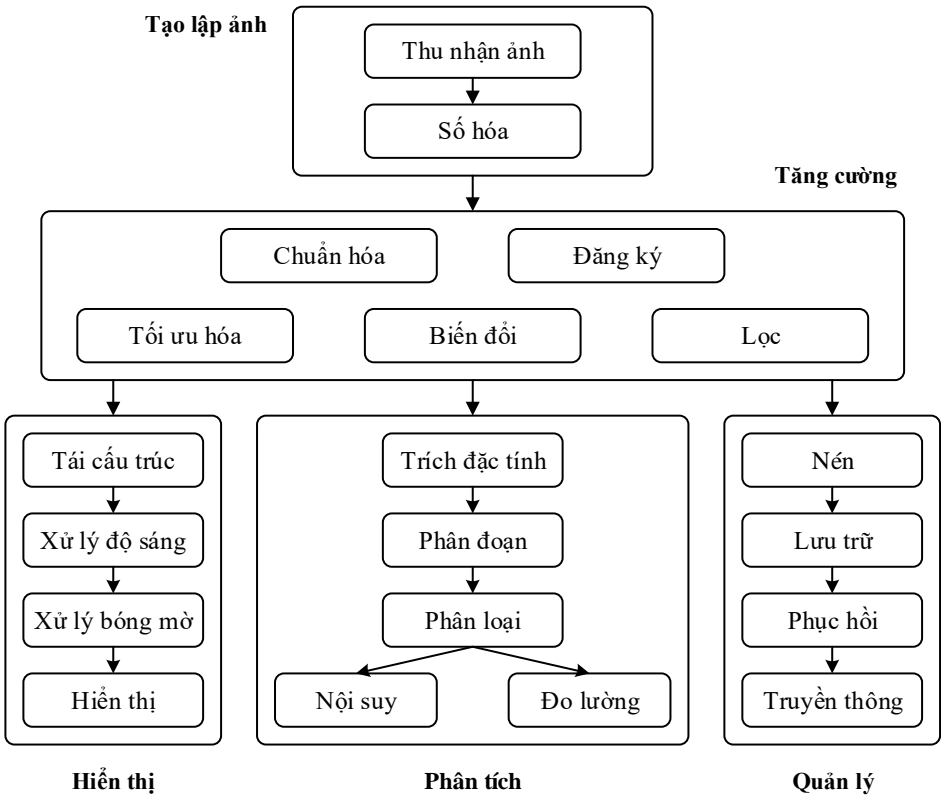
Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng nhằm nâng cao chất lượng bằng các phương pháp phân tích được nghiên cứu trong giai đoạn thiết bị phần cứng bị hạn chế, chẳng hạn như nâng cao độ sáng hay độ phân giải của hình ảnh... Về sau, sự xuất hiện và phát triển mạnh của máy tính đã tạo điều kiện hơn nữa cho quá trình thực hiện các thuật toán xử lý ảnh. Ứng dụng của xử lý ảnh ngày càng được mở rộng sang các lĩnh vực khác, chẳng hạn như viễn thông, điều khiển tự động, giao thông thông minh, kỹ thuật y sinh... Đặc biệt, trong các thiết bị kỹ thuật số có màn hình hiển thị luôn ưu tiên một phần tài nguyên dành cho việc xử lý hình ảnh: máy tính xách tay, camera kỹ thuật số, điện thoại thông minh, tivi thông minh...

Thông thường, xử lý ảnh số bao gồm nhiều vấn đề khác nhau, tuy nhiên, có thể chia thành bốn lĩnh vực chính như hình 1.1: tạo lập ảnh, biểu diễn, phân tích và quản lý. Bên cạnh đó, các thuật toán tăng cường ảnh có thể được xem xét như là một bước tiền hay hậu xử lý trong tất cả các lĩnh vực trên.

- Tạo lập ảnh: bao gồm tất cả các bước từ quá trình thu nhận ảnh đến hình thành một ảnh số dưới dạng ma trận.
- Hiển thị: tất cả các dạng xử lý trên ma trận để ngó ra là tối ưu.
- Phân tích: gồm tất cả các bước xử lý được dùng cho quá trình đo lường định lượng. Những bước này yêu cầu trước hết là hiểu biết bản chất và nội dung ảnh, do đó, các tiến trình phân tích ảnh rất rõ ràng và các thuật toán phát triển có thể được chuyển đổi nhanh chóng sang một môi trường ứng dụng khác một cách trực tiếp.

- Quản lý: các kỹ thuật cung cấp khả năng lưu trữ, truyền thông, phục hồi và truy nhập hết sức hiệu quả. Trong y khoa, việc lấy thông tin hình ảnh từ xa cũng được xem là một phần của lĩnh vực quản lý.

Ngược với phân tích ảnh, thường được nghĩ là xử lý ảnh nâng cao, xử lý ảnh cấp thấp thường đề cập đến các kỹ thuật thông thường hay tự động, nghĩa là hoàn toàn có thể nhận biết mà không cần trang bị nhiều kiến thức chuyên sâu về nội dung ảnh. Các thuật toán dạng này có ảnh hưởng không đáng kể đến nội dung của ảnh. Ví dụ, sự mở rộng lược đồ của các tấm phim chụp X-quang giúp tăng cường độ tương phản tương tự như với các tấm ảnh đời thường. Do đó, các phương pháp xử lý ảnh thường có giá trị như các chương trình dành cho tăng cường ảnh.

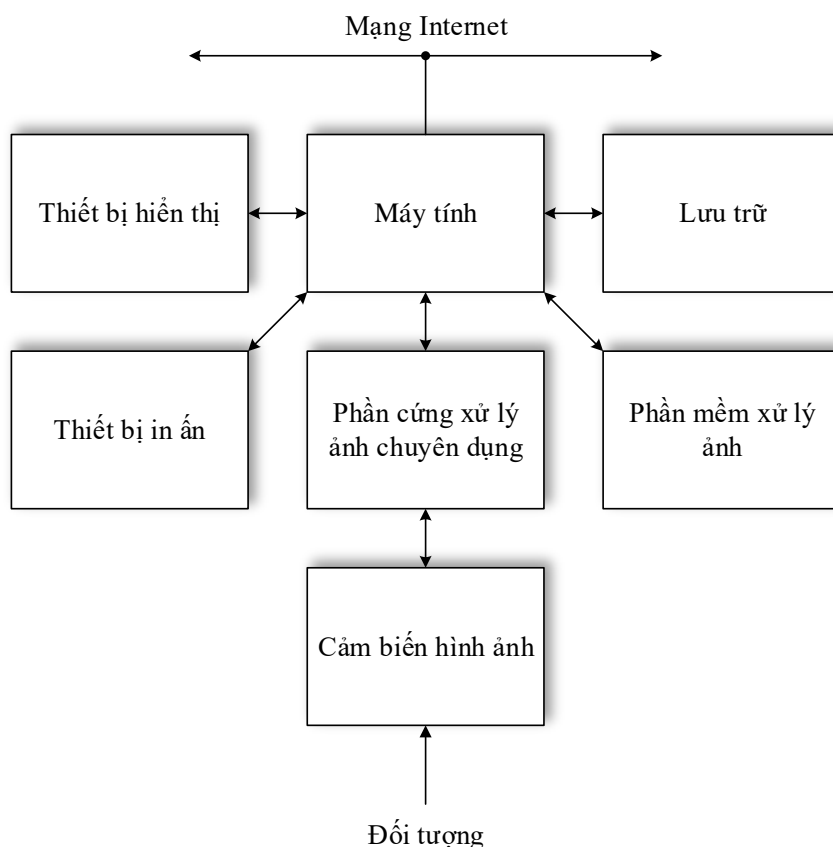


Hình 1.1. Các lĩnh vực nghiên cứu chính trong xử lý ảnh số

Các thành phần trong một hệ thống xử lý ảnh

Thông thường, một hệ thống xử lý ảnh có thể được xem như một hệ thống đa năng gồm các thành phần cơ bản như: cảm biến hình ảnh, phần cứng xử lý ảnh chuyên dụng, máy phần mềm xử lý, bộ nhớ, màn hình hiển thị, in ấn và mạng như hình 1.2.

- *Cảm biến*: hai yếu tố được yêu cầu để tạo lập một tấm ảnh số: đầu tiên là thiết bị vật lý nhạy với năng lượng bức xạ bởi vật thể mà chúng ta mong muốn ghi lại, thứ hai là bộ số hóa, hay còn gọi là thiết bị chuyển đổi ngõ ra của thiết bị cảm biến thành dạng số. Ví dụ, trong các camera số, các cảm biến xuất ngõ ra dạng tín hiệu điện tương ứng với cường độ sáng. Bộ số hóa sẽ chuyển đổi các ngõ ra này sang dạng dữ liệu số.



Hình 1.2. Các thành phần cơ bản trong hệ thống xử lý ảnh đa dụng

- *Phần cứng xử lý ảnh chuyên dụng* thường bao gồm các bộ số hóa đã đề cập và phần cứng xử lý các thông số ban đầu khác, chẳng hạn như đơn vị số học (ALU) – chuyên thực hiện các phép toán logic song song với nhiều ngõ vào. Một ví dụ cho thấy cách ALU được dùng để tính trung bình các ảnh nhằm mục đích triệt nhiễu. Dạng phần cứng này cũng thường được gọi là hệ thống phụ trợ đầu cuối, và đặc trưng phân biệt là tốc độ. Hay nói cách khác, các đơn vị thực hiện các chức năng yêu cầu bằng thông dữ liệu lớn (số hóa và lấy trung bình các ảnh trong video thường là 30 khung hình/giây) mà các máy tính dùng cho chức năng này không thể điều khiển bằng tay như thông thường.

- *Máy tính* trong hệ thống xử lý ảnh là các máy tính đa chức năng và chúng có thể là máy tính cá nhân hay thậm chí là siêu máy tính. Trong các ứng dụng cụ thể, các máy tính dân dụng có thể được dùng để đạt được hiệu năng cần thiết, nhưng điều chúng ta quan tâm ở đây là các hệ thống xử lý ảnh đa dụng. Trong những hệ thống này, phần lớn các thiết bị dạng máy tính cá nhân đều có thể phù hợp cho các nhiệm vụ xử lý ảnh ngoại tuyến (off-line).

- *Phần mềm* dành cho xử lý ảnh bao gồm các phần chuyên thực hiện các nhiệm vụ đặc trưng. Các gói này cũng bao gồm các đoạn mã dựng sẵn nhằm giúp cho người dùng có thể tạo ra các module riêng. Nhiều gói phần mềm phức tạp cho phép tích hợp những module này và các câu lệnh trong các phần mềm đa dụng từ ít nhất một ngôn ngữ máy tính.

- *Khả năng lưu trữ* là vấn đề cần quan tâm trong các ứng dụng xử lý ảnh. Một ảnh với kích thước 1024×1024 điểm ảnh, mà trong đó, mỗi điểm ảnh yêu cầu mã hóa 8 bit, thì có thể thấy dung lượng cần thiết cho việc lưu trữ là một megabytes trong trường hợp ảnh chưa được nén. Vì thế, với hàng triệu tấm ảnh thì việc cung cấp không gian lưu trữ trở thành một thách thức lớn. Các kiểu lưu trữ thường rơi vào một trong ba dạng cơ bản sau: lưu trữ tạm thời trong thời gian xử lý, lưu trữ trực tuyến với khả năng trích xuất nhanh chóng và lưu trữ dùng cơ sở dữ liệu.

- *Thiết bị hiển thị* thường được dùng hiện nay là màn hình tivi (dạng phẳng). Các màn hình được điều khiển bởi các ngõ ra hình ảnh và bo mạch hiển thị đồ họa mà xét về cơ bản thì chúng là một phần trong hệ thống máy tính. Trong các trường hợp khác, thiết bị hiển thị còn có khả năng hiển thị dạng lập thể theo không gian ba chiều.

- *Các thiết bị in ấn* dành cho việc ghi lại các hình ảnh bao gồm máy in laser, máy chụp dạng phim, các thiết bị cảm biến nhiệt, máy in phun, hay các thiết bị số như đĩa quang hay đĩa nén. Trong tất cả các dạng kể trên, phim cung cấp độ phân giải cao nhất, nhưng giấy được xem như môi trường thông dụng nhất. Để hiển thị, hình ảnh được đưa lên các tấm phim trong suốt hay trong môi trường số hóa nếu sử dụng các máy chiếu dạng phim.

- *Mạng* gần như là chức năng mặc định trong bất kỳ hệ thống máy tính hiện nay. Do một lượng lớn dữ liệu thuộc các ứng dụng xử lý ảnh, vấn đề cần xem xét ở đây chính là băng thông. Với một mạng dành riêng, nó không thật sự là vấn đề, nhưng truyền thông với các trang điều khiển qua Internet thì không phải lúc nào cũng hiệu quả. May mắn thay, trường hợp này đang được nâng cấp bằng cáp quang và các công nghệ băng rộng khác.

1.2. XỬ LÝ ẢNH SỐ DÙNG MATLAB

MATLAB (MATrix LABoratory), một ngôn ngữ mở, từ lâu đã trở thành ngôn ngữ lập trình dành riêng cho các công việc yêu cầu tính toán nhiều với khả năng tính toán, hiển thị, lập trình trong môi trường rất dễ sử dụng. Có thể xét một số chức năng chính của MATLAB như:

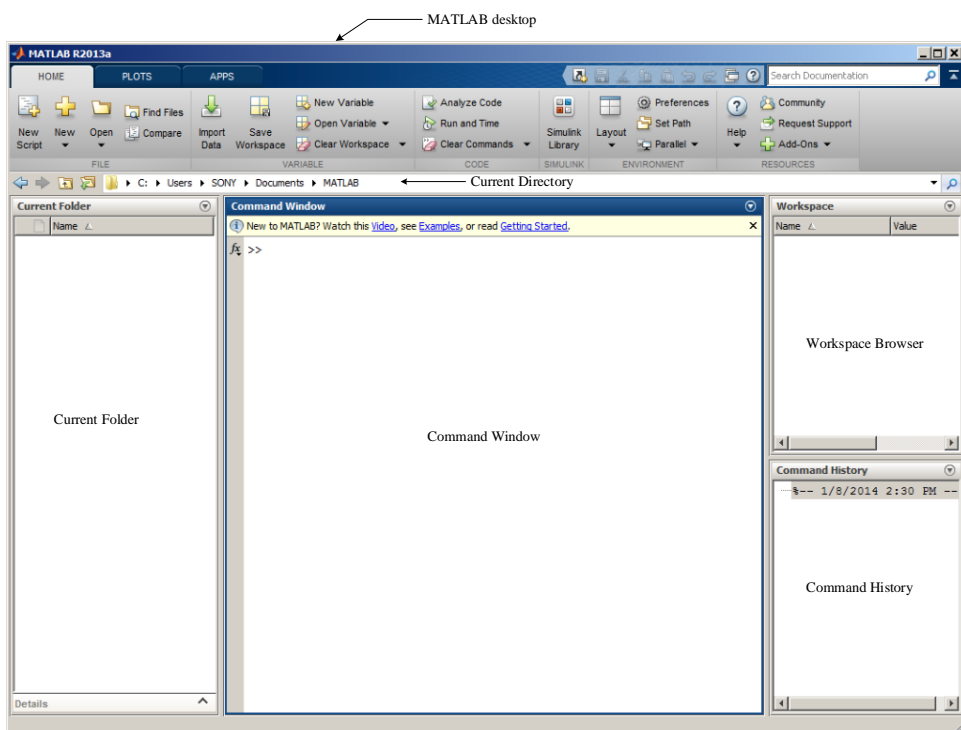
- Toán học và các phép tính
- Phát triển thuật toán
- Thu thập dữ liệu
- Mô hình hóa, mô phỏng và tạo bản mẫu
- Phân tích dữ liệu, khảo sát và hiển thị
- Đồ họa khoa học và kỹ thuật
- Phát triển ứng dụng, bao gồm các giao diện người dùng

MATLAB là một hệ thống tương tác với các thành phần dữ liệu cơ bản là các mảng không yêu cầu phải định kích thước. Điều này cho phép trình bày các giải pháp dành cho nhiều vấn đề tính toán trong kỹ thuật, đặc biệt là bao gồm việc biểu diễn ma trận, thường thì phải chia nhỏ thời gian thực hiện đối với các chương trình không hỗ trợ tương tác như C hay Fortran.

Trong môi trường đại học, MATLAB được xem như công cụ tính toán tiêu chuẩn cho các khóa học cơ bản cũng như nâng cao về toán học, kỹ thuật và khoa học. Hơn nữa, MATLAB còn là công cụ tính toán dành cho các nghiên cứu, phân tích và phát triển. MATLAB thường được bổ sung bằng các hộp công cụ (toolbox) nhằm giải quyết các vấn đề theo hướng ứng dụng, trong đó có Image Processing Toolbox. Đây là tập các hàm MATLAB (hay còn gọi là file M) được biên soạn dành cho việc giải quyết từng vấn đề cụ thể liên quan đến xử lý ảnh. Các toolbox khác đôi khi được dùng hỗ trợ, chẳng hạn như Signal Processing, Neural Network, Fuzzy Logic hay Wavelet Toolbox.

1.2.1. Môi trường làm việc trong MATLAB

Cửa sổ làm việc chính trong MATLAB luôn là MATLAB desktop như hình 1.3, với các cửa sổ nhỏ bao gồm: the Command Window, the Workspace Browser, the Current Directory Window, the Command History, the Current Folder và nhiều cửa sổ hiển thị hình ảnh là Figure Windows (chỉ cho phép hiển thị hình ảnh và đồ thị).



Hình 1.3. Cửa sổ làm việc trong MATLAB và các thành phần cơ bản

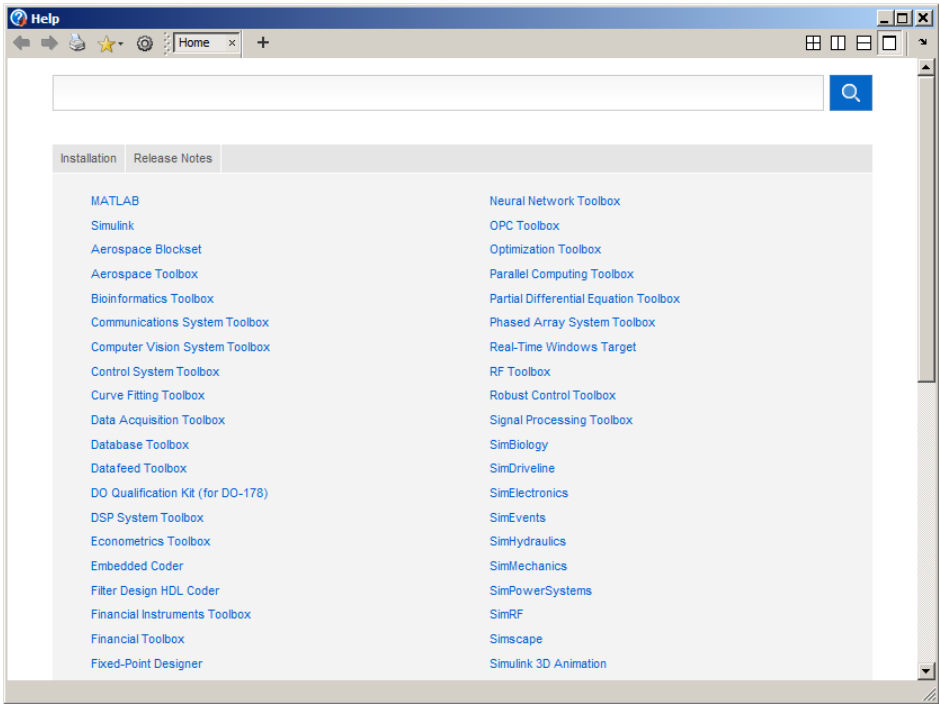
1.2.2. MATLAB Editor

MATLAB Editor bao gồm phần biên soạn dưới dạng text để tạo M-file và bộ debugger – gỡ rối dạng đồ họa. Phần biên soạn này có thể xuất hiện trong cửa sổ của chính nó hay dưới dạng một cửa sổ con bên trong desktop. Với các M-file thường được đánh dấu bởi phần đuôi .m, chẳng hạn như baitap.m. Cửa sổ soạn thảo MATLAB có nhiều menu với các chức năng khác nhau như lưu, xem trước, hay gỡ rối file. Do nó thực hiện một vài kiểm tra đơn giản và thường dùng các màu khác nhau giúp phân biệt các thành phần của đoạn mã nên bộ biên soạn này được đề xuất như là công cụ cho tạo mới và chỉnh sửa các hàm trong MATLAB. Để mở nó, gõ lệnh `edit` ngay tại cửa sổ Command Window. Tương tự, có thể gõ `edit filename` để mở tập tin đã có sẵn bên cửa sổ Current Directory.

1.2.3. Trợ giúp trong MATLAB

Phương pháp truyền thống để truy cập vào phần Help của MATLAB mà với từng cửa sổ riêng lẻ hoặc bằng cách nhấp vào biểu tượng dấu chấm hỏi (?) trên thanh công cụ hoặc gõ `helpbrowser` ngay bên trong cửa sổ Command Window. Một cách đơn giản và nhanh chóng khác là nhấn phím F1 khi ở bất kỳ cửa sổ nào của MATLAB. Phần Help gồm hai phần nhỏ:

ô dành cho tìm kiếm thông tin và ô dành hiển thị thông tin tìm được như hình 1.4.



Hình 1.4. Giao diện cửa sổ Help của MATLAB

Bên trong giao diện cửa sổ Help, người dùng có thể tùy chọn việc tìm kiếm theo hai cách: (1) có thể gõ vào nội dung cần tham khảo ô tìm kiếm, (2) tìm theo từng Toolbox với các chủ đề biết trước (thường được sử dụng như là một phương pháp tự học).

1.2.4. Thực hành xử lý ảnh cơ bản với MATLAB

Ví dụ 1.1: Đọc, hiển thị và lấy thông tin ảnh.

```
f=imread('peppers.png');  
Info_f=imfinfo('peppers.png');
```

Ngõ ra của hàm `imfinfo` cho phép truy vấn thông tin của ảnh. Với ảnh `peppers.png`, kết quả hiển thị tại cửa sổ Command Window:

```
Info_I =  
  
    Filename: [1x65 char]  
    FileModDate: [1x20 char]  
    FileSize: 287677  
    Format: 'png'  
    FormatVersion: []
```



```

        Width: 512
        Height: 384
        BitDepth: 24
        ColorType: [1x9 char]
FormatSignature: [1x8 double]
        Colormap: []
        Histogram: []
InterlaceType: 'none'
        Transparency: 'none'
SimpleTransparencyData: []
        BackgroundColor: []
        RenderingIntent: []
        Chromaticities: []
        Gamma: []
        XResolution: []
        YResolution: []
        ResolutionUnit: []
        XOffset: []
        YOffset: []
        OffsetUnit: []
SignificantBits: []
        ImageModTime: [1x26 char]
        Title: []
        Author: []
        Description: [1x13 char]
        Copyright: [1x29 char]
        CreationTime: []
        Software: []
        Disclaimer: []
        Warning: []
        Source: []
        Comment: []
        OtherText: []

```

Hàm `imread` cho phép đọc được nhiều định dạng ảnh khác nhau trong máy tính (tham khảo bảng 1.1) với tên file phải bao gồm cả phần mở rộng được đặt trong dấu nháy đơn ('). Ảnh sau khi đọc sẽ được lưu dưới dạng ma trận có tên `f` (người dùng hoàn toàn có thể dùng biến khác nhau

miễn không trùng lặp với những biến đã có trước đó). Hàm `imread` có thể đọc một file ảnh với đường dẫn cụ thể.

```
imread('D:\Baitap\baitap.png');
```

Bảng 1.1. Một số định dạng ảnh được hỗ trợ bởi hàm `imread`.

Tên định dạng	Mô tả	Phần mở rộng
TIFF	Tagged Image File Format	.tif,.tiff
JPEG	Joint Photographic Experts Group	.jpg,.jpeg
JPEG 2000	Joint Photographic Experts Group	.jpg,.jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd
HDF4	Hierarchical Data Format	.hdf
ICO	Icon File	.ico
PCX	Windows Paintbrush	.pcx
PGM	Portable Graymap	.pgm
PPM	Portable Pixmap	.ppm
RAS	Sun Raster	.ras

❖ Lưu ý:

- Nên sử dụng hàm `clear all` để xóa các tham số trong cửa sổ Workspace khi bắt đầu một chương trình mới để tránh việc nhầm lẫn với tham số của chương trình trước đó.
- Kết quả của hàm `imfinfo` là một cấu trúc có nhiều trường, vì thế có thể truy cập bất kỳ trường nào bằng cách trích xuất cụ thể. Ví dụ:

```
Info_f.FileName
Info_f.FileSize
Info_f.Format
...
```

- Truy cập nhanh kích thước của ảnh hay của ma trận, có thể dùng hàm `size` hoặc lấy thông tin của ma trận bằng hàm `whos`

```
[M N]=size(f);
whos f;
```

Ví dụ 1.2: *Hiển thị ảnh trong môi trường MATLAB*

```
imshow(f);
```

❖ Lưu ý:

- Ảnh sẽ được trình bày dưới dạng một hình (Figure) trong MATLAB. Để hiển thị đầy đủ các ảnh liên tiếp thì cần sử dụng hàm `figure` trước khi dùng `imshow` để đảm bảo hiển thị đầy đủ các ảnh mong muốn. Cấu trúc đơn giản dành cho việc hiển thị nhiều ảnh với nhiều cửa sổ hình khác nhau:

```
figure; imshow(f);
```

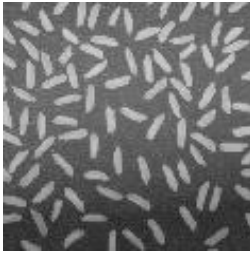


Hình 1.5. Ảnh *'peppers.png'* được hiển thị bằng cách dùng hàm `imshow`

- Có thể hiển thị nhiều ảnh trong cùng một hình với hàm `subplot` dùng chỉ định vị trí thứ tự cửa sổ hiển thị.

```
A=imread('rice.png');  
B=imread('cameraman.tif');  
C=imread('trees.tif');  
figure;  
subplot(1,3,1)  
imshow(A)  
subplot(1,3,2)
```

```
imshow(B)
subplot(1,3,3)
imshow(C)
```



Rice



Cameraman



Trees

Hình 1.6. *Hiển thị nhiều ảnh trên cùng một hình*

Ví dụ 1.3: *Lưu ma trận ảnh thành file đồ họa trong Window*

```
clear all;
f=imread('peppers.png');
f_gs=rgb2gray(f);
imwrite(f_gs,'pepper_gray.png','png');
```

❖ Lưu ý:

- Khi dùng hàm `imwrite` cần chỉ rõ tên file và định dạng file hình ảnh. File ảnh sau khi lưu dưới dạng file đồ họa sẽ lưu trong thư mục hiện hành (xem bên cửa sổ Current Window). Hoàn toàn có thể lưu với đường dẫn chỉ rõ như khi dùng hàm `imread`.
- Hàm `imwrite` cho phép lưu ảnh dưới nhiều định dạng khác nhau như tif, gif, jpg, bmp...

```
imwrite(f_gs,'pepper_gray.tif','tif');
imwrite(f_gs,'pepper_gray.png','png');
imwrite(f_gs,'pepper_gray.jpg','jpg','Quality',50);
```

Khi lưu ảnh dưới định dạng JPEG thì cần chỉ rõ hệ số chất lượng của ảnh ngõ ra. Hệ số này tương ứng với tỷ lệ nén ảnh với kỹ thuật nén có tổn hao JPEG.



(a)



(b)



(c)



(d)

Hình 1.7. Kết quả lưu ảnh dưới định dạng JPEG theo các tỷ số nén khác nhau: (a) 80%; (b) 60%; (c) 40% và (d) 20%

Hoàn toàn có thể tính được tỷ lệ nén dựa trên tỷ số dung lượng lưu trữ của ảnh gốc trước và sau khi nén (thông thường đây là tỷ số giữa kích thước ảnh lưu trên bộ nhớ máy tính). Trong MATLAB, tỷ số này được tính dựa trên các thông số của ảnh. Ví dụ, với ảnh trong hình 1.7(d) sau khi truy xuất thông tin dùng hàm `imfinfo`:

```
Infor =  
  
    Filename: [1x51 char]  
    FileModDate: [1x20 char]  
    FileSize: 7237  
    Format: 'jpg'  
    FormatVersion: ''  
    Width: 512  
    Height: 384  
    BitDepth: 8
```

```

        ColorType: 'grayscale'
FormatSignature: ''
NumberOfSamples: 1
        CodingMethod: 'Huffman'
        CodingProcess: 'Sequential'
        Comment: {}

```

Trong đó, *FileSize* là kích thước của ảnh dưới dạng byte. Kích thước của ảnh gốc khi chưa được nén được tính theo công thức:

$$I_{bytes} = Width \times Height \times BitDepth / 8 \quad (1.1)$$

Theo đó, tỷ lệ nén được tính như sau:

$$CompressionRatio = I_{bytes} / FileSize \quad (1.2)$$

Tỷ số nén dành cho ảnh trong hình 1.7(d) được tính cụ thể như sau:

$$CompressionRatio = (512 \times 512 \times 8 / 8) / 11831 \approx 22.15 \quad (1.3)$$

Tùy vào từng ứng dụng cụ thể mà yêu cầu chất lượng của ảnh sau khi nén vẫn phải được đảm bảo. Ngoài việc giảm được dung lượng cần thiết trong quá trình lưu trữ thì nén cũng giúp việc truyền tải thông tin dưới dạng hình ảnh nhanh hơn nhiều so với khi chưa được nén. Người đọc cần tham khảo thêm các thuộc tính khi dùng hàm `imwrite` với từng định dạng khác nhau. Một ví dụ đối với việc lưu ảnh dưới dạng JPEG luôn có thể tùy chọn một số thuộc tính như trong bảng 1.2.

Bảng 1.2. Các thuộc tính có thể tùy chỉnh khi dùng hàm `imwrite` ứng với định dạng JPEG

Thông số	Giá trị	Mặc định
'Bitdepth'	Đại lượng vô hướng chỉ ra độ sâu số hay có thể hiểu là lượng bit dành mã hóa 1 pixel. Với ảnh trắng đen thì có thể là 8, 12 hay 16; còn với ảnh màu thì chỉ có thể là 8 hoặc 12.	8 dành cho ảnh trắng đen 8 dành cho ảnh màu
'Comment'	Là một ma trận các ký tự. Hàm <code>imwrite</code> ghi từng hàng của ngõ vào như là một chú thích trong file JPEG	Trống
'Mode'	Chỉ rõ dạng nén được sử dụng: 'lossy' hay 'lossless'	'lossy'

'Quality'	Là một số nguyên từ 0 đến 100, 75
	thông số này càng lớn nghĩa là ảnh
	ngõ ra chất lượng càng cao (ảnh ít bị
	suy hao do ảnh hưởng của nén) và
	kích thước ảnh cũng lớn theo.

Ví dụ 1.4: Thay đổi kích thước ảnh

```
clear all;  
f=imread('peppers.png');  
f_gs=rgb2gray(f);  
f_256=imresize(f,0.5);  
f_128=imresize(f_gs,[128 128]);
```

Hàm `imresize` cho phép thay đổi kích thước của ảnh bằng cách chỉ rõ kích thước ảnh ngõ ra [width height] (trong ví dụ trên là [128 128]) hay hệ số tỷ lệ (0.5). Hàm `imresize` dùng được với ảnh ngõ vào là ảnh đen trắng và ảnh màu.

Với ảnh đen trắng:

```
size(f_128)
```

```
ans =
```

```
128 128
```

Với ảnh màu:

```
size(f_256)
```

```
ans =
```

```
256 256 3
```

Mặc dù chúng ta thường làm việc với các biến kiểu `integer` nhưng, trong MATLAB, giá trị các điểm ảnh không bị giới hạn bởi chỉ với kiểu `integer`. Bảng 1.3 trình bày các kiểu dữ liệu khác nhau được hỗ trợ bởi MATLAB và Image ProcessingToolbox.

Bảng 1.3. Các kiểu dữ liệu được hỗ trợ trong MATLAB

Tên	Mô tả
double	Các giá trị có dấu chấm động có độ chính xác kép nằm trong khoảng xấp xỉ -10^{308} đến 10^{308} (dùng 8 byte để biểu diễn cho một phần tử)
uint8	Số nguyên không dấu trong khoảng [0 255] (1 byte cho một phần tử)
uint16	Số nguyên không dấu trong khoảng [0 65535] (2 byte cho một phần tử)
uint32	Số nguyên không dấu trong khoảng [0 4294967295] (4 byte cho một phần tử)

int8	Số nguyên có dấu trong khoảng [-128 127] (1 byte cho một phần tử)
int16	Số nguyên có dấu trong khoảng [-32768 32767] (2 byte cho một phần tử)
int32	Số nguyên có dấu trong khoảng [-2147483648 2147483648] (4 byte cho một phần tử)
single	Các giá trị có dấu chấm động với độ chính xác đơn nằm trong khoảng xấp xỉ 10^{-38} đến 10^{38} (dùng 4 bytes để biểu diễn cho một phần tử)
char	Kiểu ký tự (2 byte cho một phần tử)
logical	Giá trị logic là 0 hay 1 (1 byte cho một phần tử)

Như vậy, ta có thể thao tác trên một hay nhiều ảnh với sự hỗ trợ rất thuận tiện của phần mềm MATLAB. Các ví dụ vừa trình bày nêu những hướng cơ bản để giúp người đọc tiếp cận vào việc xử lý ảnh dùng MATLAB. Các chương về sau sẽ phân tích các phương pháp trong xử lý ảnh.

BÀI TẬP CHƯƠNG 1

- 1.1. Trình bày tên các hàm hỗ trợ xử lý ảnh trong Image Processing Toolbox.
- 1.2. Mở ảnh ‘peppers.png’ và lưu ảnh này thành các định dạng khác nhau. So sánh kích thước file các định dạng này.
- 1.3. Mở ảnh ‘cameraman.tif’ và thay đổi kích thước ảnh này. Hiện thị lại ảnh đã bị thay đổi kích thước và nhận xét.

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

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.nxbdhqgthcm.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

Mã số ISBN: 978-604-73-2582-5

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.