BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ISO 9001:2015

ĐỒ ÁN TỐT NGHIỆP

NGHÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

XÂY DỰNG ỨNG DỤNG PHÁT HIỆN PHẦN KHÁC BIỆT GIỮA HAI ẢNH

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

XÂY DỰNG ỨNG DỤNG

PHÁT HIỆN PHẦN KHÁC BIỆT GIỮA HAI ẢNH

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện : Nguyễn Tiến Dũng

Mã sinh viên : 1512111022

Giáo viên hướng dẫn : TS. Ngô Trường Giang.

HẢI PHÒNG - 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

-------o0o-------

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: **Nguyễn Tiến Dũng** Mã sinh viên: **1512111022**

Lớp: **CT1901C** Ngành**: Công nghệ Thông tin**

Tên đề tài: "Xây dựng ứng dụng phát hiện phần khác biệt giữa hai ảnh"

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp a. Nội dung:

- Tìm hiểu tổng quan về đối sánh ảnh
- Tìm hiểu phương pháp căn chỉnh ảnh và trừ ảnh.
- Tìm hiểu một số hàm cơ bản trong thư viện OpenCV sử dụng cho phát hiện phần khác biệt giữa hai ảnh.

b. Các yêu cầu cần giải quyết

- Trình bày tổng quan về đối sánh ảnh
- Hiểu và trình bày phương pháp căn chỉnh ảnh và trừ ảnh .
- Xây dựng ứng dụng phát hiện phần khác biệt giữa hai ảnh sử dụng các hàm trong OpenCV.

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: **Ngô Trường Giang**

Học hàm, học vị: **Tiến sĩ.**

Cơ quan công tác: **Khoa Công nghệ Thông tin**

Nội dung hướng dẫn:

- Tìm hiểu tổng quan về đối sánh ảnh
- Tìm hiểu phương pháp căn chỉnh ảnh và trừ ảnh.
- Tìm hiểu một số hàm cơ bản trong thư viện OpenCV sử dụng cho phát hiện phần khác biệt giữa hai ảnh.

Người hướng dẫn thứ hai:

Họ và tên: …….....

Học hàm, học vị……

Cơ quan công tác: ……

Nội dung hướng dẫn: …….....

……..

……..

Đề tài tốt nghiệp được giao ngày 01 tháng 7 năm 2019

Yêu cầu phải hoàn thành trước ngày 21 tháng 9 năm 2019

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên	Cán bộ hướng dẫn Đ.T.T.N
Nguyễn Tiến Dũng	**Ngô Trường Giang**

Hải Phòng, ngày .....tháng.....năm 2019

HIỆU TRƯỞNG

GS.TS.NGUT Trần Hữu Nghị

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

PHIẾU NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẪN TỐT NGHIỆP

Họ và tên: **Ngô Trường Giang**

Cơ quan công tác: **Khoa Công nghệ Thông tin**

Họ tên sinh viên: **Nguyễn Tiến Dũng**

Ngành: **Công nghệ Thông tin**

Nội dung hướng dẫn:

- Tìm hiểu tổng quan về đối sánh ảnh
- Tìm hiểu phương pháp căn chỉnh ảnh và trừ ảnh.
- Tìm hiểu một số hàm cơ bản trong thư viện OpenCV sử dụng cho phát hiện phần khác biệt giữa hai ảnh.

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

- Sinh viên chủ động tìm đọc các tài liệu liên quan tới đề tài
- Chấp hành nghiêm túc kế hoạch, tiến độ đề ra.

2. Đánh giá chất lượng của đồ án (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp trên các mặt lý luận, thực tiễn, tính toán số liệu..):

- Về mặt lý thuyết: Đồ án trình bày các vấn đề cơ bản về đối sánh ảnh, phương pháp căn chỉnh ảnh dựa vào đối sánh đặc trưng, kỹ thuật trừ ảnh.
- Về mặt thực nghiệm: Đồ án đã cài đặt được chương trình phát hiện và đánh dấu các phần ảnh đã bị thêm/bớt so với ảnh gốc, sử dụng các hàm của OpenCV, và mới dừng lại ở mức các ảnh có biến dạng ít.
- Về hình thức: Báo cáo trình bày sáng sủa, bố cục hợp lý.
- Đồ án đáp ứng được yêu cầu đề ra.

3. Ý kiến của cán bộ hướng dẫn:

Đạt ☐ Không đạt ☐ Điểm:……

Ngày 25 tháng 9 năm 2019

Cán bộ hướng dẫn

TS. Ngô Trường Giang

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA CÁN BỘ CHẤM PHẢN BIỆN

Họ và tên giảng **viên: TS. Đỗ Văn Chiểu.**

Đơn vị công tác: Khoa Công nghệ Thông tin – Trường Đại học Dân lập Hải Phòng

Họ và tên sinh viên**: Nguyễn Tiến Dũng** Ngành: **Công nghệ Thông tin**

Đề tài tốt nghiệp:

"Xây dựng ứng dụng phát hiện phần khác biệt giữa hai ảnh"

1. Phần nhận xét của giảng viên chấm phản biện

Đồ án có bố cục tương đối tốt, hợp lý , trình bày rõ ràng về nội dung, cở sở lý thuyết bám sát đề tài; nêu khá đầy đủ các lý thuyết liên quan

Phần chương trình thể hiện trong chương 3 sử dụng thư viện OpenCV cho kết quả khá chính xác với một số ảnh ít biến động về phép biến đổi hình học

2. Những mặt còn hạn chế

Chương 3 không nên để là chương trình thử nghiệm vì tên đề tài là xây dựng ứng dụng

Phần chương trình (chương 3) không nêu rõ môi trường phát triển các thuật toán sử dụng, thiếu các giao diện trực quan, chưa nêu rõ các điều kiện đầu vào, thuật toán sử dụng cũng như các hạn chế (nếu có)

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm:……

Hải Phòng, ngày .. tháng 10 năm 2019

Cán bộ chấm phản biện

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN **5**

MỞ ĐẦU **6**

CHƯƠNG 1: TỔNG QUAN VỀ ĐỐI SÁNH ẢNH **7**

1.1 Ảnh số 7

1.2 Một số kỹ thuật tiền xử lý ảnh 7

1.2.1 Biến đổi ảnh 7

1.2.2 Nhị phân hóa 8

1.2.3 Lọc nhiễu 9

1.2.4 Kỹ thuật tìm biên 13

1.2.5 Kỹ thuật trừ ảnh 16

CHƯƠNG 2: CĂN CHỈNH ẢNH DỰA TRÊN ĐỐI SÁNH ĐẶC TRƯNG 26

2.1 Giới thiệu căn chỉnh ảnh 26

2.2 Các phép biến đổi đồ họa 27

2.2.1 Các phép biến đổi hình học 2 chiều 27

2.2.2 Phép biến đổi hình học 3D 35

2.3 Căn chỉnh ảnh dựa trên đối sánh đặc trưng 40

2.3.1 Thuật toán ORB 40

2.3.2 Tìm ma trận tương đồng 50

CHƯƠNG 3: CHƯƠNG TRÌNH THỬ NGHIỆM Error! Bookmark not defined.

3.1 Giới thiệu OpenCV 55

3.2 Phát triển ứng dụng phát hiện phần ảnh sai khác 57

3.2.1 Phát biểu bài toán 57

3.2.2 Triển khai sử dụng hàm trong OpenCV 57

3.2.3 Một số kết quả 67
KẾT LUẬN 71
TÀI LIỆU THAM KHẢO 72

DANH SÁCH CÁC HÌNH ẢNH

Hình 1.1 Một số kiểu đường biên thông dụng 13
Hình 1.2 Toán tử Sobel 15
Hình 1.3 Toán tử Prewitt 15
Hình 1.4 Toán tử Robert 16
Hình 1.5 Các ảnh có cùng biểu đồ màu nhưng nội dung khác nhau 23
Hình 2.1 Phép biến đổi vị trí 29
Hình 2.2 Các phép đối xứng trên 2D 30
Hình 2.3 Phép biến dạng theo trục oy 31
Hình 2.4 Phép biến dạng theo trục ox 31
Hình 2.5 Phép quay trên 2D 32
Hình 2.6 Phép tịnh tiến 34
Hình 2.7 Phép tỷ lệ 34
Hình 2.8 Phép xoay 35
Hình 2.9 Phép tịnh tiến trên 3D 37
Hình 2.10 Phép tỷ lệ 37
Hình 2.11 Các phép biến dạng trên 3D 38
Hình 2.12 Xác định góc quay dương trên 3 trục toạ độ 38
Hình 2.13 Quay quanh trục ox 39
Hình 2.14 Ví dụ về kết quả đối sánh ảnh sử dụng thuật toán ORB 41
Hình 2.15 Đồ thị cường độ nhiễu của ảnh 44
Hình 2.16 Sự phân phối cân bằng các vector thuộc tính 46
Hình 2.17 Xác định tập con các điểm kiểm tra nhị phân 47
Hình 2.18 Hiệu suất đối sánh của SIFT, SURF, BRIEF với FAST và ORB . 48
Hình 2.19 Thao tác đối sánh có nhiễu cho SIFT và rBRIEF 49
Hình 2.20 Ví dụ thực tế về đối sánh ảnh ORB 49
Hình 2.21 Phép chiếu Homography 51
Hình 3.1 Ảnh gốc bên trái và ảnh cần căn chỉnh bên phải (vùng khoanh đỏ là phần khác biệt giữa 2 ảnh) 57
Hình 3.2 Ảnh thứ 2 đã được căn chỉnh về xám 59
Hình 3.3. Ảnh chứa các đặc trưng tương đồng (match feature) 60
Hình 3.4 Ảnh đã được căn chỉnh và ánh xạ gần tọa độ ảnh gốc nhất 61
Hình 3.5 Ảnh sau khi dùng hàm diff 62
Hình 3.6 Ảnh sau khi sử dụng hàm diff, phân ngưỡng, lọc nhiễu để highlight phần khác biệt (bằng 2 màu trắng đen) 64

Hình 3.7 Đây là hai ảnh sau khi đã tìm được sự khác biệt và được khoanh vùng bởi hình chữ nhật màu đỏ..... 67
Hình 3.8 Ảnh gốc 67
Hình 3.9 Ảnh cần kiểm tra 68
Hình 3.10 Ảnh cần kiểm tra đã được căn chỉnh và ánh xạ gần tọa độ ảnh gốc nhất..... 68
Hình 3.11 Ảnh chứa các đặc trưng tương đồng (match feature) 69
Hình 3.12 Ảnh sau khi sử dụng hàm diff, phân ngưỡng, lọc nhiễu để highlight phần khác biệt giữa 2 ảnh..... 69
Hình 3.13 Đây là hai ảnh sau khi đã tìm được sự khác biệt và được khoanh vùng bởi hình chữ nhật màu đỏ..... 70

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Dân Lập Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy giáo TS. Ngô Trường Giang, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của Thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt là các bạn trong lớp CT1901C đã luôn gắn bó, cùng học tập và giúp đỡ em trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 21 tháng 09 năm 2019

Sinh viên

Nguyễn Tiến Dũng

MỞ ĐẦU

Trong các ngành sản xuất, hình ảnh thường được sử dụng cho mục đích kiểm soát chất lượng. Trong các ứng dụng cũng như vậy, nếu chất lượng sản phẩm tốt, thì hình ảnh của sản phẩm đó hoàn toàn giống với hình ảnh của một sản phẩm có chất lượng tốt. Do đó, so sánh các hình ảnh là một nhiệm vụ cơ bản trong kiểm soát chất lượng dựa trên hình ảnh. Tuy nhiên, vấn đề này phức tạp theo nghĩa là hình ảnh quan sát thường có nhiễu và hình ảnh được xử lý cần phải khớp với nhau về mặt hình học vì hình ảnh của các sản phẩm có thể không khớp về mặt hình học, bởi vì do thực tế sản phẩm để cập nhập lên sẽ có rất nhiều góc chụp khác nhau và hình ảnh sản phẩm thường không hoàn toàn giống nhau. Vấn đề đầu tiên được gọi là căn chỉnh hình ảnh và vấn đề thứ hai được gọi là trừ ảnh.

Để giải quyết vấn đề trên em trình bày một ứng dụng phát hiện phần khác biệt giữa hai ảnh được ứng dụng trong kiểm tra tự động sản phẩm lỗi.

Nội dung của đề tài bao gồm ba chương:

- Chương 1: Tổng quan về đối sánh ảnh
- Chương 2: Căn chỉnh ảnh dựa trên đối sánh thuộc tính
- Chương 3: Chương trình thử nghiệm

CHƯƠNG 1: TỔNG QUAN VỀ ĐỐI SÁNH ẢNH

1.1 Ảnh số

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả gần với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặc điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn. Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster.

- Mô hình Vector: Ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng hiển thị và in ấn, các ảnh biểu diễn theo mô hình vector còn có ưu điểm cho phép dễ dàng lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này thì kỹ thuật biểu diễn vector tỏ ra ưu việt hơn. Trong mô hình này, người ta sử dụng hướng vector của các điểm ảnh lân cận để mã hóa và tái tạo lại hình ảnh ban đầu. Các ảnh vector được thu nhận trực tiếp từ các thiết bị số hóa như Digitalize hoặc được chuyển đổi từ các ảnh Raster thông qua các chương trình vector hóa.
- Mô hình Raster: là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster thuận lợi cho việc thu nhận, hiển thị và in ấn. Các ảnh được sử dụng trong phạm vi của đề tài này cũng là các ảnh được biểu diễn theo mô hình Raster

1.2 Một số kỹ thuật tiền xử lý ảnh

1.2.1 Biến đổi ảnh

Thuật ngữ biến đổi ảnh thường được dùng để nói tới một lớp các ma trận đơn vị và các kỹ thuật dùng để biến đổi ảnh. Cũng như các tín hiệu một

chiều được biểu diễn bởi một chuỗi các hàm cơ sở, ảnh cũng có thể được biểu diễn dưới một số chuỗi rời rạc các ma trận cơ sở gọi là ảnh cơ sở. Phương trình ảnh cơ sở có dạng:

$$A^*_{k,l} = a_k * a_l * T \qquad (1\text{-}1)$$

Với ak là cột thứ k cửa ma trận A. A là ma trận đơn vị. Có nghĩa là AA*T=1. Các A*k,l được định nghĩa ở trên với k,l = 0, 1, 2, …, N-1 là ảnh cơ sở. Có nhiều loại biến đổi được dùng như:

- Biến đổi Fourier, Sin, Cosin, Hadamard….
- Tích Kronecker.
- Biến đổi KL (Krhumen loeve).

Do phải xử lý nhiều thông tin, các phép toán nhân và cộng trong khai triển là quá lớn. Do vậy các phép biến đổi trên nhằm giảm thứ nguyên của ảnh để việc xử lý ảnh được hiệu quả hơn.

1.2.2 Nhị phân hóa

Là quá trình biến đổi một ảnh xám thành ảnh nhị phân.

- Ta gọi giá trị cường độ sáng tại một điểm ảnh là I(x,y).
- INP(x,y) là cường độ sáng của điểm ảnh trên ảnh nhị phân.
- (Với 0 < x < image.width) và (0 < y < image.height).

Để biến đổi ảnh xám thành ảnh nhị phân. Ta so sánh giá trị cường độ sáng của điểm ảnh với một ngưỡng nhị phân **T**.

- Nếu I(x,y) > **T** thì INP(x, y) = 0 (0).
- Nếu I(x,y) > **T** thì INP(x, y) = 255 (1).

Chú ý

- Bạn có thể hoàn toàn chọn giá trị **T** từ 0 đến 255, nhưng thông thường nhiều người hay chọn một giá trị đó là 128 tức là giá trị trung bình của max(255) và min(0) của cường độ sáng (Intensity) của điểm ảnh.
- Bạn có thể dễ dàng nhận thấy với mỗi **T** thì có một ảnh nhị phân khác nhau (Khác nhau ở đây là cường độ sáng của các tấm ảnh nhị phân với mỗi giá trị **T**).

1.2.3 Lọc nhiễu

Thông thường ảnh thu nhận có nhiễu cần phải loại bỏ nhiễu hay ảnh không sắc nét bị mờ hoặc cần làm rõ các chi tiết như các đường biên ảnh. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân nhóm theo công dụng: làm trơn nhiễu, nổi biên. Để làm trơn nhiễu 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).Từ bản chất của nhiễu (thường tương ứng với tần số cao) và từ cơ sở lý thuyết lọc là: bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua do đó, để lọc nhiễu người ta thường dùng lọc thông thấp (theo quan điểm tần số không gian) hay lấy tổ hợp tuyến tính để san bằng (lọc trung bình). Để làm nổi biên (ứng với tần số cao), người ta dùng các bộ lọc thông cao, lọc Laplace

Để hiểu rõ hơn các kỹ thuật áp dụng, cần phải phân biệt các loại nhiễu can thiệp trong quá trình xử lý ảnh. Trên thực tế tồn tại khá nhiều loại nhiễu như sự thay đổi độ nhạy của cảm biến, sự biến đổi của môi trường, sai số của quá trình lượng tử hóa, sai số của kênh truyền…; tuy nhiên người ta thường xem xét 3 loại nhiễu chính và phổ biến là: nhiễu cộng, nhiễu nhân và nhiễu xung:

- Nhiễu cộng (Additive noise): thường phân bố khắp ảnh và được biểu diễn bởi:

$$Y=X+n \tag{1-2}$$

- Nhiễu nhân: cũng thường phân bố khắp ảnh và được biểu diễn bởi:

$$Y = X * n \tag{1-3}$$

- Chú ý: với Y: ảnh quan sát, X: ảnh gốc và n là nhiễu.
- Nhiễu xung (Impulse noise): là một loại nhiễu khá đặc biệt có thể sinh ra bởi nhiều lý do khác nhau chẳng hạn: lỗi truyền tín hiệu, lỗi bộ nhớ, hay lỗi định thời trong quá trình lượng tử hóa. Nhiễu này thường gây đột biến tại một số điểm ảnh.

1.2.3.1 Lọc trung bình không gian

Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số các điểm lân cận và được định nghĩa như sau:

$$v(m,n) = \sum_{(k,1)\in W} a(k,l)y(m-k,n-l) \tag{1-4}$$

Nếu trong kỹ thuật lọc trên, ta chọn các trọng số bằng nhau, phương trình trên sẽ có dạng:

$$v(m,n) = \frac{1}{N}\sum_{(k,1)\in W} a(k,l)y(m-k,n-l) \tag{1-5}$$

Với:

- y(m,n): ảnh đầu vào
- V(m,n):ảnh đầu ra

- A(k,l):là trọng số lọc
- $a_{k.l} = \frac{1}{N}$ và $N_w$là số điểm ảnh trong cửa sổ lọc W

Lọc trung bình có trọng số chính là thực hiện chập ảnh đầu vào với nhân chập H. Nhân chập H trong trường hợp này có dạng:

$$H = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \quad (1\text{-}6)$$

Trong lọc trung bình, thường người ta ưu tiên cho các hướng để bảo vệ biên của ảnh khỏi bị mờ khi làm trơn ảnh. Các kiểu mặt nạ được sử dụng tùy theo các trường hợp khác nhau. Các bộ lọc trên là bộ lọc tuyến tính theo nghĩa là điểm ảnh ở tâm cửa số sẽ được thay bởi tổ hợp các điểm lân cận chập với mặt nạ.

Giả sử ảnh đầu vào biểu diễn bởi ma trận:

$$I = \begin{bmatrix}4 & 7 & 3 & 7 & 1\\ 5 & 7 & 1 & 7 & 1\\ 6 & 6 & 1 & 8 & 3\\ 5 & 7 & 5 & 7 & 1\\ 5 & 7 & 6 & 1 & 2\end{bmatrix} \quad (1\text{-}7)$$

Ảnh số thu được bởi lọc trung bình Y=H⊗I có dạng:

$$Y = \frac{1}{9}\begin{bmatrix}23 & 26 & 31 & 19 & 16\\ 35 & 39 & 46 & 31 & 27\\ 36 & 43 & 49 & 34 & 27\\ 36 & 48 & 48 & 34 & 22\\ 24 & 35 & 33 & 22 & 11\end{bmatrix} \quad (1\text{-}8)$$

Lọc trung bình trọng số là một trường hợp riêng của lọc thông thấp

1.2.3.2 Lọc trung vị

Khái niệm trung vị được viết bởi công thức:

v(m,n) = (y(m-k, n-l) với (k, l) thuộc W *(1-9)*

Kỹ thuật này đòi hỏi giá trị các điểm ảnh trong cửa sổ phải xếp theo thứ tự Tăng hay giảm dần so với giá trị trung vị. Kích thước cửa số thường được chọn sao cho số điểm ảnh trong cửa số là lẻ. Các cửa sổ hay dùng là cửa sổ có kích thước 3x3, hay 5x5 hay 7x7. Ví dụ: Nếu y(m) = {2, 3, 8, 4, 2} và cửa sổ W = (-1, 0, 1) thì ảnh kết quả thu được sau lọc trung vị là v(m) = {2, 3, 4, 4, 2}. do đó:

v[0] = 2 <giá trị biên> v[3] = Trungvi(8, 4, 2) = 4

v[1] = Trungvi(2, 3, 8) = 3 v[4] = 2 <giá trị biên>

v[2] = Trungvi(3, 8, 4) = 4

Tính chất của lọc trung vị:

- Lọc trung vị là loại lọc phi tuyến. Điều này được thể hiện:

  Trungvi(x(m) + y(m)) ≠ Trungvi(x(m)) + Trungvi(y(m)). *(1-10)*

- Có lợi cho việc loại bỏ các điểm ảnh hay các hàng mà vẫn bảo toàn bộ phân giải.
- Hiệu quả giảm khi số điểm trong cửa sổ lớn hay bằng một nửa số điểm trong cửa sổ. Điều này dễ giải thích vì trung vị là (Nw+1)/2 giá trị lớn nhất nếu N w lẻ. Lọc trung vị cho trường hợp 2 chiều coi như lọc trung vị tách được theo từng chiều.

1.2.4 Kỹ thuật tìm biên

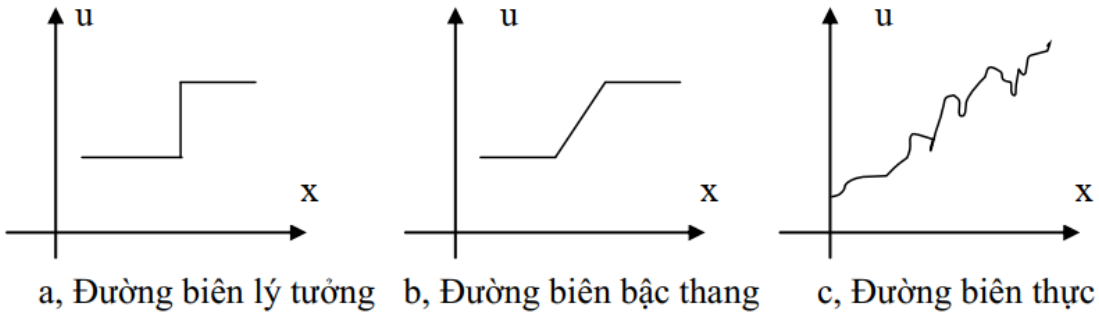
1.2.4.1 Một số khái niệm về biên

Điểm biên: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.

Đường biên: là tập hợp các điểm biên liên tiếp tạo thành một đường biên.

Ý nghĩa của đường biên trong xử lý: Thứ nhất, đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.

Đường biên là nơi mà các điểm ảnh lân cận nhau có cường độ thay đổi mạnh một cách đột ngột. Một số kiểu đường biên hay gặp trên thực tế được minh họa trên hình 1.1.



Hình 1.1 Một số kiểu đường biên thông dụng

1.2.4.1.1 Các kiểu biên cơ bản

Các phương pháp phát hiện biên truyền thống thường dựa trên kết quả của phép nhân chập giữa bức ảnh cần nghiên cứu f(x,y) và một bộ lọc 2D

h(x, y) thường được gọi là mặt nạ (mask). Công thức phát hiện biên được trình bày như bên dưới:

$$h(x,y) * f(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(k_1,k_2) f(x-k_1, y-k_2) dk_1 dk_2$$

Cấu trúc và giá trị của các toán tử phát hiện biên sẽ xác định hướng đặc trưng mà toán tử nhạy cảm với biên. Có một số toán tử thích hợp cho các đường biên có hướng nằm ngang, một số toán tử lại thích hợp cho việc tìm kiếm biên dạng thẳng đứng hay theo hướng đường chéo. Có nhiều phương pháp phát hiện biên đang được áp dụng, tuy nhiên ta có thể phân thành hai nhóm cơ bản là phát hiện biên dùng Gradient và phương pháp Laplacian.

Phương pháp Gradient.

Phương pháp phát hiện biên dùng Gradient (sử dụng các toán tử Roberts, Prewitt, Sobel, Canny) dựa vào tính giá trị cực đại và cực tiểu của đạo hàm bậc nhất của ảnh. Đạo hàm bậc nhất theo hướng ngang và dọc được tính theo công thức sau:

$$\Delta f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1\text{-}11)$$

Biên độ của vector gradient hay độ lớn tổng cộng của giá trị đạo hàm nằm tại biên là kết hợp của cả hai giá trị này theo công thức sau:

$$\Delta f = |\Delta f| = \sqrt{G_x^2 + G_y^2} \quad (1\text{-}12)$$

Hướng của vector gradient được xác định theo công thức sau:

$$\nabla f = \tan^{-1}\theta\left(\frac{Gx}{Gy}\right) \quad (1\text{-}13)$$

Hướng của biên sẽ vuông góc với hướng của vector gradient này.

Toán từ Sobel

Trên thực tế Sobel sử dụng hai mặt nạ có kích thước [3 x 3] trong đó một mặt nạ chỉ đơn giản là sự quay của mặt nạ kia đi một góc 90° như ở hình 1-6. Các mặt nạ này được thiết kế để tím ra các đường biên theo chiều đứng và chiều ngang một cách tốt nhất. Khi thực hiện phép tích chập giữa ảnh và các mặt nạ này ta nhận được các gradient theo chiều đứng và chiều ngang Gx, Gy. Toán tử Sobel có dạng như hình 1.2.

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Hình 1.2 Toán tử Sobel

Toán tử Prewitt

Phương pháp Prewitt gần giống với Sobel. Đây là phương pháp lâu đời nhất, cổ điển nhất. Toán tử Prewitt được mô tả trên hình 1.3.

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Hình 1.3 Toán tử Prewitt

Toán tử Robert

Tương tự như Sobel, ta tính đường biên ngang và dọc một cách riêng rẽ dùng 2 mặt nạ như hình 1.4, sau đó tổng hợp lại để cho đường biên thực của ảnh.

0	0	0
0	−1	0
0	0	1

0	0	0
0	0	−1
0	1	0

Hình 1.4 Toán tử Robert

1.2.5 Kỹ thuật trừ ảnh

Hiểu theo nghĩa hẹp, trừ hai ảnh có cùng kích thước là việc xây dựng ảnh mới từ sự khác biệt của hai ảnh. Theo nghĩa rộng hơn, trừ ảnh là việc tính toán độ chênh lệch giữa hai ảnh trên một đặc trưng ảnh nào đó như cường độ, màu sắc, texture (kết cấu), shape (hình dáng), chuyển động…

- Dựa vào khối: Chia ảnh thành các miền và so sánh các miền tương ứng.
- Dựa vào biểu đồ: So sánh sự phân bố của thuộc tính nào đó của ảnh.

Kết hợp các loại này với các thuộc tính so sánh của ảnh, ta sẽ có nhiều kỹ thuật trừ ảnh khác nhau.

1.2.5.1 Dựa vào so sánh điểm ảnh?

Đây là phương pháp để tính toán sự sai khác giữa hai frame bằng việc tính các giá trị, nó mô tả mọi thay đổi về cường độ điểm ảnh trong các ảnh. Có nhiều phương pháp để tính sự sai khác này, Nagasaka và Tanaka đã đưa ra một phương pháp tính tổng toàn bộ những thay đổi khác nhau về cường độ điểm ảnh giữa hai khung hình như là độ chênh lệch khung $D(f_{1,}f_2)$.

$$D(f_1, f_2) = \frac{1}{X \times Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |f_1(x,y) - f_2(x,y)| \quad (1\text{-}14)$$

Sau khi tính được độ chênh lệch D, tiến hành so sánh D với ngưỡng chuyển cảnh T xác định xem có chuyển cảnh hay không.

Nhược điểm của phương pháp này là:

- Không thể phân biệt được thay đổi lớn cho vùng ảnh nhỏ và thay đổi nhỏ cho vùng ảnh lớn. Ví dụ như các cắt cảnh rất dễ bị bỏ sót khi một phần nhỏ của khung hình có sự thay đổi lớn hoặc nhanh.
- Nhạy với nhiễu và các di chuyển camera.

Một bước phát triển hơn dược Otsuji đề xuất đó là thay vì tính toán trực tiếp tổng những điểm khác biệt lớn về cường độ thực tế, tiến hành dếm các số điểm ảnh có thay đổi lớn hơn một ngưỡng nào đó, so sánh tổng đó với ngưỡng khác để phát huện chuyển cảnh.

$$DP(x,y) = \begin{cases} 1, \text{Nếu } |f_1(x,y) - f_2(x,y)| > T_1 \\ 0, \quad \text{ngược lại} \end{cases} \quad (1\text{-}15)$$

$$D(f_1, f_2) = \frac{1}{X \times Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} DP(x,y) \quad (1\text{-}16)$$

Nếu tỉ lệ số diểm ảnh thay đổi $D(f_1,f_2)$ lớn hơn ngưỡng T thì ta đã có chuyển cảnh do cắt. Tuy các thay đổi không liên quan trong khung hình đã được loại bớt nhưng phương pháp này vẫn nhãy cảm với những di chuyển camera và di chuyển của đối tượng khi camera quay hướng theo đối tượng, rất nhiều điểm ảnh thay đổi dù chỉ một số ít điểm ảnh dịch chuyển.

Một nhược điểm nữa của phương pháp phân biệt điểm ảnh là tính nhạy cảm những thay đổi về độ sáng của ảnh, ví dụ điển hình là các chớp sáng (đèn flash).

Giá trị độ xám nhảy lên mức cao khi chớp sáng xuất hiện. Điều này sẽ trở lại bình thường sau một số frame do các thay đổi mở của camera. Nhưng với một cảnh thật, phân bố màu sẽ không trở lại mức ban đầu. Người ta dùng tỉ lệ khác biệt màu qua frame và khác biệt màu long term để phát hiện flash. Tỉ lệ này được định nghĩa:

$$\mathrm{Fr(i)} = \mathrm{D(i, i-1)/D(i+\delta, i-1)} \qquad \textit{(1-17)}$$

Trong đó i là frame đang xét, và δ là chiều dài trung bình của thay đổi mở của camera. Nếu Fr(i) nhỏ hơn một ngưỡng cho trước thì một chớp sáng được phát hiện tại vị trí frame thứ i và ngược lại. Khi đố người ta điều chỉnh độ sai khác giá trị điểm ảnh bằng cách chia nó cho cường độ của điểm ảnh trên khung hình thứ hai

$$\mathbf{D(f_1, f_2)} = \frac{\mathbf{1}}{\mathbf{X \times Y}} \sum_{\mathbf{x=0}}^{\mathbf{X-1}} \sum_{\mathbf{y=0}}^{\mathbf{Y-1}} \frac{\mathbf{|f_1(x,y) - f_2(x,y)|}}{\mathbf{f_2(x,y)}} \qquad \textit{(1-18)}$$

Phương pháp trừ giá trị điểm ảnh cơ bản là tính toán từ các giá trị cường độ, nhưng có thể mở rộng với các ảnh màu. Ví dụ với ảnh màu RGB, ta tính tổng có trọng số các sai khác của ba gia trị Red, Green và Blue của các điểm ảnh

$$\mathrm{D(f_1, f_2)} = \frac{1}{\mathrm{X \times Y}} \sum_{\mathrm{x=0}}^{\mathrm{X-1}} \sum_{\mathrm{y=0}}^{\mathrm{Y-1}} \sum_{\mathrm{i \in \{R,G,B)}} \mathrm{w_i |f_{1i}(x,y) - f_{2i}(x,y)|} \qquad \textit{(1-19)}$$

1.2.5.2 Dựa vào khối

Trái ngược với hướng tiếp cận sử dụng các đặc tính toàn cục của cả khung hình, hướng tiếp cận phân phối sử dụng các đặc tính cục bộ nhằm tăng tính độc lập với các di chuyển của camera và đối tượng. Mỗi khung hình được chia thành b khối. Các khối trên khung hình f_1 được so sánh với khối trương ứng trên khung hình f_2. Về cơ bản, độ chênh lệch giữa hai khung hình được tính như sau:

$$D(f_1, f_2) = \sum_{k-1}^{b} C_k \,.\, DP(f_1, f_2, k) \qquad (1\text{-}20)$$

Trong đó C_k là hệ số cho trước, $DP(f_1,f_2,k)$ là độ chênh lệch giữa khối thứ k của hai khung hình f_1 và f_2.

Kasturi so sánh các khối tương ứng áp dụng công thức:

$$\lambda_k = \frac{\left[\frac{\sigma_{1k}\sigma_{2k}}{2} + \left(\frac{\mu_{1k} - \mu_{2k}}{2}\right)^2\right]^2}{\sigma_{1k}.\sigma_{2k}} \qquad (1\text{-}21)$$

Trong đó μ_{1k}, μ_{2k} là giá trị cường độ trung bình của khối thứ k, và σ_{1k}, σ_{2k} là độ chênh lệch tương ứng với hai khối đó.

$$DP(f_1, f_2, k) = \begin{cases} 1 & \text{Nếu } \lambda > T_1 \\ 0 & \text{Nếu ngược lại} \end{cases} \qquad (1\text{-}22)$$

Một cắt cảnh xảy ra khi số các khối thay đổi đủ lớn, nghĩa là $D(f_1,f_2)>T_2$ và $C_k=1$ cho tất cả các khối. Phương pháp này chậm đi theo độ phức tạp của hàm thống kê. Phương pháp này có một bất lợi là các chuyển shot sẽ bị bỏ qua trong trường hợp hai khối rất khác nhau có thể có cùng hàm mật độ. Tuy nhiên trường hợp đó cũng ít xảy ra.

Một hướng tiếp cận khác với kỹ thuật trừ ảnh phân khối do Shahraray đưa ra. Shahararay đã chia khung hình thành 12 miền và tìm miền thích hợp nhất cho mỗi miền ở khung hình kia. Độ chênh lệch tính bằng kỹ thuật trừ ảnh dựa vào điểm ảnh của từng miền được sắp xếp, tổng có trọng số của các chênh lệch được sắp xếp cho ta kết quả D cuối cùng.

Xong phát triển phương pháp trừ ảnh, gọi là so sánh thực, phát hiện chuyển cảnh do ngắt chỉ bằng việc so sánh một phần của ảnh. Phương pháp này chỉ ra rằng, sai sót mắc phải hoàn toàn có thể bỏ qua nếu ít hơn một nửa các cửa sổ cơ sở (các ô vuông không chồng nhau) đều được kiểm tra. Trong trường giữa hai khung hình có sự biến đổi lớn hơn kích thước của các cửa sổ được chọn đủ lớn để bất biến với các thay đổi không làm vỡ và đủ nhỏ để có thể chứa thông tin về không gian nhiều chừng nào có thể. Các cửa sổ cơ sở được so sánh và tính độ chênh lệch mức xám hoặc giá trị màu của các điểm ảnh. Khi giá trị chênh lệch lớn hơn một ngưỡng nào đó thì xem như miền đang xét đã thay đổi, khi số miền thay đổi lớn hơn một ngưỡng khác thì sự chuyển cảnh do ngắt đã xảy ra. Thực nghiệm đã chứng minh rằng hướng tiếp cận này cho tốc độ nhanh hơn phương pháp so sánh từng cặp điểm, thậm chí cả phương pháp biểu đồ xét dưới đây.

1.2.5.3 Dựa vào so sánh biểu đồ

Phương pháp đo sự khác biệt giữa các frame dưới dạng giá trị màu không mạnh do chuyển động của camera và đối tượng có thể gây ra sự khác biệt giá trị điểm ảnh quá lớn. Có thể dùng biểu đồ màu hoặc biểu đồ mức xám để tính toán sự sai khác giữa hai khung hình vì sự phân bố màu giữa các frame liên tục không bị ảnh hưởng nhiều bởi chuyển động của camera và chuyển động của đối tượng

Biểu dồ màu (mức xám) của khung hình i là một vecto G chiều $H_i=(H_i(1),H_i(2),\ldots,H_i(G))$. Trong đó G là số màu (mức xám), $H_i(j)$ là số điểm ảnh của khung hình i có màu (mức xám)j.

a) Biểu đồ toàn cục

Phương pháp đơn giản nhất là tính tổng sự sai khác các cột của biểu đồ.

$$D(f_1,f_2)=\sum_{k=0}^{G} w(k)|H_1(k)-H_2(k)| \quad (1\text{-}23)$$

Trong đó w(k) là trọng số ứng với giá tị màu (mức xám) k.

Swain và Ballard lại sử dụng sự giao nhau của biểu đồ được so sánh:

Vùng biểu đồ chung nhau,cho biết độ tương tự về nội dung hai ảnh có thể được định nghĩa như sau:

$$S(f_1,f_2)=\sum_{k=0}^{G} \min(H_1(k)-H_2(k)) \quad (1\text{-}24)$$

Độ tương tự còn có thể được định nghĩa như sau:

$$S(f_1,f_2)=\frac{\sum_{k=0}^{G}\min(H_1(k)-H_2(k))}{\sum_{k=0}^{G}\max(H_1(k)-H_2(k))} \quad (1\text{-}25)$$

Như vậy có thể tính độ chênh lệch biểu đồ hai khung hình theo công thức

$$S(f_1,f_2)=1-S(f_1,f_2)=1=\frac{\sum_{k=0}^{G}\min(H_1(k)-H_2(k))}{\sum_{k=0}^{G}\max(H_1(k)-H_2(k))} \quad (1\text{-}26)$$

Phương pháp khác biệt về biểu đồ màu được sử dụng nhiều và thông dụng nhất vì nó tính toán nhanh, đơn giản và hiệu quả trong việc phát hiện

chuyển cảnh đột ngột, chuyển cảnh rõ ràng, hoặc có sự dịch chuyển nhỏ của đối tượng và sự dịch chuyển của camera. Nhưng với chuyển cảnh dần dần, các ảnh từ từ mờ đi, đan xen lẫn nhau, làm cho khác biệt về đặc trưng giữa các khung hình liên tiếp tương đối nhỏ, không đủ để vượt qua ngưỡng xác định đã đặt ra, dẫn đến khó phát hiện được chuyển cảnh. Nếu như ngưỡng xác định thấp quá thì sẽ tìm ra nhiều đoạn dư thừa,nếu đặt ngưỡng quá cao thì không phát hiện được chuyển cảnh. Mặt khác, trong trường hợp camera hay đối tượng chuyển động nhanh cũng tạo ra sự khác biết tương đối lớn giữa các khung hình. Để giải quyết vấn đề này, một phương pháp được đề xuất là sử dụng hai ngưỡng để tăng mức độ tìm đúng và phát hiện được chuyển cảnh dần dần, đó chính là phương pháp so sánh cặp.

Phương pháp so sánh cặp tính toán chênh lệch tích lũy giữa các khung hình sử dụng một ngưỡng lớn hơn cho chuyển cảnh trực tiếp T_h và một ngưỡng nhõ hơn cho chuyển cảnh dần dần T_l.

Trước tiên sử dụng T_h để phát hiện chuyển cảnh do cắt cảnh. Sau đó sử dụng T_l để phát hiện vị trí khung hình F_s có thể là khung hình đầu tiên của chuyển cảnh dần dần, khung F_s này được đem so sánh với các khung tiếp theo, công việc so sánh tích lũy vì trong suốt quá trình biến đổi dần dần độ chênh lệch sẽ tăng lên. Khung hình cuối cùng của chuyển cành dần dần sẽ được phát hiện khi độ chênh lệch giảm xuống thấp hơn mức ngưỡng T_l trong đó so sánh tích ũy vượt ngưỡng T_h.

Nếu độ chệnh lệch giữa các khung hình liên tiếp giảm xuống dưới ngưỡng T_l mà so sánh tích lũy chưa vượt T_h thì bỏ qua vị trí F_s và việc tìm kiếm bắt đầu với một biến đổi dần dần khác. Tuy nhiên, có nhiều biến đổi dần dần mà chệnh lệch giữa các khung hình liên tiếp đều nhỏ hơn ngưỡng bé T_l.

Vấn đề này có thể giải quyết dễ dàng bằng cách đặt giá trị chấp nhận được cho phép chỉ một số lượng nhất định các khung hình liên tiếp có chênh lệch thấp trước khi loại trường hợp biến đổi đang xét. Như vậy, phương pháp so sánh cặp có thể phát hiện chuyển cảnh đột ngột và chuyển cảnh dần dần cùng một lúc. Qua kết quả thực nghiệm cài đặt thuật toán và so sánh các kỹ thuật phân đoạn khác nhau và thấy rằng so sánh cặp là phương pháp đơn giản và phân đoạn rất tốt.

Nhược điểm của phương pháp này là:

- Biểu đồ chỉ mô tả sự phân bố các giá trị điểm ảnh(màu hay mức xám) chứ không chứa đựng các thông tin nào về không gian. Như vậy hai ảnh có cùng biểu đồ màu nhưng vẫn có thể có nội dung khác nhau do không gian phân bố khác nhau

Hình 1.5 Các ảnh có cùng biểu đồ màu nhưng nội dung khác nhau

- Với những vùng cảnh nhỏ, khi thay đổi vẫn gây ra chú ý nhưng lại không đóng vai trò quan trọng biểu đồ và như vậy thì rất dễ bị bỏ qua khi tiến hành kĩ thuật trừ ảnh để tìm ra sự sai khác.

b) Biểu đồ cục bộ

Biểu đồ cục bộ là biểu đồ mô tả sự phân phối các giá trị điểm ảnh trên một phần của khung hình. Như đã đề cập ở trên, phương pháp trừ ảnh dựa vào biểu đồ là phương pháp ít chịu ảnh hưởng của nhiễu và sự di chuyển đối

tượng. Tuy nhiên với biểu đồ toàn cục thì vẫn gặp một số trở ngại, để khắc phục những nhược điểm của biểu đồ toàn cục, chúng ta sẽ kết hợp trừ ảnh dựa vào biểu đồ với kỹ thuật trừ ảnh phân khối. Trừ ảnh phân khối quan tâm đến thông tin về không gian. Về cơ bản phương pháp này tốt hơn việc so sánh từng cặp diểm ảnh, nhưng nó vẫn chịu tác động của sự di chuyển của camera và đối tượng và cũng tốn kém. Còn kỹ thuật trừ ảnh dựa vào biểu đồ không chịu ảnh hưởng nhiều của camera hay sự di chuyển của đối tượng, nhưng lại có nhược điểm là không chứa đựng thông tin về không gian. Như vậy việc kết hợp hai phương pháp này sẽ bù đắp được những thiếu sót cho nhau. Bằng cách kết hợp hai ý tưởng, chúng ta vừa có thể giảm được tác động của sự di chuyển camera và đối tượng, vừa được sử dụng thông tin về không gian ảnh, và cho kết quả phân đoạn tốt hơn.

Ý tưởng là, ta sẽ chia khung hình thành b khối, đánh sô từ 1 đến b. So sánh biểu đồ của các khối tương ứng rồi tính tổng chệnh lệch để có kết quả trừ ảnh cuối cùng.

$$D(f_1, f_2) = \sum_{k-1}^{b} DP(f_1, f_2, k) \qquad (1\text{-}27)$$

Với

$$D(f_1, f_2) = \sum_{j=0}^{G} |H_1(j,k) - H_2(j,k)| \qquad (1\text{-}28)$$

Trong đó H(j,k) là giá trị biểu đồ mức xám j ứng với khối thứ k.

Hướng tiếp cận khác trong kỹ thuật trừ ảnh dựa vào biểu đồ cục bộ được Swanberg đưa ra. Sự chệnh lệch DP(f1,f2,k) giữa các khối được tính bằng cách so sánh biểu đồ màu RGB sử dụng công thức sau:

$$DP(f_1, f_2, k) = \sum_{c\in\{R,G,B)} \sum_{j=0}^{G} \frac{(H_1^c(j,k) - H_2^c(j,k))^2}{H_2^c(j,k)} \quad (1\text{-}29)$$

1.2.5.4 Dựa vào phương pháp thống kê

Phương pháp sai khác thống kê dựa vào phương pháp trừ giá trị điểm ảnh, nhưng thay vì tính tổng sự sai khác của tất cả các điểm ảnh, ta chia ảnh thành các miền rồi so sanh các đại lượng thống kê điểm ảnh của các miền đó.

Ta sử dụng thống kê tỷ lệ số điểm ảnh thay đổi trên toàn bộ khung hình, sử dụng một giá trị d là ngưỡng sai khác được tính giữa hai điểm ảnh tương ứng

Gọi S là tập các điểm ảnh có độ sai khác lớn hơn d:

$$S = \{(x,y)\setminus |f_1(x,y) - f_2(x,y)| > d\} \quad (1\text{-}30)$$

Độ sai khác giữa hai khung hình được tính bằng tỷ lệ các điểm ảnh có độ chệnh lệch lớn hơn d.

$$D(f_1, f_2) = \frac{S.\,count}{X * Y} \quad (1\text{-}31)$$

Chúng ta có thể sử dụng cách khác là dùng các đại lượng thống kê cho từng miền, như biểu đồ chẳng hạn. Phương pháp này có khá nhiều sai sót trong phát hiện cảnh phim.

CHƯƠNG 2: CĂN CHỈNH ẢNH DỰA TRÊN ĐỐI SÁNH ĐẶC TRƯNG

2.1 Giới thiệu căn chỉnh ảnh

Thuật toán căn chỉnh hình ảnh là một trong số lâu đời nhất và được sử dụng rộng rãi nhất trong thị giác máy tính. Căn chỉnh hình ảnh tốc độ khung hình được sử dụng trong mọi máy quay phim có tính năng "ổn định hình ảnh".

Một ví dụ ban đầu của thuật toán căn chỉnh hình ảnh được sử dụng rộng rãi vì đó là bản dịch dựa trên bản vá kỹ thuật căn chỉnh (dòng quang) được phát triển bởi Lucas và Kanade (1981). Các biến của thuật toán này được sử dụng trong hầu hết các chương trình nén video bù chuyển động như MPEG và H.263 (Lê Gall 1991). Các thuật toán ước lượng chuyển động tham số tương đồng đã tìm thấy rất nhiều về các ứng dụng, bao gồm tóm tắt video, ổn định video và nén video. Các thuật toán đăng ký hình ảnh tinh vi hơn cũng đã được phát triển để chụp ảnh y tế và viễn thám cho một số khảo sát trước đây về kỹ thuật căn chỉnh hình ảnh.

Trong phim ảnh, các máy ảnh đặc biệt đã được phát triển vào đầu thế kỷ để chụp ảnh toàn cảnh góc cực rộng, thường bằng cách phơi phim qua khe dọc khi máy quay trên trục của nó (Meehan 1990). Vào giữa những năm 1990, các kỹ thuật căn chỉnh hình ảnh bắt đầu được áp dụng để xây dựng ảnh toàn cảnh liền mạch với góc rộng từ các máy ảnh cầm tay thông thường. Một công việc gần đây trong lĩnh vực này đã giải quyết nhu cầu sự tính toán sắp xếp thống nhất toàn cầu, việc loại bỏ bóng mờ do thị sai, chuyển động của đối tượng và đối phó với các mức phơi nhiễm khác nhau (Benosman và Kang 2001). Những

kỹ thuật này đã tạo ra một số lượng lớn các sản phẩm ở khâu thương mại, trong đó đánh giá và so sánh có thể được tìm thấy trên Web.

Mặc dù hầu hết các kỹ thuật trên hoạt động bằng cách giảm thiểu trực tiếp sự khác biệt giữa pixel và pixel, một lớp thuật toán khác hoạt động bằng cách trích xuất một tập hợp các tính năng thưa thớt và sau đó khớp với các tính năng này với nhau. Phương pháp tiếp cận dựa trên tính năng có lợi thế là mạnh mẽ hơn đối với chuyển động cảnh và có khả năng nhanh hơn, nếu được thực hiện đúng cách. Tuy nhiên, ưu điểm lớn nhất của chúng là khả năng nhận diện toàn cảnh tức là, để tự động khám phá các mối quan hệ kề (chồng chéo) giữa một bộ ảnh không có thứ tự, điều này làm cho chúng phù hợp lý tưởng để ghép ảnh toàn cảnh tự động hoàn toàn được chụp bởi người dùng thông thường (Brown và Lowe 2003).

2.2 Các phép biến đổi đồ họa

Trước khi chúng ta có thể đăng ký và căn chỉnh hình ảnh, chúng ta cần thiết lập các mối quan hệ toán học ánh xạ tọa độ pixel từ hình ảnh này sang hình ảnh khác. Một loạt các mô hình chuyển động tham số như vậy có thể, từ các phép biến đổi 2D đơn giản, đến các mô hình phối cảnh phẳng, xoay camera 3D, ống kính biến dạng và ánh xạ tới các bề mặt không phẳng

2.2.1 Các phép biến đổi hình học 2 chiều

2.2.1.1 Phép biến đổi Affine (Affine Transformations)

Phép biến đổi Affine là phép biến đổi tuyến tính tọa độ điểm đặc trưng của đối tượng thành tập tương ứng các điểm mới để tạo ra các hiệu ứng cho toàn đối tượng.

Ví dụ: phép biến đổi tọa độ với chỉ 2 điểm đầu cuối của đoạn thẳng tạo thành 2 điểm mới mà khi nối chúng với nhau tạo thành đoạn thẳng mới. Các

điểm nằm trên đoạn thẳng sẽ có kết quả là điểm nằm trên đoạn thẳng mới với cùng phép biến đổi thông qua phép nội suy.

2.2.1.2 Các phép biến đổi đối tượng

Các đối tượng phẳng trong đồ hoạ 2 chiều mô tả tập các điểm phẳng. Điểm trong đồ hoạ 2 chiều biểu diễn thông qua toạ độ, viết dưới dạng ma trận gọi là vectơ vị trí.

Có 2 dạng biểu diễn:

Một hàng và 2 cột: [x y]

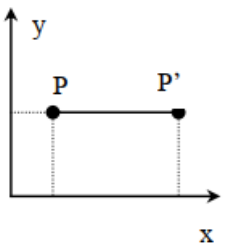
Hai hàng và 1 cột: $\begin{bmatrix} x \\ y \end{bmatrix}$

Trong giáo trình chúng ta chọn biểu diễn điểm là ma trận hàng (một hàng và 2 cột). Tập các điểm được lưu trữ trong máy tính sẽ được viết dưới dạng ma trận vị trí của chúng. Chúng có thể là đường thẳng, đường cong, ảnh....thật dễ dàng kiểm soát các đối tượng thông qua các phép biến đổi chúng, thực chất các phép biến đổi đồ hoạ này được mô tả dưới dạng các ma trận.

2.2.1.2.1 Phép biến đổi vị trí

Giả sử ta có điểm P = [x y] trong mặt phẳng với [x y] là vectơ vị trí của P, kí hiệu là [P]. Gọi ma trận T là ma trận biến đổi sẽ có dạng:

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad (2\text{-}1)$$



Hình 2.1 Phép biến đổi vị trí

Ta có điểm P sau phép biến đổi thành P' có giá trị [x' y']. Thực thi phép biến đổi đúng trên 1 điểm ảnh sẽ đúng với toàn bộ đối tượng.

$$[\mathbf{X}] * [\mathbf{T}] = [\mathbf{x\ y}] * \begin{bmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{bmatrix} = [(\mathbf{ax} + \mathbf{cy})(\mathbf{bx} + \mathbf{dy})] = [\mathbf{x}'\ \mathbf{y}'] \quad (2\text{-}2)$$

Hay ta có: x' = ax + cy

y' = bx + dy

Xét ma trận biến đổi T:

Phép bất biến:

- Khi đó: a = d =1 và b = c = 0 và ma trận cho phép bất biến là:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2\text{-}3)$$

$$T[X] * [T] = [x\ \ y] * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x\ \ y] = [x'\ y'] \quad (2\text{-}4)$$

Vậy x'=x và y = y' hay là P' = P chứng tỏ bất biến qua phép biến đổi.

Phép biến đổi tỷ lệ (Scaling)

- Nếu d=1 và b = c = 0 thì ma trận biến đổi là

$$\mathbf{T} = \begin{bmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (2\text{-}5)$$

x' = ax

y' = y

$$[X] * [T] = [x\ \ y] * \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = [(ax)\ \ y] = [x'\ y'] \quad (2\text{-}6)$$

P’ dịch chuyển theo trục x với tỷ lệ a xác định.

- Nếu b = c =0 thì ma trận biến đổi là:

$$\mathbf{T} = \begin{bmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{0} & \mathbf{d} \end{bmatrix} \tag{2-7}$$

$$[\mathbf{X}] * [\mathbf{T}] = [\mathbf{x}\ \mathbf{y}] * \begin{bmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{0} & \mathbf{d} \end{bmatrix} = [\mathbf{ax}\ \ \mathbf{dy}] = [\mathbf{x}'\ \mathbf{y}'] \tag{2-8}$$

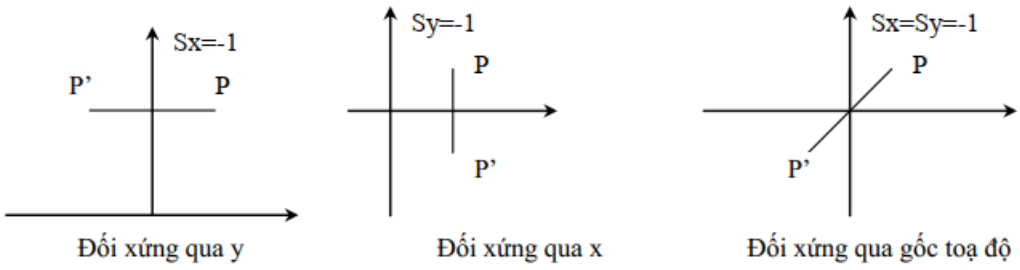
Hay tổng quát hơn gọi Sx, Sy lần lượt là tỷ lệ theo trục x và trục y, thì ma trận tỷ lệ sẽ là:

$$\mathrm{T} = \begin{bmatrix} \mathrm{Sx} & 0 \\ 0 & \mathrm{Sy} \end{bmatrix} \tag{2-9}$$

Khi Sx, Sy >1 gọi là phép phóng to

Khi Sx, Sy <1 gọi là phép thu nhỏ

- Các trường hợp đặc biệt:



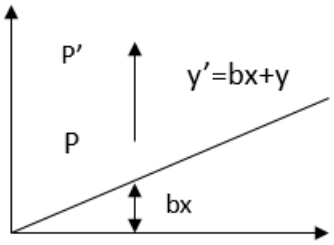
Hình 2.2 Các phép đối xứng trên 2D

Phép biến dạng

Khi a = d = 1 thì toạ độ của P’ phụ thuộc vào thay đổi của b và c

- Xét c=0

$$[\mathrm{X}] * [\mathrm{T}] = [\mathrm{x}\ \mathrm{y}] * \begin{bmatrix} 1 & \mathrm{b} \\ 0 & 1 \end{bmatrix} = [\mathrm{x} \quad \mathrm{bx} + \mathrm{y}] = [\mathrm{x}'\ \mathrm{y}'] \tag{2-10}$$

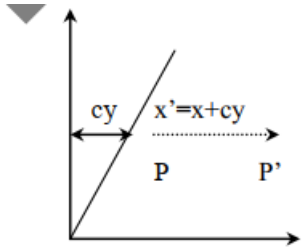


Hình 2.3 Phép biến dạng theo trục oy

Có P' không thay đổi giá trị toạ độ x, còn y' thay đổi phụ thuộc vào cả b và x

- Xét b=0

$$[X] * [T] = [x \;\; y] * \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = [x + cy \quad y] = [x' \; y'] \tag{2-11}$$

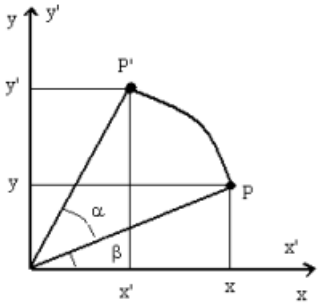


Hình 2.4 Phép biến dạng theo trục ox

Chú ý: điểm gốc toạ độ P[0 0] bất biến với mọi phép biến đổi

Phép quay

Có α>0 ngược chiều kim đồng hồ



Hình 2.5 Phép quay trên 2D

Ta có:

$$[X'] = [X] * [T] = [(x\cos\alpha - y\sin\alpha)\ (x\sin\alpha + y\cos\alpha) \quad (2\text{-}12)$$

Vậy T tổng quát khi quay đối tượng quanh gốc toạ độ 1 góc α bất kỳ là:

$$T = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \quad (2\text{-}13)$$

2.2.1.2.2 Phép biến đổi tổng hợp

Phương pháp biến đổi sử dụng phép nhân ma trận với toạ độ điểm thông qua các vectơ vị trí thật sự hiệu quả và đem lại công cụ mạnh về đồ hoạ cho người sử dụng. Nhưng thực tế các thao tác thường cần không chỉ một mà nhiều phép biến đổi khác nhau. Ta có phép hoán vị khi nhân ma trận là không thực hiện nhưng khả năng tổ hợp các phép nhân lại cho phép tạo ra một ma trận biến đổi duy nhất. Làm giảm bớt đáng kể khối lượng tính toán trong quá trình biến đổi, làm tăng tốc các chương trình ứng dụng và tạo điều kiện cho việcquản lý các biến đổi trong ứng dụng.

Phép xoay quanh một điểm gốc (pivotal point)

- Thực hiện phép tịnh tiến tất cả các điểm theo vector $(-x_c,-y_c)$
- Xoay quanh gốc trục tọa độ

- Thực hiện phép tịnh tiến tất cả các điểm về vị tríban đầu theo vector (x_c,y_c)

2.2.1.3 Tọa đồ đồng nhất và các phép biến đổi

2.2.1.3.1 Tọa độ đồng nhất

Thế nào là phương pháp biểu diễn toạ độ đồng nhất ? là phương pháp biểu diễn mở rộng thông qua toạ độ đồng nhất của các vectơ vị trí không đồng nhất [x y] là ứng dụng của phép biến đổi hình học mà ở đó toạ độ điểm được mô tả dưới ma trận [x* y* h] với x=x*/h, y=y*/h có h là một số thực tuỳ ý. Vậy một vectơ vị trí [xy] bất kỳ trên mặt phẳng xoay bằng tập vô số các điểm đồng nhất [hx hyh].

Phương pháp đưa ra cái nhìn hợp nhất của các phép biến đổi dưới phép nhân ma trận, hỗ trợ cho việc xử lý bằng cả phần cứng và phần mềm. Cho phép kết hợp với cả các phép biến đổi đặc biệt không tuyến tính khác (non-affine) như: Phép chiếu phối cảnh (Perspective projections), uốn (bends), vuốt (tapers)...Kết hợp các các phép biến đổi tạo thành ma trận tích đơn giản duy nhất. Tránh nhầm lẫn về thứ tự của các phép nhân khi sử dụng.

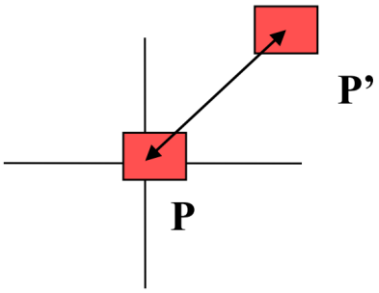
2.2.1.3.2 Phép biến đổi với toạ độ đồng nhất

$$[T] = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix} \qquad (2\text{-}14)$$

Ma trận biến đổi đồng nhất

Phép tịnh tiến

- Phép tịnh tiến dùng để dịch chuyển đối tượng từ vị trí này sang vị trí khác.



Hình 2.6 Phép tịnh tiến

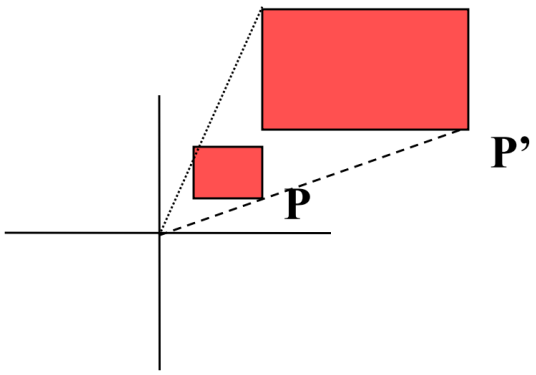
Có a=d=1 và b=c=0

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} \quad (2\text{-}15)$$

$$[x' \; y' \; 1] = [x \quad y \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = [x + m \quad y + n \quad 1] \quad (2\text{-}16)$$

Chú ý: trong mặt phẳng toạ độ, mọi điểm kể cả gốc toạ độ đều có thể biến đổi.

Phép tỷ lệ

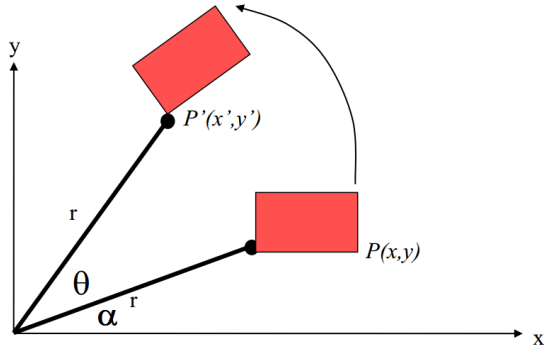


Hình 2.7 Phép tỷ lệ

$$[T] = \begin{bmatrix} S1 & 0 & 0 \\ 0 & S2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}17)$$

$$[x'\ y'\ 1] = [x\ \ y\ \ 1]\begin{bmatrix} S1 & 0 & 0 \\ 0 & S2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [x.S1 \quad y.S2 \quad 1] \tag{2-18}$$

Phép xoay



Hình 2.8 Phép xoay

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-19}$$

$$[x'\ y'\ 1] = [x\ \ y\ \ 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-20}$$

$$= [x.\cos\theta - y.\sin\theta \quad x.\sin\theta + y.\cos\theta \quad 1]$$

2.2.2 Phép biến đổi hình học 3D

Phép biến đổi hình học 3D là sự mở rộng của phép biến đổi 2D. Tương tự nó cũng có các phép: tịnh tiến, tỷ lệ, biến dạng và quay. Phức tạp nhất là

phép quay, nên chúng ta khảo sát lần lượt từ đơn giản đến phức tạp: quay đối tượng quanh các trục toạ độ, quanh 1 trục song song với trục toạ độ, quanh 1 trục bất kỳ....Do khảo sát các phép biến đổi affine với biểu diễn dạng ma trận đồng nhất (4x4 với 3D) nên công việc khá đơn giản và nhất quán. Lưu ý phép tịnh tiến và quay có chung thuộc tính là: sau khi biến đổi hình dạng và kích thước của đối tượng không thay đổi mà chúng chỉ bị thay đổi vị trí và định hướng trong không gian.

2.2.2.1 Biểu diễn điểm trong không gian 3 chiều

[x* y* z* h] hay[x*/h y*/hz*/h 1]

Viết gọn hơn:[x y z 1]

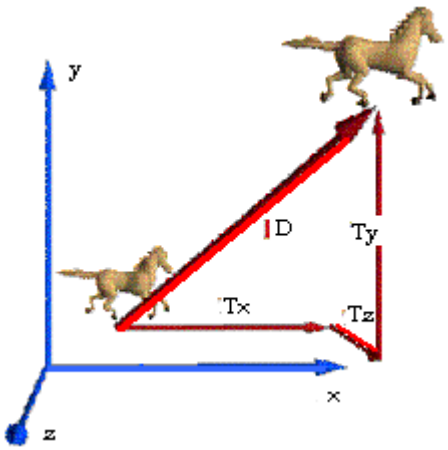
Ma trận biến đổi tổng quát trong không gian 3D với tọa độ đồng nhất (4x4):

$$[T] = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix} \qquad (2\text{-}21)$$

2.2.2.2 Phép tịnh tiến

Đây là phép biến đổi đơn giản nhất, mở rộng từ phép biến đổi trong không gian 2D ta có:

$$[T(dx, dy, dz)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{bmatrix} \qquad (2\text{-}22)$$

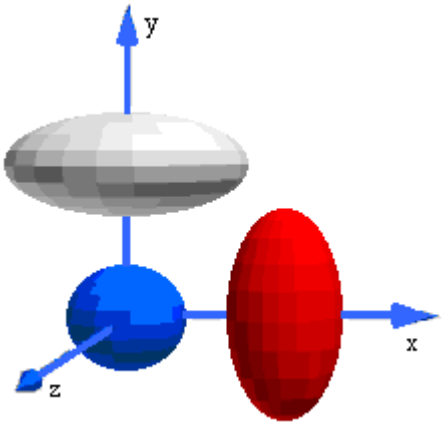


Hình 2.9 Phép tịnh tiến trên 3D

2.2.2.3 Phép tỷ lệ

Tương tự trong 2D ta có phép tỉ lệ trong 3D là :

$$[Ts] = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}23)$$



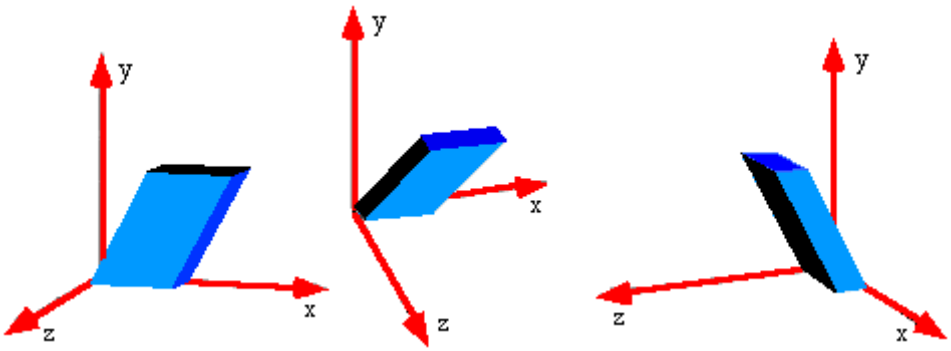
Hình 2.10 Phép tỷ lệ

2.2.2.4 Phép biến dạng

- Ta có tất cả các phần tử nằm trên đường chéo chính bằng 1.
- Các phần tử chiếu và tịnh tiến bằng 0.

$$[x' \quad y' \quad z'] = \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2\text{-}24)$$

$$= [x + yd + gz \quad bx + y + iz \quad cx + fy + z \quad 1]$$

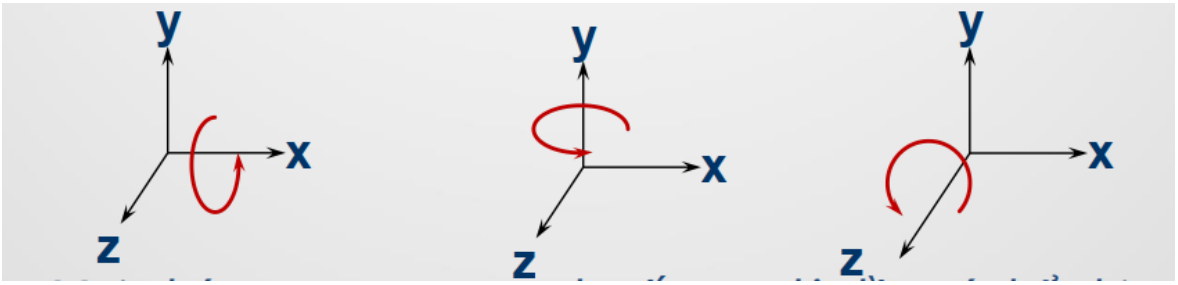


Hình 2.11 Các phép biến dạng trên 3D

2.2.2.5 Phép quay 3 chiều

2.2.2.5.1 Quay quanh các trục độ

- Trong 2D, phép xoay chỉ được thực hiện xung quanh gốc tọa độ
- Trong 3D, một đối tượng có thể xoay theo 3 trục: trục x, trục y, trục z
- Xoay theo chiều dương là ngược chiều kim đồng hồ và xoay theo chiều âm là theo chiều kim đồng hồ

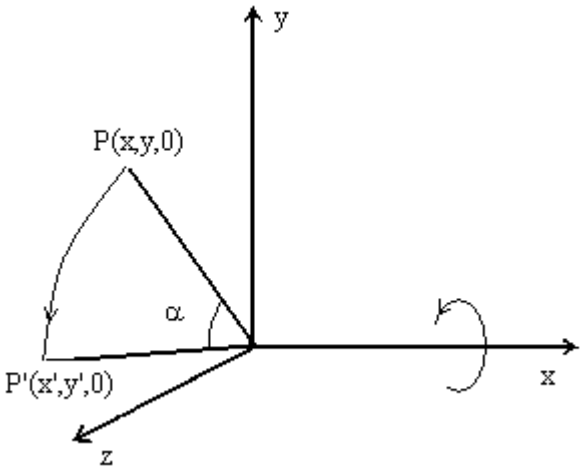


Hình 2.12 Xác định góc quay dương trên 3 trục toạ độ

Chú ý: Mọi phép xoay xung quanh gốc tọa độ đều có thể chia

thành phép quay quanh trục x, sau đó quay quanh trục y và sau đó quay quanh trục z (Định lý Euler)

Quay quanh trục ox



Hình 2.13 Quay quanh trục ox

$$[\mathrm{Tx}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\emptyset & \sin\emptyset & 0 \\ 0 & -\sin\emptyset & \cos\emptyset & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}25)$$

Quay quanh trục oy

$$[\mathrm{Ty}] = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}26)$$

Quay quanh trục oz

$$[\mathrm{Tz}] = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}27)$$

2.3 Căn chỉnh ảnh dựa trên đối sánh đặc trưng

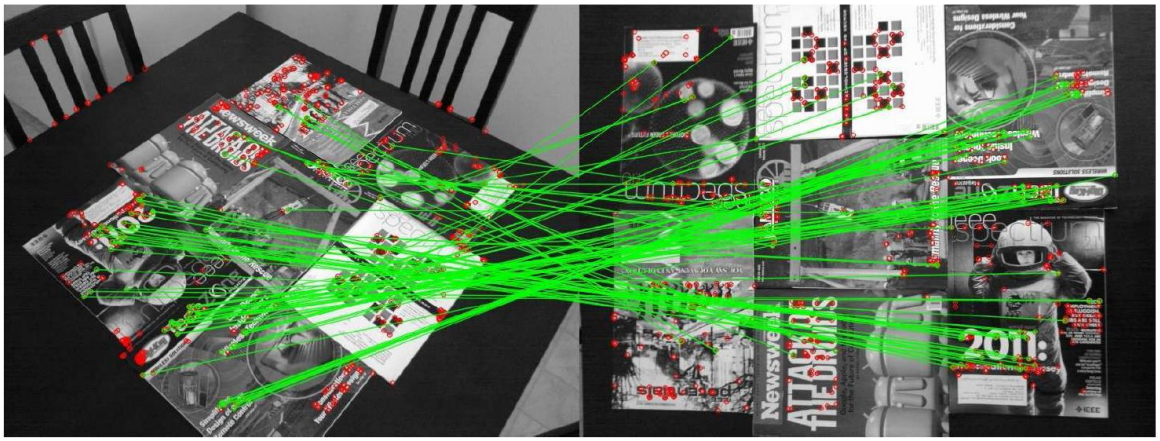
2.3.1 Thuật toán ORB

Đối sánh thuộc tính là cơ sở của nhiều vấn đề thị giác máy tính, chẳng hạn như nhận dạng đối tượng hoặc cấu trúc từ sự chuyển động. Phương pháp hiện nay dựa trên việc mô tả gây tốn kém về nhận dạng và đối sánh. Trong báo báo này ta nghiên cứu một mô tả nhị phân dựa trên BRIEF gọi là ORB, đó là vòng xoay bất biến và có khả năng chống nhiễu. Các thí nghiệm đã chứng mình rằng ORB đứng ở vị trí thứ hai về độ lớn và nhanh hơn SIFT trong khi nó hoạt tốt trong nhiều tình huống. Hiệu quả được thử nghiệm trên một số ứng dụng thế giới thực, bao gồm phát hiện đối tượng và theo dõi trên điện thoại thông minh.

Thuộc tính ORB đề xuất được xây dựng trên các bộ dò keypoint nổi tiếng FAST và gần đây là bộ mô tả BRIEF và gọi là ORB (Oriented Fast and Rotated BRIEF). Cả hai công nghệ này rất hấp dẫn vì cách thực hiện tốt của nó và chi phí thấp. Trong luận văn này, ta giải quyết một số hạn chế của những công cụ liên quan đến SIFT, đáng kể nhất là thiếu bất biến trong phép quay BRIEF. Cụ thể là:

- Thêm vào FAST và gán hướng chính xác cho các thành phần của FAST
- Các tính toán hiệu quả thuộc tính hướng BRIEF.
- Phân tích phương sai và tương quan của thuộc tính hướng BRIEF.

- Một phương pháp học giảm tương ứng thuộc tính BRIEF dưới bất biến quay, dẫn đến thực hiện tốt hơn các ứng dụng láng giềng gần.



Hình 2.14 Ví dụ về kết quả đối sánh ảnh sử dụng thuật toán ORB

Để hiểu hơn về ORB ta thực hiện các thí nghiệm kiểm tra các thuộc tính của ORB liên quan đến SIFT và SURF, cả khả năng đối sánh thô và hiệu suất trong các ứng dụng đối sánh hình ảnh. Kiểm nghiệm hiệu quả của ORB bằng cách thực hiện một ứng dụng theo dõi trên bản vá trên điện thoại thông minh. Một lợi ích nữa của ORB là nó là miễn phí trong khi SIFT và SURF thì không.

2.3.1.1 Công trình nghiên cứu liên quan

Keypoint FAST và các biến thể của nó là phương pháp lựa chọn việc tìm kiếm keypoint trong các hệ thống thời gian thực phù hợp với thuộc tính thị giác, ví dụ theo dõi vết trên bản đồ. Để tìm góchợp lý cho keypoint cần tăng cường tỉ lệ của lược đồ kim tự tháp và trong trường hợp này bộ lọc góc Harris lọc loại bỏ đường biên.

Nhiều phép dò keypoint bao gồm một thao tác gán hướng (SIFT và SURF là hai ví dụ nổi bật) nhưng FAST thì không. Có nhiều cách khác nhau để mô tả hướng của một keypoint và có nhiều cách liên quan đến biểu đồ của

các tính toán gradient, ví dụ như trong SIFT và sự tương đối của mô hình khối trong SURF. Những phương pháp này là một trong hai yêu cầu tính toánhoặc trong trường hợp của SURF thì tính toán xấp xỉ kém. Các tài liệu tham khảo của Rosin đưa ra một phân tích khác nhau để đo hướng của các góc và sử dụng kỹ thuật trọng tâm. Không giống như các thao tác gán hướng trong SIFT, có thể có nhiều giá trị trên một keypoint duy nhất, các thao tác trọng tâm của Rosin cho một kết quả duy nhất.

Mô tả BRIEF là một mô tả thuộc tính mới có sử dụng các kiểm tra nhị phân đơn giản giữa các điểm ảnh trong hình ảnh của một bản vá nhẵn. Hiệu quả của nó tương tự như SIFT ở nhiều khía cạnh, trong đó rất tốt với ánh sáng, làm mờ và hướng nhìn méo. Tuy nhiên, nó rất nhạy cảm với chuyển động quay trong mặt phẳng.

BRIEF phát triển từ nghiên cứu có sử dụng các kiểm tra nhị phân để huấn luyện một tập hợp các cây phân loại. Một tập huấn luyện 500 hoặc nhiều keypoint tiêu biểu, cây có thể được dùng trả về dấu hiệu cho bất kỳ keypoint đặc biệt nào. Bằng cách tương tự ta tìm kiếm các kiểm tra nhị phân ít nhạy với hướng. Lớp phương thức kinh điển để tìm kiếm tra không tương ứng là PCA, ví dụ như nó hiển thị PCA cho SIFT có thể giúp bỏ một lượng lớn thông tin dư thừa. Tuy nhiên không gian kiểm tra nhị phân là quá lớn cho thực hiện PCA và tìm kiếm vét cạn được sử dụng thay thế.

2.3.1.2 FAST: hướng của keypoint FAST

Thuộc tính FAST mở rộng được sử dụng vì đặc tính tính toán của nó. Tuy nhiên thuộc tính FAST không có thành phần hướng. Trong phần này ta nghiên cứu thêm về tính toán hướng cách hiệu quả cho FAST.

2.3.1.2.1 Bộ dò FAST

Ta bắt đầu dò điểm FAST trong bức ảnh. FAST lấy một tham số là cường độ giữa điểm tâm và những điểm trong hình tròn của tâm.

Ta sử dụng FAST-9 (bán kính tròn là 9) có kết quả thực hiện tốt nhất

FAST không tạo ra phép đo góc mà ta phải tìm góc mà nó đáp ứng mạnh dọc theo các biên. Ta tiến hành nhúng góc đo Harris để đặt các keypointFAST. Với đích là N keypoint đầu tiên chúng ta thiết lập các ngưỡng thấp, đủ để nhận được nhiều hơn N keypoint, sau đó đặt chúng theo phép đo Harris và chọn điểm đầu trong N điểm.

FAST không sinh ra các thuộc tính đa tỉ lệ mà sử dụng một kim tự tháp tỉ lệ của hình ảnh và tạo ra các thuộc tính FAST (được lọc bởi Harris) ở mỗi cấp của kim tự tháp.

2.3.1.2.2 Hướng của cường độ trọng tâm

Cường độ trọng tâm được giả định là cường độ của một góc được tương đương từ trung tâm của nó và vector này có thể được sử dụng để quy cho một hướng. Rosin định nghĩa một bản vá của phép quay như sau:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \qquad (2\text{-}28)$$

và với những phép quay này chúng ta có thể tìm thấy các trọng tâm:

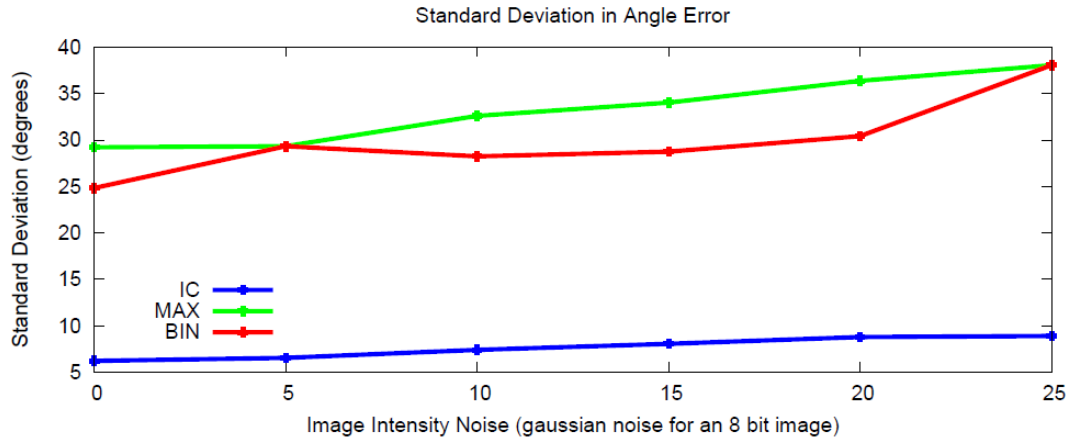
$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \qquad (2\text{-}29)$$

Ta cấu trúc 1 vector từ tâm O của góc tới các trọng tâm O_c. Hướng của các bản vá đơn giản là:

$$\theta = a\tan 2\,(m_{01}, m_{10}) \qquad (2\text{-}30)$$

Ở đây atan2 là phiên bản góc phần tư của arctan. Rosin đề cập lấy loại góc cho dù góc tối hoặc có ánh sáng. Tuy nhiên, mục đích của ta là có thể bỏ qua điều này miễn là biện pháp đo góc không liên quan đến loại góc.

Để cải tiến phép quay bất biến phép đo này ta chắc chắn phép quay được tính toán với X và Y còn lại trong phạm vi hình tròn bán kính r. Ta quan sát thử nghiệm và chọn r là kích thước bản vá, vì thế x và y chạy từ [-r, r]. Vì thế C tiếp cận O, phép đo trở thành không ổn định. Với góc Fast FAST ta thấy rằng nó hiếm khi gặp trường hợp này.



Hình 2.15 Đồ thị cường độ nhiễu của ảnh

2.3.1.3 Phép quay:rBRIEF

Phần báo cáo này sẽ giới thiệu về một mô tả hướng BRIEF, sự tính toán hiệu quả của nó và nhược điểm của nó là kém hiệu quả với phép xoay. Để giải quyết vấn đề này ta phải tìm các kiểm tra nhị phân ít tương quan để rBRIEF mô tả tốt hơn, sau đó so sánh với SIFT và SURF.

2.3.1.3.1 Hiệu quả của phép quay BRIEF

Tổng quan về BRIEF

Bộ mô tả BRIEFlà một mô tả chuỗi bit của một bản vá hình ảnh được xây dựng từ một tập hợp các kiểm tra cường độ nhị phân. Với một bản vá hình ảnh được làm mịn p. Một thử nghiệm nhị phân t được xác định bởi:

$$\tau(p;x;y) \coloneqq \begin{cases} 1: p(x) < p(y) \\ 0: p(x) \geq p(y) \end{cases} \quad (2\text{-}31)$$

Trong đó p(x) là cường độ của p tại điểm x. Thuộc tính được định nghĩa như một vector của n kiểm tra nhị phân

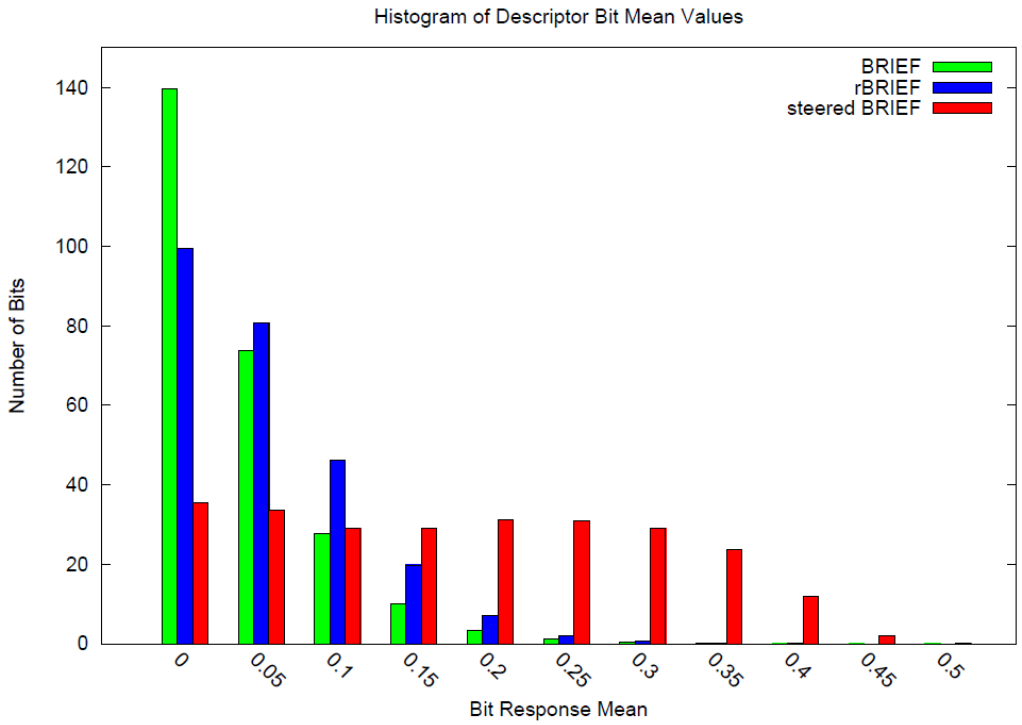
$$f_n(p) \coloneqq \sum_{1 \leq i \leq n} 2^{i-1} \tau(p;x_i;y_i) \quad (2\text{-}32)$$

Ở đây ta sử dụng một trong những cách biểu diễn tốt nhất là phân phối Gaussian quanh trung tâm của các bản vá và chọn chiều dài vector n = 256.

Điều quan trọng là làm mịn hình ảnh trước khi thực hiện các kiểm tra. Trong việc thực hiện này, việc làm mịn dùng một ảnh tích hợp và mỗi điểm kiểm tra là một cửa sổ con 5×5 của bản vá 31x31.

Hướng BRIEF

Ta giả sử rằng BRIEF là bất biến trong chuyển động quay mặt phẳng. Việc thực hiện đối sánh của BRIEF giảm mạnh trong mặt phẳng quay một vài độ (xem Hình 2.23). Calonder đề nghị tính toán một mô tả BRIEF cho một tập hợp các phép quay và biến dạng cong của mỗi miếng vá



Hình 2.16 Sự phân phối cân bằng các vector thuộc tính

Nhưng giải pháp này rõ ràng là tốn kém. Một phương pháp hiệu quả hơn là hướng BRIEF theo hướng của keypoint. Đối với bất kỳ thuộc tính nào của bộ n kiểm tra nhị phân ở vị trí (xi, yi), định nghĩa các ma trận 2×n

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix} \quad (2\text{-}33)$$

Sử dụng bản vá hướng θ và ma trận xoay tương ứng R_θ ta xây dựng một "hướng" phiên bản S_θ của S:

$$S_\theta = R_\theta S \quad (2\text{-}34)$$

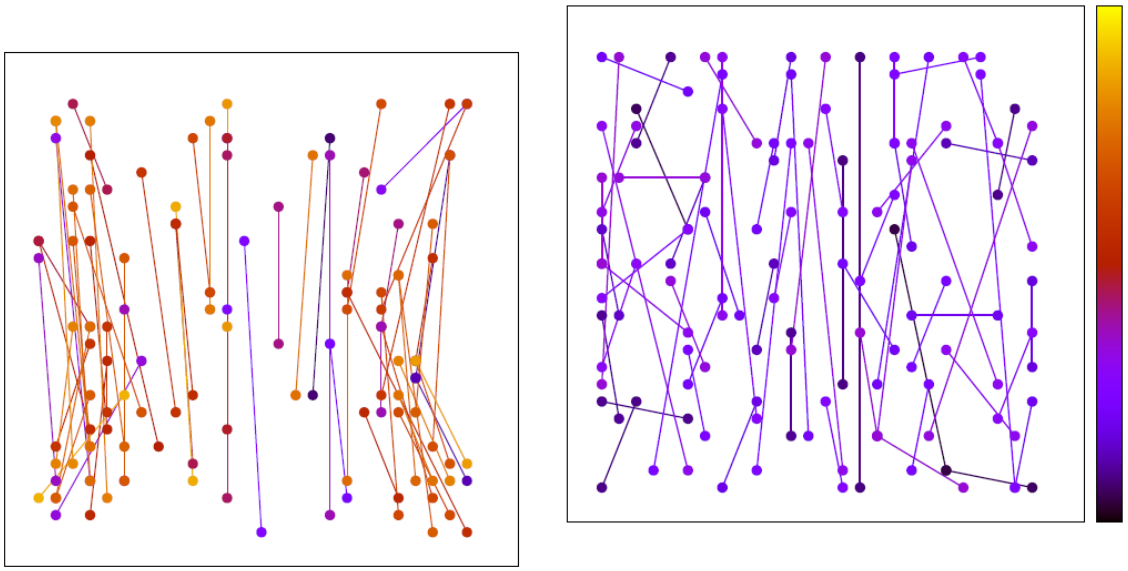
Bây giờ các thao tác hướng BRIEF trở thành

$$g_n(p, \theta) := f_n(p) | (x_i, y_1) \epsilon S_\theta \quad (2\text{-}35)$$

Sau đó rời rạc góc để có số gia 2π/30 (12 độ) và xây dựng một bảng tra cứu của BRIEF. Hướng keypoint θ là nhất quán trên các hướng nhìn, các tập điểm chính xác của điểm Sθ sẽ được dùng để tính toán bộ mô tả của nó

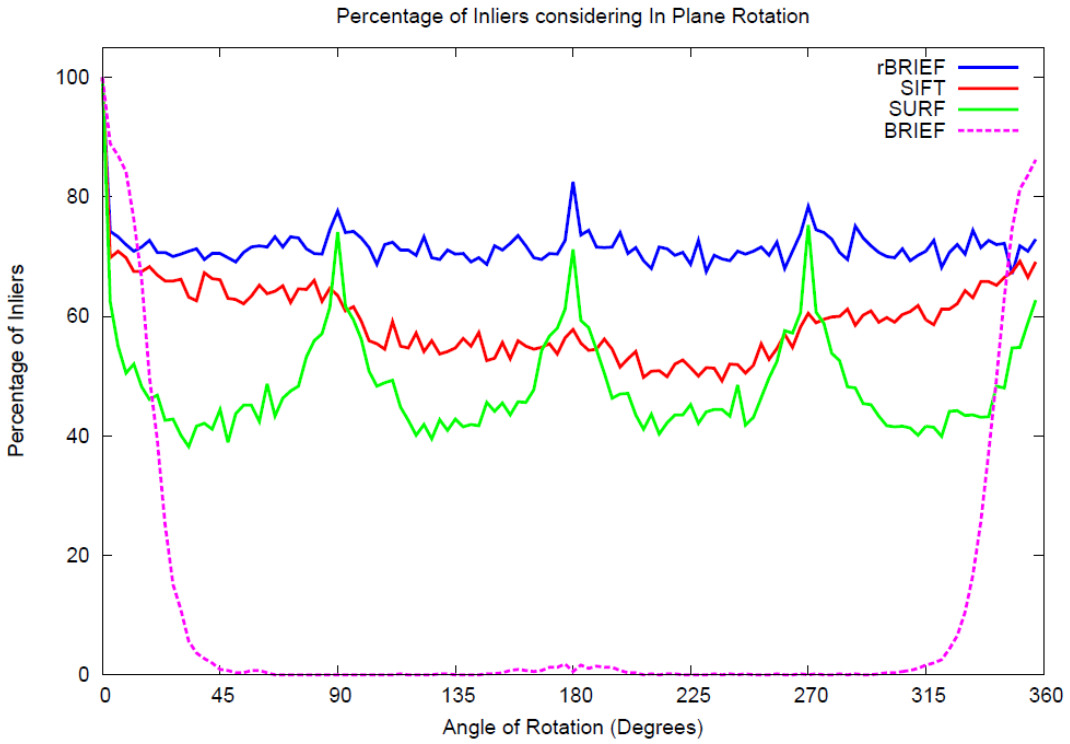
2.3.1.4 Đánh giá

Ta đánh giá sự kết hợp của oFAST và rBRIEF gọi là ORB sử dụng hai bộ dữ liệu: hình ảnh bị xoay trong mặt phẳng và thêm nhiễu Gaussian và một tập dữ liệu thật của hình ảnh phẳng với vân ảnh được chụp từ nhiều hướng nhìn khác nhau. Đối với mỗi hình ảnh tham khảo, chúng ta tính toán các keypoint oFAST và các thuộc tính rBRIEF nhắm mục tiêu 500 keypoint cho mỗi hình ảnh. Đối với mỗi hình ảnh thử nghiệm (xoay hoặc thay đổi hướng nhìn), sau đó thực hiện đối sánh để tìm điểm tương đồng tốt nhất.



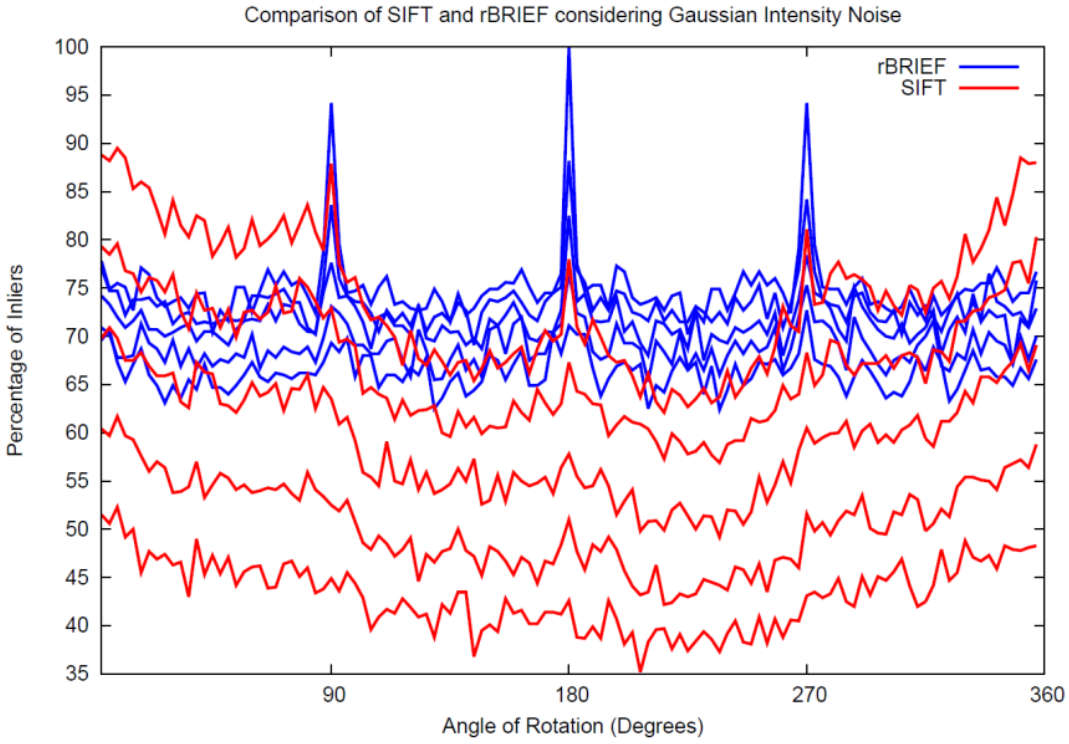
Hình 2.17 Xác định tập con các điểm kiểm tra nhị phân

Các kết quả được đưa ra trong giới hạn tỷ lệ phần trăm các đối sánh chính xác và khắc phục được những góc quay

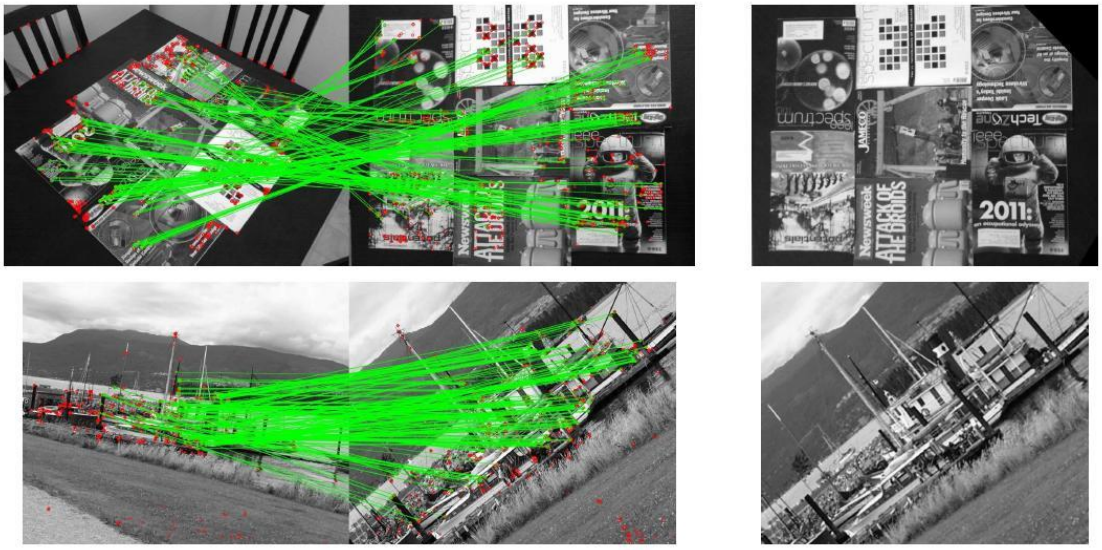


Hình 2.18 Hiệu suất đối sánh của SIFT, SURF, BRIEF với FAST và ORB

ORB chống lại nhiễu ảnh Gaussian, không như SIFT. Nếu chúng ta xét tính hiệu quả với nhiễu, SIFT là 10% với mỗi lần tăng nhiễu thêm 5. ORB cũng giảm nhưng với tốc độ thấp hơn nhiều (Hình 2.20).



Hình 2.19 Thao tác đối sánh có nhiễu cho SIFT và rBRIEF



Hình 2.20 Ví dụ thực tế về đối sánh ảnh ORB

2.3.2 Tìm ma trận tương đồng

2.3.2.1 Vài nét về Homography

Trong toán học, Homography là sự dịch chuyển sử dụng phép chiếu hình học, hay nói cách khác nó là sự kết hợp của cặp điểm trong phép chiếu phối cảnh. Ảnh thực trong không gian ba chiều có thể biến đổi về không gian ảnh bằng phép chiếu thông qua ma trận biến đổi Homography hay còn gọi là ma trận H. Các phép chiếu biến đổi thông qua ma trận Homography không đảm bảo về kích thước và góc của vật được chiếu nhưng lại đảm bảo về tỷ lệ.

Trong lĩnh vực thị giác máy, Homography là một ánh xạ từ mặt phẳng đối tượng đến mặt phẳng ảnh. Ma trận homography thường có liên quan đến các công việc xử lý giữa hai ảnh bất ký và có ứng dụng rất sộng rãi trong các công tác sửa ảnh, ghép ảnh, tính toán sự chuyển động, xoay hay dịch chuyển giữa hai ảnh.

Ta có công thức sau:

HX=sX’

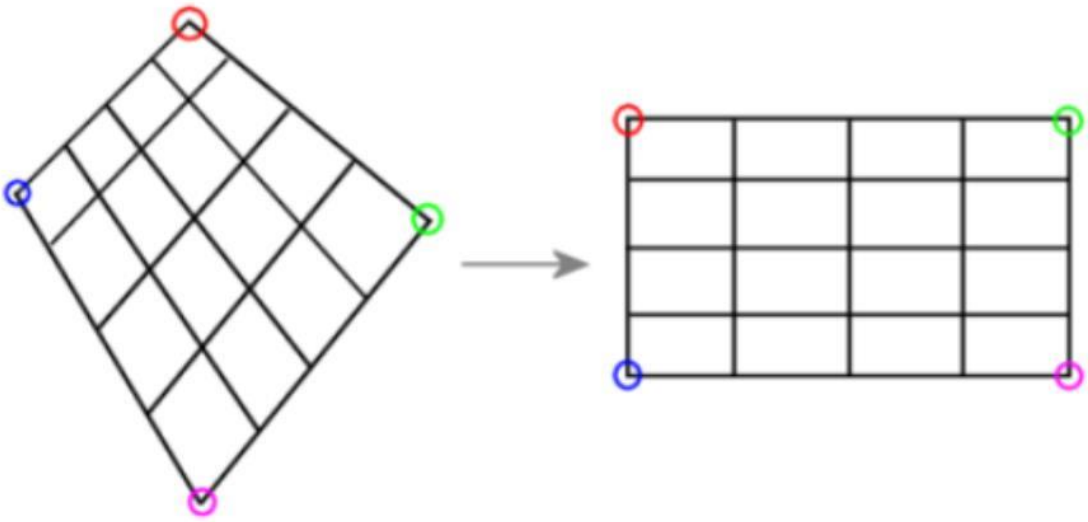
Trong đó:

S là hằng số tỉ lệ của phép chiếu và khác 0

X’ là kết quả của phép ánh xạ

H là ma trận Homography, là một ma trận khả nghịch

Vì H là một ma trận khả nghịch, cho nên trong trường hợp muốn tái tạo ảnh X từ X’, ta chỉ cần xác định được ma trận Homography.



Hình 2.21 Phép chiếu Homography

2.3.2.2 Mô hình chuyển động

2.3.2.2.1 Phân rã giá trị đơn SVD (Singular Value Decomposition)

Phân rã giá trị đơn SVD là phương pháp đại số được sử dụng nhiều trong các bài toán yêu cầu việc tính toán ma trận vốn sẽ cho ra kết quả sai số lớn nếu như sử dụng các phương pháp thông thường như khử Gauss hay phân tích LU.

2.3.2.2.2 Tính ma trận Homography bằng phương pháp Direct Linear Transform

Để tính ma trận Homography từ các cặp điểm tương ứng, người ta dùng phương pháp DLT (Direct Linear Transform), gồm có 2 bước: Đầu tiên, từ các cặp điểm tương ứng, ta chuyển về dạng ma trận Aih = 0. Sau đó, áp dụng phân rã SVD để tính ma trận H.

Phương pháp DLT với các điểm nổi bật được tìm thấy từ thuật toán Harris:

Trong tọa độ không đồng nhất, ta có công thức:

$$c\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}h_1 & h_2 & h_3\\h_4 & h_5 & h_6\\h_7 & h_8 & h_9\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix} \quad (2\text{-}36)$$

Lần lượt chia dòng thứ nhất của công thức trên cho dòng thứ ba và dòng thứ hai cho dòng thứ ba, ta có

$$\begin{aligned}-h_1x-h_2y-h_3+(h_7x+h_8y+h_9)_u=0\\-h_4x-h_5y-h_6+(h_7x+h_8y+h_9)_u=0\end{aligned} \quad (2\text{-}37)$$

Viết dưới dạng ma trận ta có:

$$A_ih=\begin{bmatrix}-x & -y & -1 & 0 & 0 & 0 & ux & uy & u\\0 & 0 & 0 & -x & -y & -1 & vx & vy & v\end{bmatrix}(h_1h_{2\dots}h_9)^T \quad (2\text{-}38)$$

Với mỗi cặp điểm tương ứng, ta có hai biểu thức. Mặt khác, do ma trận là ma trận có bậc tự do (D.O.F) là 8 nên chỉ cần 4 cặp điểm tương ứng là ta có thể xác định được nó.

Áp dụng công thức phân rã SVD cho ma trận [A] ta có:

$$A=U\sum V^T=\sum_{i=1}^{9}s_iu_iv_i^T \quad (2\text{-}39)$$

Với s_i là các giá trị đơn và được sắp xếp nhỏ dần, nên s_9 là giá trị nhỏ nhất. Khi đó, giá trị của h_i bằng giá trị cuối cùng của cột v_i.

Theo lý thuyết, với 4 cặp điểm tương ứng sẽ tìm được một ma trận H với D.O.F bằng 8. Tuy nhiên, trong thực tế các ảnh đầu vào có thể có gốc tọa độ ở góc trái của ảnh, cũng có thể gốc tọa độ nằm ở tâm ảnh. Nếu để tình trạng như vậy sẽ ảnh hưởng đến các kết quả biến đổi về sau như khi nhân ảnh với một hệ số hay các biến đổi tương tự, affin. Do vậy, Hartley và Zisserman [8] đã đưa ra bước chuẩn hóa để đảm bảo rằng kết quả của thuật toán sẽ cho ra kết quả chính xác. Các ảnh cần phải chuẩn hóa bằng phép biến đổi quay và dịch chuyển.

2.3.2.2.3 Thuật toán RANSAC

RANSAC, đại diện cho cụm từ "RANdom SAmple Consensus", tức là"đồng thuận mẫu ngẫu nhiên", là thuật toán khử nhiễu được công bố bởi Fischler và Bolles vào năm 1981.

Ý tưởng chính của RANSAC như sau: Từ tập dữ liệu ban đầu, ta sẽ có hai loại dữ liệu nhiễu và không nhiễu (outlier và inlier), vì thế ta phải đi tính toán để tìm ra mô hình tốt nhất cho tập dữ liệu. Việc tính toán và chọn ra mô hình tốt nhất sẽ được lặp đi lặp lại k lần, với giá trị k được chọn sao cho đủ lớn để đảm báo xác suất p (thường rơi vào giá trị 0.99) của tập dữ liệu mẫu ngẫu nhiên không chứa dữ liệu nhiễu.

Với ma trận Homography được tính từ bốn cặp điểm ngẫu nhiên, ta có d là khoảng cách đo mức độ gần nhau của các cắp điểm đã được so sánh đối chiếu. Với cặp điểm nổi bật tương đồng (x,x') và $d(\vec{a}, \vec{b})$ là khoảng cách của hai vector, ta có công thức khoảng cách như sau:

$$\mathrm{d} = \left(\vec{\mathrm{x}}, \mathrm{H}\overrightarrow{\mathrm{x}'}\right) + \mathrm{d}(\overrightarrow{\mathrm{x}'}, \mathrm{H}\vec{\mathrm{x}}) \qquad (2\text{-}40)$$

Thuật toán chi tiết:

- Khởi tạo số vòng lặp k, ngưỡng distance, max_{inlier} và p = 0.
- for(i = 1:k), thực hiện các bước sau:
- Bước 1: Chọn 4 cặp điểm tương đồng ngẫu nhiên
- Bước 2: Kiểm tra xem các điểm có nằm trên cùng một dường thẳng hay ko.
- Bước 3: Tính ma trận Homography H từ 4 điểm sử dụng phương pháp DLT chuẩn hóa.
- Bước 4: Tính khoảng cách d của các cặp điểm nổi bật tương đồng
- Bước 5: Tính số lượng m các cặp điểm không ngẫu nhiên (inlier) thỏa điều kiện: d_i < distance
- Bước 6: Nếu m > max_{inlier} thi max_{inlier} = m, ma trận Homography H= $H_{curr.}$
- Tiếp tục tính lại ma trận H cho tất cả các cặp điểm tương đồng được coi là không nhiễu (inlier) bằng phương pháp DLT.

CHƯƠNG 3: KỸ THUẬT XÂY DỰNG ỨNG DỤNG

3.1 Giới thiệu OpenCV

Đây là một thư viện mã nguồn mở về thị giác máy tính và học máy. Thư viện được xây dựng để cung cấp nền tảng cho các ứng dụng thị giác máy tính nhằm đẩy mạnh sự phát triển về hàm lượng tri thức máy tính trong các sản phẩm thương mại. Nhờ giấy phép bản quyền BSD và được nhiều công ty lớn hàng đầu thế giới như Google, Yahoo, Microsoft, Intel, IBM,…cùng đóng góp xây dựng thư viện, OpenCV là một trong những công cụ mạnh và được sử dụng rộng rãi trong trường học cũng như các công ty khởi nghiệp.

OpenCV bao gồm nhiều giao diện dành cho C++, C, Python, Java, MATLAB và hỗ trợ cho các hệ điều hành khác nhau như Windows, Linux, Android, MacOS. Trong phiên bản OpenCV 3.1, giao diện sử dụng cho CUDA và OpenCL cũng đã được phát triển hoàn thiện. OpenCV được viết nguyên bản bằng ngôn ngữ C++.

OpenCV có rất nhiều chức năng. Sau đây là những tóm tắt cơ bản về hệ thống các về chức năng của các hàm trong OpenCV

- Image and Video I/O.

Những giao diện này sẽ giúp ta đọc được dự liệu ảnh từ file hoặc trực tiếp từ video. Ta cũng có thể tạo các file ảnh và video với giao diện này.

- Các thuật toán xử lý ảnh và thị giác máy (General computer-vision and image-processing algorithms (mid – and low level APIs)).

Sử dụng những giao diện này, ta có thể thực hành với rất nhiều chuẩn thị giác máy mà không cần phải có mã nguồn của chúng.

- Đồ họa.

Những giao diện này giúp ta viết chữ và vẽ trên hình ảnh. Thêm vào đó những chức năng này được sử dụng nhiều trong ghi nhãn và đánh dấu. Ví dụ nếu ta viết một chương trình cần nhận dạng nhiều đối tượng thì nó sẽ rất có ích cho tạo nhãn ảnh (label image) với kích thước và vị trí.

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

- Nhận diện hình ảnh
- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y tế
- Tìm kiếm và phục hồi hình ảnh/video
- Phim - cấu trúc 3D từ chuyển động
- Nghệ thuật sắp đặt tương tác

Với Window, khi cài đặt OpenCV, nó sẽ copy file OpenCV vào thư mục mà ta chọn. Cách thức lựa chọn trong đường dẫn hệ thống chứa mã nhị phân OpenCV, đăng kí một vài bộ lọc DirectX. Mặc định nó cài đặt đến C:/Program Files/OpenCV/<version>

Trong thư mục OpenCV có chứa một vài thư mục khác. Thư mục docs chứa file văn bản html cho toàn bộ các hàm OpenCV và kiểu dữ liệu. Từ file văn bản này ta có thể làm các ví dụ, ta cũng có thể muốn xem thư mục “samples”. Những file header sẽ cần thiết khi ta dịch chương trình sử dụng OpenCV. Ta có thể xác định file header bằng cách tìm kiếm trong thư mục cài đặt và những thư mục khác những file có dạng *.h, *.hpp. Header dùng cho tất

cả các module trừ HighGUI được tách riêng trong "include". Ta có thể bỏ header trong thư mục "src".

3.2 Phát triển ứng dụng phát hiện phần ảnh sai khác

3.2.1 Phát biểu bài toán

Đưa hai hình ảnh có cấu trúc giống nhau. Ảnh thứ 2 là ảnh gốc được thêm bớt một số nội dung như minh họa trong hình 3.1. Hệ thống tự động tìm ra các phần khác biệt trên 2 ảnh đó và đóng khung lại.

Hình 3.1 Ảnh gốc bên trái và ảnh cần căn chỉnh bên phải (vùng khoanh đỏ là phần khác biệt giữa 2 ảnh)

Để thực hiện yêu cầu trên, cần phải trải qua các bước sau:

Bước 1: Căn chỉnh để hai ảnh về cùng kích thước, và giống nhau nhất về bố cục.

Bước 2. Thực hiện kỹ thuật trừ hai ảnh cho nhau để tìm ra vùng khác biệt

3.2.2 Triển khai sử dụng hàm trong OpenCV

3.2.2.1 Căn chỉnh ảnh

Bước 1: Tiền xử lý

Bước này gồm 2 thao tác cần thực hiện : đọc ảnh và chỉnh xám

Đọc hình ảnh

Đọc 2 ảnh một ảnh mẫu và một ảnh cần được căn chỉnh đã có sẵn trong thư viện OpenCV

```
// Read reference image
string refFilename("images\\or1.png");
cout << "Reading reference image : " << refFilename << endl;
Mat imReference = imread(refFilename);


// Read image to be aligned
string imFilename("images\\mo1.png");
cout << "Reading image to align : " << imFilename << endl;
Mat im = imread(imFilename);
```

Chỉnh xám

Trong OpenCV có thể sử dụng hàm CvtColor để thực hiện chuyển ảnh đầu vào từ ảnh màu qua ảnh xám. Cấu trúc của hàm CvtColor trong OpenCV được trình bày như sau:

```
Mat im1Gray, im2Gray;
cvtColor(im1, im1Gray, CV_BGR2GRAY);
cvtColor(im2, im2Gray, CV_BGR2GRAY);
```

Trong đó:

- im1 là đối số đầu vào
- im1Gray là đối số đầu ra
- CV_BGR2GRAY là code quy định sẽ chuyển đổi từ mã màu nào sang mã màu nào. Ở đây ta sẽ thực hiện việc chuyển đổi từ mã ảnh màu sang ảnh xám.

Hình 3.2 Ảnh thứ 2 đã được căn chỉnh về xám

Bước 2: Phát hiện đặc trưng (Dectect feature)

Chúng ta cần phải tìm các keypoints (feature points) trong mỗi hình ảnh. Ở đây mình sẽ sử dụng ORB detect feature bởi vì SIFT hay SUFT nếu muốn dùng phải trả phí. Lí do đó cho nên đã phải chuyển ảnh về ảnh xám ở bước đầu.

Mặc dù chỉ cần 4 đặc trưng để tính toán đồng nhất, nhưng thông thường hàng trăm đặc trưng được phát hiện trong hai hình ảnh. Vậy nên cần kiểm soát số lượng tính năng bằng cách sử dụng tham số MAX_FEATURE

```
//Ptr<Feature2D> f2d = xfeatures2d::ORB create();
Ptr<Feature2D> orb = ORB::create(MAX_FEATURES);
orb->detectAndCompute(im1Gray, Mat(), keypoints1, descriptors1);
orb->detectAndCompute(im2Gray, Mat(), keypoints2, descriptors2);
```

Bước 3: Đối sánh đặc trưng (Matching Feature)

Dựa vào mô tả của các điểm thuộc tính của ảnh thứ nhất và thứ hai, ta sẽ tiến hành đối sánh hai ảnh dựa trên các tập điểm đó. Cuối cùng hiển thị các kết

quả khớp trên hình ảnh như hình 3.3 và ghi tệp vào đĩa để kiểm tra trực quan. Sau đó sử dụng thuật toán đo khoảng cách hamming tương tự như một thước đo giữa hai thuộc tính mô tả. Các đặc trưng phù hợp được hiển thị trong hình bên dưới bằng cách vẽ một đường nối chúng.

Hình 3.3. Ảnh chứa các đặc trưng tương đồng (match feature)

```
48      // Match features.
49      std::vector<DMatch> matches;
50      Ptr<DescriptorMatcher> matcher = DescriptorMatcher::create("BruteForce-Hamming");
51      matcher->match(descriptors1, descriptors2, matches, Mat());
52
53      // Sort matches by score
54      std::sort(matches.begin(), matches.end());
55
56      // Remove not so good matches
57      const int numGoodMatches = matches.size() * GOOD_MATCH_PERCENT;
58      matches.erase(matches.begin() + numGoodMatches, matches.end());
59
60      // Draw top matches
61      Mat imMatches;
62      drawMatches(im1, keypoints1, im2, keypoints2, matches, imMatches);
63      imwrite("matches.jpg", imMatches);
64
65      // Extract location of good matches
66      std::vector<Point2f> points1, points2;
67      //int th = matches.size() / 2;
68      for (size_t i = 0; i < matches.size(); i++)
69      {
70          points1.push_back(keypoints1[matches[i].queryIdx].pt);
71          points2.push_back(keypoints2[matches[i].trainIdx].pt);
72      }
```

Lưu ý, sẽ có nhiều kết quả khớp nhau nhưng không chính xác và do đó sẽ cần sử dụng một phương pháp để tính toán homography trong bước tiếp theo.

Bước 4: Tính toán ma trận Homography

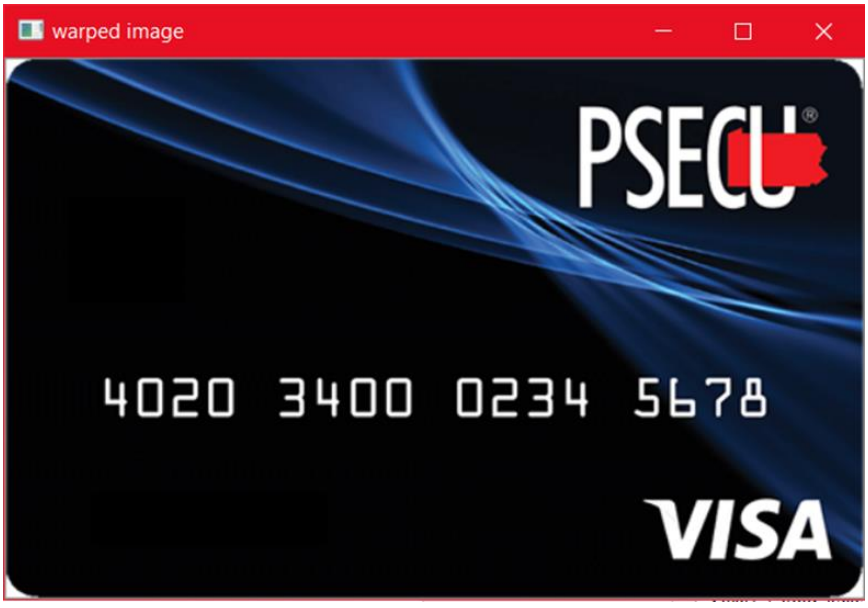
Một homography có thể được tính khi chúng ta có 4 điểm tương ứng trở lên trong hai hình ảnh. Kết hợp thuộc tính tự động được giải thích trong phần trước không phải lúc nào cũng tạo ra kết quả khớp chính xác 100%.

Không có gì lạ khi 20-30% không chính xác. May mắn thay, phương pháp findHomography sử dụng một kỹ thuật ước lượng mạnh mẽ được gọi là Đồng thuận mẫu ngẫu nhiên (RANSAC) tạo ra kết quả đúng ngay cả khi có số lượng lớn các kết quả khớp xấu.

```
74      // Find homography
75      h = findHomography(points1, points2, RANSAC);
```

Bước 5:

Khi đã tìm được ma trận homography chúng ta dùng warpPerspective để ánh xạ nó về gần với tọa độ của ảnh gốc nhất



Hình 3.4 Ảnh đã được căn chỉnh và ánh xạ gần tọa độ ảnh gốc nhất

Tuy ảnh không được giống 100% như ảnh gốc nhưng với kết quả đạt được so với ảnh ban đầu đã góp phần tăng độ chính xác lên kha khá.

3.2.2.2 Trừ ảnh

Bước 1. Trừ ảnh

Thao tác tiếp theo cần thực hiện là trừ ảnh. Sẽ cần lấy ảnh đầu vào hay còn gọi là ảnh thứ 2 trừ ảnh gốc để ra ảnh có phần khác biệt của 2 ảnh (giá trị tuyệt đối). Ở đây hàm abdiff của openCV sẽ được sử dụng. Cấu trúc hàm như sau:

```
89      absdiff(im1, im2, errorImage);
```

Trong đó:

- im1 là ảnh thứ 1 hay ảnh gốc
- im2 là ảnh thứ 2
- error Image là ảnh được lấy từ ảnh 2 trừ ảnh gốc để ra những phần khác nhau giữa 2 ảnh

Hình 3.5 Ảnh sau khi dùng hàm diff

Bước 2:Phân ngưỡng

Tiếp theo xác định đường viền sử dụng ngưỡng, thao tác này chuyển qua ảnh nhị phân, hay còn gọi là phân ngưỡng. Mục đích của thao tác phân ngưỡng ảnh là dùng để phục vụ cho việc tách các đối tượng của các thuật toán tiếp theo.

Để thực hiện việc phân ngưỡng, hàm threshold trong OpenCV sẽ được sử dụng, hàm được trình bày như sau:

```
91      threshold(errorImage, thresh, 0, 255, THRESH_BINARY);
```

Trong đó:

- errorImage là ảnh (sau khi sử dụng hàm diff) xám đầu vào
- thresh là ảnh đầu ra
- 0 là giá trị ngưỡng được gán nếu pixel giá trị nhỏ hơn giá trị ngưỡng
- 255 là giá trị được gán nếu pixel giá trị lớn hơn giá trị ngưỡng
- THRESH_BINARY là hằng số xác định cách phân ngưỡng. Tùy theo các loại phân ngưỡng mà pixel được gán giá trị khác nhau, ví dụ:
- THRESH_BINARY: Nếu giá trị pixel lớn hơn ngưỡng thì gán bằng
- maxval. Ngược lại bằng gán bằng 0. Phân đoạn ảnh dựa trên thuật toán nở vùng
- THRESH_BINARY_INV: Nếu giá trị pixel lớn hơn ngưỡng thì gán bằng 0. Ngược lại bằng gán bằng maxval
- THRESH_TRUNC: Nếu giá trị pixel lớn hơn ngưỡng thì gán giá trị

bằng ngưỡng. Ngược lại giữ nguyên giá trị

Bước 3:Lọc nhiễu

Tiếp theo đó là lọc nhiễu cho ảnh. Để lọc nhiễu cho bức ảnh này, hàm sử dụng bộ lọc làm mờ Blur sẽ được sử dụng (cụ thể là median blur).

Trong OpenCV hàm lọc được trình bày như sau:

```
92      medianBlur(thresh, imdiff, 9);
```

Trong đó:

- thresh là ảnh đầu vào (ảnh sau khi đã được phân ngưỡng)
- Imdiff là ảnh sau khi thực hiện phép lọc
- 9 là kích thước ma trận lọc và chắc chắn phải là số lẻ

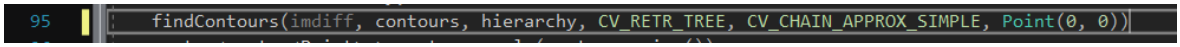
Hình 3.6 Ảnh sau khi sử dụng hàm diff, phân ngưỡng, lọc nhiễu để highlight phần khác biệt (bằng 2 màu trắng đen)

Bước 4: Xác định Contours

Contours là đường bao kết nối tất cả các điểm liền kề nhau có cùng màu sắc hoặc độ tương phản. Chính vì đặc tính này, contours thường được dùng trong xác định vật thể, nhận dạng, ...

Trong trường hợp này chúng ta cũng sẽ dùng thuật toán để tìm đường viền để có thể cho chúng vào xung quanh hình chữ nhật của các khu vực đã được xác định là khác nhau và lưu chúng vào vectơ contour và hiearchy

Để thực hiện được việc tìm biên của các đối tượng chúng ta sẽ sử dụng hàm findContours. Trong OpenCV hàm tìm biên được trình bày như sau:



Trong đó:

- Imdiff là ảnh (sau khi thực hiện phép lọc) cần tìm biên
- Contour là lưu trữ các đường biên tìm được, mỗi đường biên, sẽ được lưu trữ dưới dạng một vector của các điểm
- Hiearchy: chứa thông tin về hình ảnh như số đường viền, xếp hạng các đường viên theo kích thước, trong ngoài…
- CV_RETR_TREE: khi sử dung cờ này nó lấy tất cả các đường biên và tạo ra một hệ thống phân cấp đầy đủ của những đường lồng nhau
- CV_CHAIN_APPROX_SIMPLE: nó sẽ nén đường viên trước khi lưu trữ, nén phân đoạn theo chiều ngang, chiều dọc và chéo.

Sau khi đếm số biên bằng hàm findContours chúng ta sẽ có được số lượng biên tương ứng với số đối tượng.

Bước 5: Vẽ hình

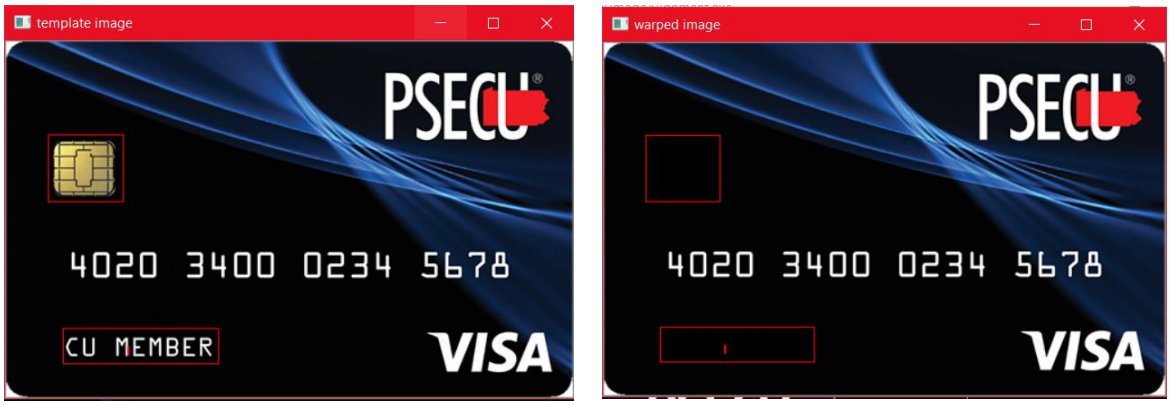
Trước khi vẽ hình chữ nhật chúng ta cần phải tìm đường biên của 2 hình ảnh. Ở đây, chương trình sẽ sử dụng hàm xấp xỉ hình chữ nhật với độ chính xác +-3 và đường cong phải là đường cong kín. Khi đã tìm được boundingRect cho mỗi hình chữ nhật và lưu vào boundingRect (boundingRect là hàm được viết sẵn trong OpenCV để tạo được hình chữ nhật bao quanh contour)

```
96      vector<vector<Point> > contours_poly(contours.size());
97      vector<Rect> boundRect(contours.size());
98      for (int i = 0; i < contours.size(); i++)
99      {
100         Scalar color = Scalar(0, 0, 255);
101         approxPolyDP(Mat(contours[i]), contours_poly[i], 3, true);
102         boundRect[i] = boundingRect(Mat(contours_poly[i]));
103         rectangle(imReference, boundRect[i].tl(), boundRect[i].br(), color, 1, 8, 0);
104         rectangle(imReg, boundRect[i].tl(), boundRect[i].br(), color, 1, 8, 0);
105     }
```

Rồi sau đó sử dụng các giá trị có được ở trên để vẽ một hình chữ nhật màu đỏ trên mỗi hình ảnh với hàm rectangle

Trong đó:

- ImReference và imReg là ảnh đầu vào
- boundRect[i].tl là điểm trên cùng góc bên trái của hình chữ nhật thứ i
- boundRect[i].br là điểm dưới cùng góc bên phải của hình chữ nhật thứ i
- color là màu của hình chữ nhật
- 1 là độ dày
- 8 là loại dòng kẻ



Hình 3.7 Đây là hai ảnh sau khi đã tìm được sự khác biệt và được khoanh vùng bởi hình chữ nhật màu đỏ

3.2.3 Một số kết quả

Ví dụ về ảnh khác:

Hình 3.8 Ảnh gốc

Hình 3.9 Ảnh cần kiểm tra

Hình 3.10 Ảnh cần kiểm tra đã được căn chỉnh và ánh xạ gần tọa độ ảnh gốc nhất

Hình 3.11 Ảnh chứa các đặc trưng tương đồng (match feature)

Hình 3.12 Ảnh sau khi sử dụng hàm diff, phân ngưỡng, lọc nhiễu để highlight phần khác biệt giữa 2 ảnh

Hình 3.13 Đây là hai ảnh sau khi đã tìm được sự khác biệt và được khoanh vùng bởi hình chữ nhật màu đỏ

KẾT LUẬN

Trong quá tìm hiểu tài liệu và thực hiện đồ án dưới sự định hướng của thầy hướng dẫn em đã đạt được một số kết quả như sau:

- Tìm hiểu được một cách tổng quan các vấn đề về đối sánh ảnh. Đưa ra nhận xét, đánh giá một số kỹ thuật tiền xử lý ảnh và có lựa chọn phù hợp đối với từng kỹ thuật.
- Đặc biệt phần nào hiểu rõ hơn về căn chỉnh ảnh và áp dụng thuật toán phù hợp cho bài tập.
- Đối với một số fornt chữ mà có các đặc tính giống nhau có thể gây ra sự nhầm lẫn trong việc ánh xạ các đặc tính giữa hai bức ảnh như chữ Nhật hay Trung Quốc.
- Bị hạn chế khi xoay góc độ của ảnh, làm cho ảnh bị lỗi không, hiện phần sai khác không đúng.
- Đối với những ảnh có nhiều chi tiết hơn sẽ xảy ra trường hợp chương trình khó tìm thấy những phần khác nhau giữa 2 ảnh, đôi khi sai lệch.
- Xây dựng một ứng dụng phát hiện phần khác biệt giữa hai ảnh dựa theo phương pháp căn chỉnh ảnh và trừ ảnh đã trình bày.
- Ngoài ra, trong quá trình tìm hiểu em cũng tự tích lũy thêm cho mình các kiến thức về toán học, về kỹ thuật lập trình,…Tuy mới chỉ là bước đầu, nhưng những kết quả này sẽ giúp ích cho em trong những tìm hiểu sau này để thu được những kết quả tốt hơn. Dựa trên những kết quả đã đạt được, em sẽ tiếp tục tìm hiểu đề xuất một số cải tiến phương pháp căn chỉnh ảnh và trừ ảnh hiệu quả hơn trong tương lai

TÀI LIỆU THAM KHẢO

[1.] [1] Richard Szeliski, " Image Alignment and Stitching",Last update December 10,2006.

[2.] [2]Satya Mallick, "Image Aligment (Feature Based)using OpenCV(C++/Python)",March 11, 2008.

[3.] [3]Adrian Rosebrock, "Image Difference with OpenCV and Python", June 19,2007.

[4.] [4]Deepanshu Tyagi, "Introduction to ORB(Oriented FAST and Rotated BRRIEF", Jan 1.