



LÊ MỸ HÀ

GIÁO TRÌNH
**THỊ GIÁC MÁY TÍNH
VÀ ỨNG DỤNG**



NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

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



TS. LÊ MỸ HÀ

GIÁO TRÌNH THỊ GIÁC MÁY TÍNH VÀ ỨNG DỤNG

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA
THÀNH PHỐ HỒ CHÍ MINH - 2019

Giáo trình
**THỊ GIÁC MÁY TÍNH
VÀ ỨNG DỤNG**

TS. LÊ MỸ HÀ

NHÀ XUẤT BẢN

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

Khu phố 6, Phường Linh Trung, Quận Thủ Đức, TP Hồ Chí Minh
Đường C, số 10-12 Đinh Tiên Hoàng, Phường Bến Nghé,
Quận 1, TP Hồ Chí Minh

ĐT: 028 6272 6361 – 028 6272 6390

E-mail: vnuhp@vnuhcm.edu.vn

TRUNG TÂM SÁCH ĐẠI HỌC

Đường C, số 10-12 Đinh Tiên Hoàng, Phường Bến Nghé,
Quận 1, TP Hồ Chí Minh

ĐT: 028 6681 7058 - 028 6272 6390 - 028 6272 6351

Website: <https://trxbvnuhcm.edu.vn>

Chịu trách nhiệm xuất bản

ĐỖ VĂN BIẾN

Xuất bản năm 2019

Chịu trách nhiệm nội dung

ĐỖ VĂN BIẾN

Tổ chức bản thảo và chịu trách nhiệm về tác quyền

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

Website: <http://hcmute.edu.vn>

Biên tập

LÊ THỊ MINH HUỆ

Sửa bản in

THANH HÀ

Trình bày bìa

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

Website: <http://hcmute.edu.vn>

Số lượng 300 cuốn,
Khổ 16 x 24 cm,
ĐKKHXB số: 1989-2019/CXBIPH/
05-103/ĐHQGTPHCM,
Quyết định XB số 106/QĐ-ĐHQGTPHCM
của NXB ĐHQG-HCM
cấp ngày 02/7/2019.
In tại: Công ty TNHH In &
bao bì Hưng Phú
Đ/c: 162A/1 – KP1A – P. An Phú –
TX. Thuận An – Bình Dương
Nộp lưu chiểu: Quý III/2019

ISBN: 978-604-73-7083-2



ISBN: 978 – 604 – 73 – 7083 – 2

Giáo trình

THỊ GIÁC MÁY TÍNH VÀ ỨNG DỤNG

TS. LÊ MỸ HÀ

Bản tiếng Việt ©, TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM, NXB ĐHQG-HCM và CÁC TÁC GIẢ.

Bản quyền tác phẩm đã được bảo hộ bởi Luật Xuất bản và Luật Sở hữu trí tuệ Việt Nam. Nghiêm cấm mọi hình thức xuất bản, sao chép, phát tán nội dung khi chưa có sự đồng ý của Trường Đại Học Sư Phạm Kỹ Thuật TP. HCM và Tác Giả.

ĐỂ CÓ SÁCH HAY, CẦN CHUNG TAY BẢO VỆ TÁC QUYỀN!

MỤC LỤC

LỜI NÓI ĐẦU	9
Chương 1: XỬ LÝ ẢNH CƠ BẢN VÀ NGÔN NGỮ LẬP TRÌNH PYTHON	13
1.1. PIL - THƯ VIỆN HÌNH ẢNH TRONG PYTHON	13
1.1.1. Chuyển đổi hình ảnh sang định dạng khác.....	14
1.1.2. Tạo hình thu nhỏ.....	15
1.1.3. Sao chép và dán các vùng.....	15
1.1.4. Thay đổi kích thước và xoay	15
1.2. MATPLOTLIB	15
1.2.1. Vẽ hình ảnh, điểm và đường	16
1.2.2. Đường biên ảnh và lược đồ	17
1.2.3. Chú thích tương tác	19
1.3. NUMPY	19
1.3.1. Thể hiện hình ảnh dưới dạng mảng.....	20
1.3.2. Biến đổi cấp độ xám.....	21
1.3.3. Cân bằng lược đồ ảnh.....	23
1.3.4. Phép tính trung bình pixel ảnh	25
1.3.5. Phân tích thành phần chính (PCA) của hình ảnh.....	26
1.3.6. Sử dụng Module Pickle	29
1.4. SCIPY.....	30
1.4.1. Làm mờ ảnh.....	30
1.4.2. Tính đạo hàm của ảnh (Image derivatives)	32
1.4.3. Các module SciPy hữu ích khác.....	36
1.5. VÍ DỤ NÂNG CAO: GIẢM NHIỀU HÌNH ẢNH.....	37
1.6. BÀI TẬP	41
Chương 2: MÔ TẢ ĐẶC TRƯNG HÌNH ẢNH	42
2.1. XÁC ĐỊNH ĐẶC TRƯNG GÓC HARRIS.....	42
2.2. ĐẶC TRƯNG BẤT BIẾN VỚI SỰ THAY ĐỔI TỶ LỆ ẢNH (SCALE-INVARIANT FEATURE TRANSFORM- SIFT).....	49
2.2.1. Lập trình xác định đặc trưng SIFT trong ảnh	53
2.2.2. Đối sánh các mô tả của điểm đặc trưng.....	54
2.3. BÀI TẬP	60

Chương 3: PHÉP ẢNH XẠ HÌNH ẢNH	61
3.1. HOMOGRAPHIES.....	63
3.1.1. Thuật toán biến đổi tuyến tính trực tiếp.....	64
3.1.2. Biến đổi Affine.....	69
3.2. Nấn ảnh.....	66
3.2.1. Đặt một ảnh trong một ảnh khác.....	67
3.2.2. Nấn ảnh Affine theo từng cặp.....	70
3.2.3. Hiển thị hình ảnh.....	74
3.3. Tạo ảnh toàn cảnh.....	79
3.3.1. Thuật toán RANSAC.....	79
3.3.2. Ước tính hệ tọa độ đồng nhất bền vững.....	81
3.3.3. Ghép các hình ảnh lại với nhau.....	85
3.4. BÀI TẬP.....	88
Chương 4: MÔ HÌNH MÁY ẢNH VÀ THỰC TẾ ẢO	89
4.1. MÔ HÌNH MÁY ẢNH LỖ KIM (PIN- HOLE CAMERA).....	89
4.1.1. Điểm chiếu 3D.....	91
4.1.2. Hệ số ma trận camera.....	93
4.1.3. Tính toán điểm trung tâm camera.....	94
4.2. HIỆU CHỈNH MÁY ẢNH.....	94
4.3. ƯỚC LƯỢNG VỊ TRÍ CAMERA TỪ MẶT PHẪNG VÀ ĐIỂM MỐC.....	96
4.4. THỰC TẾ ẢO.....	100
4.4.1. PyGame và PyOpenGL.....	101
4.4.2. Từ ma trận camera đến định dạng OpenGL.....	102
4.4.3. Đặt các đối tượng ảo trong ảnh.....	104
4.4.4. Kết nối tất cả ảnh.....	105
4.4.5. Tải mô hình.....	107
4.5. BÀI TẬP.....	109
Chương 5: HÌNH HỌC ĐA GÓC NHÌN (MULTIPLE VIEW GEOMETRY)	110
5.1. EPIPOLAR GEOMETRY.....	110
5.1.1. Một tập dữ liệu mẫu.....	112
5.1.2. Vẽ dữ liệu 3D với Matplotlib.....	114
5.1.3. Epipole và các đường thẳng cpipole.....	117

5.2. TÍNH TOÁN VỚI MÁY ẢNH VÀ CẤU TRÚC 3D.....	119
5.2.1. Tính toán ma trận camera từ các điểm 3D.....	121
5.2.2. Tính toán ma trận camera từ ma trận cơ bản.....	123
5.2.3. Trường hợp không được hiệu chỉnh - tái cấu trúc hình chiếu.....	123
5.2.4. Trường hợp được hiệu chỉnh - tái cấu trúc chính xác mô hình.....	124
5.3. TÁI CẤU TRÚC NHIỀU GÓC NHÌN.....	125
5.3.1. Ước lượng ma trận cơ bản bền vững.....	125
5.3.2. Ví dụ tái tạo 3D.....	128
5.3.3. Phần mở rộng và nhiều hơn hai góc nhìn.....	131
5.4. ẢNH STEREO.....	132
5.5. BÀI TẬP.....	138
Chương 6: PHÂN CỤM ẢNH	139
6.1. THUẬT TOÁN PHÂN CỤM K-MEANS.....	139
6.1.1. Thư viện phân cụm SciPy.....	140
6.1.2. Phân cụm ảnh.....	141
6.1.3. Biểu diễn trực quan ảnh các thành phần chính.....	143
6.1.4. Phân cụm các Pixels.....	154
6.2. PHÂN CỤM THEO THỨ BẬC.....	146
6.3. Phân cụm ảnh.....	150
6.4. PHÂN CỤM PHỒ.....	155
6.5. BÀI TẬP.....	159
Chương 7: TÌM KIẾM ẢNH	160
7.1. TRUY XUẤT ẢNH DỰA TRÊN NỘI DUNG.....	160
7.2. TỪ NGỮ TRỰC QUAN (VISUAL WORDS).....	161
7.3. LẬP CHỈ MỤC ẢNH.....	165
7.3.1. Thiết lập cơ sở dữ liệu.....	165
7.3.2. Thêm ảnh.....	166
7.4. TÌM KIẾM CƠ SỞ DỮ LIỆU CHO ẢNH.....	168
7.4.1. Tìm kiếm bằng ảnh.....	170
7.4.2. Kiểm chuẩn và biểu diễn kết quả.....	171
7.5. XẾP HẠNG KẾT QUẢ BẰNG HÌNH HỌC.....	173
7.6. XÂY DỰNG CÁC DEMO VÀ ỨNG DỤNG WEB.....	176
7.7. BÀI TẬP.....	179

Chương 8: PHÂN LOẠI NỘI DUNG HÌNH ẢNH	181
8.1. K-NEAREST NEIGHBORS.....	181
8.1.1. Một ví dụ 2D đơn giản	182
8.1.2. Đặc trưng hình ảnh dùng dense SIFT (SIFT dày đặc).....	185
8.1.3. Phân loại hình ảnh - nhận dạng cử chỉ tay.....	187
8.2. BỘ PHÂN LOẠI BAYES.....	190
8.3. SUPPORT VECTOR MACHINES.....	195
8.3.1. Sử dụng LibSVM	196
8.3.2. Nhận dạng cử chỉ tay lại bằng phương pháp SVM	198
8.4. NHẬN DẠNG KÝ TỰ QUANG HỌC (OPTICAL CHARACTER RECOGNITION).....	200
8.4.1. Huấn luyện một bộ phân loại	200
8.4.2. Chọn đặc trưng	201
8.4.3. SVM nhiều lớp	201
8.4.4. Trích xuất các hàng và nhận dạng ký tự.....	202
8.4.5. Chỉnh sửa hình ảnh.....	205
8.5. BÀI TẬP	206
Chương 9: PHÂN ĐOẠN ẢNH	208
9.1. PHƯƠNG PHÁP CẮT ĐỒ THỊ (GRAPH CUTS).....	208
9.1.1. Đồ thị từ các hình ảnh	210
9.1.2. Phân đoạn với đầu vào của người dùng.....	215
9.2. PHÂN ĐOẠN BẰNG CÁCH SỬ DỤNG PHÂN CỤM.....	217
9.3. PHƯƠNG PHÁP BIẾN PHÂN.....	222
9.4. BÀI TẬP	224
Chương 10: GIỚI THIỆU THƯ VIỆN OpenCV	226
10.1. GIAO DIỆN PYTHON OPENCV	226
10.2. KHÁI NIỆM CƠ BẢN VỀ OPENCV	226
10.2.1. Đọc và viết hình ảnh.....	227
10.2.2. Không gian màu.....	227
10.2.3. Hiển thị hình ảnh và kết quả.....	228
10.3. XỬ LÝ VIDEO	231
10.3.1. Đầu vào video.....	231
10.3.2. Đọc video vào mảng NumPy.....	233
10.4. THEO VẾT (TRACKING)	233

10.4.1. Luồng quang học	233
10.4.2. Thuật toán Lucas-Kanade.....	236
10.4.3. Sử dụng generators	241
10.5. CÁC VÍ DỤ KHÁC.....	242
10.5.1. Vá lỗi ảnh (Inpainting)	243
10.5.2. Phát hiện đường thẳng với phép biến đổi Hough	244
10.6. BÀI TẬP.....	245
Chương 11: ỨNG DỤNG HÌNH HỌC ĐA GÓC NHÌN VÀO XÂY DỰNG MÔ HÌNH 3D KẾT HỢP PHÂN TÍCH NGỮ NGHĨA VÀ PHƯƠNG PHÁP TỐI ƯU LỖI	247
11.1. GIỚI THIỆU.....	247
11.2. MÔ TẢ THUẬT TOÁN.....	248
11.2.1. Phân tích ngữ nghĩa	249
11.2.2. Xây dựng lại cảnh quan trong trường hợp đặc biệt	253
11.2.3. Đối sánh theo phương pháp cặp ảnh stereo	259
11.3. THỰC NGHIỆM TRÊN TẬP DỮ LIỆU THỰC.....	260
11.4. KẾT LUẬN	263
Chương 12: ỨNG DỤNG HÌNH HỌC ĐA GÓC NHÌN TRONG ĐỊNH VỊ CHO MÁY ẢNH ĐA HƯỚNG	264
12.1. GIỚI THIỆU.....	264
12.2. MÔ TẢ THUẬT TOÁN.....	264
12.2.1. Ước tính các cặp chuyển động xoay.....	265
12.2.2. Biến đổi camera đa hướng trong mô hình cầu.....	266
12.2.3. Trích đặc trưng đường	267
12.2.4. Phát hiện điểm vô cực dựa trên phương pháp RANSAC.....	270
12.2.5. Tìm các điểm vô cực tương đồng dựa trên phương pháp ước lượng chuyển động xoay	271
12.3. TÍNH TOÁN GÓC XOAY TOÀN CỤC CỦA CAMERA....	273
12.4. SỰ PHỤC HỒI CHUYỂN ĐỘNG CỦA CAMERA	273
12.4.1. Điểm tương đồng trên gương cầu.....	273
12.4.2. Chuyển động của camera với ma trận xoay đã biết.....	273
12.4.3. Tối ưu hóa trong mô hình hình cầu	275
12.4.3. Tối ưu giảm lỗi sử dụng ngưỡng giữa	277
12.5. THỰC NGHIỆM.....	278

12.6. KẾT LUẬN	284
Chương 13: ỨNG DỤNG THỊ GIÁC MÁY TÍNH KẾT HỢP	
BIÊN GPS TRONG ĐỊNH VỊ ROBOT	281
13.1. GIỚI THIỆU.....	281
13.2. MÔ TẢ THUẬT TOÁN.....	284
13.3. HỆ THỐNG ĐO LƯỜNG BẰNG THỊ GIÁC DỰA TRÊN CAMERA ĐA HƯỚNG.....	286
13.3.1. Phân tích mô hình di chuyển	286
13.3.2. Hệ thống đo lường bằng hình ảnh	290
13.4. SỰ HIỆU CHỈNH SAI SỐ ĐO VỊ TRÍ CỦA HỆ THỐNG NHỜ VÀO THÔNG TIN DẪN ĐƯỜNG QUA VỆ TINH (GPS)....	293
13.4.1. Đánh giá giá trị đo từ hệ thống định vị toàn cầu	293
13.4.2. Hiệu chỉnh vị trí dựa trên hệ thống định vị toàn cầu	295
13.5. KẾT QUẢ THỰC NGHIỆM	295
13.6. KẾT LUẬN	298
Phụ lục A.....	300
Phụ lục B.....	306
TÀI LIỆU THAM KHẢO.....	310

LỜI NÓI ĐẦU

Thị giác máy tính có thể được hiểu một cách khái quát là làm thế nào lập trình một máy tính và sử dụng các thuật toán để hiểu những nội dung thu được từ dữ liệu hình ảnh tương tự như cách làm việc của não bộ và đôi mắt con người. Thị giác máy tính hiện nay thu hút rất nhiều sự quan tâm của các nhà khoa học trên thế giới cũng như tại Việt Nam. Cùng với sự phát triển mạnh mẽ phần cứng xử lý của máy tính và trí tuệ nhân tạo, thị giác máy tính ngày càng thể hiện vai trò then chốt trong hầu hết các lĩnh vực thuộc khoa học, công nghệ, kỹ thuật cũng như những ứng dụng thiết thực trong đời sống hiện nay. Một vài ứng dụng cụ thể có thể kể ra như truy vấn ảnh, phân loại ảnh, xây dựng mô hình ba chiều, điều hướng các Robot, phân tích hình ảnh y tế, v.v.

Cuốn sách này ra đời với mục đích giúp người học dễ dàng tiếp cận và thực hành thị giác máy tính. Cơ sở lý thuyết và thuật toán cơ bản được trình bày theo cách đơn giản nhất để sinh viên, nhà nghiên cứu và những đối tượng khác có cùng đam mê trong lĩnh vực này dễ dàng thực hành trên những ứng dụng cụ thể. Ngôn ngữ lập trình Python được lựa chọn sử dụng trong cuốn sách này, đi kèm với nhiều mô hình đã xây dựng sẵn để thực nghiệm.

Để tiếp cận nhanh những kiến thức đề cập trong cuốn sách này, người đọc cần tự trang bị thêm một số kiến thức nền tảng về lập trình cơ bản, cần biết cách sử dụng trình soạn thảo và chạy tập lệnh, cách cấu trúc chương trình cũng như các kiểu dữ liệu cơ bản. Làm quen với Python hoặc các ngôn ngữ khác như Ruby hoặc Matlab sẽ giúp ích rất nhiều cho việc thực hành. Ngoài ra, kiến thức nền tảng về toán học cũng cần phải được trang bị thêm ví dụ như hiểu biết về ma trận, vector, các hàm và khái niệm đạo hàm, phương pháp gradient để thực hiện các bài toán tối ưu.

Các chương trong cuốn sách trình bày những lĩnh vực quan trọng nhất của thị giác máy tính và các chương trình mẫu để thực hiện nhận dạng đối tượng, truy xuất hình ảnh dựa trên nội dung, tìm kiếm hình ảnh, nhận dạng ký tự, dòng quang học (optical flow), theo vết (tracking), tái tạo hình ảnh ba chiều, định vị camera, phân đoạn hình ảnh, khử nhiễu, phân nhóm hình ảnh và nhiều hơn nữa. Ở phần cuối quyển sách này, một số ứng dụng cụ thể của thị giác máy tính trong ước lượng vị trí cũng như xây dựng mô hình ba chiều trên tập dữ liệu lớn cũng được giới thiệu.

Sơ lược về các chương

Chương 1 giới thiệu các công cụ cơ bản để làm việc với hình ảnh và các mô hình Python được sử dụng trong sách. Chương này cũng bao gồm nhiều ví dụ cơ bản cần thiết cho các chương còn lại.

Chương 2 giải thích các phương pháp phát hiện các điểm đặc trưng bất biến và cách sử dụng chúng để tìm các điểm và vùng tương đồng giữa các hình ảnh.

Chương 3 trình bày các phép biến đổi cơ bản giữa hình ảnh và các phương pháp để tính toán. Một số ví dụ cụ thể từ đơn giản như nắn ảnh cho đến phức tạp hơn như tạo hình ảnh panorama.

Chương 4 giới thiệu cách viết phương trình mô hình máy ảnh, các phép chiếu từ không gian 3D đến điểm đặc trưng trong ảnh và ước lượng góc nhìn của máy ảnh.

Chương 5 giải thích cách làm việc với một số hình ảnh chụp cùng một cảnh quan, các nguyên tắc cơ bản của hình học đa góc nhìn và tính toán tái tạo mô hình ba chiều (3D) từ nhiều hình ảnh khác nhau.

Chương 6 giới thiệu một số phương pháp phân cụm và chỉ ra cách sử dụng chúng để nhóm và sắp xếp hình ảnh dựa trên sự tương đồng hoặc nội dung ảnh.

Chương 7 trình bày kỹ thuật truy xuất hình ảnh hiệu quả, có thể lưu trữ các biểu diễn ảnh và tìm kiếm hình ảnh dựa trên nội dung của chúng.

Chương 8 mô tả các thuật toán để phân loại nội dung hình ảnh và cách sử dụng chúng để nhận dạng các đối tượng trong hình ảnh.

Chương 9 giới thiệu các kỹ thuật khác nhau để phân chia một hình ảnh thành các vùng có ý nghĩa bằng cách sử dụng phân cụm, tương tác, hoặc mô hình ảnh.

Chương 10 trình bày cách sử dụng giao diện Python cho thư viện thị giác máy tính OpenCV phổ biến và cách làm việc với dữ liệu đầu vào là video.

Chương 11 trình bày ứng dụng của hình học đa góc nhìn kết hợp phân tích ngữ nghĩa và tối ưu lỗi để thực hiện xây dựng mô hình ba chiều trong không gian rộng lớn.

Chương 12 giới thiệu phương pháp tính toán vị trí của camera đa hướng trên tọa độ cầu.

Chương 13 trình bày ứng dụng định vị cho các thiết bị di chuyển trên vùng làm việc rộng lớn ngoài trời kết hợp thị giác máy tính và các loại cảm biến khác để hiệu chỉnh sai số vị trí cho ra kết quả định vị chính xác nhất.

Trong phụ lục A và B, người dùng sẽ được hướng dẫn cách cài đặt chương trình và các tập dữ liệu ảnh phổ biến trên thế giới để có thể tải về và tự làm thực nghiệm.

Trong quá trình biên soạn sẽ không tránh những thiếu sót, tác giả rất mong nhận được những ý kiến đóng góp từ bạn đọc thông qua hộp thư điện tử halm@hcmute.edu.vn để lần tái bản sau sách được tốt hơn.

Trân trọng

Lê Mỹ Hà

Chương 1

XỬ LÝ ẢNH CƠ BẢN VÀ NGÔN NGỮ LẬP TRÌNH PYTHON

Chương này giới thiệu những kiến thức cơ bản về xử lý hình ảnh và ngôn ngữ Python dùng để lập trình. Các ví dụ cụ thể trong chương này được dùng để giải thích các gói thư viện Python phổ biến dùng trong xử lý ảnh. Một số công cụ cơ bản để bắt đầu lập trình xử lý ảnh trên ngôn ngữ Python ví dụ như đọc hình ảnh, chuyển đổi và nhân rộng hình ảnh, tính toán đạo hàm, vẽ hoặc lưu kết quả sẽ được giới thiệu trong phần này.

1.1. PIL - THƯ VIỆN HÌNH ẢNH TRONG PYTHON

Thư viện hình ảnh Python (PIL) cung cấp chung cho việc xử lý hình ảnh và nhiều thao tác hình ảnh cơ bản hữu ích như thay đổi kích thước, cắt xén, xoay, chuyển đổi màu sắc và các tác vụ khác. Thư viện PIL có thể tải miễn phí tại trang web <http://www.pythonware.com/products/pil/>.

Với thư viện PIL, có thể đọc và ghi các hình ảnh từ hầu hết các định dạng phổ biến nhất hiện nay. Một trong những module quan trọng nhất trong xử lý hình ảnh là Image module. Để đọc một hình ảnh có thể sử dụng đoạn chương trình sau:

```
from PIL import Image
pil_im = Image.open('empire.jpg')
```

Giá trị trả về, `pil_im`, là một đối tượng hình ảnh.

Chuyển đổi màu được thực hiện bằng lệnh `convert()`. Để đọc một hình ảnh và chuyển đổi nó thành thang độ xám, chỉ cần thêm `convert('L')` như sau:

```
pil_im = Image.open('empire.jpg').convert('L')
```

Dưới đây là một số ví dụ được lấy từ tài liệu PIL, có sẵn tại địa chỉ là <http://www.pythonware.com/library/pil/handbook/index.htm>. Đầu ra từ các ví dụ được mô tả trong Hình 1.1.



Hình 1.1: Ví dụ về xử lý ảnh bằng PIL

1.1.1. Chuyển đổi hình ảnh sang định dạng khác

Sử dụng lệnh `save()`, PIL có thể lưu hình ảnh ở hầu hết các định dạng tệp hình ảnh. Ở đây, một ví dụ có thể lấy tất cả các tệp hình ảnh trong một danh sách tên tệp (danh sách tệp) và chuyển đổi hình ảnh thành tệp JPEG.

```
from PIL import Image
import os

for infile in filelist:
    outfile = os.path.splitext(infile)[0] + ".jpg"
    if infile != outfile:
        try:
            Image.open(infile).save(outfile)
        except IOError:
            print "cannot convert", infile
```

Hàm PIL `open()` tạo một đối tượng hình ảnh PIL và lệnh `save()` lưu hình ảnh vào một tệp với tên tệp đã cho trước. Thay vào đó, tên tệp mới sẽ giống như tên gốc với đuôi tệp là ".jpg". PIL dù thông minh để xác định định dạng hình ảnh từ phần mở rộng tệp tin. Có một kiểm tra đơn giản rằng tệp chưa phải là tệp JPEG và thông báo sẽ được in ra màn hình nếu chuyển đổi không thành công.

Trong suốt cuốn sách này, chúng ta thông thường sẽ xử lý một tập nhiều hình ảnh. Dưới đây sẽ trình bày cách có thể tạo một danh sách tên tệp cho tất cả các hình ảnh trong một thư mục. Tạo một tệp `imtools.py` để lưu trữ một số đường dẫn và thêm hàm sau:

```
import os

def get_imlist(path):
    """ Returns a list of filenames for
    all jpg images in a directory. """
    return [os.path.join(path, f) for f in os.listdir(path) if f.endswith('.jpg')]
```

1.1.2. Tạo hình thu nhỏ

Sử dụng PIL để tạo hình thu nhỏ khá đơn giản. Lệnh `thumbnail()` lấy một bộ chỉ định kích thước mới và chuyển đổi hình ảnh thành hình ảnh thu nhỏ với kích thước vừa với bộ dữ liệu. Để tạo hình thu nhỏ với cạnh dài nhất 128 pixel, hãy sử dụng cú pháp như sau:

```
pil_im.thumbnail((128,128))
```

1.1.3. Sao chép và dán các vùng

Cắt một vùng từ một hình ảnh được thực hiện bằng lệnh `crop()`.

```
box = (100,100,400,400)
region = pil_im.crop(box)
```

Vùng được xác định bởi 4 thông số, trong đó tọa độ là (trái, trên, phải, dưới). PIL sử dụng hệ tọa độ có (0, 0) ở góc trên bên trái. Ví dụ, vùng được trích xuất có thể được xoay và sau đó đặt lại bằng lệnh `paste()` như sau:

```
region = region.transpose(Image.ROTATE_180)
pil_im.paste(region,box)
```

1.1.4. Thay đổi kích thước và xoay

Để thay đổi kích thước hình ảnh, gọi hàm `resize()` bằng một bộ thông số cho kích thước mới.

```
out = pil_im.resize((128,128))
```

Để xoay hình ảnh, sử dụng các góc ngược chiều kim đồng hồ và xoay () như sau:

```
out = pil_im.rotate(45)
```

Một số ví dụ được hiển thị trong Hình 1.1. Hình ảnh ngoài cùng bên trái là bản gốc, tiếp theo là phiên bản thang độ xám, hình cắt và hình ảnh thu nhỏ.

1.2. MATPLOTLIB

Khi làm việc với toán học và vẽ đồ thị hoặc vẽ các điểm, đường thẳng và đường cong trên hình ảnh, Matplotlib là một thư viện đồ họa tốt với các tính năng mạnh hơn nhiều so với đồ thị có sẵn trong PIL. Matplotlib tạo ra các hình ảnh chất lượng cao giống như nhiều hình minh họa được sử dụng

trong cuốn sách này. Giao diện Matplotlib PyLab là tập hợp các chức năng cho phép người dùng tạo các đồ thị. Matplotlib là chương trình nguồn mở và có sẵn miễn phí tại <http://matplotlib.sourceforge.net/>. Tại đây cũng có tài liệu và hướng dẫn chi tiết về cách dùng các công cụ trong đó. Dưới đây là một số ví dụ cho thấy hầu hết các chức năng sẽ cần trong cuốn sách này.

1.2.1. Vẽ hình ảnh, điểm và đường

Có thể tạo ra các đồ thị cột, lược đồ hình tròn, các lược đồ phân tán,... với một vài lệnh cho hầu hết các mục đích của thị giác máy tính. Quan trọng nhất, muốn hiển thị những điểm đặc trưng, sự đối xứng và các đối tượng được phát hiện bằng cách sử dụng điểm và đường. Dưới đây là một ví dụ về vẽ đồ thị với một vài điểm và đường thẳng.

```
from PIL import Image
from pylab import *

# read image to array
im = array(Image.open('empire.jpg'))

# plot the image
imshow(im)

# some points
x = [100,100,400,400]
y = [200,500,200,500]

# plot the points with red star-markers
plot(x,y,'r*')

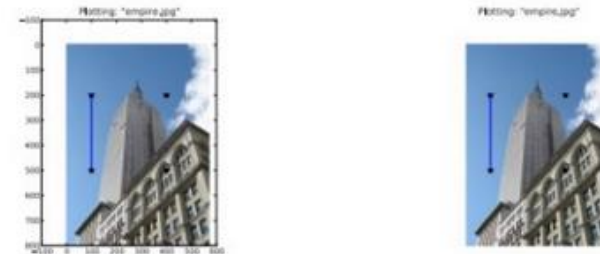
# line plot connecting the first two points
plot(x[:2],y[:2])

# add title and show the plot
title('Plotting: "empire.jpg"')
show()
```

Đoạn chương trình này hiển thị hình ảnh, sau đó vẽ bốn điểm ký hiệu bằng dấu sao màu đỏ với tọa độ x và y cho trước, và cuối cùng vẽ một đường (màu xanh theo mặc định) giữa hai điểm đầu tiên. Hình 1.2 cho thấy kết quả của đoạn chương trình. Lưu ý rằng PyLab sử dụng gốc tọa độ ở góc trên cùng bên trái như thông thường cho hình ảnh. Trục tọa độ rất hữu ích để gỡ lỗi (debug), nhưng nếu muốn một lược đồ đẹp hơn có thể thêm lệnh:

```
axis('off')
```

Lệnh này sẽ xóa trục đồ thị như hình bên phải thay vì hình bên trái trong Hình 1.2.



(a) (b)

Hình 1.2: Ví dụ vẽ đồ thị với Matplotlib.

Hình (a) hiển thị trục tọa độ. Hình (b) không hiển thị trục tọa độ

Có nhiều tùy chọn để định dạng màu sắc và kiểu dáng khi vẽ. Hữu ích nhất là các lệnh ngắn được hiển thị trong Bảng 1.1, 1.2 và 1.3. Sử dụng chúng như sau:

```
plot(x,y) # default blue solid line
plot(x,y,'r*') # red star-markers
plot(x,y,'go-') # green line with circle-markers
plot(x,y,'ks:') # black dotted line with square-markers
```

1.2.2. Đường biên ảnh và lược đồ

Sau đây xem xét hai ví dụ về các đồ thị đặc biệt: đường biên hình ảnh và lược đồ hình ảnh. Việc quan sát được đường biên ảnh cung cấp nhiều thông tin rất hữu ích. Điều này cần hình ảnh thang độ xám (gray scale) vì các đường biên được thực hiện trên ảnh một lớp. Sau đây là một số tùy chọn để hiển thị đồ thị.

Bảng 1.1: Các lệnh định dạng màu cơ bản để vẽ đồ thị với PyLab

color	
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white